AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 55(2) (2023) 243-262
DOI: 10.22060/miscj.2023.22575.5330

Path Planning and Tracking for a Quadrotor with the Aim of Obstacle Avoidance Using the BUG2 and Predictive Control

Shahriar Ansari, Saeed Nasrollahi *

Faculty of Electrical & Computer Engineering, Malek Ashtar University of Technology, Iran

ABSTRACT: One of the main topics these days is the use of quadrotors to transport commodities in the urban environment, and the main challenge in quadrotors for route planning in urban areas is obstacles. Quadrotors are not suitable to fly at high altitudes, because due to economic limitations, this will be a challenge. Therefore, most quadrotors in the urban environment must fly at a low altitude, and for this reason, there will be obstacles in their path. Some of the obstacles are predetermined, and some others are unpredictable. A new method to interact with these unexpected obstacles has been presented In this paper. This method combines the BUG2 or online-BUG2 path planning methods in robotics and model predictive control, which is intended to guide the quadrotor. In this method, first, the desired path of the quadrotor is determined with the help of the BUG2 and online-BUG2 algorithms, and then, with the help of model predictive control using the predictive functional control, the control law required to change the direction of the quadrotor to this reference path is obtained. According to the three scenarios implemented with the help of the introduced method, it can be seen the integrated approach has been able to detect them well and guide the quadrotor to the target by bypassing the obstacles.

Review History:

Received: Jul. 23, 2023
Revised: Oct. 17, 2023
Accepted: Nov. 11, 2023
Available Online: May, 30, 2024

Keywords:

Path Planning
BUG2
Online-BUG2
Obstacle Avoidance
Predictive Functional Control
Quad Rotor

1- Introduction

In today's era, many studies have been done on autonomous cars, drones, robots, etc., and some have been exploited in industry. One of the main challenges for self-driving these types of devices is the lack of complete knowledge of the path and obstacles on their route because even with previous predictions, an unpredictable obstacle such as other vehicles, drones, or animals may be placed in the way of these devices, and the art of self-driving designers is to deal with these type of unpredictable obstacles.

One of great interest nowadays is the economic and civilian use of quadrotors so that packages can be delivered to the home or the desired person. As mentioned before, there is a possibility of the existence of undetermined obstacles for which a solution should be thought of.

Determining the path to avoid obstacles in unmanned aerial vehicles is mainly done by assuming knowledge of all the obstacles. By considering the location of the obstacles, the best possible route to reach the destination has been obtained. For example, in [1],[BinKai, 2021 #42] the method of finding the optimal path to avoid obstacles is done with the help of the improved artificial potential field method. In [2], the optimization of the optimal route to avoid obstacles has been done with the help of an ant colony algorithm. In [3], finding the optimal route using the improved rapid exploration random tree algorithm is explained, which, such as the previous methods, requires a map of the location of obstacles. Also, in [4], an approach has been introduced in which the optimal route is determined with the help of a set of images, and then the drone corrects its course by taking pictures of places during the flight.

In all the mentioned cases and even more that were not investigated, the approaches have been determined based on the knowledge of the location of the obstacles. In other words, the obstacles in the path were determined in advance, and then the drones tried to find the optimal way to reach the goal. However, the introduced methods for the case where there are no knowledge barriers are primarily done in robotics and two-dimensional space and less about aerial issues. Among these cases, it may be possible to refer to the study done in [5] that the initial path was made with the help of the ant's algorithm assuming predetermined obstacles for the unmanned surface vehicle and the artificial potential field algorithm was used to avoid the predetermined obstacles. One of the new methods proposed to prevent obstacles in recent years is the free configuration Eigenspace method, which requires the presence of a sensor on the device. This method was implemented in [6] for a mobile robot. In [7], a new approach for finding the path to reach the goal by avoiding obstacles called M-BUG has been introduced. In this approach, when encountering obstacles, by considering the slope of the obstacles at the moment of reaching them and

*Corresponding author's email: nasrollahi.saeid@gmail.com

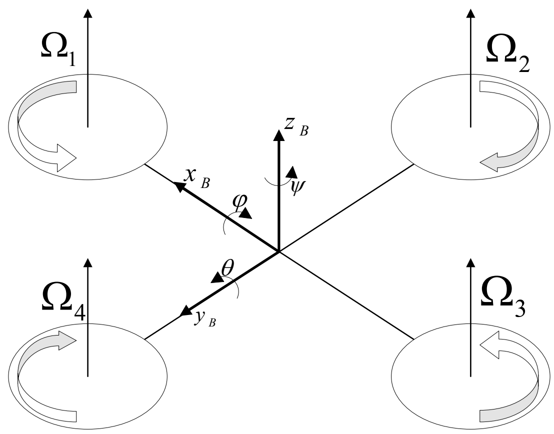

Fig. 1. Configuration of quadrotor

choosing the shortest path, it tries to bypass the obstacle. In [8], the BUG1 method and the robot's presence are used in a multi-agent way to avoid the obstacle. In this method, two robots help each other in bypassing obstacles with the help of the BUG1 method and by sharing information.

Apart from the issue of finding the path, finding a control law to follow the determined path is also one of the challenges of this problem. Many control methods have been proposed to follow the path of the quadrotor. For example, in [9, 10], the sliding mode method controls the quadrotor to be in a specific direction. In [11], sliding mode combined with a neural network has been used to control the quadrotor. In meta-heuristics topics, we can refer to [12], which uses the particle swarm optimization method to control the quadrotor so that it is placed in a specific direction. In the field of multivariable control, we can refer to [13], which uses a MIMO-PID controller to find a suitable control law. Also [14], is an example of using the fuzzy law to control the quadrotor.

But among the available control methods, model predictive control is more suitable for this problem, because the future path of the quadrotor is determined by the guidance law, and the future steps of the quadrotor to move in it are known. Therefore, according to the nature of the problem, predictive control is considered to find the control law. In recent years, predictive control has been widely used for quadrotor control, and its effectiveness has been proven. In [15], model predictive control is used to track the quadrotor path with the help of linearizer feedback. In [16], the combination of nonlinear estimation methods and nonlinear model predictive control has been used to track the path of an unmanned aerial vehicle in the presence of its four actuator faults. Also, [17] has introduced a model predictive control method to track the path in the fact of control input limitations and disturbance.

In this paper, finding the path to reach the goal by avoiding obstacles will be discussed using the BUG2 and online-BUG2 algorithms, and the control law used to apply the input to track the path will be the model predictive control law using the predictive Functional Control (PFC). The innovative aspects of this article can be expressed in the combination of the PFC method with the route tracking obtained through the obstacle avoidance path planning algorithms called BUG2 and online-BUG2 so that the presence and absence of obstacles are unpredictable. The introduced algorithm has been investigated in three different scenarios, in which the height of the starting and ending points are assumed to be the same. In these scenarios, the goal is to move along the horizon and maintain altitude by the quadrotor. The map examined in each of these scenarios is different from the others.

This paper is prepared in 5 parts, the second part examines the mathematical equations and the structure of the used methods such as BUG2, online-BUG2, and PFC, along with their integration. Of course, in this chapter, the assumed quadrotor sample is also introduced, and its parameters, equations, and structure are described. The third part of this article presents three scenarios that are considered to check the performance of the presented approach. The fourth part examines the results obtained from each scenario, and finally, the last part deals with the conclusion and suggestions for future studies.

2- Mathematical equations

2- 1- System model

A quadrotor consists of four motors, and all kinds of motion modes are based on these motors. Fig1 shows a quadrotor with a body coordinate system. The quadrotor moves upwards If all engines work in the indicated direction. If the second and fourth engines work in opposite directions, the quadrotor will revolve around the x-axis. If the first and third engines work in opposite directions, the quad will revolve around the y-axis. Also, for the rotation of the quadrotor around the z-axis, the first and third engines must work in the same direction while the second and fourth engines work in the opposite direction.

Based on the stated relationships, and the relationship between the torque and the angular speed of the motors, the control inputs of the quadrotor can be expressed as follows [18].

$$U_1 = k_f\left(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2\right) \quad (1)$$

$$U_2 = k_f\left(-\Omega_2^2+\Omega_4^2\right) \quad (2)$$

$$U_3 = k_f\left(\Omega_1^2-\Omega_3^2\right) \quad (3)$$

$$U_4 = k_f\left(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2\right) \quad (4)$$

Where k_f is a constant coefficient equal to the product of the thrust force coefficient along the length of the arm. Ω_1, Ω_2, Ω_3, and Ω_4 are the angular velocities of the first, second, third, and fourth motor, respectively, which is shown in the Fig1 in the direction of positive movement of each of them.

