IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 25, 2019, accepted March 13, 2019, date of publication March 20, 2019, date of current version April 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906345

Autonomous Moving Target-Tracking for a UAV Quadcopter Based on Fuzzy-PI

MOHAMMED RABAH[1,2], ALI ROHAN[1], SHERIF A. S. MOHAMED[3], AND SUNG-HO KIM[4]

[1]School of Electronics and Information Engineering, Kunsan National University, Gunsan 54150, South Korea
[2]Department of Electronics and Telecommunication Engineering, Al-Safwa High Institute of Engineering, Cairo 11837, Egypt
[3]Department of Future Technologies, University of Turku, 20014 Turku, Finland
[4]Department of Control and Robotics Engineering, Kunsan National University, Gunsan 54150, South Korea

Corresponding author: Mohammed Rabah (mohamedmostafamousa1991@gmail.com)

ABSTRACT Moving target-tracking is an attractive application for quadcopters and a very challenging, complicated field of research due to the complex dynamics of a quadcopter and the varying speed of the moving target with time. For this reason, various control algorithms have been developed to track a moving target using a camera. In this paper, a Fuzzy-PI controller is developed to adjust the parameters of the PI controller using the position and change of position data as input. The proposed controller is compared to a gain-scheduled PID controller instead of the typical PID controller. To verify the performance of the developed system and distinguish which one has better performance, several experiments of a quadcopter tracking a moving target are conducted under the varying speed of the moving target, indoor and outdoor and during day and night. The obtained results indicate that the proposed controller works well for tracking a moving target under different scenarios, especially during night.

INDEX TERMS Target-tracking, Fuzzy-PI, gain-scheduled PID, quadcopter, MATLAB/SIMULINK.

I. INTRODUCTION

Quadcopter is a type of Unmanned Aerial Vehicles (UAV), which is lifted and propelled by four rotors [1], [2]. The quadcopter has been increasingly used in education and research area due to its low cost, and high maneuverability. Furthermore, a quadcopter is capable of handling complex tasks in cramped and crowded environments, as well as it has a simple control mechanism compared to the other types of UAVs [3]. Because of this, there has been increasing interest in the quadcopter applications such as Target-Tracking, object detection and landing to drone station [4]–[8].

Especially, Target-Tracking is an essential task for quadcopters, as it has been proven to be an effective tool in search and rescue surveillance [9], following [10], and providing aerial based video of sports events. Also, it has become a relevant field for the correct development of many multidisciplinary applications, such as traffic supervision, autonomous robot navigation, mapping, and aerial photography.

The autonomous Moving Target-Tracking system for quadcopter can be divided into three processes; 1) Taking off, 2) detection of the moving target, and 3) tracking the moving target. Since the quadcopters are dynamically unstable, under-actuated, and nonlinear systems, it is very challenging to design a controller for a quadcopter to track a moving target especially under the varying speed of the moving target and under environment effect. This is due to that any error occurs in any of them can lead to severe damage to the quadcopter.

Many control algorithms for tracking a moving target have been developed to deal with these difficulties. Huang *et al.* [11] present a tracking scheme based on Global positioning system (GPS) and CamShift algorithm. Human detection and tracking method based on image processing on a Raspberry Pi environemnt is presented in [12]. Pei *et al.* [13] developed a trajectory control algorithm based on a Linear Quadratic Integral (LQI) optimal controller with a feedback linearization. In [14], a target-tracking control algorithm based on Fuzzy Logic Controller (FLC) has been developed. A PD and PID controllers are used to develop a tracking control algorithm for a quadcopter [15]. An autonomous control algorithm for a quadcopter based on Fuzzy-PD is proposed in [16]. A linear quadratic regulator (LQR) control method and a PD controller have been introduced in [17]. A back-stepping controller and a sliding mode controller were proposed in [18] to overcome the under-actuated

The associate editor coordinating the review of this manuscript and approving it for publication was Xudong Zhao.

problem of a quadcopter. Chen *et al.* [19] (2018), proposed a tracking method based on fusing the inertial measurements from the Inertial Measurements Unit data (IMU), GPS readings, and images captured by the camera. Samir *et al.* [20] (2017), proposed a trajectory tracking control algorithm using State-Feedback control with integral action. A Fuzzy-PID controller is designed for quadcopter to follow different trajectories [21]. A target detection algorithm based on a PID controller and a safe landing algorithm based on a FLC is proposed in [22]. Wang *et al.* [23], Wang *et l.* [24], and Yin *et al.* [25] developed an adaptive fuzzy tracking controller for nonlinear systems.

Most of the prior studies show good performances and results. However, they are either using a GPS to detect the position of the target which provides inaccurate localization in indoor environments or they are using sensors fusion. Other researchers focused more on the computer vision part, ignoring the effect of the outside disturbances such as environment effect. Moreover, running computer vision algorithms in real-time requires an expensive and higher specification CPUs. Other authors dedicated only on tracking a moving target with a constant speed using a PID and FLC controllers. Although they are working well, the results show slower response and higher overshot. Some developed techniques are based on image processing techniques by using an RPi. Though, the RPi has its drawbacks when it comes to real-time tracking, especially with UAVs because of its limited memory. The remaining authors concentrated on the simulation results without any practical test neglecting the effect of the environmental disturbances.

The current research focuses on the problems of previous research works. It focuses on replacing the high specification and expensive CPUs with a cheaper one. Also, it focuses on decreasing the number of the used sensors, so only the IR-Camera output is used as input to the proposed controller. Moreover, since the objects are often distant, small and their signatures have low contrast against the background during the night, the traditional object detection algorithms do not perform very well at night. Therefore, an IR-Camera is used to detect objects during day and night. This IR-Camera can detect any object that is equipped with IR LEDs, which make it simple, cheap and easy to use to detect a fixed or moving object, especially during the night.

To overcome the problem of environment effect such as wind, varying speed of the moving target, and detecting the target during the night, a novel Moving Target-Tracking algorithm based on Fuzzy-PI is developed. The proposed Fuzzy-PI controller has its own rule base which can either stabilize the quadcopter over a non-moving target or track a moving target under varying speed. The proposed system is equipped with an IR-Camera that is connected to an Arduino DUE for detecting the moving target. Then it calculates the required Roll and Pitch angles that are responsible for tracking the moving target and send the acquired data to the flight controller.

The main contributions of this paper lie in the following: 1) Overcome the instability and nonlinearity of the quadcopter system by using a Fuzzy-PI controller, 2) Be able to stabilize or track a target during day and night by knowing the position and position's change of the detected target, and finally 3) Implement the proposed algorithm in a cheap MCU.

To verify the system performance, MATLAB/SIMULINK is used to compare the output of the proposed controller to a typical PID controller and a Fuzzy Logic Controller (FLC). Simulations are carried out under different trajectories. Afterward, a Gain-Scheduled PID controller is used instead of the typical PID controller, as it has been proven its ability to track the moving target under varying speed. Several experiments were conducted under different scenarios are demonstrated.

The practical configuration of the proposed system is discussed in detail in section 2. Section 3 presents a brief description of the Gain-Scheduled PID controller and the Fuzzy-PI controller. The proposed moving target-tracking algorithm is explained in section 4. Section 5 demonstrates the simulation studies of the proposed system, followed by its experimental results of the proposed system. Finally, the conclusion is drawn in section 6.

II. CONFIGURATION OF THE PROPOSED MOVING TARGET-TRACKING FOR QUADCOPTER

The proposed moving target-tracking for quadcopter using an IR-Camera system consists of the following components: flight controller, Arduino DUE, IR-Camera, ESC module, RC receiver/transmitter, BLDC motor, a LIDAR sensor, and a 2D camera gimbal with quadcopter mainframe. A block diagram of the overall system is shown in Fig.1.

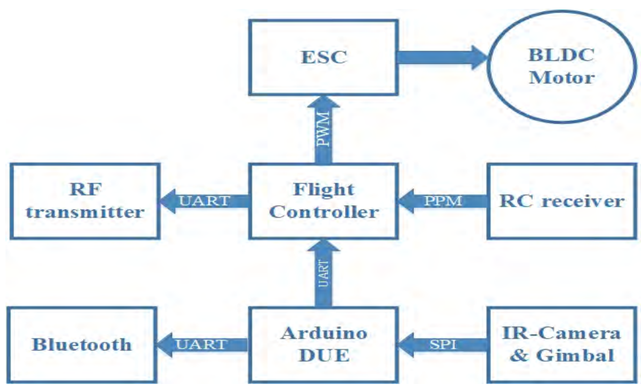


FIGURE 1. Block diagram of the overall system.

