



Original Research Paper

Automatic Navigation of Aerial Robot Based on Obstacle Avoidance Using Improved 3D Vector Field Histogram Algorithm

M. Navabi¹  and Amir Hossein Emtiazipur^{2*} 

Faculty of New Technologies Engineering, Shahid Beheshti University, Tehran, Iran

ARTICLE INFO**Article History:**

Received 03 September 2025

Revised 11 November 2025

Accepted 13 December 2025

Available Online 16 December 2025

Keywords:

Autonomous navigation

Aerial robot

Obstacle avoidance algorithm

3D vector field histogram

PID Control

ABSTRACT

In this paper, autonomous navigation of an aerial robot based on obstacle avoidance for outdoor applications has been developed. First, the dynamic model of a quadrotor as an aerial robot is presented using equations that determine linear and angular accelerations derived from Newton's and Euler's equations of motion. The performance of the quadrotor is then evaluated using transient response parameters such as settling time, rise time, and overshoot percentage. Subsequently, a linear Proportional–Integral–Derivative (PID) controller is designed for altitude and attitude control of the quadrotor. To determine the optimal control gains, the criteria of Integral of Squared Error (ISE) and Integral of Absolute Error (IAE) are employed. Furthermore, for autonomous navigation of the quadrotor, an improved three-dimensional vector field histogram method is used to develop an obstacle avoidance algorithm. Through simulation, the validity of the developed dynamic model is first verified, followed by the implementation of the altitude and attitude controllers. The results show that the quadrotor achieves the desired altitude and attitude with favorable transient characteristics. Finally, the obstacle avoidance algorithm is simulated with distance thresholds of 0.5 meters and 1 meter from obstacles. The simulation results demonstrate that, with a one-meter threshold, the algorithm ensures autonomous navigation of the quadrotor without collisions. Therefore, the developed method can be suitable for outdoor environments with small obstacles.

*Corresponding Author's E-mail: s.emtiazipur@sbu.ac.ir.**How to Cite this Article:**

M. Navabi and A.H. Emtiazipur, "Automatic navigation of aerial robot based on obstacle avoidance using improved 3d vector field histogram algorithm," *Journal of Technology in Aerospace Engineering*, Vol. 9, No. 4, pp. 83-95, 2025, <http://doi.org/10.22034/jtae.2025.9.4.7>

COPYRIGHTS

Authors retain the copyright and full publishing rights.

Published by ARI. This article is an open-access article licensed under [the Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)



مسیریابی خودکار ربات هوایی مبتنی بر اجتناب از موانع با استفاده از الگوریتم هیستوگرام میدان برداری سه بعدی بهبودیافته

محمد نوابی^۱ و سیدامیرحسین امتیازی پور^{۲*} 

۱- دانشکده مهندسی فناوری‌های نوین، دانشگاه شهید بهشتی، تهران، ایران

۲- دانشجوی دکتری، دانشکده مهندسی فناوری‌های نوین، دانشگاه شهید بهشتی، تهران، ایران

چکیده

در این مقاله، مسیریابی خودکار ربات هوایی مبتنی بر اجتناب از موانع برای کاربردهای فضای باز توسعه داده شده است. بدین منظور، ابتدا مدل دینامیکی کوادروتور به عنوان یک ربات هوایی با استفاده از معادلات تعیین کننده شتاب‌های خطی و زاویه‌ای با بهره‌گیری از معادلات حرکت نیوتن و اویلر ارائه شده است. سپس عملکرد آن با استفاده از پارامترهای حالت گذرا مانند زمان نشست، زمان برخاست و درصد فراجهش ارزیابی شده است. در ادامه، کنترل کننده خطی از نوع «تناسبی، مشتقی، انتگرالی» برای کنترل ارتفاع و کنترل وضعیت کوادروتور طراحی شده است. برای تعیین ضرایب کنترلی بهینه از معیارهای انتگرال مربع خطا و انتگرال زمانی مطلق خطا استفاده شده است. همچنین به منظور مسیریابی خودکار کوادروتور یک الگوریتم اجتناب از موانع با استفاده از روش هیستوگرام میدان برداری سه بعدی بهبود یافته در پنج مرحله توسعه داده شده است. با استفاده از شبیه‌سازی ابتدا صحت مدل دینامیکی توسعه داده شده برای کوادروتور اعتبارسنجی می‌شود، بعد از آن کنترل ارتفاع و وضعیت ارائه شده پیاده‌سازی شده است. نتایج نشان می‌دهد کوادروتور به ارتفاع و وضعیت مطلوب با ویژگی‌های گذرای مناسب دست یافته است. در آخر الگوریتم اجتناب از موانع با در نظر گرفتن آستانه فاصله نیم متر و یک متر با موانع شبیه‌سازی شده است نتایج شبیه‌سازی نشان می‌دهد این الگوریتم با در نظر گرفتن آستانه فاصله یک متر با موانع، مسیریابی خودکار کوادروتور را بدون برخورد با موانع تضمین می‌کند. بنابراین، روش توسعه داده شده می‌تواند برای کاربرد در محیط‌های باز با موانع کوچک مناسب باشد.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۲ شهریور ۱۴۰۴
بازنگری ۲۰ آبان ۱۴۰۴
پذیرش ۲۲ آذر ۱۴۰۴
اولین انتشار ۲۵ آذر ۱۴۰۴