Ω_r is the residual angular velocity of the propellers and can be calculated as (5) [18]:

$$\Omega_r = -\Omega_1+\Omega_2-\Omega_3+\Omega_4 \quad (5)$$

The dynamic equations of the quadrotor are obtained with the help of Euler-Lagrange formulation based on potential energy and kinematics, which will include 12 state strain equations. By considering states as Eq.(6) [19]:

$$X = \begin{bmatrix}\varphi & \dot{\varphi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} & x & \dot{x} & y & \dot{y}\end{bmatrix}^T \quad (6)$$

State-space equations will be as[19]:

$$\dot{X} = f(X,U) = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta}\dot{\psi}a_1+\dot{\theta}a_2\Omega_r+b_1U_2 \\ \dot{\theta} \\ \dot{\varphi}\dot{\psi}a_3+\dot{\varphi}a_4\Omega_r+b_2U_3 \\ \dot{\psi} \\ \dot{\theta}\dot{\varphi}a_5+b_3U_4 \\ \dot{z} \\ g-\dfrac{u_zU_1}{m} \\ \dot{x} \\ \dfrac{u_xU_1}{m} \\ \dot{y} \\ \dfrac{u_yU_1}{m} \end{bmatrix} \quad (7)$$

Where φ, θ, and ψ are the Euler angles of roll, pitch, and yaw and x, y, and z indicate displacement along the quadrotor's body axes, and u_x, u_y, u_z are the direction cosines of the total thrust vector along the x_B, y_B, z_B axes, respectively[19]:

$$\begin{cases} u_x = \cos\varphi\sin\theta\cos\psi+\sin\varphi\sin\psi \\ u_y = \cos\varphi\sin\theta\sin\psi-\sin\varphi\cos\psi \\ u_z = \cos\varphi\cos\theta \end{cases} \quad (8)$$

According to the dynamics of the issue, the angle of the quadrotor should not exceed 80 degrees, because the quadrotor will lose its stability and will not be controllable anymore. constant Coefficients are[20]:

$$\begin{aligned} a_1 &= \left(I_{yy}-I_{zz}\right)/I_{xx} \\ a_2 &= J_r/I_{xx} \\ a_3 &= \left(I_{zz}-I_{xx}\right)/I_{yy} \\ a_4 &= J_r/I_{yy} \\ a_5 &= \left(I_{xx}-I_{yy}\right)/I_{zz} \\ b_1 &= l/I_{xx} \\ b_2 &= l/I_{yy} \\ b_3 &= 1/I_{zz} \end{aligned} \quad (9)$$

Quadrotor specifications are as shown in Table 1 [20]:

2- 2- BUG2 and online-BUG2 algorithms

One of the available and valuable approaches in robotics for path planning in case of obstacle existence, and a lack of prior knowledge of the road map is the BUG2 algorithm. BUG algorithms are among the earliest and simplest sensor-

Table 1. Quadrotor specifications

Name	Parameter	Value	Unit
mass	m	0.650	kg
inertia on x-axis	I_{xx}	7.5e-3	kg.m^2
inertia on y-axis	I_{yy}	7.5e-3	kg.m^2
inertia on z-axis	I_{zz}	1.3e-2	kg.m^2
thrust coefficient	k_f	3.13e-5	N.s^2
drag coefficient	d	7.5e-7	N.m.s^2
propeller radius	R_{rad}	0.15	m
propeller chord	c	0.04	m
pitch of incidence	θ_0	0.26	rad
twist pitch	$\theta_{t\omega}$	0.045	rad
rotor inertia	J_r	6e-5	kg.m^2
arm length		0.23	m

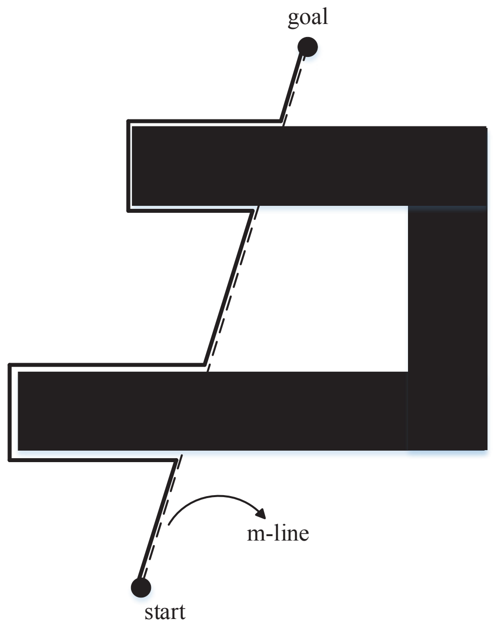

Fig. 2. Moving the particle from the start to the goal using the BUG algorithm

based planners with provable guarantees. These algorithms assume the robot is a point operating in the plane with a contact sensor or a zero-range sensor to detect obstacles.

We name the line from the start point to the goal as m-line, which is the line that determines the robot's initial direction. The steps of this algorithm can be represented as follows [21, 22].

- while not at the goal
  Follow the m-line
  while the path to the goal is blocked
  follow obstacle edges
  if the m-line met, then break
  end while
- end while

Fig2 explains an example of robot movement using the BUG2 algorithm. In this scenario, the robot moves from the starting point towards the target along the m-line until it reaches the obstacle. At this moment, the robot starts to turn around the obstacle until it is located along the same

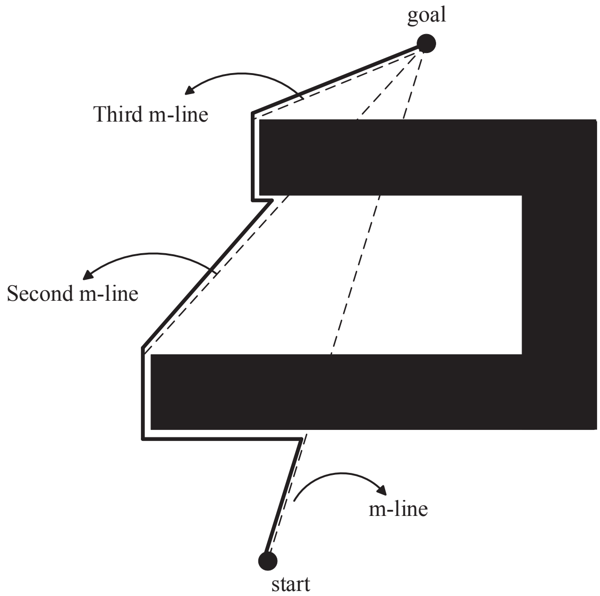

Fig. 3. Moving the particle from the start to the goal using the BUG algorithm

m-line. At this moment, it returns to its path toward the target along the m-line until it reaches the obstacle again. The robot repeats the previous process and finally reaches the goal.

This algorithm is derived for 2D path planning problems. For our problem, we will assume the quadrotor will control itself in the same attitude, so it can be simplified as a two-dimensional problem.

The online-BUG2 algorithm works such as BUG2, with one difference. BUG2 algorithm has one m-line, but in online-BUG2, the m-line will update in each movement. The algorithm of this method can be stated in superficial as follows.

- while not at the goal
    Follow the m-line
    while the path to the goal is blocked
        follow obstacle edges
        update m-line
    end while
- end while

The movement path shown in the previous example will be changed in Fig. 3 with the help of this algorithm. As seen in this example, with the help of the online-BUG2 method, the distance traveled by the robot has been reduced.

2- 3- PFC Control

In this section, the PFC controller design process will be explained. If the state-space equations of the system are considered as follows[23]:

$$\dot{X} = AX + BU \tag{10}$$

$$Y = CX + D \tag{11}$$

It responds to [23]

$$X_m(k+1) = A_m X_m(k) + B_m U \tag{12}$$

m shows the number of parameters involved in the control law; that is, the influence of m control samples will be considered in predicting the future behavior of the system. Assuming the system is complete, the output will be as (13).

$$y_m(k) = C_m X_m(k) \tag{13}$$

If two consecutive sentences are subtracted from the state space, [23]:

$$\begin{aligned} X_m(k+1) - X_m(k) = \\ A_m\left(X_m(k) - X_m(k-1)\right) + \\ B_m\left(u_m(k) - u_m(k-1)\right) \end{aligned} \tag{14}$$

$$y_m(k+1) - y_m(k) = C_m\left(X_m(k+1) - X_m(k)\right) \tag{15}$$

Or, to put it more succinctly,[23]:

$$\Delta X_m(k+1) = A_m \Delta X_m(k) + B_m \Delta u_m(k) \quad (16)$$

$$\Delta y_m(k+1) = C_m A_m \Delta X_m(k) + C_m B_m \Delta u_m(k) \quad (17)$$

New equations define in (18) and (19) [23]:

$$\begin{bmatrix} \Delta X_m(k+1) \\ Y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta X_m(k) \\ Y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (18)$$

$$y(k) = \begin{bmatrix} 0_m & 1 \end{bmatrix} \begin{bmatrix} \Delta X_m(k) \\ y(k) \end{bmatrix} \quad (19)$$

By writing equations for future predictions, the final result can be represented under the matrix Eq. (20).

$$Y = Fx(k) + \Phi U \quad (20)$$

in Eq(20)[23]:

$$Y = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p) \end{bmatrix} \quad (21)$$

$$U = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m-1) \end{bmatrix} \quad (22)$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \quad (23)$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & CA^{p-3}B & \cdots & CA^{p-m}B \end{bmatrix} \quad (24)$$

In the above relationships, m is the control horizon, p is the prediction horizon, and the condition must be met.

Now, if the cost function is defined as (25) where R and Q are the coefficients related to the importance of the size of the control law and tracking the output path [23].

$$J = \left(R_s - Y\right)^T Q \left(R_s - Y\right) + \Delta U^T R \Delta U \quad (25)$$

In this case, by minimizing this cost function, the optimal control law will be obtained as (26) [23].

$$\frac{\partial J}{\partial \Delta U} = 0 \Rightarrow \Delta U = \left(\Phi^T \Phi + R\right)^{-1} \Phi^T Q \left(R_s - Fx(k)\right) \quad (26)$$

As seen in section 2-1, the equations of the quadrotor were nonlinear, but the PFC method is linear, so it is necessary to perform linearization for all the points where the quadrotor is placed, according to the state of all parameters of the existing state space, and the law A control corresponding to that point should be calculated which is done in the implementation stage of this process.

2- 4- Proposed algorithm

One of the most critical tools needed to implement the proposed algorithm is the distance sensor. The need to have a distance to the obstacle is the main reason for suggesting the distance sensor instead of the proximity sensor because the proximity sensor only announces the presence of the obstacle and does not calculate the distance to the obstacle.