A. FLIGHT CONTROLLER

The flight controller is the brain of the quadcopter. It reads all data coming from the sensors and calculates the best commands, and finally sends it to the ESC module to control the speed of each BLDC motors. The flight controller used in this work is a Pixhawk flight controller. The Pixhawk flight controller has an ARM Cortex M4 CPU with a clock frequency of 168 MHz. It is equipped with a 10 DOF-IMU, to measure roll, pitch, yaw, and altitude. It also has eight PWM outputs which can support up to eight BLDC motors. It also has several connectivity options for additional peripherals like UART, I2C, CAN, SPI, and ADC, etc. The main reason for utilizing the Pixhawk flight controller is because it can be interfaced easily with the ArduCopter, which is an open source code that is written in C++ and is free to use.

B. ARDUINO DUE

The Arduino Due is a microcontroller board based on the ARM Cortex-M3 CPU, with a clock speed of 84 MHz. Arduino DUE is used because both Roll and Pitch data are updated inside the flight controller every 10 msec. Therefore, it's necessary to use a microcontroller with fast execution time.

C. IR-CAMERA AND GIMBAL

The camera used in this work is a Pixy camera with a filter and IR lens installed on it. The Pixy camera is fast, easy to use and cheap. It supports several interfaces such as SPI, UART, and I2C. It can detect IR object and gives out x *and* y coordinates of the center of the detected object in pixels. This data is given out every 20 msec. The IR-Camera is connected to the Arduino DUE through SPI, and it's attached to a 2D-gimbal, that is used for allowing the camera to always face the ground while the quadcopter is hovering around.

III. CONTROLLERS DESCRIPTION

Figure 2 shows the block diagram of the proposed controller for both Gain-Scheduled PID controller and Fuzzy-PI controller. As it is shown in Fig.2, the IR-Camera gives out the x *and* y position of the detected object in pixels. This position (pos) is assumed to be the error $e(t)$ between the position of the detected object and the centroid of the image since the centroid is always zero. After that, change of position (Δpos) is obtained by taking the difference between the current position and the previous position, while maximum change of position ($Max\_\Delta pos$) is a fixed value that is obtained by storing the maximum change of position of the moving target in pixels. pos *and* Δpos are used as inputs for Fuzzy-PI controller, and Gain-Scheduled PID controller.

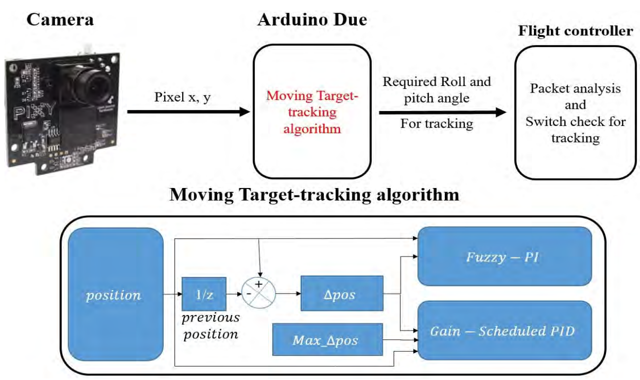


FIGURE 2. Block diagram of the proposed controllers.

A. GAIN-SCHEDULED PID CONTROLLER

Gain scheduling is one of the most popular approaches to nonlinear control design, as it has a better performance than robust ones [26]. Therefore, a Gain-Scheduled PID controller is used instead of the typical PID controller with fixed gains to track the target. In Fig.2, the Gain-Scheduled PID controller has three inputs, pos, Δpos, *and* $Max\_\Delta pos$. Equation (1) indicates how to calculate the required Roll and Pitch angle for tracking.

$$y(t) = \left(1+\left|\frac{\Delta pos}{Max_{\Delta} pos}\right|\right)K_p e(t) + K_i\int_0^t e(\tau)\,d\tau + K_d\frac{de(t)}{dt} \quad (1)$$

where the error $e(t) = pos$, $y(t)$ is the output of the system, K_p, K_i *and* K_d are the proportional, Integral, and derivative gains. As it is seen in the previous Equation, the proportional gain of the PID controller can be increased according to the ratio between Δpos, and $Max\_\Delta pos$, so if the target increased its speed suddenly, the proportional gain will also increase to be able to track the moving target.

B. FUZZY-PI CONTROLLER

The designed Fuzzy Proportional-Integral (Fuzzy-PI) controller is a hybrid controller, utilizing two sets of PI gains to achieve a non-linear response [27]. Two inputs are given to the Fuzzy-PI, pos and Δpos. It gives out two outputs, K_{pf} *and* K_{if} which are determined by a set of fuzzy rules that are used to adapt the K_p *and* K_i gains of the PID controller as in Equation (2, 3), where G is a gain value that can be used to for tuning the output of the Fuzzy-PI, P is the sum of the proportional gain K_p and the K_{pf} gain, and I is the sum of the integral gain K_i and the K_{if} gain. The output of the Fuzzy-PI is described in Equation (4), where $e(t) = pos$, $y(t)$ is the output of the system, and K_d is a fixed derivative gain.

$$P = (G * K_{pf}) + K_p \quad (2)$$
$$I = (G * K_{if}) + K_i \quad (3)$$

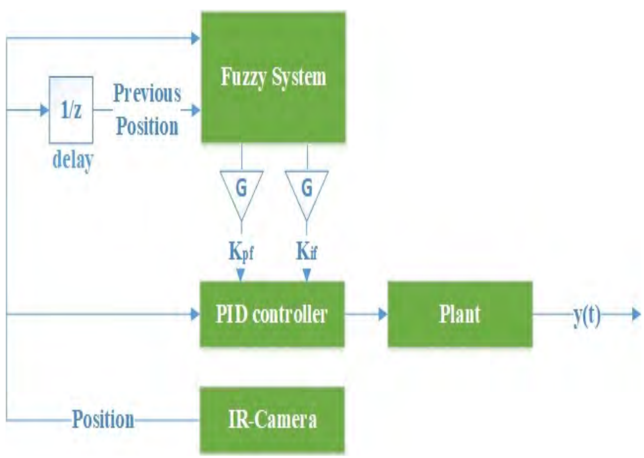


FIGURE 3. Fuzzy-PI basic structure.

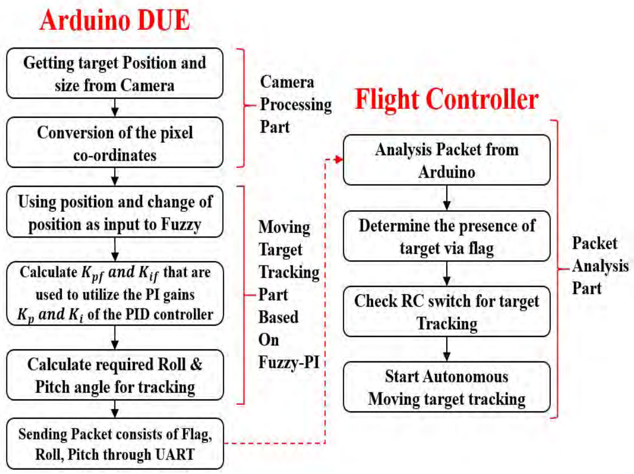


FIGURE 4. Moving target-tracking algorithm.

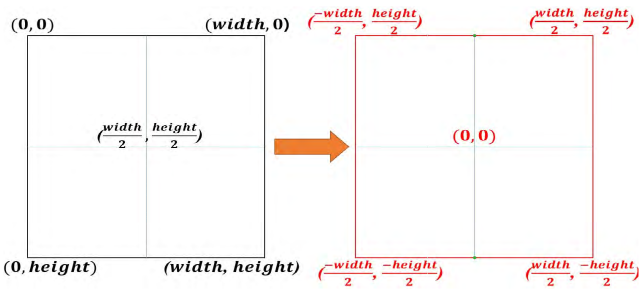


FIGURE 5. Conversion of the pixel coordinates of the IR-Camera.

$$y(t) = (P * e(t)) + (I * \int_0^t e(\tau)\, d\tau) + K_d \frac{de(t)}{dt} \tag{4}$$

Figure 3 depicts the basic structure of the Fuzzy-PI controller.

IV. AUTONOMOUS MOVING TARGET-TRACKING ALGORITHM FOR QUADCOPTER

The proposed algorithm can be divided into three parts as shown in Fig.4:

1) Camera process part.
2) Moving target-tracking part (based on Fuzzy-PI).
3) Data packet processing part.

In the first part, the position of the target given by the IR-Camera is represented as pixel coordinates (x, y). These co-ordinates is changed so that the center co-ordinates are converted from $(\frac{width}{2}, \frac{height}{2})$ to $(0, 0)$ as demonstrated in Fig.5, by using Equation (5, 6).

$$X = x - \frac{width}{2} \tag{5}$$

$$Y = y - \frac{height}{2} \tag{6}$$

where X *and* Y are the new x *and* y position in pixels, *width and height* are the resolution of the IR-Camera. These new positions are assumed to be the $e(t)$ for the proposed controller.

FIGURE 6. Block diagram of the proposed fuzzy logic controller.

