Strojnícky časopis – *Journal of* MECHANICAL ENGINEERING,
VOL 75 (2025), NO 2, 111 – 122

THE EFFECTIVE ALGORITHM FOR NAVIGATION OF QUADROTOR UAV IN INDOOR ENVIRONMENT

VAN-LONG Trinh*, NGOC-TIEN Tran

School of Mechanical and Automotive Engineering, Hanoi University of Industry, 298 Caudien Street, Hanoi 100000, Vietnam, e – mail: longtv@haui.edu.vn

Abstract: The quadrotor is an extensively researched and popular unmanned aerial vehicle in various fields, with navigation being one of the most widely studied areas. This paper focuses on setting the path for the quadrotor to move in a predefined 3D map environment. A real-time mathematical model for the quadrotor's motion was developed, incorporating the moment of resistance and the rotor rolling coefficient using improved Lagrange dynamic modeling. Two navigation scenarios were then built using Dijkstra and A star (A*) algorithms to evaluate the efficiency of the mathematical model based on criteria such as distance, time, and speed. The results indicate that Dijkstra's algorithm resulted in a 0.55% shorter travel distance for the quadrotor than the A* algorithm, while the quadrotor's flight time of the A* algorithm is 0.52% longer. However, the maximum speed of the quadrotor guided by the A* algorithm is 7.87% higher than that of Dijkstra's algorithm. These findings provide a basis for developing an optimal navigation strategy for the quadrotor to meet specific requirements.

KEYWORDS: Quadrotor UAV; navigation; dynamic modeling; Dijkstra algorithm; star algorithm.

1 Introduction

Robotics has emerged as a highly promising avenue for 4.0 industry [1-4]. The design of unmanned aerial vehicles (UAV) has rapidly evolved to meet the demands of various critical missions, including surveillance, fire protection, search and rescue, military, and civilian applications [5-7]. The quadrotor stands out as a versatile rotorcraft that can hover, fly forward, and take off and land vertically, making it an ideal research and application platform due to its adaptability, flexibility, and ease of construction [8, 9]. Many research groups have been focused on controlling and applying UAV in practical applications such as developing a navigation algorithm for quadrotor UAVs to guide themselves autonomously through unfamiliar tunnel-like environments [10], introducing an algorithm for autonomous navigation of a quadrotor UAV in a GPS-denied and unknown environment [11], using a vision-guided autonomous navigation method for quadrotor UAVs [12], introducing a robust geometric navigation algorithm for a quadrotor, based on the special Euclidean group, and formulated using the Newton-Euler method with ability of robustness against parameter uncertainty and disturbances, as well as convergence to the sliding manifold in a finite time and asymptotic convergence of the trajectory tracking [13]. A framework based on duelling double-deep recurrent Q-learning for achieving autonomous navigation in semi-known environments that is designed to handle situations where only a portion of the obstacles can be identified and located in advance [14]. It is suggested using two sensors to function as a flying odometer for estimating the forward velocity of a UAV [15]. The estimated velocity is then combined with INS using an Extended Kalman Filter to improve the precision of navigation. The authors suggested an improved navigation system for UAVs by integrating a monocular camera, an Ultra-Wideband (UWB) system, and INS [16]. They explore this integration in scenarios where UWB anchor density is low, with the aim of reducing the infrastructure cost associated with UWB system. Many methods have been employed for UAV path planning in indoor environments such as

DOI: 10.2478/scjme-2025-0028, Print ISSN 0039-2472, On-line ISSN 2450-5471

UAV capabilities for indoor operations [17, 18], graph construction [19], rapidly exploring random trees [20], and Dijkstra's algorithm for optimizing UAV path [21]. Some groups have been developed simulation skill to evaluate the working ability for improving the service quality of robot [22] and autonomous vehicles [23]. This study deals with the creation of a path planning and obstacle avoidance system for a quadrotor UAV operating in an indoor environment. A Euler-Lagrange equation-based motion dynamics model for the quadrotor is developed in the study. The two navigation algorithms for quadrotor are used to assess the method's efficacy.

2 Materials and Methods

A quadrotor is an unmanned aerial vehicle that utilizes four rotors to generate lift. Its structure resembles a plus sign with two pairs of reversible rotors, where opposite pairs of rotors rotate in the same direction. In addition to contributing to the aircraft's thrust to gain altitude, these rotors also create motion around the center of gravity as shown in Figure 1. The quadrotor's movement is controlled by adjusting the speed of each rotor independently. By increasing the speed of two opposing rotors and decreasing the speed of two others, a moment is created that causes the quadrotor to tilt in a specific direction. This mechanism is utilized to control the quadrotor's motion in various directions.