On the other hand, among the types of distance-measuring sensors such as ultrasonic sensors, infrared sensors, or Light Detection and Ranging (LIDAR) sensors, due to the need for more than one sensor and the higher cost of LIDAR sensors than other sensors, this type of sensor is not recommended for this type of economic use. Also, Ultrasonic sensors are dependent on the shape of the obstacle, and there may be an error in calculating the distance despite the protrusion of the obstacle. Therefore, infrared sensors are recommended for use in this type of use.

In this method, the sensor range is assumed to be the same as the prediction horizon (which is responsible for detecting the presence of an obstacle here). Therefore, the sensor detects the obstacles based on the detectable distance in front of it. In this way, the relevant location for the next step is provided to the quadrotor. Then, with the help of its model predictive controller, the quadrotor applies the appropriate control law to reach this location.

In a more general case, the task of determining the quadrotor guidance law is determined by the BUG2 algorithm with the sensor horizon equal to the prediction horizon. The necessary control law to follow the guidance law is extracted by the predictive control method. At each step, the quadrotor moves towards the target, and the new location of the quadrotor is considered the new origin of the guidance law. It is evident that the controlled quadrotor has an error and will not fully implement the determined guidance law, the reasons for which can be the limitation of the bird's dynamics or the

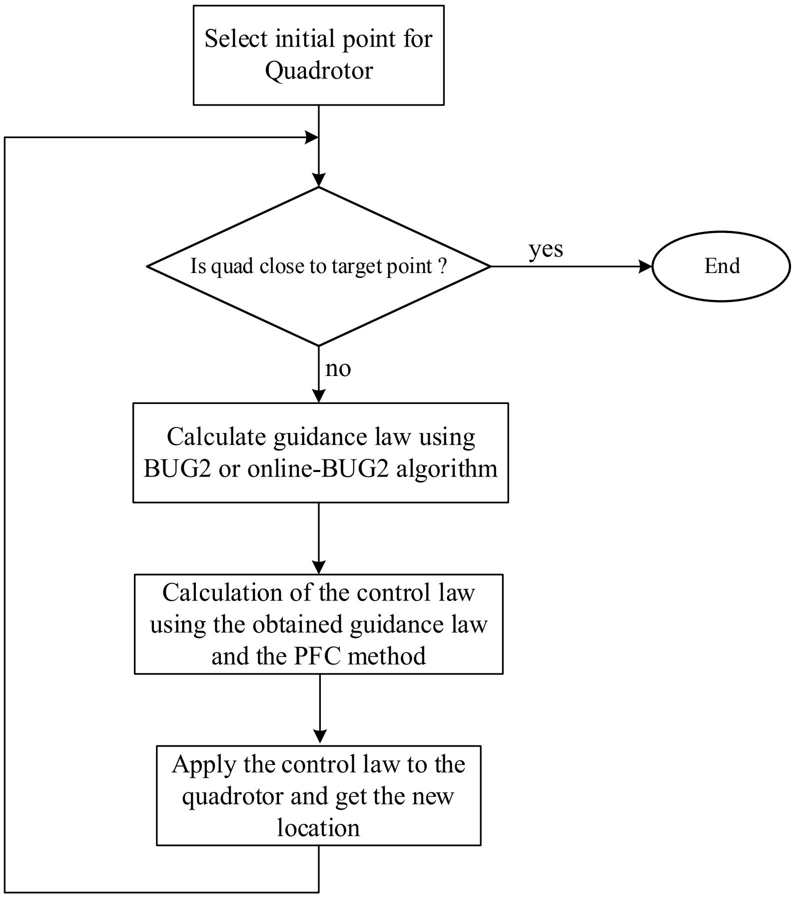

Fig. 4. Block diagram of the introduced method

presence of disturbances such as wind.

However since in each stage, the quadrotor is considered the main source, and the guidance law is generated based on it, it is expected that this error will not increase and will be acceptable for this problem. To consider the location of the quadrotor as the origin, it is evident that a GPS sensor will be needed. The block diagram related to the integration of this method can be seen in Fig 4.

3- Simulations

In the simulations, the sampling time is 0.02 seconds, the obstacle prediction distance is 0.2 meters, the prediction horizon is 20, and the control horizon is 10. All the simulations performed in MATLAB were performed with a Corei5 8250U series processor and 20.0 GB DDR 4 RAM. It should be noted that all the calculations performed, such as determining the path and determining the predictive control law, were done in less than 0.02 seconds, which indicates the real-time of the algorithm for this processor and conditions.

For a more detailed examination of three types of maps and paths in the presence of obstacles, it is considered that the desired quadrotor must continue its route from the starting point to the endpoint without encountering obstacles and reach the desired goal. These scenarios will be discussed further.

3- 1- First scenario

In the first scenario, it is assumed that the range of changes of x and y is from zero to 10, and the height also changes in the same range. Obstacles are determined in the environment of this area, and some protrusions can be seen in Fig 5.

In this scenario, we will move from the starting point of $(1,1,5)$ to the endpoint of . The detection range of the obstacle by the sensors is 0.4 meters, the sampling time is 0.02 seconds, the prediction horizon is 20, and the control horizon is 10.

3- 2- Second scenario

The conditions of the second scenario are considered similar to the first scenario, with the difference that the map of obstacles and the route has been completely changed. As shown in Fig 6, four rectangular obstacles are used in the permitted movement range. In this map, similar to the map of the first scenario, the entire spatial range is in the range of 0 to 10. The distribution of obstacles in the direction of height or z is uniform and has different shapes only in the x and y range.

In this scenario, the goal of the quadrotor is to move from the starting point $(1,1,5)$ to the target point $(9,9,5)$ so that it does not collide with obstacles.

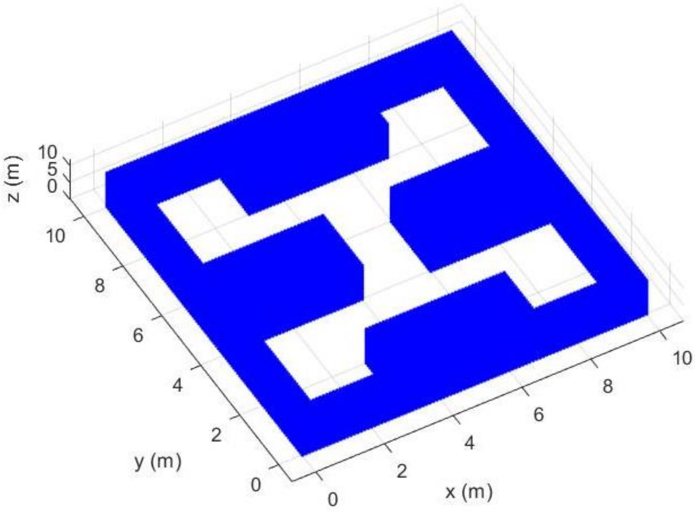

Fig. 5. The shape of the perimeter of the quadrotor view in the first scenario

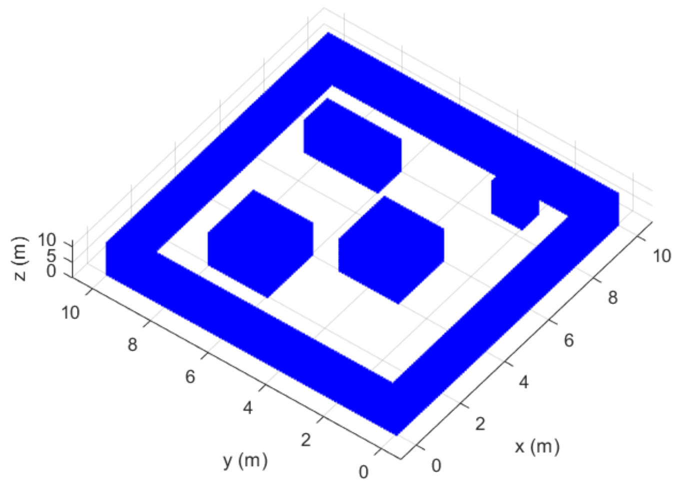

Fig. 6. The shape of the surroundings of the quadrotor in the second scenario

3- 3- Third scenario

In this scenario, the favorite paths of robotic routing methods, i.e., spiral, have been used. The starting point is assumed at coordinates (1.5, 7.5, 5), and the endpoint at coordinates (9, 2, 5). Fig 7 shows this map in three dimensions. The range of three-dimensional changes for this map is considered to be 0 to 10 meters.

4- Results

In this section, the results obtained for each of the defined scenarios are stated.

4- 1- First scenario results

If the sampling time is 0.02 seconds, the prediction horizon is 20, the control horizon is 10, and the obstacle distance by sensors is assumed to be 0.4 meters, the motion path of the quadrotor based on the dynamic equations of the quadrotor will be as shown in Fig 8.

The details related to the reference location of the quadrotor by the BUG2 algorithm and the path traveled by the quadrotor by PFC control law are shown in Fig 9. As is evident, the control method has been able to meet the requirements of the problem to follow the path.