A. MOVING TARGET-TRACKING ALGORITHM BASED ON FUZZY-PI

When the quadcopter detects the moving target, the Target-Tracking algorithm is activated. In this work, a Fuzzy-PI controller is utilized. The proposed controller can be thought as an auto-tune for PID controller, where the gains are continuously changed according to the value of the two inputs *pos and* Δpos of the Fuzzy-PI controller.

Generally, the Fuzzy-PI controller is made up of fuzzification, rule base, and the de-fuzzification process [28]. The Fuzzy-PI input, output membership functions, and rule base are designed in MATLAB/ SIMULINK. Fig.6 illustrates the block diagram of the proposed Fuzzy-PI controller.

1) FUZZIFICATION

Fuzzy-PI input-output membership functions and rule-base are designed in MATLAB/SIMULINK using the Simulink fuzzy toolbox. The Universe of Discourse (UD) of each fuzzy input and output is carefully selected according to the desired ranges. The Fuzzy-PI controller has two inputs and two output. 49 rules are made for each output for optimum tracking control. Input and output membership functions are shown in Fig. 7 and Fig.8. The UD of each membership function defines the operation range of that specific fuzzy linguistic variables. These ranges are adjustable and act similar to PID gains. These fuzzy linguistic variables are defined as OUT_POS_NB (OUT range Position Negative Big), OUT_POS_NM (OUT range Position Negative Medium), IN_POS_ NS (IN range Position Negative Small), IN_ZE (IN range Zero), IN_POS_PS (IN range Position Positive Small), OUT_POS_PM (OUT range Position Positive Medium), and OUT_POS_PB (OUT range Position Positive Big) for the corresponding position input for both *X and Y pos* as in Fig (7-a & 7-b) Fuzzy linguistic variables for the second input Δpos is defined as NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZE (Zero), PS (Positive Small), PM (Positive Medium), and PB (Positive Big) as shown in Fig. (7-c)Δpos input parameters are the most important parameter in this controller, because it's responsible for controlling the tracking speed and angle of the quadcopter. The fuzzy set 'ZE' indicates a fixed position of the quadcopter. Whereas in Fig. (8-a & 8-b), NB, NM, NS, ZE, PS, PM, and PB fuzzy linguistic functions are for tuning the PI gains. Negative Δpos like NB, NM, and NS indicates a big/medium/small change of position in the negative direction of the detected target. Similarly, positive Δpos like PB, PM, and PS indicates a big/medium/small change of position in the positive direction of the detected target.

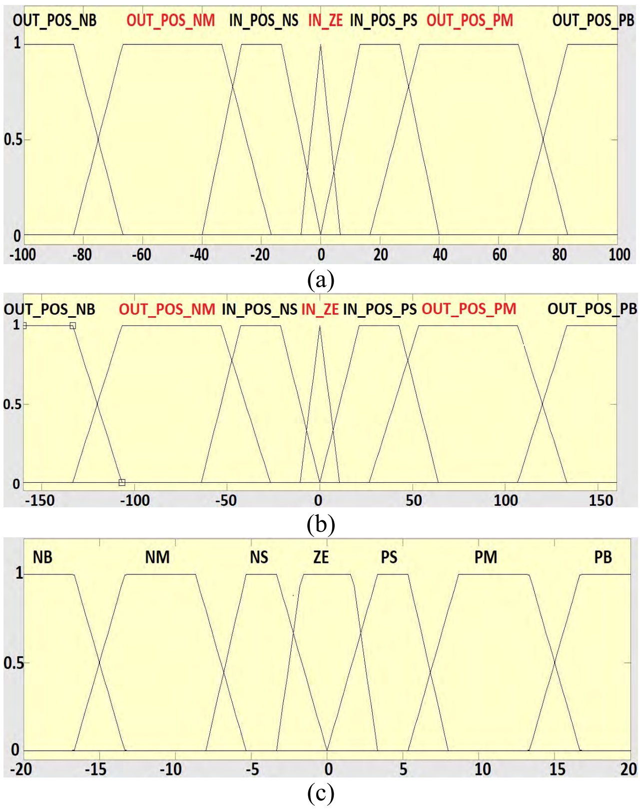


FIGURE 7. **(a) Fuzzy input *X* (pixels), (b) fuzzy input *Y* (pixels), and (c) fuzzy input Δ*pos* for *X and Y* (pixels).**

Equation (7, 8) explains Δpos calculation formula for both X *and* Y *position*.

$$\Delta pos_x = \frac{X_{current} - X_{pre}}{\Delta t} \tag{7}$$

$$\Delta pos_y = \frac{Y_{current} - Y_{pre}}{\Delta t} \tag{8}$$

where $\Delta pos_{x/y}$ is the change of position of the quadcopter in X/Y positions in pixels, $X/Y_{current}$ is the current position in X/Y axis in pixels, X/Y_{pre} is the previous position in X/Y axis, and Δt is the time difference between the current position and previous position.

Position range is taken from -160 to 160 pixels for Y, and -100 to 100 for X, while the Δpos ranges from -20 to 20 for both X *and* Y. Any values above or below the previous values will be replaced by the maximum and minimum value in X, Y, *and* Δpos. The position input of the Fuzzy-PI consists of two ranges, OUT and IN ranges. The OUT works as a brake for the quadcopter by decreasing its incoming angle and push it back to the center of the detected target avoid losing the target, while the IN ranges is responsible for the Target-Tracking, and also to stabilize the quadcopter above the target if it stopped moving. The Output of the Fuzzy-PI controller for K_{pf} gain ranges from -1 to 5, and for K_{if} gain, it ranges from -1 to 1. The "±" sign indicates the change in both P and I gains, since K_{pf} *and* K_{if} are either added or subtracted to the fixed K_p *and* K_i gains, as in

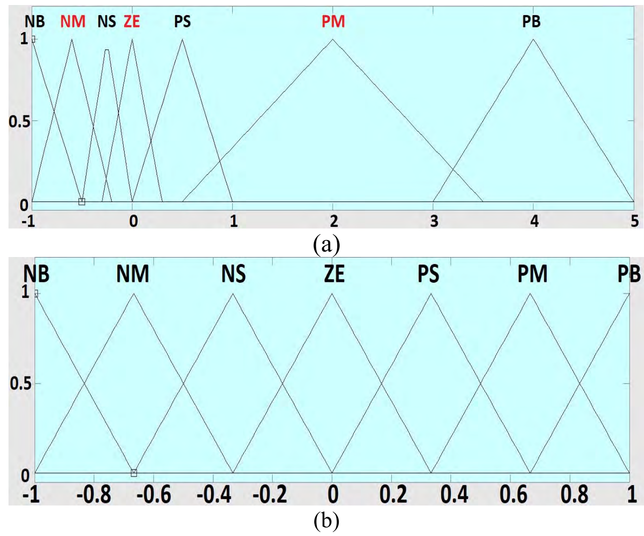


FIGURE 8. **(a) Fuzzy output (K_{pf}) and (b) Fuzzy output (K_{if}).**

Equation (2, 3). These ranges for each parameter can be changed in fuzzy logic or gains can be used for more precise tuning.

2) DESIGNING THE RULE BASE

Designing the rules-base is the most crucial part of the fuzzy logic controller, as the whole function of this controller depends on the rule base. Therefore, after several practical tests and experiments, in addition to experience, the rule base was designed. Table.1 and Table.2 show the 49 rules designed to utilize the *PI* gains that is responsible for tracking the moving target. This rule base works in two different Modes, Mode A and Mode B.

Mode A: This mode operates in the OUT region where the position of the quadcopter and its change of position are monitored to give the suitable K_{pf} *and* K_{if} responsible for utilizing the *PI* gains of the PID controller for tracking the moving target. This mode can be thought of a brake where it works in two different ways. First, if the quadcopter is moving away from the center of the detected target, a positive P gain and negative I gain is used to brake the quadcopter and stop it from losing the target and return it back to track the moving target. Second, if the quadcopter is moving towards the center of the target with either medium or big Δpos, a negative P gain only is used to decrease the angle and speed of the quadcopter. Figure 9 demonstrates how the quadcopter operates in Mode A. Following are examples of the operation of the quadcopter in Mode A:

- When *pos* is OUT_POS_NB and Δpos is NM, K_{pf}

will be PB, while K_{if} will be NB. This will increase the value of the total *P* gain of the PID controller and decrease the value if the *I* gain to quickly brake the quadcopter and stop it from losing the target and move in the opposite direction towards the center of the detected target. Similarly, if the *pos*

TABLE 1. Fuzzy-PI rule base for P gain.

pos / *Δpos*	Mode A		Mode B			Mode A	
	OUT_POS_NB	OUT_POS_NM	IN_POS_ NS	IN_ZE	IN_POS_ PS	OUT_POS_PM	OUT_POS_PB
NB	PB	PB	PS	ZE	NB	NM	NM
NM	PB	PB	PS	ZE	NM	ZE	ZE
NS	PM	PM	PS	ZE	NM	PS	PS
ZE	PM	PM	PS	ZE	PS	PM	PM
PS	PS	PS	NM	ZE	PS	PM	PM
PM	NM	ZE	NM	ZE	PS	PB	PB
PB	NB	NM	NB	ZE	PS	PB	PB

TABLE 2. Fuzzy-PI rule base for I gain.

pos / *Δpos*	Mode A		Mode B			Mode A	
	OUT_POS_NB	OUT_POS_NM	IN_POS_ NS	IN_ZE	IN_POS_ PS	OUT_POS_PM	OUT_POS_PB
NB	NB	NB	PB	PM	NM	ZE	ZE
NM	NB	NB	PM	PM	NS	ZE	ZE
NS	NB	NM	PS	PS	NS	ZE	NS
ZE	NM	NS	PS	ZE	PS	NS	NM
PS	NS	ZE	NS	PS	PS	NM	NB
PM	ZE	ZE	NS	PM	PM	NB	NB
PB	ZE	ZE	NM	PM	PB	NB	NB

is OUT_POS_PM and Δpos is PS, the output of the Fuzzy-PI will be PM for K_{pf}, and NM for K_{if}.