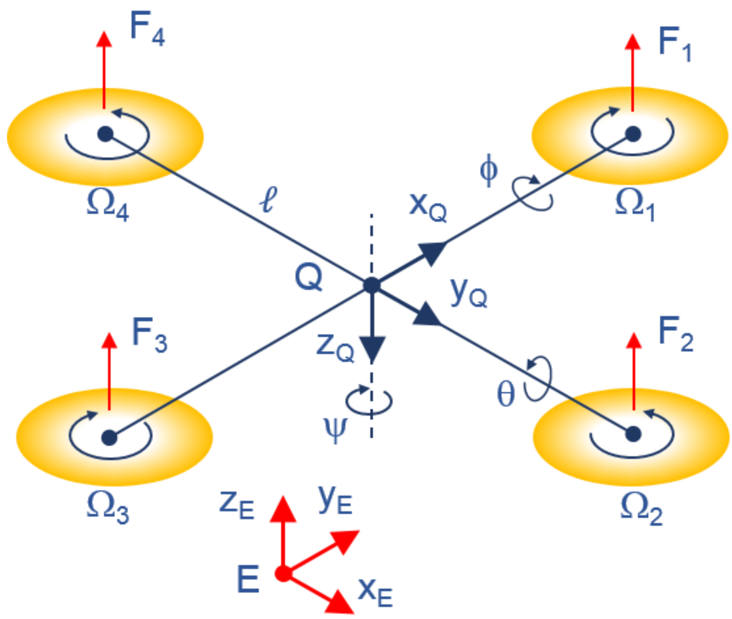


Fig. 1 The body-fixed frame and the inertial reference frame of quadrotor

Figure 1 illustrates the relationship between the inertial frame of reference, denoted as frame $\Psi(Ex_E y_E z_E)$, and the body-fixed frame, denoted as frame $\Upsilon(Qx_Q y_Q z_Q)$. It also shows the distance between the rotor and the origin of the body-fixed frame, denoted as ℓ, and the lift forces, denoted as F_1, F_2, F_3, F_4, produced by the rotors rotating at an angular velocity denoted as $\Omega_1, \Omega_2, \Omega_3, \Omega_4$.

The UAV quadrotor's movement in the world frame is expressed using Euler rotation sequence of Z-X-Y. This convention is used to construct the rotation matrix from frame Ψ to frame Υ:

$$\mathbf{R}_{\Upsilon}^{\Psi} = \begin{bmatrix} c_{\psi}c_{\theta} & c_{\psi}s_{\theta}s_{\phi} - s_{\psi}c_{\phi} & c_{\psi}s_{\theta}c_{\phi} + s_{\psi}s_{\phi} \\ s_{\psi}c_{\theta} & s_{\psi}s_{\theta}s_{\phi} + c_{\psi}c_{\phi} & s_{\psi}s_{\theta}c_{\phi} - c_{\psi}s_{\phi} \\ -s_{\theta} & c_{\theta}s_{\phi} & c_{\theta}s_{\phi} \end{bmatrix} \tag{1}$$

where $\begin{cases} c_* \triangleq \cos(*) \\ s_* \triangleq \sin(*) \end{cases}$, the angular velocity components of the robot in the body frame are denoted as ω_x, ω_y, ω_z:

$$\omega_{\Upsilon}^{\Psi} = \omega_x \mathbf{x}_Q + \omega_y \mathbf{y}_Q + \omega_z \mathbf{z}_Q \tag{2}$$

Here, $\mathbf{x}_Q$, $\mathbf{y}_Q$, $\mathbf{z}_Q$ are unit vectors in the Υ reference system, corresponding to the coordinate axes x_Q, y_Q, z_Q. The values of ω_x, ω_y, ω_z are obtained by differentiating the roll, pitch, and yaw angles using the following formula:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{s}_{\theta} \\ 0 & c_{\phi} & c_{\phi}s_{\phi} \\ 0 & -s_{\phi} & \mathrm{c}_{\theta}\, c_{\phi} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3}$$

This section models the quadrotor's rotational dynamics using the Euler-Lagrange equation. The general dynamic model of the quadrotor is given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} = \tilde{\mathbf{Q}} \tag{4}$$

where $\mathbf{q}$ is the generalized coordinate, $\tilde{\mathbf{Q}}$ represents the generalized force, and Lagrangian is defined as L:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T - U \tag{5}$$

where $U = mgz$ is the potential energy (m is the mass of the quadrotor, g is the acceleration of gravity, z is the rotorcraft altitude), and the kinetic energy T is given by:

$$T = T_{rot} + T_{trans} = \frac{1}{2}\mathbf{\Omega}^{\mathrm{T}}\mathbf{I}\mathbf{\Omega} + \frac{m}{2}\dot{\boldsymbol{\xi}}^{\mathrm{T}}\dot{\boldsymbol{\xi}} \tag{6}$$