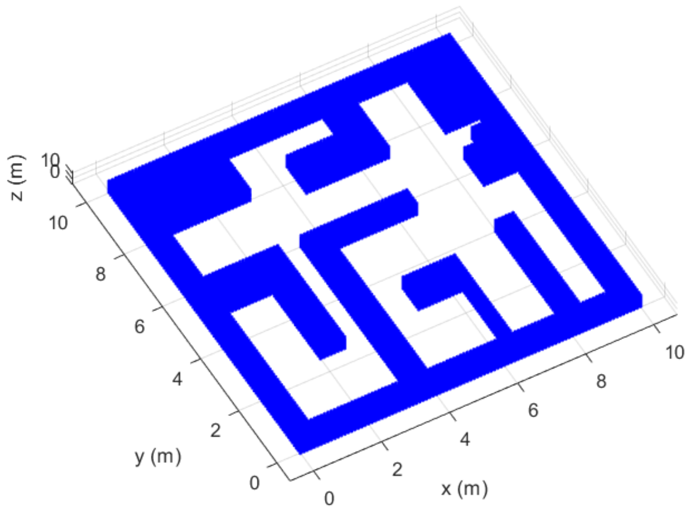

Fig. 7. The shape of the surroundings of the quadrotor in the third scenario

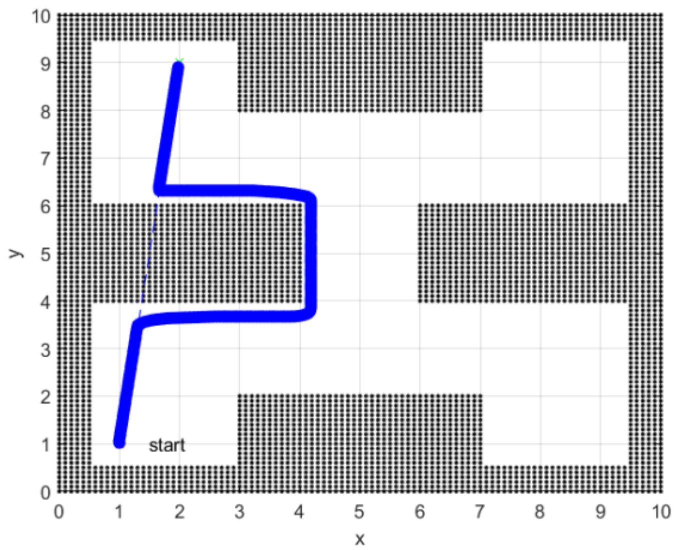

Fig. 8. The path of the quadrotor from the top view in the first scenario and BUG2 path planning

The plot of the angles of the quadrotor over time (Fig 10) shows that the quadrotor moved within the range of set angles (elevation angle less than 80 degrees) and did not change much in the angles.

Also, the result obtained for the route taken for this scenario by considering the online-BUG2 algorithm is obtained as shown in Figs11-13.

As can be seen from Figs 8-13, the angle of the quadrotor and the changes of the path in the y direction for two algorithms, the results obtained by the online-Bug2 algorithm are more acceptable than the Bug2 algorithm because both the path is shorter and the angle change rate has decreased.

4- 2- Second scenario results

By simulating the introduced approach, it can be seen that the quadrotor could move from the starting point and reach the endpoint by avoiding obstacles. The reason for not choosing a shorter route by the quadrotor is the lack of knowledge of the areas where obstacles are present. In other

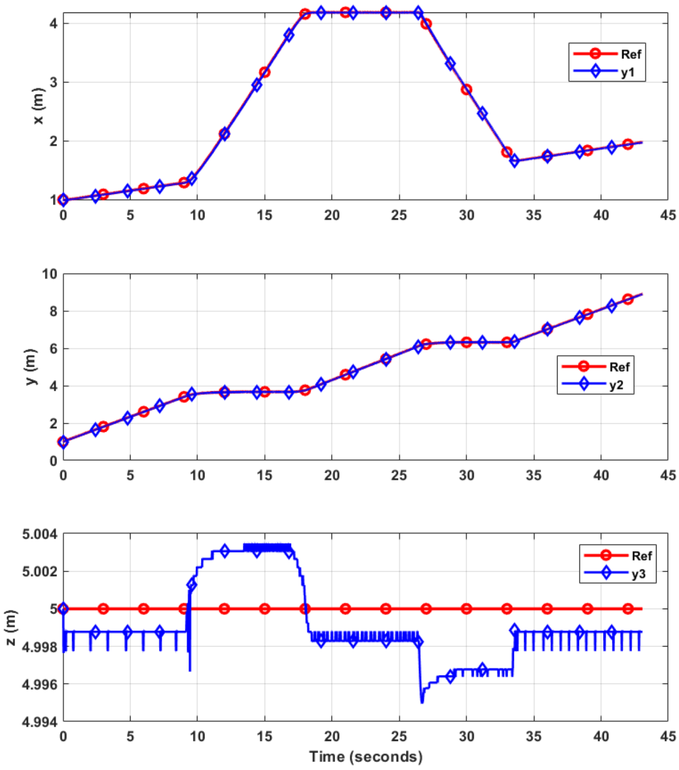

Fig. 9. The quadrotor reference path by BUG2 algorithm and the quadrotor traveled path by PFC control law in the first scenario and BUG2 path planning

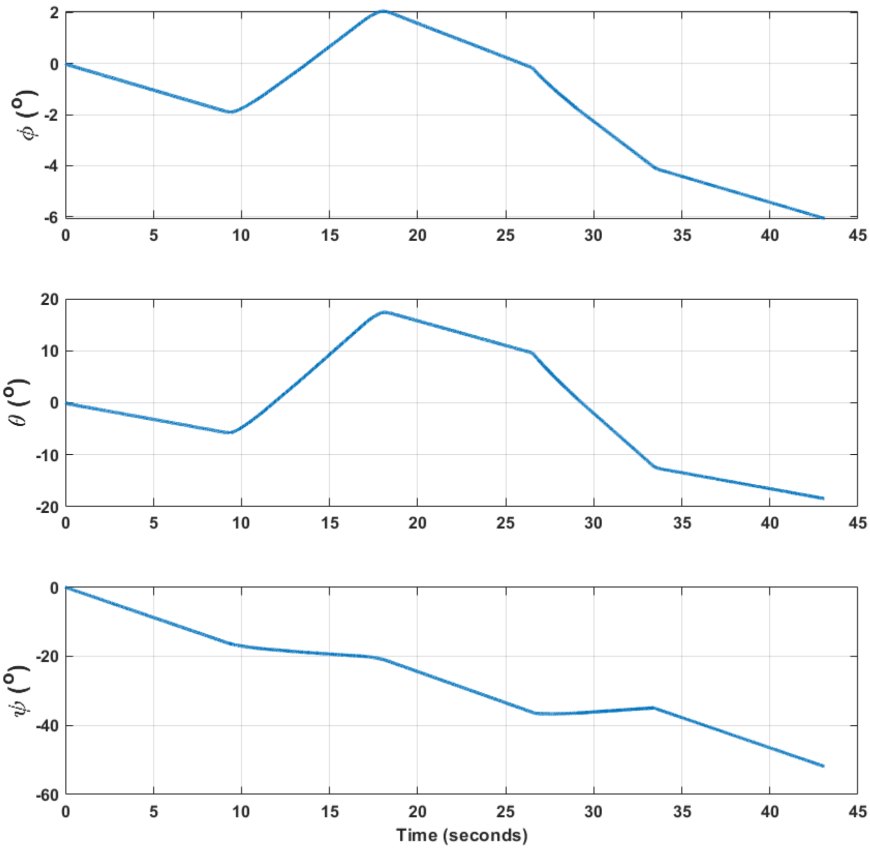

Fig. 10. Quadrotor angle over time in the first scenario and BUG2 path planning

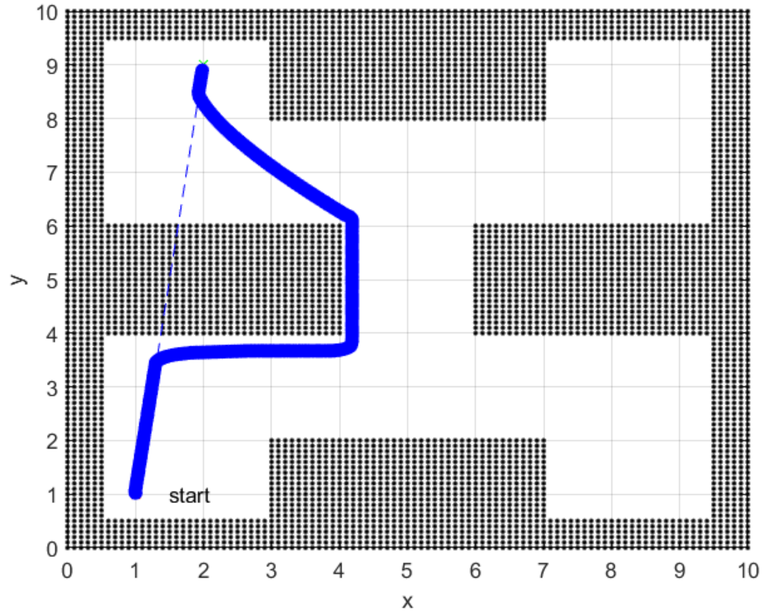

Fig. 11. The path of the quadrotor from the top view in the first scenario and online-BUG2 path planning