- When pos is OUT_POS_NM and Δpos is PS, K_{pf} will be PS, while K_{if} will be ZE. The reason for this, is because when quadcopter is moving towards the center of the detected target with small Δpos, it will need more P gain to push it into the IN region, to keep tracking the center of the detected target.
- When pos is OUT_POS_PB and Δpos is NM, K_{pf} will be NM, while K_{if} will be ZE. This will decrease the P gain with a medium value which in return will decrease the angle and speed of the quadcopter moving towards the center of the detected target.

Mode B: This scenario occurs when the quadcopter approaches the IN range, where it has to either decrease the quadcopter angle and speed or try to stop the quadcopter from reaching the OUT region. This mode is responsible for tracking and stabilizing the quadcopter above the center of the detected target. Figure 9 shows how the quadcopter operates in Mode B, followed by some examples of the operation of the quadcopter in Mode B.

- When pos is IN_POS_NS and Δpos is NB, K_{pf} will be PS, while K_{if} will be PB. This will slightly increase the P gain of the PID controller and also increase the I gain to stop the quadcopter from reaching the OUT region and return it back towards the center of the detected target.
- When pos is IN_ZE and Δpos is NB, K_{pf} will be ZE, while K_{if} will be PM. In this situation, the quadcopter will try to stabilize above the center of the detected target by using the total value of I gain and the fixed K_p gain.
- When pos is IN_POS_PS and Δpos is NB, K_{pf} will be NB, while K_{if} will be NM. This gains are used to rapidly decrease the quadcopter angle and speed, so it won't pass the IN_ZE position and try to stabilize the quadcopter and continue tracking the center of the detected target.

3) DEFUZZIFICATION

When the quadcopter is tracking the moving target, the corresponding rules are used to generate the K_{pf} *and* K_{if} that are used to adjust the PI gains of the PID controller. There are many defuzzification methods for this. In the current study, the Center of Gravity (CoG) method [23] is utilized. K_{pf} *and* K_{if} values calculated from the CoG method are added or subtracted to K_p *and* K_i as in Equation (2-4). Finally, the output of the Fuzzy-PI controller is given to the flight controller as shown in Fig.6.

B. DATA PACKET PROCESSING PART

Once the Fuzzy-PI controller gives out the K_{pf} *and* K_{if} gains responsible for adjusting the P *and* I gains discussed in Equation (2, 3), PID controller calculates the required roll and pitch angle responsible for tracking the moving target.

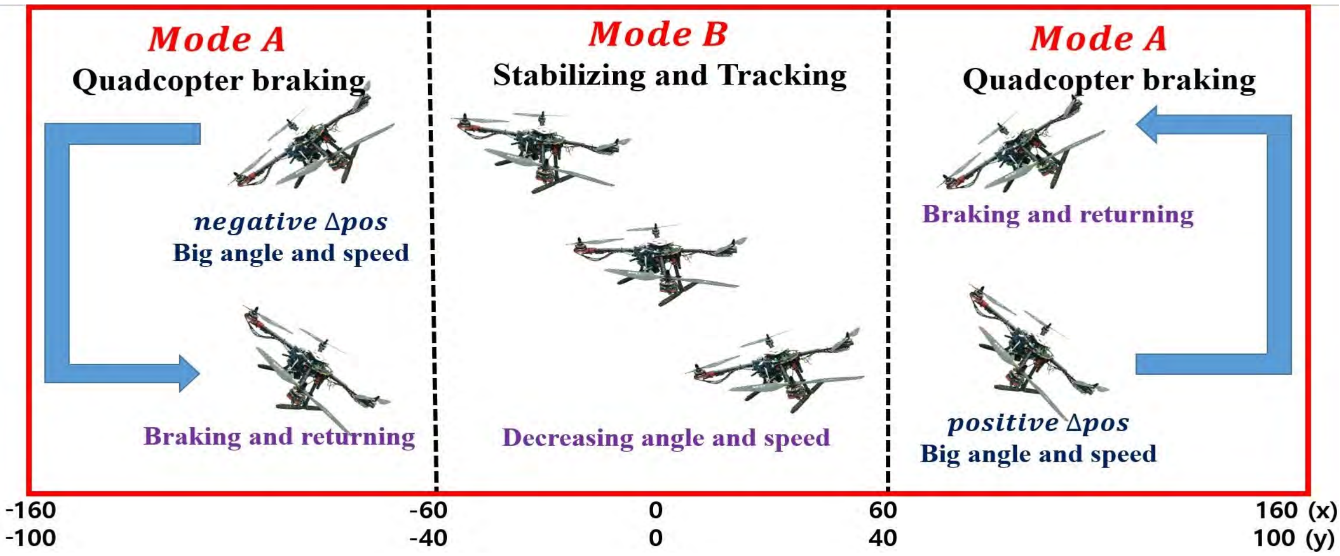


FIGURE 9. Operating regions of the proposed Fuzzy-PI controller.

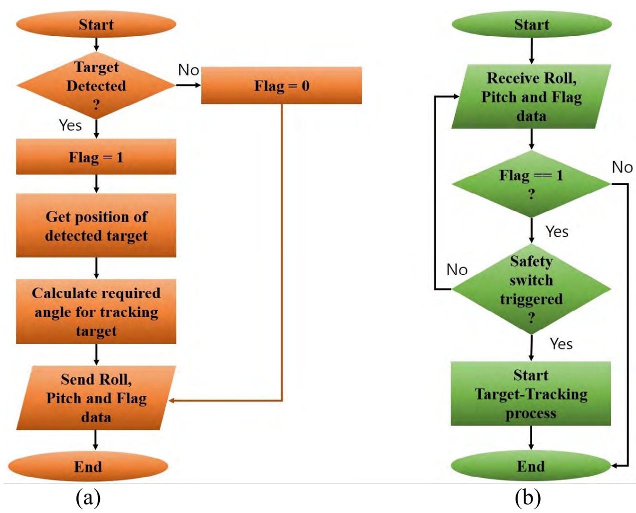


FIGURE 10. (a) Flowchart of proposed moving-target tracking algorithm and (b) flowchart of the received data processing inside of the flight controller.

The calculated roll and pitch angle are sent as a packet to the flight controller through UART communication in addition to a Flag value. This data is sent every 22 msec to the flight controller. Once the flight controller receives the data, it will check for the Flag if it's either 1 or 0. If the Flag is equal to 1, it will send commands to the quadcopter that the Moving Target-Tracking algorithm is ready to operate. Furthermore, an emergency switch on the RC transmitter is provided to support the safe operation of the quadcopter. For this, both Flag and the safety switch should be triggered to start tracking the moving target. Figure (10-a) depicts the flowchart of the Moving Target-Tracking algorithm inside the Arduino Due, while Fig. (10-b) shows the flowchart of the received data processing inside of the flight controller.