where $\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{xx} \end{bmatrix}$ is the inertia matrix, $\dot{\boldsymbol{\xi}}$ is the velocity matrix defined in the generalized coordinate:

$$\begin{cases} \mathbf{q} = (x, y, z, \phi, \theta, \psi)^{\mathrm{T}} \in \mathbb{R}^6 \\ \boldsymbol{\xi} = (x, y, z)^{\mathrm{T}} \in \mathbb{R}^3 \\ \boldsymbol{\eta} = (\phi, \theta, \psi)^{\mathrm{T}} \in \mathbb{R}^3 \end{cases} \tag{7}$$

Angular velocity $\mathbf{\Omega}$ is given by:

$$\mathbf{\Omega} = \mathbf{W}_{\eta}\dot{\boldsymbol{\eta}} \tag{8}$$

which yields $\mathbf{W}_\eta = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\phi s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix}$ and T_{rot} can be rewritten as follows:

$$T_{rot} = \frac{1}{2}\dot{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{W}_\eta^{\mathrm{T}}\mathbf{I}\mathbf{W}_\eta\dot{\boldsymbol{\eta}} = \frac{1}{2}\dot{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{J}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} \tag{9}$$

On the other hand, the generalized force $\tilde{\mathbf{Q}}$ is given by:

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{F}_\xi \\ \boldsymbol{\tau}_\eta \end{bmatrix} \tag{10}$$

which yields: $\begin{cases} \mathbf{F}_\xi = \mathbf{R}_\Upsilon^\Psi \hat{\mathbf{F}} = \mathbf{R}_\Upsilon^\Psi \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_i \end{bmatrix} \\ \boldsymbol{\tau}_\eta = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \approx \begin{bmatrix} (F_3 - F_1)\ell \\ (F_2 - F_4)\ell \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} \end{cases}$

Substituting (6) into (4) gives us the translational equation, and the rotational equation is given by:

$$\begin{cases} m\ddot{\boldsymbol{\xi}} + mg\mathbf{V}_z = \mathbf{F}_\xi \\ \mathbf{J}\ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} = \boldsymbol{\tau}_\eta \end{cases} \tag{11}$$

where $\mathbf{V}_z$ is the rotor speed, $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \dot{\mathbf{J}} - \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\eta}}\left(\dot{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{J}\right)$.

The map is a 3D probability density function that serves as a scalar field, representing the probability density of occupancy at a specific location in space. However, the map is stored discretely as a representation of the most recent occupancy probability over time. In this paper, the construction of the grid map is discussed with the state model and the measurement model. A collision map is constructed, with a size determined during the initialization of the map. A 3D collision map is created with a specified size, describing the density of the occupied parameters on the x, y, and z axes as shown in Figure 2. The list of squares is then looped through, and for each block, its corresponding position on the map and state are calculated by transforming coordinates between real and design space. The map has been arranged according to the layout depicted in Figure 2, allowing for the observation of cell occupancy density. The colors have been incorporated for ease of visual examination.

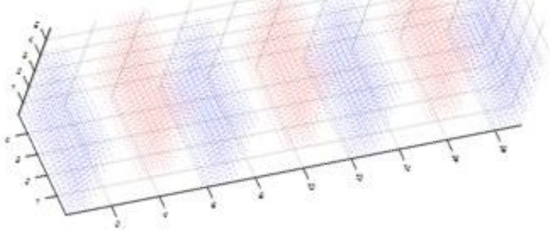


Fig. 2 Building 3D maps

The two widely used algorithms are the A* algorithm and Dijkstra algorithm. The paper involves a combination of these two algorithms to determine the optimal direction for quadrotor navigation. The A* algorithm constructs a path tree from the initial node and progressively expands paths until one leads to the target node. During each iteration of the algorithm, A* selects neighboring nodes with the lowest cost using an evaluation function.

Algorithm 1: A* algorithm

```
A*(Start, Goal)
S ← ∅
Q ← V
G(Start) ← 0 ; H(Start) ← H_c(Start, Goal) ;
F(Start) ← H(Start)
while Q ≠ ∅ do
    C_node ← EXTRACT − MIN − F(Q)
    if C_node == Goal then
    return BestPath
    for each N ⊂ C_node do
        if N ⊂ S then Nothing
        else if Calculate N_s , G , H , F
            if G(N ⊂ Q) > calculated G(N) then
            RELAX(N, N ⊂ Q, w)
            N_s_parent = C_node
            Add N to Q
            else then calculated N_s, G, H, F
            N_s_parent = C_node
            Add N to Q
            end if
        end if
    end for
    end if
end while
```