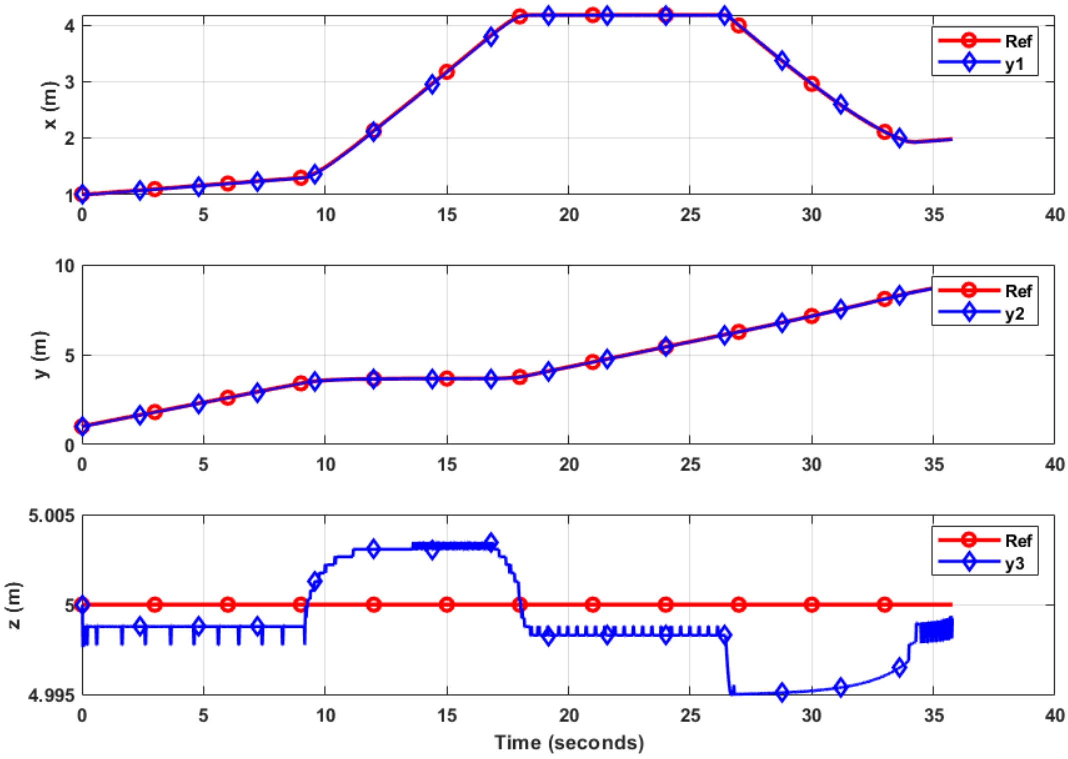

Fig. 12. The quadrotor reference path by BUG2 algorithm and the quadrotor traveled path by PFC control law in the first scenario and online-BUG2 path planning

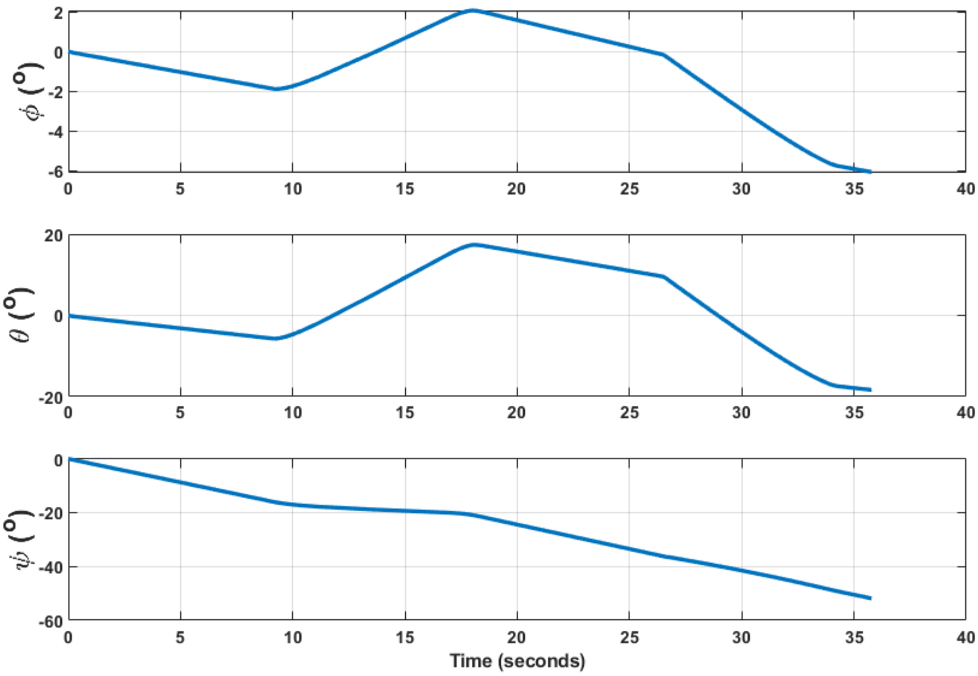

Fig. 13. Quadrotor angle over time in the first scenario and online-BUG2 path planning

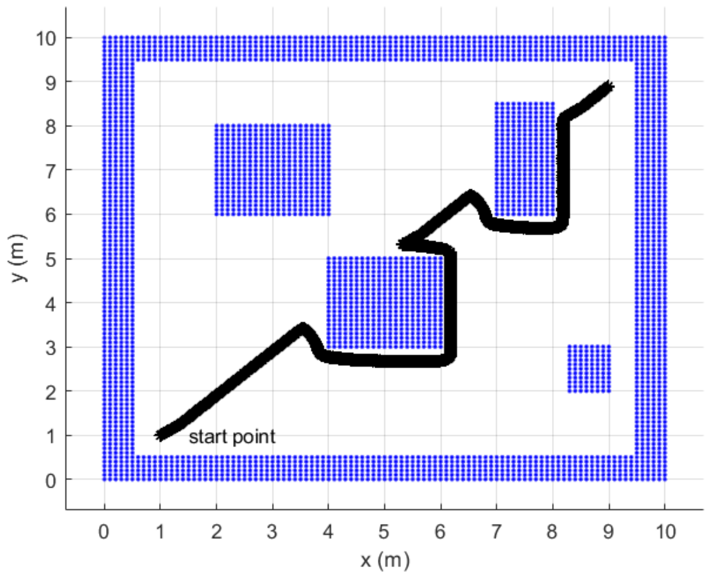

Fig. 14. The path of the quadrotor from the top view in the second scenario and BUG2 path planning

words, the quadrotor does not know in which areas of the map the obstacles are present.

The motion view of the quadrotor can be seen from the top of the self-loading section in Fig 14.

Also, the plots of the reference path determined by the BUG2 algorithm and the path traveled by the quadrotor by PFC control law are shown in Fig 15.

According to Fig 16, the movement angles of the quadrotor, despite the increase compared to the previous scenario, are within the allowed range, which indicates the acceptability of the results.

The equivalent of the results observed for the BUG2 algorithm in Figs14-16 is shown in Figs17-19 for the online-BUG2 algorithm.

According to the observations in Figs 17-19, compared to Figs 14-16, the online-BUG2 algorithm has performed better

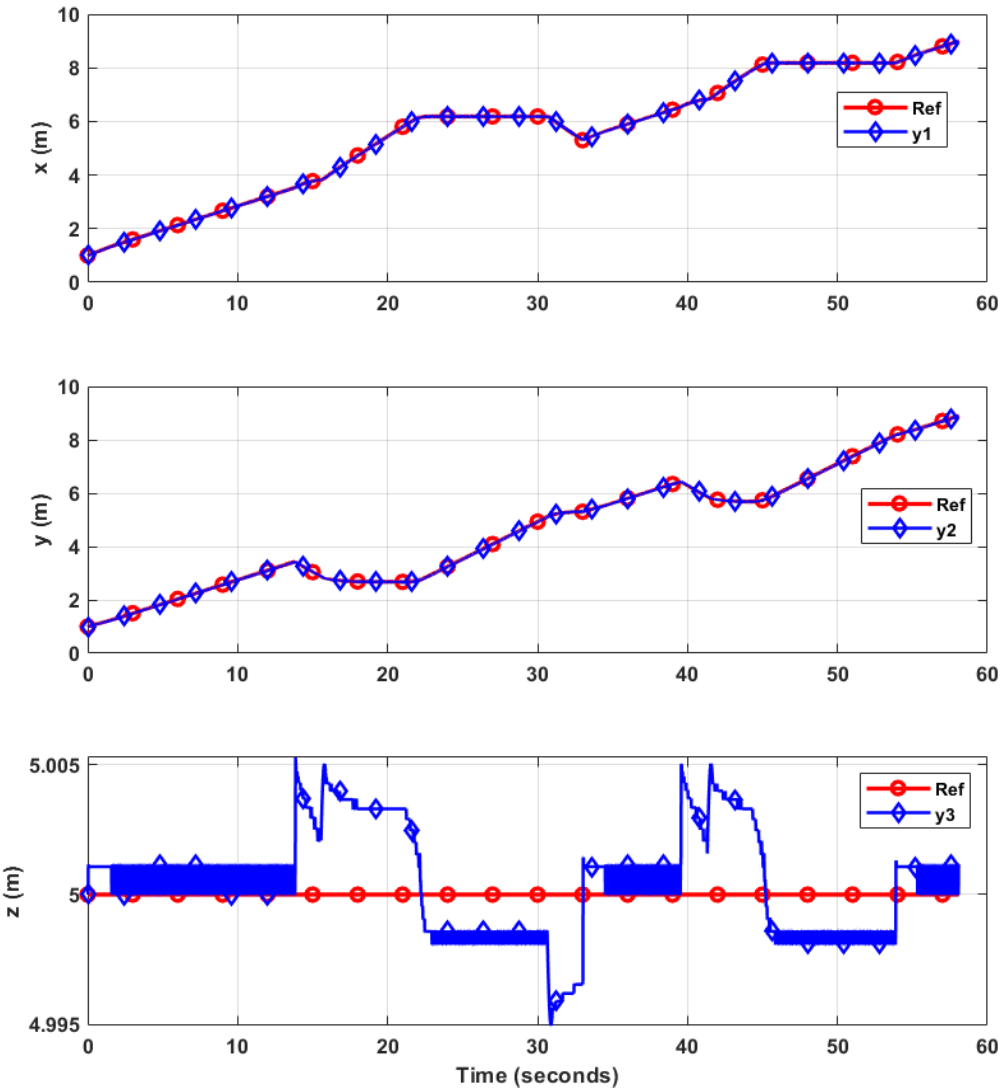

Fig. 15. The quadrotor reference path by BUG2 algorithm and the quadrotor traveled path by PFC control law in the second scenario and BUG2 path planning