V. SIMULATION AND EXPERIMENTAL STUDIES

A. SIMULATION STUDIES

MATLAB/SIMULINK is used to test the proposed system by comparing the performance of the PID controller, FLC and the Fuzzy-PI controller under two different trajectories. The used quadcopter simulation system is available on MATLAB file exchange, and it's free to use. The used FLC rule base and membership function can be found in [21]. Figure (11-a) illustrates the simulation system of the quadcopter. In this system, the position control block is responsible for taking the trajectory data (x, y, t) from the user, where x, y represent the coordinates points in a Cartesian coordinate system and t is the time. The position control block gives out the required angle responsible for controlling the quadcopter over the defined trajectory. Figure (11-b) shows the position control block, where **X/Y error e(t)** block is responsible for calculating the position error between the current position in the desired trajectory and the current position of the quadcopter, and outputs X error and Y error. The Control blocks uses the resulted X error and Y error to calculate the desired Roll/Pitch angles required for tracking based on the outputs of the two controllers. Figure (11-c) depicts the controllers used in the Control blocks to control the trajectory-tracking of the quadcopter with a manual switch block is to switch between the three controllers. A limit switch is used to limit the output angle between -15 degree and 15 degree. Figure (11-d) demonstrates the implementation of the Fuzzy-PI controller based on Equations (2-4). Figure 12 and Fig.13 shows the two trajectories defined by the user for the comparison between the PID, FLC and the Fuzzy-PI controllers. In Figures (12-a & 12-b), Fuzzy-PI shows the best response in x and y-direction while PID controller and FLC have overshot. Figure (12-c) displays the output of the three controllers in the XY plane. As it is seen in the figure, the Fuzzy-PI has the best response among the three controllers. On the other hand, the FLC has overshot while PID controller has bigger overshot and oscillation. Figures (13-a & 13-b) show that Fuzzy-PI and FLC have close results. With a small oscillation in the FLC and poor stability in the PID controller where it takes more time to settle. Figure (13-c) indicates that the Fuzzy-PI has a slightly better response than FLC.

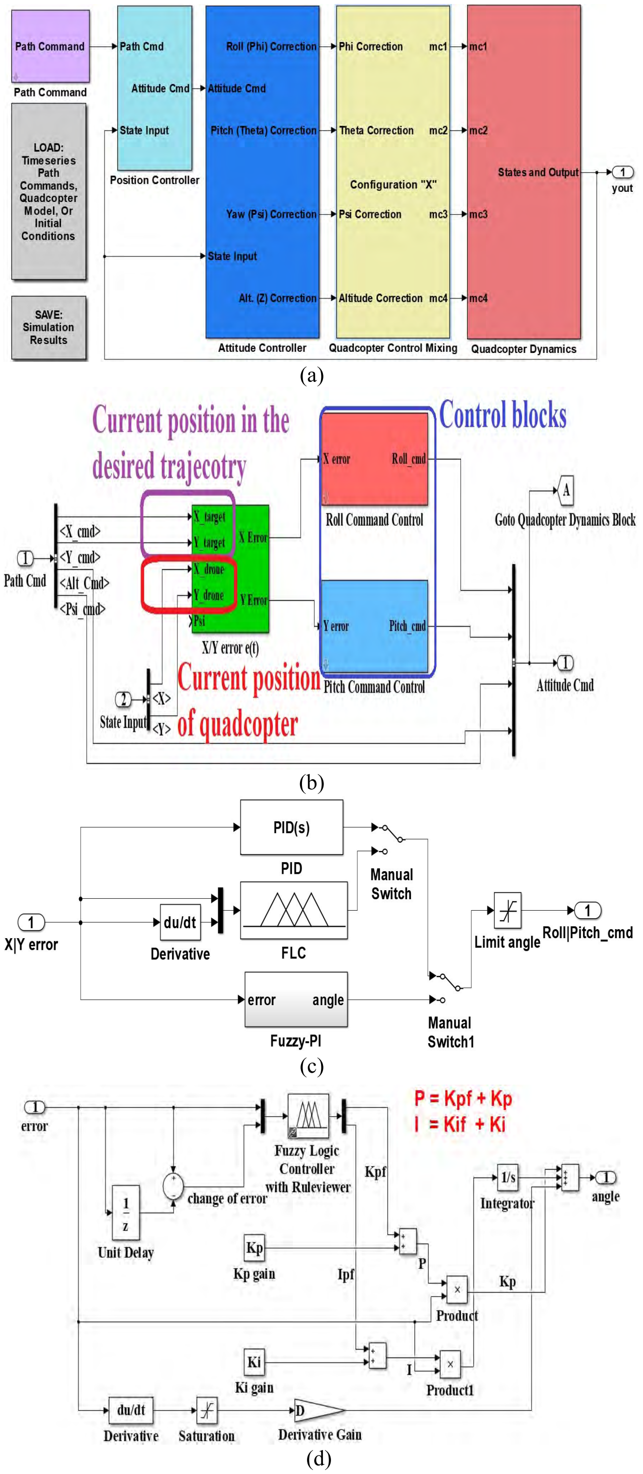


FIGURE 11. (a) Quadcopter simulation system, (b) position control block, (c) trajectory-tracking control system, and (d) fuzzy-PI controller implementation.

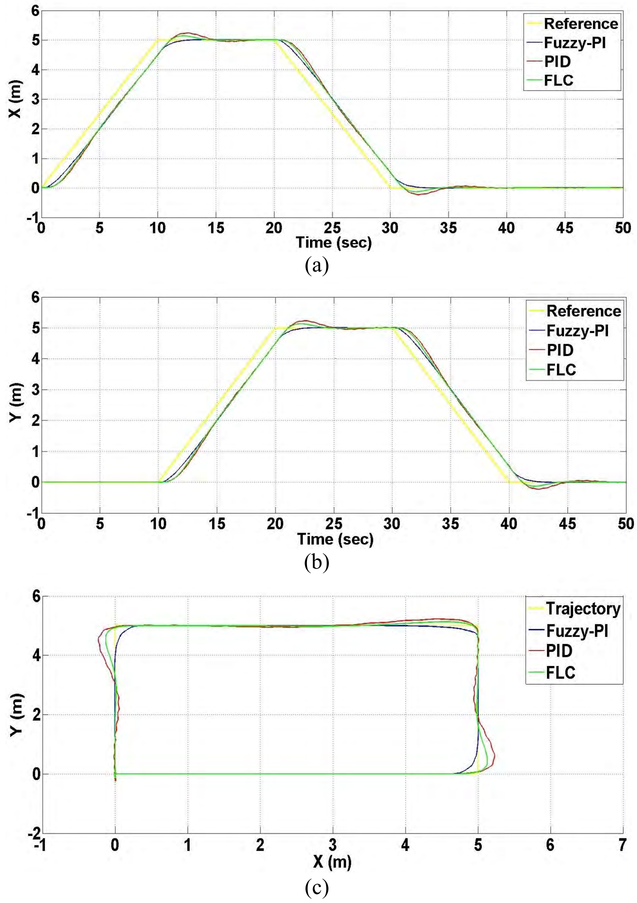


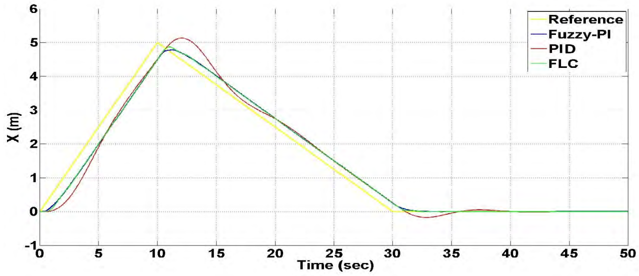
FIGURE 12. Simulation results of the three controllers in a rectangular trajectory in (a) X position, (b) Y position, and (c) XY plane.

B. EXPERIMENTAL STUDIES

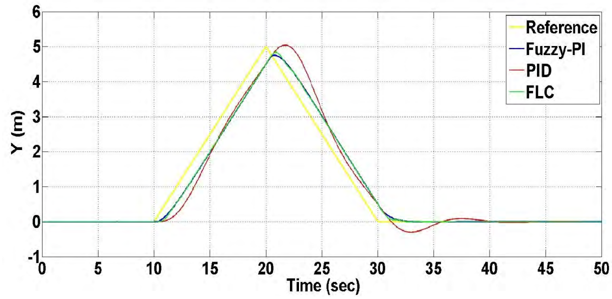
To verify the performance of the proposed Moving Target-Tracking algorithm, several experiments were performed indoor and outdoor. The experimental system is composed of a quadcopter frame with flight controller and Arduino DUE which executes the Moving Target-Tracking algorithm. A Pixy camera with IR lens installed on it, attached to a 2D gimbal is connected to the Arduino DUE. The Pixy camera gives out the position of the center of the detected target in pixels. A Bluetooth connected to the Arduino DUE is used to send the position of the detected target in pixels every 20 msec. Figure (14-a) depicts the quadcopter experimental setup, while Fig. (14-b) shows the experimental target. The Fuzzy-PI controller design is a time-consuming process when it comes to real-time calculation. It is possible to implement Fuzzy-PI controller on most high-end MCUs, but the problem occurs when a fast response is needed from the controller similar to the quadcopter tracking scenario. To overcome this problem, a look-up table based on Fuzzy-PI controller is designed to make the tracking process faster and efficient.