واژه‌های کلیدی:

مسیریابی خودکار
ربات هوایی
الگوریتم اجتناب از برخورد با موانع
الگوریتم هیستوگرام میدان برداری سه بعدی
کنترل تناسبی، مشتقی، انتگرالی

پست الکترونیکی نویسنده مسئول: s_emtiazipour@sbu.ac.ir

How to Cite this Article:

M. Navabi and A.H. Emtiazipour, "Automatic navigation of aerial robot based on obstacle avoidance using improved 3d vector field histogram algorithm," *Journal of Technology in Aerospace Engineering*, Vol. 9, No. 4, pp. 83-95, 2025, <http://doi.org/10.22034/jtae.2025.9.4.7>

COPYRIGHTS

Authors retain the copyright and full publishing rights.

Published by ARI. This article is an open-access article licensed under [the Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)



۱ مقدمه

در عصر حاضر ربات‌های هوایی به دلیل کاربردهای متنوعشان در حوزه‌های غیرنظامی و نظامی، از سوی محققان دانشگاهی و صنایع مورد توجه زیادی قرار گرفته‌اند. کاربردهای مختلف این ربات‌ها شامل موارد مختلفی چون نظارت هوایی و مرزی، نقشه برداری، عملیات جستجو و نجات، جمع‌آوری و انتقال داده‌ها، حمل و نقل و... می‌شود [۱]-[۴].

ربات‌های هوایی عمودپرواز مانند کوادروتورها به دلیل سهولت در راه‌اندازی، مانورپذیری بالا و قابلیت برخاست و فرود عمودی، از جمله پرکاربردترین سیستم‌های پروازی محسوب می‌شوند. کنترل پایدار و دقیق موقعیت و وضعیت سیستم کوادروتور برای پاسخگویی به نیازهای کاربران مختلف و الزامات حوزه‌های کاربردی متنوع، ضروری است [۳]. در مقاله [۴] یک نمای کلی از طبقه‌بندی ربات‌های هوایی و پیشرفت‌های یک دهه اخیر را ارائه می‌دهد و در ادامه به بیان دینامیک، ابزارهای شبیه‌سازی کاربردی و استراتژی‌های کنترل برای یک کوادروتور می‌پردازد. در مطالعه‌های انجام شده بر روی کوادروتورها، از کنترل‌کننده‌های پروازی مختلفی برای کنترل سیستم کوادروتور استفاده شده است. این کنترل‌کننده‌ها را می‌توان به سه دسته کلی کنترل‌کننده‌های خطی، کنترل‌کننده‌های غیرخطی و کنترل‌کننده‌های مبتنی بر یادگیری تقسیم کرد.

کنترل‌کننده خطی «تناسبی-انترگالی-مشتقی» (PID) یکی از محبوب‌ترین الگوریتم‌های کنترلی در صنعت است که به دلیل سادگی طراحی و سهولت پیاده‌سازی، بسیار مورد استفاده قرار می‌گیرد. بررسی‌ها نشان داده‌اند که کنترل‌کننده PID بیش از ۹۰٪ از سیستم‌های کنترلی صنعتی را شامل می‌شوند [۵]. این کنترل‌کننده، بسته به مقادیر پارامترهای کنترلی، عملکرد رضایت‌بخشی را برای بیشتر کاربردها فراهم می‌کند. در مرجع [۶] تنظیم پارامترهای کنترلگر PID با استفاده از الگوریتم‌های هوشمند تطبیقی و غیرتطبیقی برای کوادروتور انجام شده است.

ناوبری خودکار در ربات‌های هوایی منجر به افزایش قابل توجه قابلیت‌های عملیاتی آن‌ها می‌شود [۷] و [۸]. در مرجع [۷]، یک حرکت خودکار جدید الهام گرفته از طبیعت با استفاده از الگوریتم زنبور عسل برای ربات‌های هوایی مورد استفاده قرار گرفته است و در مرجع [۸] یک سیستم هدایت هوشمند و خودران برای ربات‌های هوایی بر اساس حرکت دسته جمعی ملخ‌ها و با استفاده از منطق فازی ارائه شده است. این سیستم با الهام از رفتار بهینه اجتماعی ملخ‌ها طراحی شده است که منجر به توسعه الگوریتمی برای بهینه‌سازی مسیر حرکت گروهی ربات‌ها می‌شود.