Algorithm 2. Dijkstra algorithm

```
DIJKSTRA(V, E, w, s)
S ← ∅
Q ← V
while Q ≠ ∅ do
    u ← EXTRACT − MIN(Q)
    S ← S ∪ {u}
    for each v ∈ Adj(u) do
        RELAX(u, v, w)
    end for
end while
```

Dijkstra's algorithm can be used to find the shortest path in a weighted, directed graph with non-negative weights. The algorithm maintains a priority queue called $MIN(Q)$, which stores the unprocessed vertices along with their shortest path estimates. It repeatedly extracts the vertex with the minimum value from MIN (Q) and relaxes all edges that are incident from u to any vertex in $MIN(Q)$. Once a vertex is extracted from $MIN(Q)$ and all its incident edges are relaxed, the algorithm considers the vertex as processed and does not touch it again. The algorithm stops when $MIN(Q)$ is empty or when every vertex is examined exactly once.

3 RESULTS AND DISCUSIONS

This section establishes a flight map for the quadrotor, which includes dimensions and obstacles depicted in Figure 3.

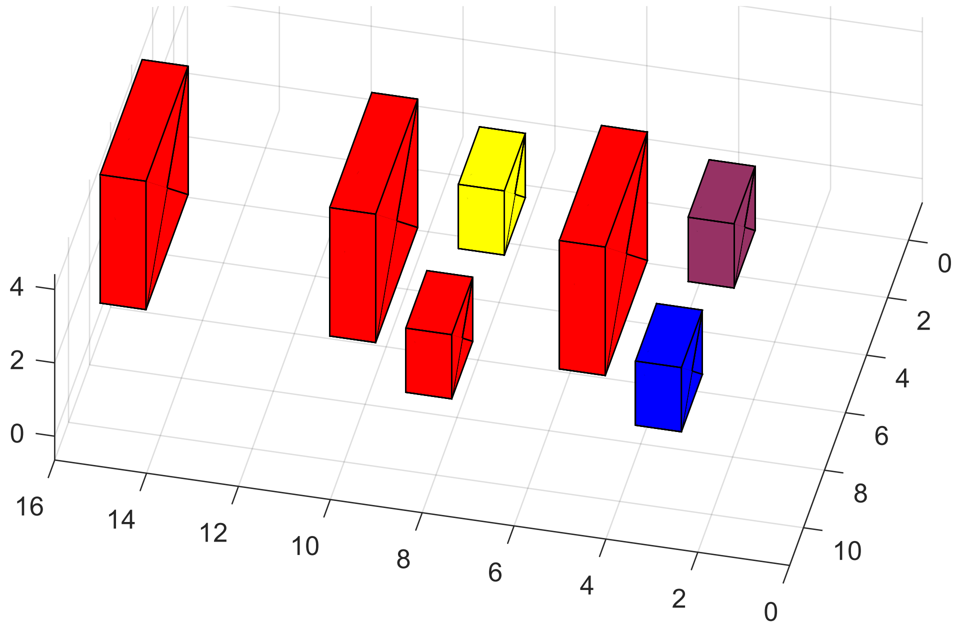


Fig. 3 Quadrotor's movement map

The quadrotor UAV travels from its initial position at coordinates (0, 6, 0) to the destination point at (14, 4, 2). The robot's mass is $m = 0.03kg$, $\ell = 0.046m$, and the maximum tilt angle is 40^0. The A* algorithm is employed to determine the actual position, velocity, and path of the quadrotor, which are represented in Figures 4-7. The Dijkstra's algorithm is utilized to illustrate the actual position, velocity, and path of the quadrotor as shown in Figures 8-11.

The results are obtained using a sampling time of 0.2 seconds during UAV flight. The quadrotor takes 9.60s to travel from the starting point to the destination using the A* algorithm, while it takes 9.55s by using Dijkstra's algorithm. Therefore, Dijkstra's algorithm results in a 0.52% faster travel time than the A* algorithm. In Figures 7 and 17, the blue line represents the actual path of quadrotor. The A* algorithm's path is 3.5% shorter than Dijkstra's algorithm path. However, according to Figure 6 and Figure 10, the quadrotor travels a distance of 18.03m with the A* algorithm and 17.93m for Dijkstra's algorithm, resulting in Dijkstra's algorithm being 0.55% shorter than the A* algorithm in terms of travel distance. However, with Dijkstra's algorithm, the quadrotor's movement is unstable due to deceleration to ensure the correct target, as shown by the path. Figures 5 and 9 illustrate that the quadrotor achieves a maximum speed of Dijkstra's algorithm of 7.87% higher than that of the A* algorithm.