1) DESIGNING THE LOOK-UP TABLE

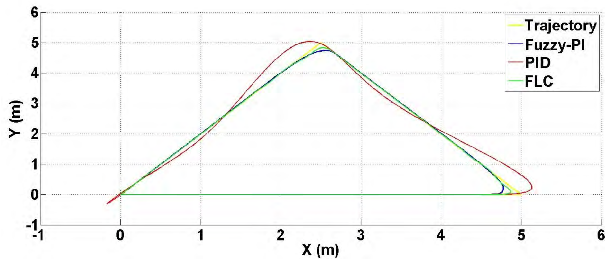
A Simulink based fuzzy logic system is designed to generate output data from the Fuzzy-PI controller. Figure 15 shows



(a)



(b)

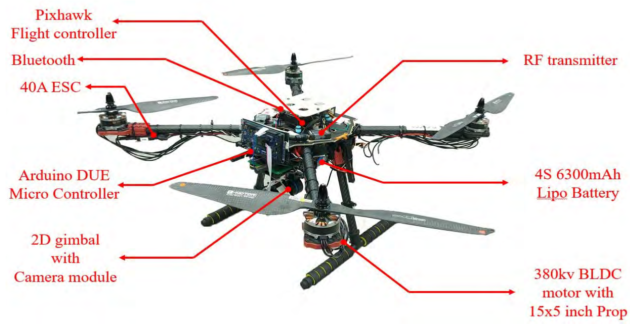


(c)

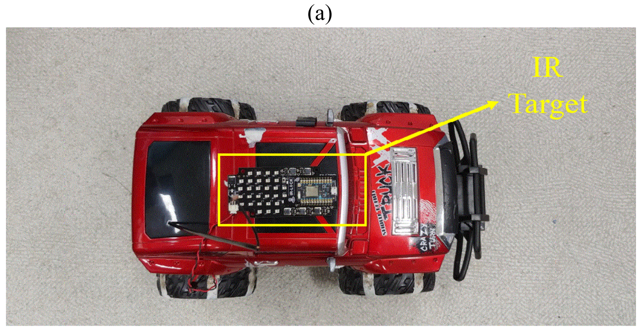
FIGURE 13. **Simulation results of the three controllers in a triangular trajectory in (a) X position, (b) Y position, and (c) XY plane.**

the Simulink based fuzzy logic system with virtual inputs. These inputs are selected according to all expected variations. Due to the X and Y positions have different input ranges, two Fuzzy-PI controllers have been executed as given in Fig.15. Variable positions are designed to be ranged from −160 to 160 for the Y axis with 8 pixels difference, and from −100 to 100 for the X axis with 5 pixels difference with a change of position value ranges from −20 to 20 with a 2-pixel difference. The position and change of positions values are generated using MATLAB; then it's used in the SIMULINK to output the required K_{pf} *and* K_{if}.

The simulation program is executed for the generated positions inputs at each change of position point. The output of the Fuzzy-PI controller is recorded. For example, position input of quadcopter in the Y-axis is 160 pixels and change of position is provided from −20 to 20 pixels with the step size of 2 pixels, corresponding output K_{pf} *and* K_{if} values are recorded. The same process is repeated for each position point from −160 to 160 pixels in Y with 8 pixels difference, and for −100 to 100 pixels in X with 5 pixels difference. Therefore, four look-up tables has been created with 40 different positions values aligned in rows with 20 different change of positions as columns which made a look-up table of



(a)



(b)

FIGURE 14. **(a) Quadcopter experimental setup and (b) experimental target.**

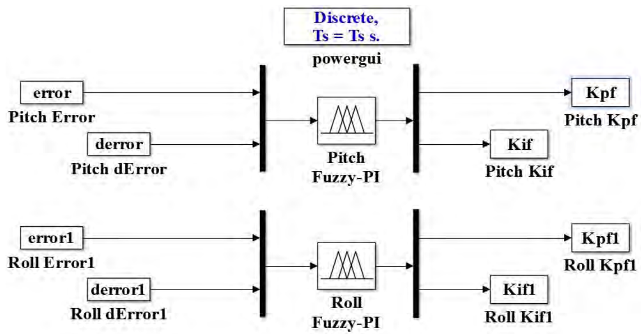


FIGURE 15. **Simulink block for generation of lookup table.**

800 values for each output. This look-up table is written inside the Arduino DUE. The Moving Target-Tracking controller measures the real-time position and change of position using the IR-camera, and then round off the data to exactly match the look-up table inputs. The rounded values are then used as input for the look-up table, which generates the corresponding K_{pf} *and* K_{if} outputs for these specific inputs. These outputs data can be multiplied by a gain value for further tuning. Then by applying Equations (2-4), the required roll and pitch angle for tracking is calculated. Finally, this data is further sent to flight controller via UART communication to update the attitude commands for tracking. These instructions are updated every 25 msec in the flight controller.

2) EXPERIMENTAL RESULTS

To verify the feasibility of the proposed Moving Target-Tracking algorithm, several experiments were performed indoor and outdoor with varying speed. Initially, the quadcopter is hovering at a fixed altitude of 2.5 meters. Once the target is detected, the IR-Camera will send the position of the detected target in pixels to the Arduino DUE every 20 msec. The received data is used as input to the Fuzzy-PI controller to calculate the required angle for tracking the moving target. This process is executed every 22 msec inside the Arduino DUE.

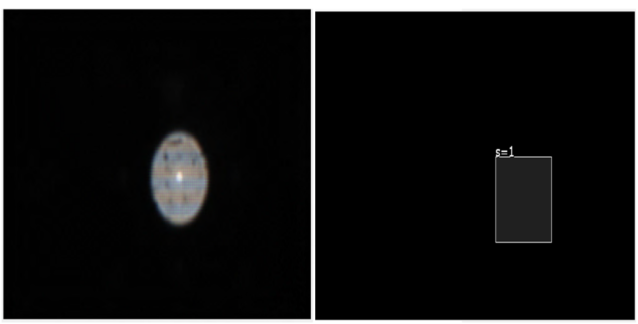
For calculating the absolute position of the quadcopter in meters, a GPS can be used. However, since the GPS provides inaccurate localization in indoor environments. A vision-based algorithm is used to obtain the absolute position between the detected target and the drone. The algorithm uses the pinhole camera model and the captured images to build a metric map. The coordinate of a point in 3D (X, Y, Z) can be computed from its projection pixel in the 2D (x, y) image and the projection matrix. Firstly, a reference object is used to measure the pixels per metric ratio. In the algorithm, a MarkOne beacon with a known width and height is used as a reference object. An image of the reference object is captured from a defined height of 2.5 meters using the IR Camera equipped on the drone. Then, the reference object is detected in the image, either based on the placement of the object or via appearances, such as its color or shape. In our case, a distinctive color is used to locate the object in the image and a contour technique is used to define the boundaries of the object, as shown in Fig.16. Thus, the width and height of the reference object in pixels are obtained. The pixels per metric can be calculated as follow:

$$\text{p} = \frac{h}{w} \tag{9}$$

where p denotes the pixels per metric ratio, h and w are the reference object width in pixels and metric, respectively. The distance between the drone and the detected target is computed as shown in Equation (10, 11):

$$\hat{x} = \frac{A1 * x + B1}{\left(C\,(x-100)^2 + D\,(y-160)^2 - 1\right)} \tag{10}$$

$$\hat{y} = \frac{A2 * y + B2}{\left(C\,(x-100)^2 + D\,(y-160)^2 - 1\right)} \tag{11}$$



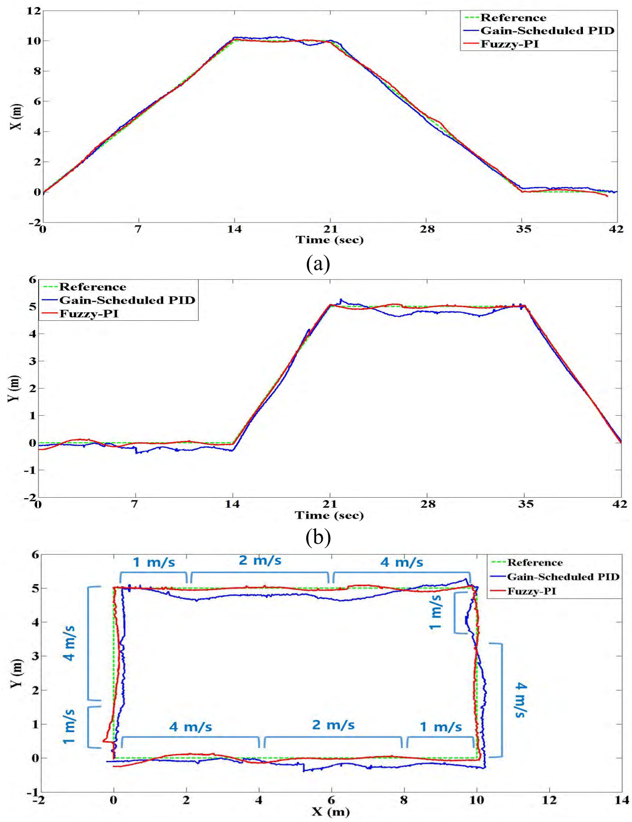
(a) (b)

FIGURE 16. **(a) Raw image captured using the IR-Camera and (b) the output of the applied Contour technique used on the captured imaged.**

TABLE 3. **Identified parameters values.**

Identified parameters	Value
A1	-0.00294
B1	0.47020
A2	-0.00305
B2	0.30568
C	4.4301e-6
D	4.7933e-6

Where $\hat{x}$ *and* $\hat{y}$ are the distance in meter, x *and* y are the distance in pixels. As shown in the previous equations, each equation has four unknowns, and therefore four known points are used to generate four equations for each axis to identifies these unknowns. Table. 3 shows the identified parameters that are obtained using regular simultaneous equation solving methods, such as substitution and elimination.