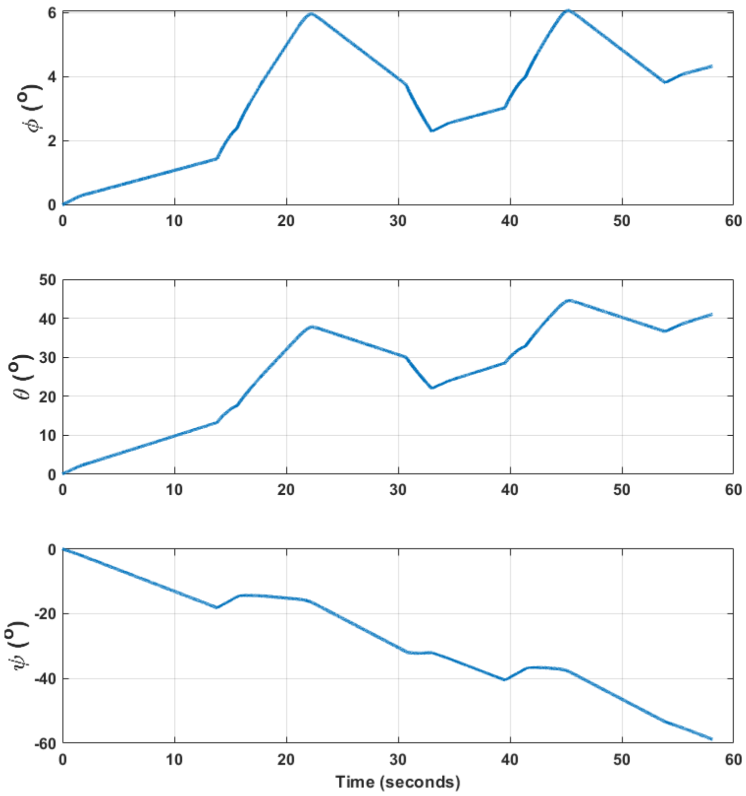

Fig. 16. Quadrotor angle over time in the second scenario and BUG2 path planning

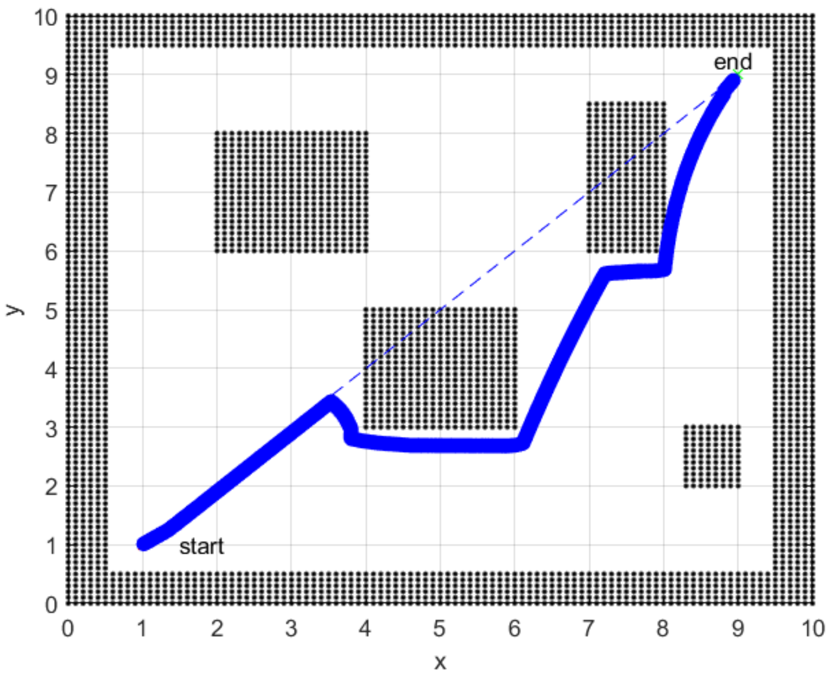

Fig. 17. The path of the quadrotor from the top view in the second scenario and online-BUG2 path planning

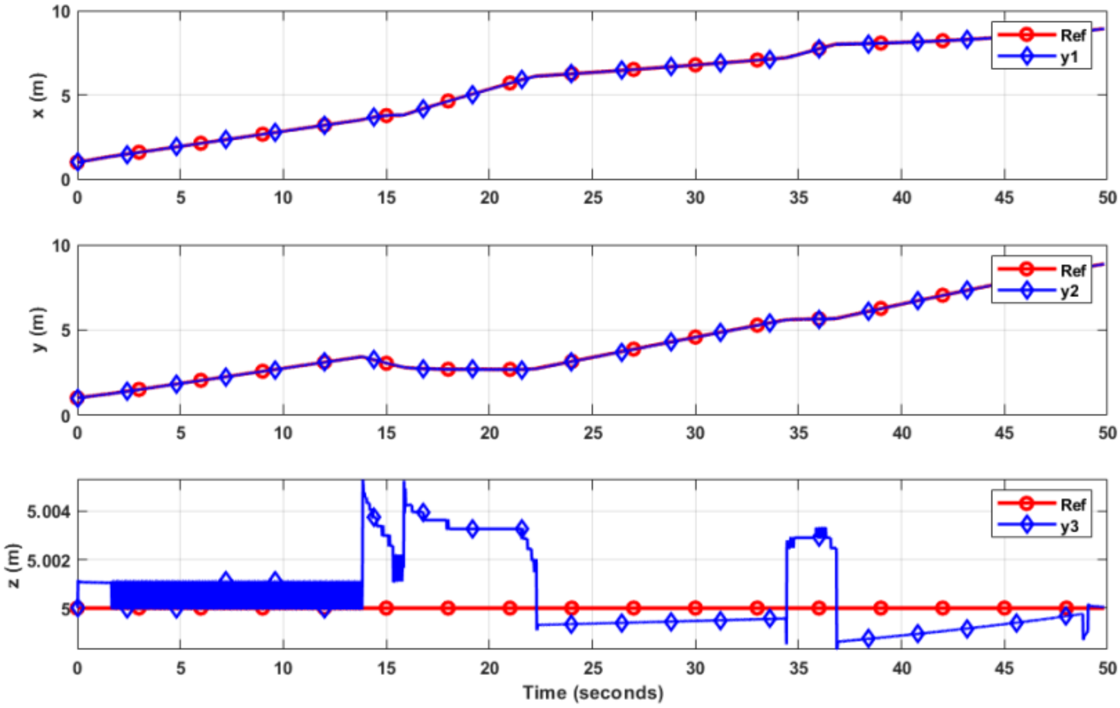

Fig. 18. The quadrotor reference path by the online-BUG2 algorithm and the quadrotor traveled path by PFC control law in the second scenario

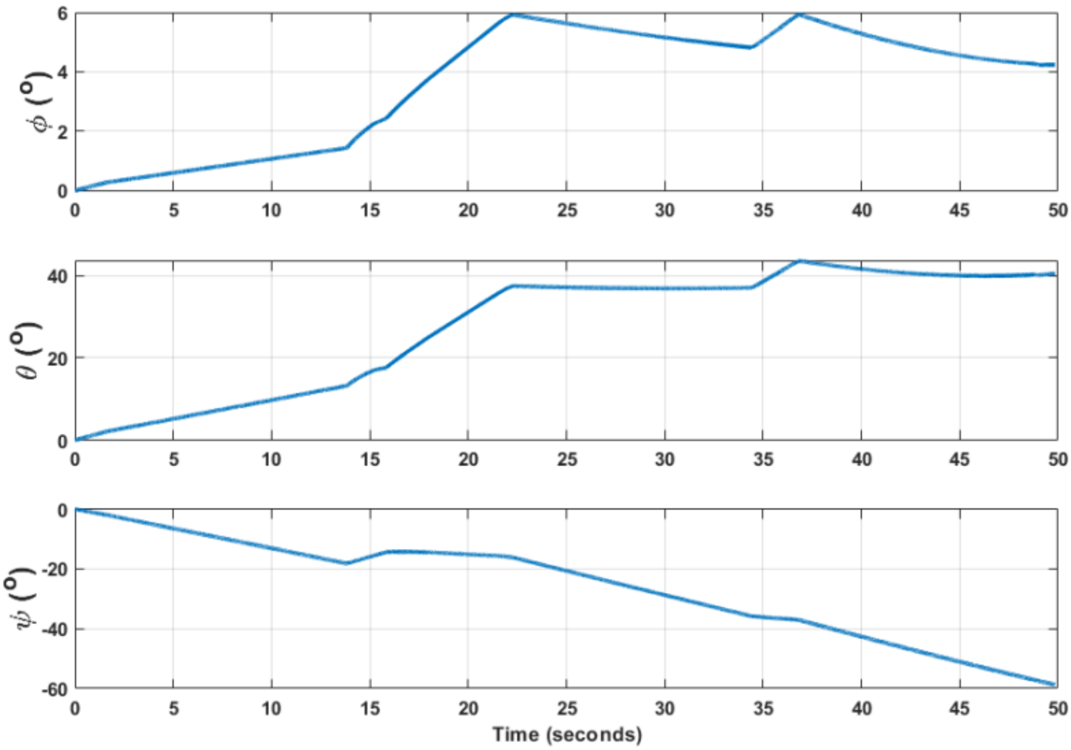

Fig. 19. Quadrotor angle over time in the second scenario and online-BUG2 path planning