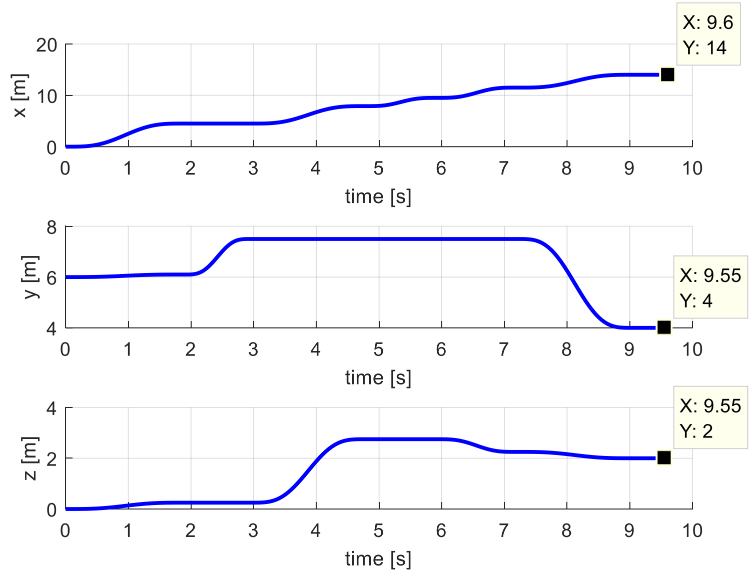


Fig. 4 Position of robot during movement using A* algorithm

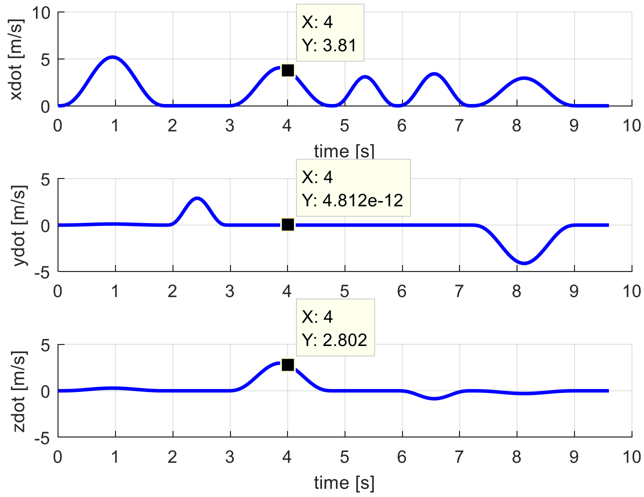


Fig. 5 Velocity of robot during movement using A* algorithm

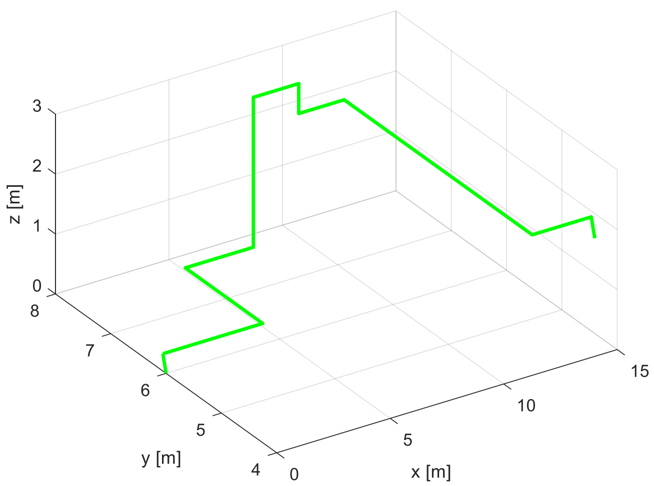
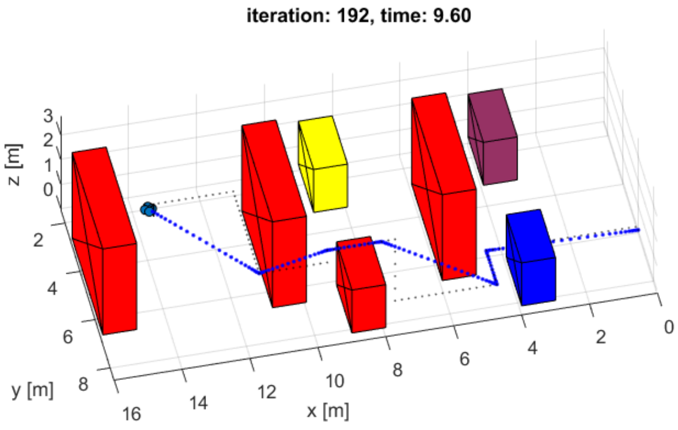
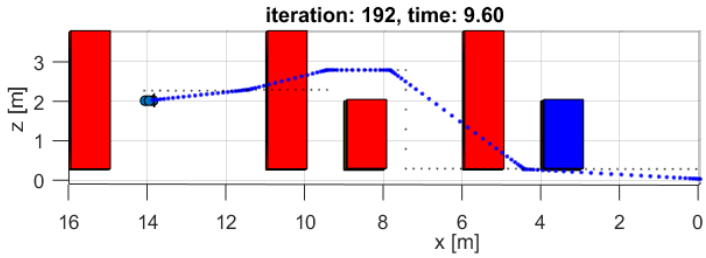