(a)

(b)

(c)

FIGURE 17. **Experimental results of the two controllers in a rectangular trajectory indoor in (a) X position, (b) Y position, and (c) XY plane.**

The experimental target is moving in two trajectories, a rectangular trajectory, and a triangular trajectory. The trials were executed indoor and outdoor. Figures (17-20) shows the results of the performed trials. Figure 17 compares between the Gain-Scheduled PID controller and the Fuzzy-PI controller under variable target speed indoor in a rectangular trajectory. Figure (17-a) shows that both controllers have

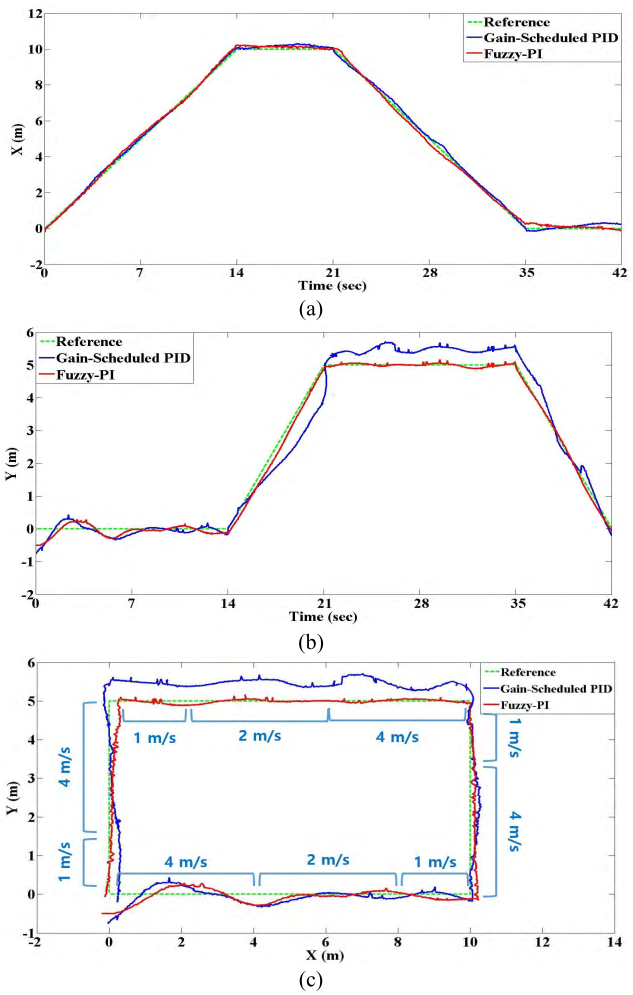


FIGURE 18. **Experimental results of the two controllers in a rectangular trajectory outdoor in (a) X position, (b) Y position, and (c) XY plane.**

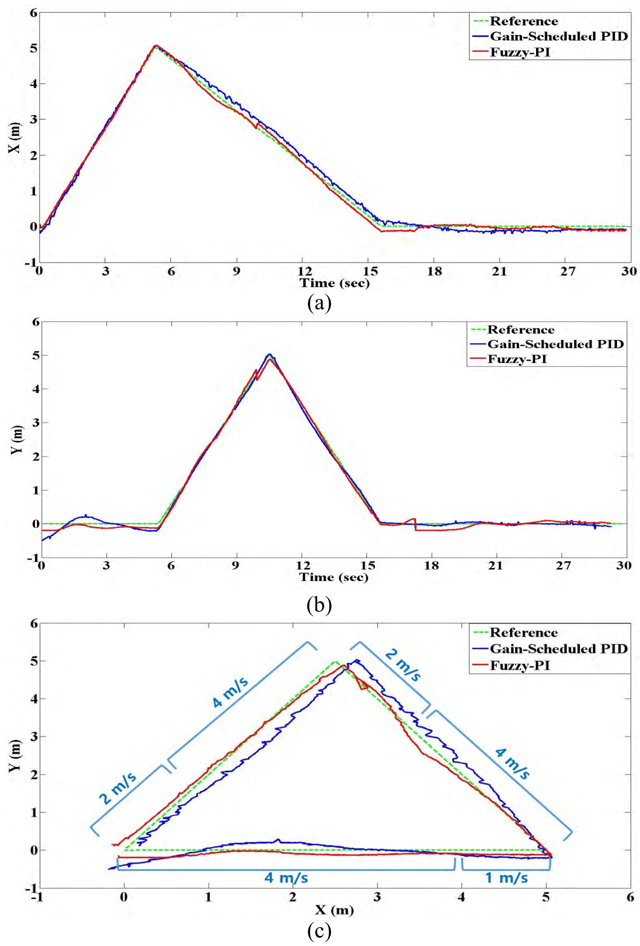


FIGURE 19. **Experimental results of the two controllers in a triangular trajectory indoor in (a) X position, (b) Y position, and (c) XY plane.**

close results in the X position. However, in Fig. (17-b), Fuzzy-PI shows a better response than the other controller in the Y position. Figure (17-c) shows the behavior of the quadcopter in the XY plane under varying speed. As shown in this figure, Fuzzy-PI has better response and faster settling time than Gain-Scheduled PID controller.

Figure 18 evaluates the performance of the Gain-Scheduled PID controller and the Fuzzy-PI controller under different target speed outdoor in a rectangular trajectory. Figure (18-a) shows that both controllers have close results in X position, while in Fig. (18-b), Gain-Scheduled PID controller depicts feeble response and stability in the Y position. Figure (18-c) illustrates the performance of the quadcopter in the XY plane under varying speed. As it is easily seen in the Figure, the Gain-Scheduled PID has poor performance, poor stability and failed to reach the center of the detected target.

Figure 19 shows comparison results between the two controllers under varying target speed indoor in a triangular trajectory. Figure (19-a & 19-b) shows that both controllers have close results in X and Y positions. However, Figure (19-c) indicates that the Fuzzy-PI controller has better response and faster settling time compared to the Gain-Scheduled PID controller in the XY plane under varying speed.

Figure 20 evaluates the performance of the Gain-Scheduled PID controller and the Fuzzy-PI controller under variable target speed outdoor in a triangular trajectory. Figure (20-a & 20-b) shows that the Gain-Scheduled PID controller has poor stability and longer settling time than the Fuzzy-PI controller in X and Y positions. Figure (20-c) demonstrates that under variable speed, Fuzzy-PI has better response and faster settling time than the Gain-Scheduled PID controller.

According to the previous evalutioan of the experimental results, the Fuzzy-PI controller has proven that it has better response, and shorter settling time compared to the Gain-Scheduled PID controller. Although the Gain-Scheduled PID controller is working well compared to the typical PID controller, since the proportional gain is continuously changed according to the position and change of position of the detected target, the output of the Gain-Scheduled PID is either increasing or decreasing linearly, which is the reason of the slow response of this controller. In contrast, the Fuzzy-PI controller has proved it has better control because according

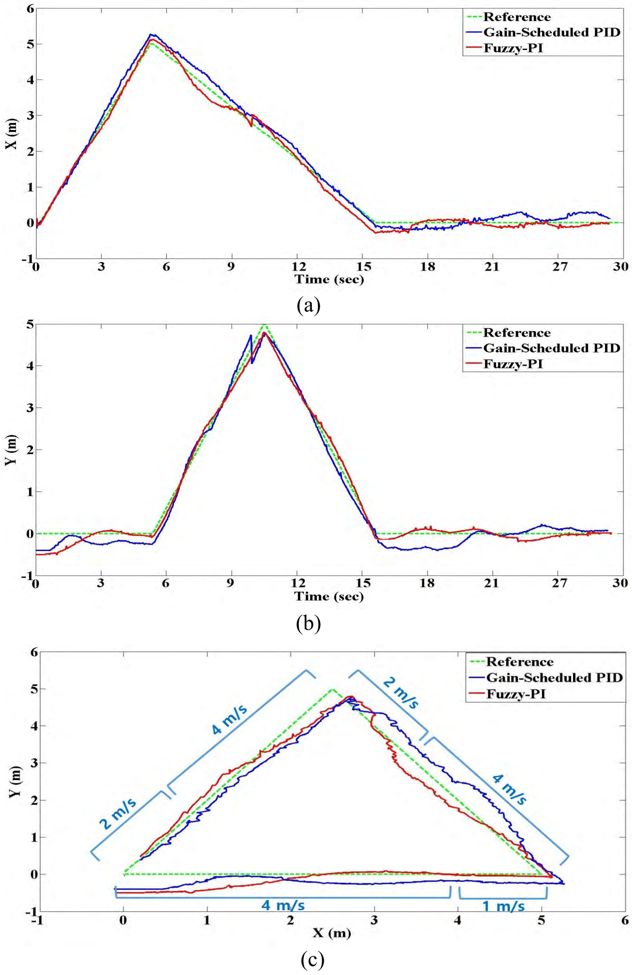


FIGURE 20. Experimental results of the two controllers in a triangular trajectory outdoor in (a) X position, (b) Y position, and (c) XY plane.

to the position and change of position values, we can control how much we need to increase or decrease the output of the Fuzzy-PI controller. Thus, it can rapidly minimize the error between the position and the center of the detected target as much as possible, where it gives us more accurate output to adjust the final *P and I* gain of the PID controller. As a result, the Fuzzy-PI can be thought of an auto-tuning method for PID controller.