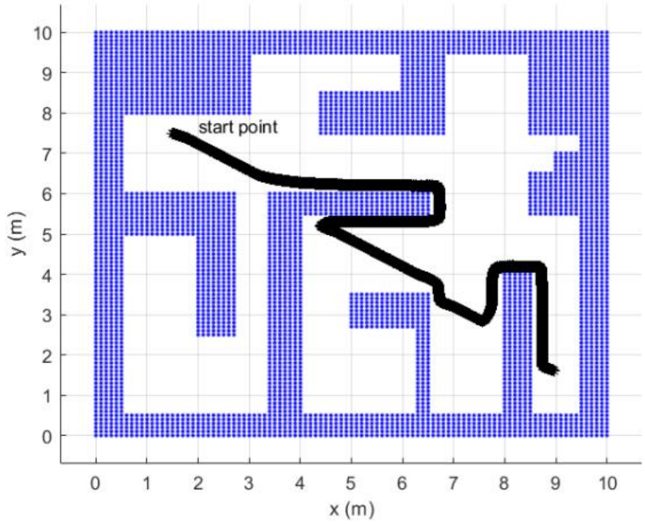

Fig. 20. The path of the quadrotor from the top view in the second scenario and online-BUG2 path planning

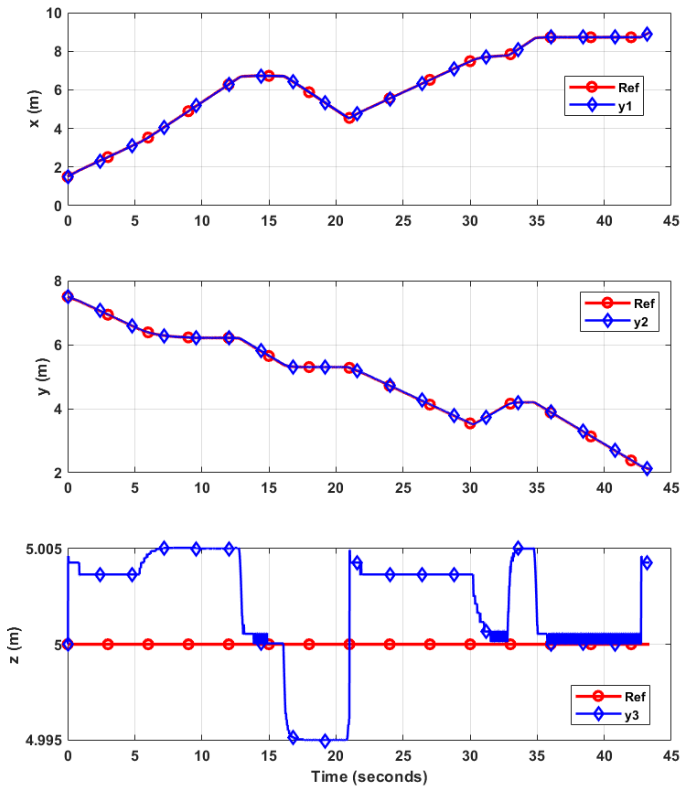

Fig. 21. The quadrotor reference path by BUG2 algorithm and the quadrotor traveled path by PFC control law in the third scenario

than the BUG2 algorithm in terms of distance traveled and angle rate changes.

4- 3- Third scenario results

The results obtained for the movement path of the third scenario are shown in Fig 20. This figure shows the movement path of this quadrotor from the top section of the screen.

As expected, the quadrotor could bypass the obstacles well and reach the destination point. Again, similar to the previous two scenarios, the guide law reference command diagram obtained with the help of BUG2, along with the quadrotor motion output obtained by applying the PFC control law to it, is plotted in a graph for the x, y, and z vectors in Fig 21. In this scenario, the applied control law met the angular limit of

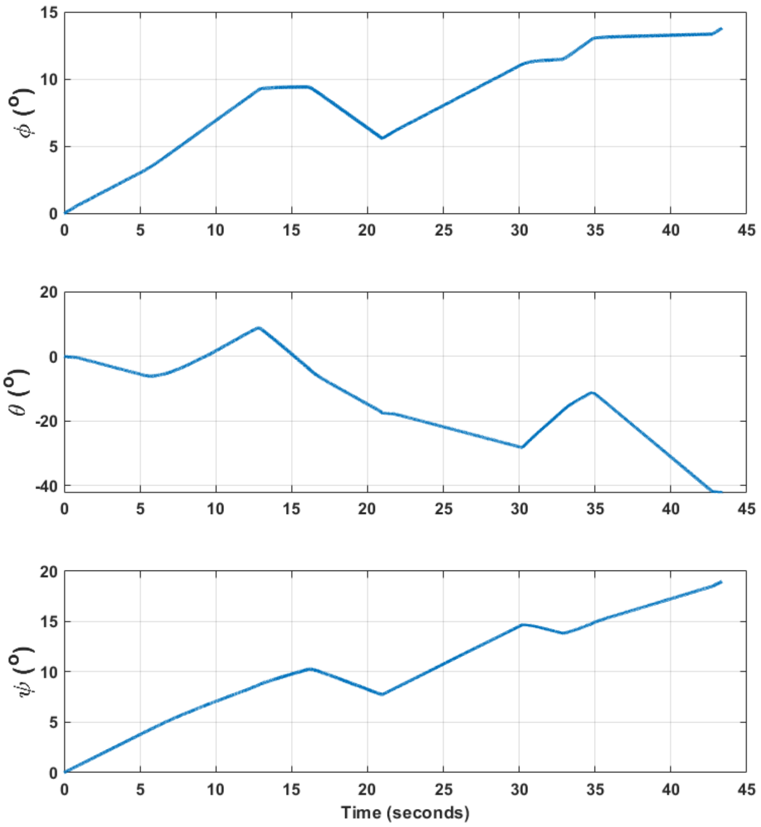

Fig. 22. Quadrotor angle over time in the third scenario and BUG2 path planning

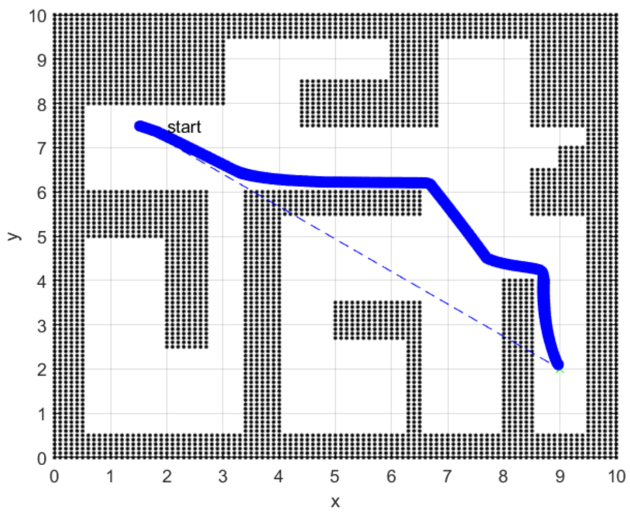

Fig. 23. The path of the quadrotor from the top view in the third scenario and online-BUG2 path planning

the elevation angle.

The equivalent of the results observed for the BUG2 algorithm in Figs 17-19 are shown in Figs 20-21 for the online-BUG2 algorithm.

As can be seen, in this scenario, similar to the previous two scenarios, the online-BUG2 method has been able to reach the target point with a shorter path and a lower angle change rate of the quadrotor, which shows the better performance of this method than BUG2.

The desired values and obtained results for each scenario are shown in Table2. The maximum error value obtained for each dimension is equal to 0.04 m, which is an acceptable value. Also, the value of the error size considering three dimensions and using the norm2 are shown in Table 3.

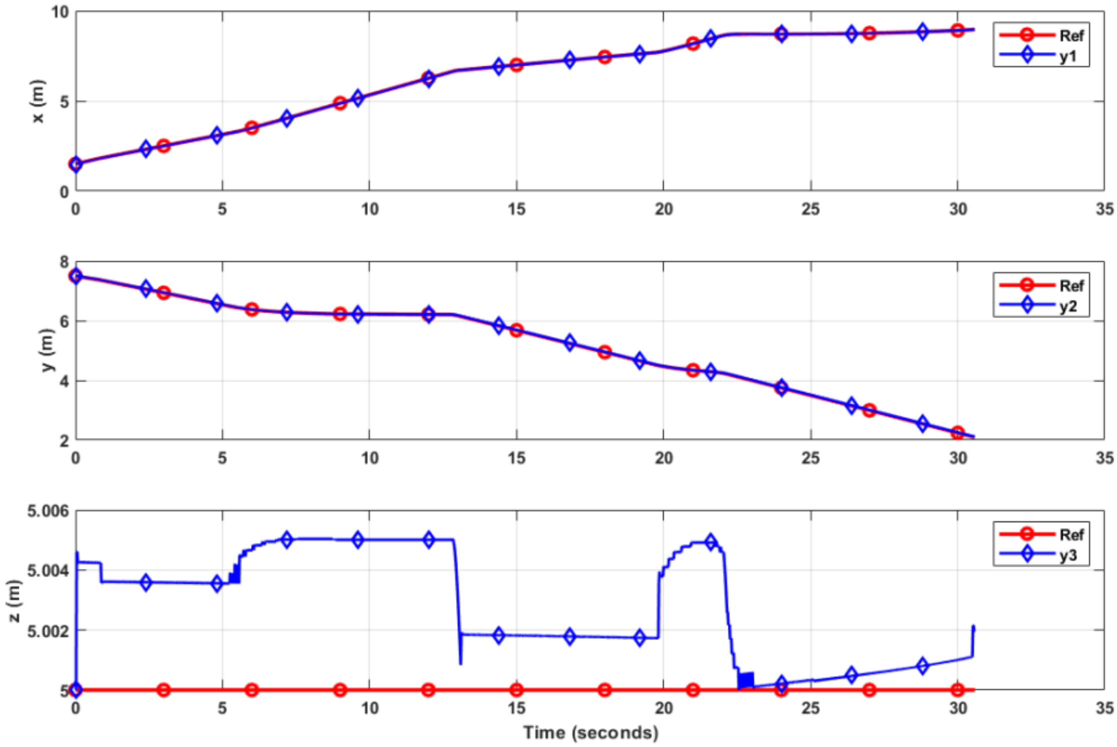

Fig. 24. The quadrotor reference path by BUG2 algorithm and the quadrotor traveled path by PFC control law in the third scenario and online-BUG2 path planning