Fig. 6 Path of the quadrotor UAV using A* algorithm



(a)



(b)

Fig. 7 Actual path of quadrotor when using A* algorithm:
a- Panoramic view; b- view angle z-x

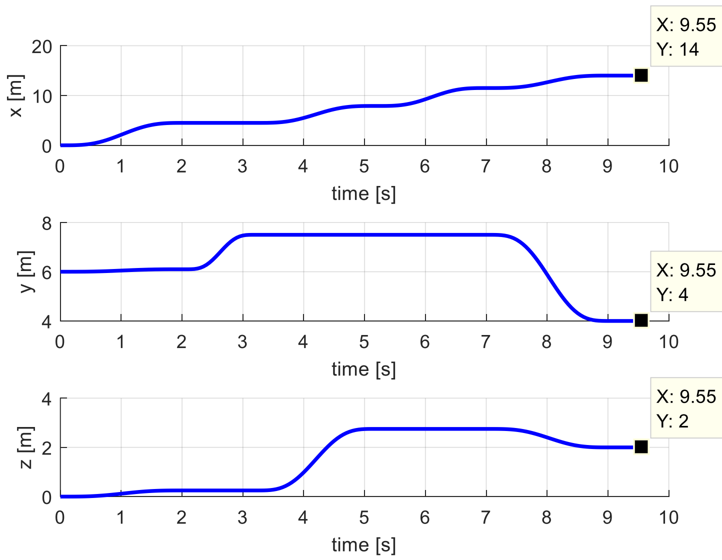


Fig. 8 Position of robot during movement using Dijkstra's algorithm

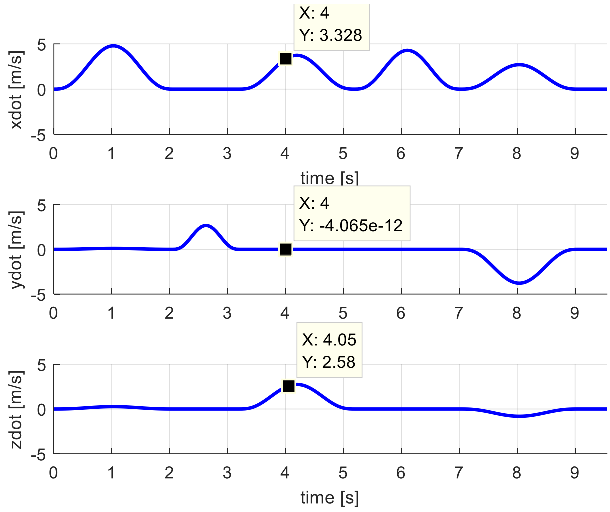


Fig. 9 Velocity of robot during movement using Dijkstra's algorithm

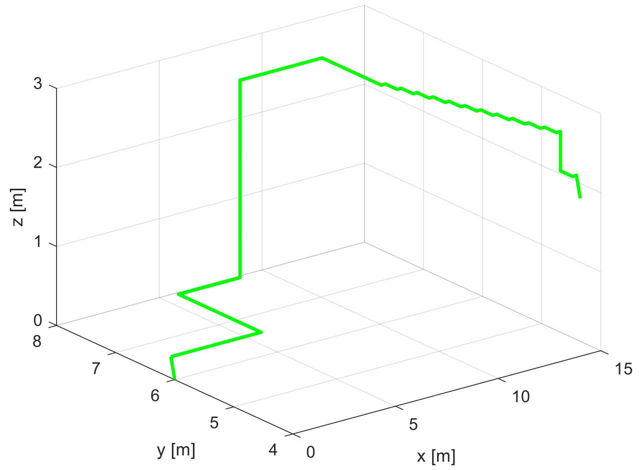
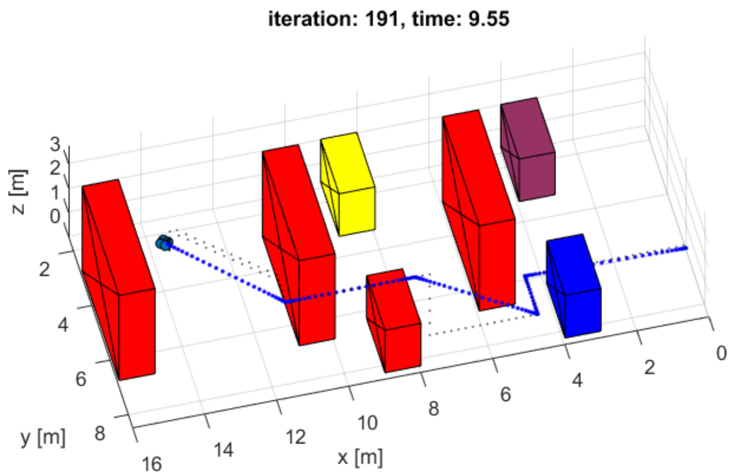
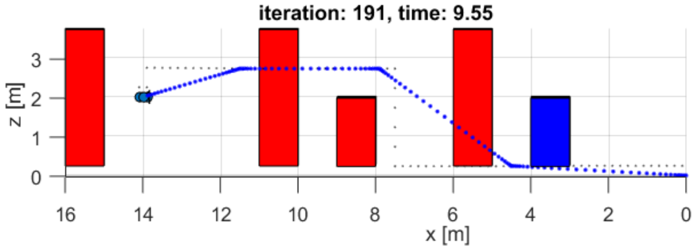


Fig. 10 Path of the quadrotor UAV using Dijkstra's algorithm



(a)



(b)
Fig. 11 Actual path of quadrotor when using Dijkstra’s algorithm:
a- Panoramic view; b- view angle z-x

CONCLUSIONS

This study investigated the performance of A* and Dijkstra algorithms in determining the path and travel time of a quadrotor UAV from a starting point to a destination. The results show that while A* algorithm resulted in a path that is 3.5% shorter than that of Dijkstra's algorithm, Dijkstra's algorithm covered a travel distance that is 0.55% shorter than A* algorithm. However, using Dijkstra's algorithm resulted in unstable movement due to deceleration to ensure the correct target. The quadrotor achieved a maximum speed that is 7.87% higher when using A* algorithm compared to when using Dijkstra's algorithm. For flight time, Dijkstra's algorithm resulted in a 0.52% faster travel time than A* algorithm. Findings suggest that the choice of algorithm depends on the specific objectives and constraints of UAV flight mission.

ACKNOWLEDGMENT

The authors thank Hanoi University of Industry, 298 Caudien Street, Hanoi 100000, Hanoi, Vietnam.

REFERENCES