VI. CONCLUSION

In this paper, a Moving Target-Tracking algorithm has been proposed. Target-Tracking algorithm based on Fuzzy-PI controller is developed to track a moving target under varying speed and during day and night. Furthermore, the relatively cheap embedded controller was used for real-time applications such as target tracking. The proposed algorithm can be used in many applications such as traffic supervision, autonomous robot navigation, and landing to a moving target. Several experiments are performed indoor and outdoor. The obtained results show that the proposed system works well indoor and outdoor for tracking the moving target. Furthermore, this system is currently being improved to land on a moving target such as a car and also an unstable target such as a ship during the night. Moreover, it can be further improved by using a deep learning algorithm such as CNN (Convolutional Neural Network) and train it to detect a target during day and night then apply the proposed algorithm for tracking and landing.

REFERENCES

[1] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proc. 23rd Digit. Avionics Syst. Conf.*, vol. 2, Oct. 2004, pp. 12.E.4–121-10.

[2] *ICAO's Circular 328 AN/190: Unmanned Aircraft Systems*, ICAO, Montreal, QC, Canada, Feb. 2016.

[3] J. Stafford, "How a Quadcopter works | Clay Allen," Univ. Alaska Fairbanks, Jan. 2015.

[4] M. Bhaskaranand and J. D. Gibson, "Low-complexity video encoding for UAV reconnaissance and surveillance," in *Proc. IEEE Military Commun. Conf.*, Nov. 2011, pp. 1633–1638.

[5] T. Tomic *et al.*, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[6] L. Merino, F. Cabalero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *J. Field Robot.*, vol. 23, nos. 3–4, pp. 165–184, 2006.

[7] J. Liu, C. Wu, Z. Wang, and L. Wu, "Reliable filter design for sensor networks using type-2 fuzzy framework," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1742–1752, Aug. 2017.

[8] M. Talha, F. Asghar, A. Rohan, M. Rabah, and S. H. Kim, "Fuzzy logic-based robust and autonomous safe landing for UAV quadcopter," *Arabian J. Sci. Eng.*, vol. 44, no. 3, pp. 2627–2639, Mar. 2019.

[9] P. Doherty and P. Rudol, "A UAV search and rescue scenario with human body detection and geolocalization," in *AI 2007: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2007, pp. 1–13.

[10] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a Quadrotor MAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2182–2189.

[11] H. Huang, L. Zhang, C. Fu, Y. Zhou, and Y. Tian, "Ground moving target tracking algorithm for multi-rotor unmanned aerial vehicle," in *Proc. IEEE Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 1214–1219.

[12] R. P. George and V. Prakash, "Real-time human detection and tracking using quadcopter," in *Intelligent Embedded Systems*, vol. 492. Singapore: Springer, 2018, pp. 301–312.

[13] C. Pei, J. Zhang, X. Wang, and Q. Zhang, "Research of a non-linearity control algorithm for UAV target tracking based on fuzzy logic systems," *Microsyst. Technol.*, vol. 24, no. 5, pp. 2237–2252, 2018.

[14] A. Joukhadar, M. AlChehabi, C. Stöger, and A. Müller, "Trajectory Tracking Control of a Quadcopter UAV Using Nonlinear Control," in *Mechanism, Machine, Robotics and Mechatronics Sciences*, vol. 58. Cham, Switzerland: Springer, 2018, pp. 271–285.

[15] H. Bolandi, H. Nemati, R. Mohsenipour, and M. Rezaei, "Attitude control of a quadrotor with optimized PID controller," *Intell. Control Automat.*, vol. 8, no. 3, pp. 335–342, 2013.

[16] D. K. Tiep and Y. J. Ryoo, "An autonomous control of fuzzy-PD controller for quadcopter," *Int. J. Fuzzy Logic Intell. Syst.*, vol. 17, no. 2, pp. 107–113, 2017.

[17] M. Islam, M. Okasha, and M. M. Idres, "Trajectory tracking in quadrotor platform by using PD controller and LQR control approach," *IOP Conf., Mater. Sci. Eng.*, vol. 260, no. 1, pp. 2451–2456, 2017.

[18] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.*, vol. 4, no. 11, pp. 2343–2355, Nov. 2010.

[19] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "Real-time object tracking on a drone with multi-inertial sensing data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 131–139, Jan. 2018.

[20] A. Samir, A. Hammad, A. Hafez, and H. Mansour, "Quadcopter trajectory tracking control using state-feedback control with integral action," *Int. J. Comput. Appl.*, vol. 168, no. 9, p. 8887, 2017.

[21] M. Rabah, A. Rohan, Y.-H. Han, and S. H. Kim, "Design of fuzzy-PID controller for quadcopter trajectory-tracking," *Int. J. Fuzzy Logic Intell. Syst.*, vol. 18, no. 3, pp. 204–213, 2018.
[22] M. Rabah, A. Rohan, M. Talha, K. H. Nam, and S. H. Kim, "Autonomous vision-based target detection and safe landing for UAV," *Int. J. Control, Autom. Syst.*, vol. 16, no. 6, pp. 3013–3025, Dec. 2018.
[23] H. Wang, B. Chen, X. Liu, K. Liu, and C. Lin, "Robust adaptive fuzzy tracking control for pure-feedback stochastic nonlinear systems with input constraints," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2093–2104, Dec. 2013.
[24] H. Wang, P. X. Liu, and B. Niu, "Robust fuzzy adaptive tracking control for nonaffine stochastic nonlinear switching systems," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2462–2471, Aug. 2018.
[25] Y. Yin, G. Zong, and X. Zhao, "Improved stability criteria for switched positive linear systems with average dwell time switching," *J. Franklin Inst.*, vol. 354, no. 8, pp. 3472–3484, May 2017.
[26] V. Veselý and I. Adrian, "Gain-scheduled PID controller design," *J. Process Control*, vol. 23, no. 8, pp. 1141–1148, 2013.
[27] A. Rohan, F. Asghar, and S. H. Kim, "Design of fuzzy logic tuned PID controller for electric vehicle based on IPMSM using flux-weakening," *J. Elect. Eng. Technol.*, vol. 13, no. 1, pp. 451–459, 2018.
[28] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, Sep. 1992.

MOHAMMED RABAH received the B.S. degree in electronics and telecommunication engineering from the AL-SAFWA High Institute of Engineering, Cairo, Egypt, in 2015, and the M.S. degree in electronics and information engineering from Kunsan National University, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include UAV's, fuzzy logic systems, and machine learning.

ALI ROHAN received the B.S. degree in electrical engineering from The University of Faisalabad, Pakistan, in 2012, and the M.S. degree in electronics and information engineering from Kunsan National University, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include renewable energy systems, power electronics, fuzzy logic, neural networks, EV systems, and flywheel energy storage systems.

SHERIF A. S. MOHAMED received the B.S. degree in electrical, electronics, and communication engineering from Ain Shams University, Egypt, in 2011, and the M.S. degree in electronics and information engineering from Kunsan National University, South Korea, in 2016. He is currently pursuing the Ph.D. degree with the University of Turku, Finland. His research interests include vision-based navigation algorithms for autonomous vehicles, embedded systems, swam intelligence, and machine learning.

SUNG-HO KIM received the B.S. degree in electrical engineering from Korea University, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from Korea University, in 1986 and 1991, respectively. He has completed the Postdoctoral Research from Japan Hiroshima University, in 1996. He is currently a Professor with Kunsan National University. His research interests include fuzzy logic, sensor networks, neural networks, intelligent control systems, renewable energy systems, fault diagnosis systems.