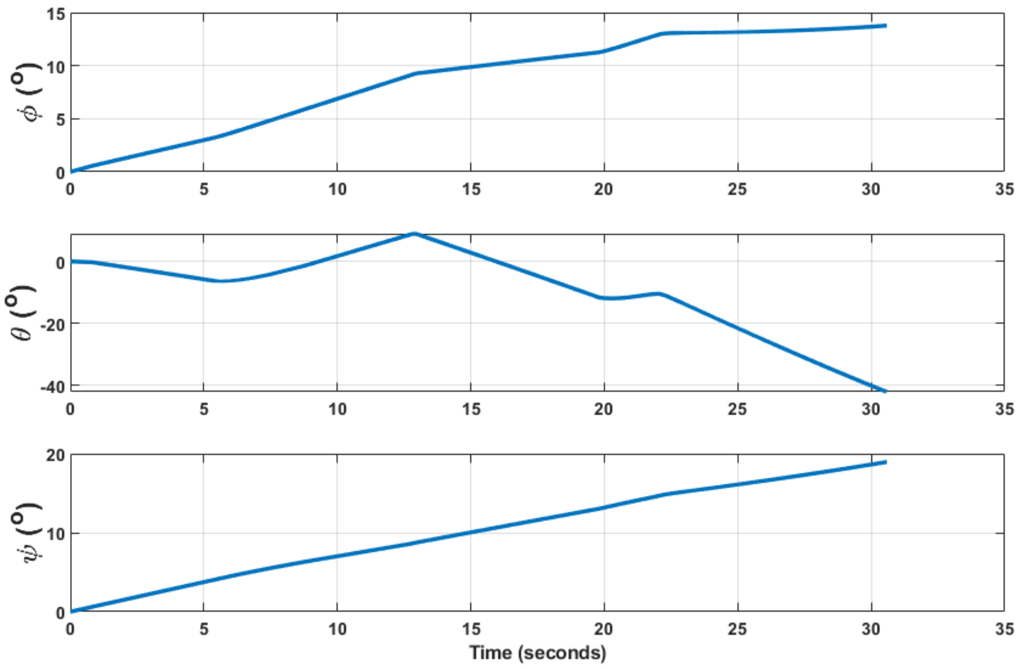

Fig. 25. Quadrotor angle over time in the third scenario and BUG2 path planning

Table 2. desired and final position of quadrotor for each scenario and path planning algorithms

First Scenario	desired position (m)	final position (m) using Bug2	final position (m) using online-Bug2
x	2	1.9701	1.9802
y	9	8.8830	8.8824
z	9	4.9988	4.9992
Second Scenario	desired position (m)	final position (m) using Bug2	final position (m) using online-Bug2
x	9	8.9929	8.9945
y	9	8.9929	8.9936
z	5	5.0012	4.9985
Third Scenario	desired position (m)	final position (m) using Bug2	final position (m) using online-Bug2
x	9	8.9686	8.9858
y	2	2.1170	2.0582
z	9	5.0043	5.0021

Table 3. Error in the final position for each scenario and path planning algorithms

Scenario	Error (m) using BUG2	Error (m) using online-BUG2
First	0.1208	0.1193
Second	0.1073	0.0086
Third	0.1212	0.0599

5- Conclusion

In this paper, two methods of path planning with quadrotor control using a predictive method were introduced. As seen from the obtained results, the online BUG method has been more effective than BUG for determining the route. As it is clear from the results of the experiments, this method has guided the quadrotor to the predetermined goal by avoiding obstacles. The result shows that the maximum error value obtained for each dimension is equal to 0.04 m, which is an acceptable value. According to the obtained results, this idea can be used to determine the direction and control of quadrotors in urban roads and even other similar cases. Of course, it goes without saying that in this direction, more research paths should be conducted, and the proposed idea should reach full maturity. Because in this idea, the predetermined obstacles were not used, and the navigation was not done in line with the third dimension, i.e., height, it is suggested that future researchers conduct the necessary research on these matters.

References

[1] Q. BinKai, L. Mingqiu, Y. Yang, W. XiYang, Research on UAV path planning obstacle avoidance algorithm based on improved artificial potential field method, in: Journal of Physics: Conference Series, IOP Publishing, 2021, pp. 012060.

[2] U. Cekmez, M. Ozsiginan, O.K. Sahingoz, Multi colony ant optimization for UAV path planning with obstacle avoidance, in: 2016 international conference on unmanned aircraft systems (ICUAS), IEEE, 2016, pp. 47-52.

[3] F. Yang, X. Fang, F. Gao, X. Zhou, H. Li, H. Jin, Y. Song, Obstacle avoidance path planning for UAV based on improved RRT algorithm, Discrete Dynamics in Nature and Society, 2022 (2022) 1-9.

[4] H.-Y. Lin, X.-Z. Peng, Autonomous quadrotor navigation with vision based obstacle avoidance and path planning, IEEE Access, 9 (2021) 102450-102459.

[5] Y. Chen, G. Bai, Y. Zhan, X. Hu, J. Liu, Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning, Ieee Access, 9 (2021) 40728-40742.

[6] S. Zaheer, T. Gulrez, I.A. Thythodath Paramabath, From sensor-space to eigenspace–a novel real-time obstacle avoidance method for mobile robots, IETE Journal of Research, 68(2) (2022) 1512-1524.

[7] A.M. Mohsen, M.A. Sharkas, M.S. Zaghlol, New Real Time (M-Bug) Algorithm for Path Planning and Obstacle Avoidance In 2D Unknown Environment, in: 2019 29th International Conference on Computer Theory and Applications (ICCTA), IEEE, 2019, pp. 25-31.

[8] J.J. Kandathil, R. Mathew, S.S. Hiremath, Development and analysis of a novel obstacle avoidance strategy for a multi-robot system inspired by the bug-1 algorithm, Simulation, 96(10) (2020) 807-824.

[9] D.J. Almakhles, Robust backstepping sliding mode control for a quadrotor trajectory tracking application, IEEE Access, 8 (2019) 5515-5525.

[10] G. Perozzi, D. Efimov, J.-M. Biannic, L. Planckaert, Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains, Journal of the Franklin Institute, 355(12) (2018) 4809-4838.

[11] S. Raiesdana, Control of quadrotor trajectory tracking with sliding mode control optimized by neural networks, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 234(10) (2020) 1101-1119.

[12] A. Kapnopoulos, A. Alexandridis, A cooperative particle swarm optimization approach for tuning an MPC-based quadrotor trajectory tracking scheme, Aerospace Science and Technology, 127 (2022) 107725.

[13] D. Wang, Q. Pan, Y. Shi, J. Hu, C. Zhao, Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment, IEEE Transactions on Cybernetics, 51(10) (2021) 5057-5068.

[14] V. Nekoukar, N.M. Dehkordi, Robust path tracking of a quadrotor using adaptive fuzzy terminal sliding mode control, Control Engineering Practice, 110 (2021) 104763.

[15] Z. Cai, S. Zhang, X. Jing, Model predictive controller for quadcopter trajectory tracking based on feedback linearization, IEEE Access, 9 (2021) 162909-162918.

[16] A. Eltrabyly, D. Ichalal, S. Mammar, Quadcopter Trajectory Tracking in the Presence of 4 Faulty Actuators: A Nonlinear MHE and MPC Approach, IEEE Control Systems Letters, 6 (2021) 2024-2029.

[17] R. Xue, L. Dai, D. Huo, Y. Xia, Robust Model Predictive Control with ESO for Quadrotor Trajectory Tracking with Disturbances, in: 2022 IEEE 17th International Conference on Control & Automation (ICCA), IEEE, 2022, pp. 192-198.

[18] M. Islam, M. Okasha, E. Sulaeman, A model predictive control (MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking, International Journal of Control, Automation and Systems, 17(11) (2019) 2819-2832.

[19] A. Eskandarpour, I. Sharf, A constrained error-based MPC for path following of quadrotor with stability analysis, Nonlinear Dynamics, 99 (2020) 899-918.

[20] B. Samir, Design and control of quadrotors with application to autonomous flying, Project report, Ecole Polytechnic, (2007).

[21] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementation ERRATA!!!!, (2007).

[22] H. Jahanshahi, N.N. Sari, Robot path planning algorithms: a review of theory and experiment, arXiv preprint arXiv:1805.08137, (2018).

[23] R. Bishop, Model Based Predictive Control–A Practical Approach, in, CRC Press, 2004.

HOW TO CITE THIS ARTICLE
Sh. Ansari, S. Nasrollahi, Path Planning and Tracking for a Quadrotor with the Aim of Obstacle Avoidance Using the BUG2 and Predictive Control, AUT J. Model. Simul., 55(2) (2023) 243-262.

DOI: 10.22060/miscj.2023.22575.5330