[1] Tran, N. T., Ngo,T. D., Nguyen, D. K., Son, P. X., Thai, N. H. "Mapping and path planning for the differential drive wheeled mobile robot in unknown indoor environments using the rapidly exploring random tree method", In Regional Conference in Mechanical Manufacturing Engineering, pp. 516 – 527, **2021**. DOI: 10.1007/978-981-19-1968-8_43

[2] Bui, T. L., Tran, N. T. "Navigation strategy for mobile robot based on computer vision and YOLOv5 network in the unknown environment", Applied Computer Science 19 (2), pp. 82 – 95, **2023**. DOI: 10.35784/acs-2023-16

[3] Pham, D. A., Bui, H. A., Nguyen, X. T., Mac, T. T., Hoang, H. H. "A case study on humanoid robot using robotics software in e-learning", In Regional Conference in Mechanical Manufacturing Engineering, pp. 761 – 770, **2021**. DOI: 10.1007/978-981-19-1968-8_63

[4] Nguyen, X. T., Bui, T. L., Bui, H. A., Pham, D. A., Miura, N. "Control the movement of mobile robot using fingers gestures based on fuzzy logic", In International Conference on Advanced Mechanical Engineering, Automation and Sustainable Development, pp. 799 – 804, **2021**. DOI: 10.1007/978-3-030-99666-6_116

[5] Alanezi, M. A., Shahriar, M. S., Hasan, M. B., Ahmed, S., Sha’aban, Y. A., Bouchekara, H. R. E. H. "Livestock management with unmanned aerial vehicles: a review", IEEE Access. 10, pp. 45001 – 45028, **2022**. DOI: 10.1109/ACCESS.2022.3168295

[6] Rivas, A., Chamoso, P., González-Briones, A., Corchado, J. M. "Detection of cattle using drones and convolutional neural networks", Sensors 18 (7), 2048, **2018**. DOI: 10.3390/s18072048

[7] Barbedo, J. G. A., Koenigkan, L. V., Santos, P. M., Ribeiro, A. R. B. "Counting cattle in UAV images—dealing with clustered animals and animal/background contrast changes", Sensors 20, 2126, **2020**. DOI: 10.3390/s20072126

[8] Xu, B., Wang, W., Falzon, G., Kwan, P., Guo, L., Chen, G., Tait, A., Schneider, D. "Automated cattle counting using mask r-cnn in quadcopter vision system", Comput. Electron. Agric. 171, 105300, **2020**. DOI: 10.1016/j.compag.2020.105300

[9] Alanezi, M. A., Sadiq, B. O., Sha, Y. A., Bouchekara, H. R. E. H. "Livestock management on grazing field: a fanet based approach", Appl. Sci. 12, 6654, **2022**. DOI: 10.3390/app12136654

[10] Elmokadem, T., Savkin, A. V. "A method for autonomous collision-free navigation of a quadrotor UAV in unknown tunnel-like environments", Robotica 40 (4), pp. 1 – 27, **2021**. DOI: 10.1017/S0263574721000849

[11] Biswas, S., Sharma, R. "Goal-aware robocentric mapping and navigation of a quadrotor unmanned aerial vehicle", Journal of Aerospace Information Systems 18 (8), pp. 488 – 505, **2021**. DOI: 10.2514/1.I010843

[12] Lin, H. Y., Peng, X. Z. "Autonomous quadrotor navigation with vision based obstacle avoidance and path planning", IEEE Access 9, pp. 102450 – 102459, **2021**. DOI: 10.1109/ACCESS.2021.3097945

[13] Garcia, O., Rojo-Rodriguez, E., Sanchez, A., Saucedo, D., Munoz-Vazquez, A. "Robust geometric navigation of a quadrotor UAV on SE(3)", Robotica 38 (6), pp. 1019 – 1040, **2020**. DOI: 10.1017/S0263574719001231

[14] Ou, J., Guo, X., Lou, W., Zhu, M. "Quadrotor autonomous navigation in semi-known environments based on deep reinforcement learning", Remote Sensing 13 (21), 4330, **2021**. DOI: 10.3390/RS13214330

[15] Zahran, S., Moussa, A., Moussa, A., El-Sheimy, N. "Enhanced uav navigation using hall-magnetic and air-mass flow sensors in indoor environment", ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 187 – 194, **2019**. DOI: 10.5194/ISPRS-ANNALS-IV-2-W5-187-2019

[16] Zahran, S., Masiero, A., Mostafa, M. M., Moussa, A., Moussa, A., Vettore, A., El-Sheimy, N. "Uavs enhanced navigation in outdoor gnss denied environment using uwb and monocular camera systems", ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4213, pp. 665 – 672, **2019**. DOI: 10.5194/ISPRS-ARCHIVES-XLII-2-W13-665-2019

[17] Khosiawan, Y., Park, Y., Moon, I., Nilakantan, J. M., Nielsen, I. "Task scheduling system for UAV operations in indoor environment", Neural Computing and Applications 31, pp. 5431 – 5459, **2019**. DOI: 10.1007/s00521-018-3373-9

[18] Grzonka, S., Grisetti, G., Burgard, W. "A fully autonomous indoor quadrotor", IEEE Transactions on Robotics 28 (1), pp. 90 – 100, **2012**. DOI: 10.1109/TRO.2011.2162999

[19] Sariff, N., Buniyamin, N. "An overview of autonomous mobile robot path planning algorithms", 2006 4th Student Conference on Research and Development, pp. 183 – 188, **2006**. DOI: 10.1109/SCORED.2006.4339335

[20] Kothari, M., Postlethwaite, I., Gu, D. W. "Multi-UAV path planning in obstacle rich environments using rapidlyexploring random trees", Joint 48th IEEE Conference on Decision and Control, pp. 15 – 18, **2009**. DOI: 10.1109/CDC.2009.5400108

[21] Knispel, L., Matousek, R. "A performance comparison of rapidly-exploring random tree and dijkstra’s algorithm for holonomic robot path planning", M. S. Thesis, Brno University of Technology, Brno, Czech Republic, **2013**.

[22] Kijovský, J., Križan, P., Hanko, L., Svátek, M., Bábics, J. "Simulation of the Workspace and Manipulability of An Industrial Robot For Research in Robotic Milling Needs," Strojnícky časopis – Journal of Mechanical Engineering 75 (1), pp. 49 – 60, **2025**. DOI: 10.2478/scjme-2025-0006

[23] Nemčík, J., Šoltés, L., Marek, G., Ivan, K. "Edge to Cloud Task Offloading Optimization in Internet of Vehicles Networks", Strojnícky časopis – Journal of Mechanical Engineering 75 (1), pp. 123 – 134, **2025**. DOI: 10.2478/scjme-2025-0013