در ناوبری خودکار کوادروتورها، ممکن است به دلایل مختلفی، از جمله شرایط نامساعد آب و هوایی، محیط عملیات و موانع متحرک، برخورد رخ دهد. بنابراین، اجتناب از موانع، یکی از اصول اساسی در ناوبری خودکار کوادروتورها محسوب می‌شود. از این رو، تحقیقات بسیاری برای اطمینان از ناوبری خودکار بدون برخورد ربات‌ها در حال انجام است [۹]-[۱۴].

در مقاله [۹] مروری جامعی بر استراتژی‌های جلوگیری از برخورد با موانع انجام شده است. همچنین در این مقاله چالش‌های موجود و مسائل باز در این حوزه بررسی شده است. ربات‌های هوایی برای تحقق اجتناب از برخورد، نیازمند سیستم‌هایی هستند که بتوانند محیط اطراف را حس کرده، داده‌های مربوطه را پردازش و مسیر بهینه را برای حرکت انتخاب کنند. این سیستم‌ها معمولاً ترکیبی از فناوری‌های حسگر و الگوریتم‌های مسیریابی پیشرفته هستند [۱۰].

مراجع [۱۱]-[۱۳] از حسگر لیدار (LiDAR) به عنوان حسگر تشخیص موانع استفاده نموده‌اند. مرجع [۱۱] به بررسی توسعه یک سیستم اجتناب از برخورد برای پهپاد هگزاکوپتر با استفاده از داده‌های لیدار در محیط‌های پویا پرداخته است. در این مقاله نویسندگان از حسگر لیدار برای جمع‌آوری داده‌های سه‌بعدی محیطی استفاده کرده‌اند تا موانع را شناسایی و موقعیت آن‌ها را تعیین کنند. سیستم پیشنهادی در محیط‌های شبیه‌سازی شده و واقعی مورد آزمایش قرار گرفته تا کارایی آن در شناسایی موانع، زمان واکنش و اجتناب از برخورد با موانع متحرک ارزیابی شود. این مقاله نشان می‌دهد که استفاده از داده‌های لیدار می‌تواند به‌طور مؤثری در توسعه سیستم‌های اجتناب از برخورد برای ربات‌های هوایی در محیط‌های پویا مفید باشد. مقاله [۱۲] به توسعه یک سیستم تشخیص و اجتناب از موانع برای کوادروتورهای کوچک پرداخته است که با استفاده از حسگر لیدار، موانع را شناسایی و از آن‌ها اجتناب می‌کند. این مقاله نشان می‌دهد که استفاده از حسگر لیدار می‌تواند به بهبود ایمنی و کارایی کوادروتورهای کوچک در محیط‌های پیچیده کمک کند. در مقاله [۱۳] یک روش سه لایه‌ای و ایمن برای جلوگیری از برخورد مبتنی بر داده‌های لیدار اتخاذ شده است و الگوریتم پیشنهادی در دو محیط با موانع ایستا و پویا برای یک و چند ربات ارزیابی شده است. در مقاله [۱۴] یک چارچوب سه‌مرحله‌ای برای اجتناب خودکار ربات‌های هوایی از موانع ارائه شده که شامل ادراک محیط، الگوریتم اجتناب و کنترل حرکت است. در این مقاله از الگوریتم هیستوگرام برداری (VFH) برای برنامه‌ریزی مسیر و اجتناب از موانع در زمان واقعی استفاده شده است. شبیه‌سازی‌ها و آزمایش‌های میدانی نشان می‌دهند که سیستم پیشنهادی عملکرد خوبی در اجتناب از موانع دارد.

الگوریتم VFH+ یکی از روش‌های پیشرفته و مؤثر در این زمینه است. این الگوریتم نسخه بهبودیافته‌ای از الگوریتم VFH اصلی است و بر پایه ساخت هیستوگرام موانع و استفاده از یک رویکرد قطبی برای

در شکل ۱ f_i نیروی پیشرانه‌ای است که بر روتور اعمال می‌شود و ω_i سرعت زاویه‌ای روتور است. نیروی پیشرانه f_i و گشتاور روتور τ_i به کمک روابط ۱ و ۲ با مربع سرعت زاویه‌ای روتور مرتبط می‌شوند:

$$f_i = b\omega_i^2 \quad (1)$$

$$\tau_i = d\omega_i^2 \quad (2)$$

۲-۱ ورودی کنترلی

برای دستیابی به حرکت موردنظر کوادروتور، نیروی پیشرانه تولید شده توسط چهار موتور با استفاده از روابط ۳ تا ۶ بیان می‌شوند:

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (3)$$

$$U_2 = b(\omega_4^2 - \omega_2^2) \quad (4)$$

$$U_3 = b(\omega_3^2 - \omega_1^2) \quad (5)$$

$$U_4 = d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \quad (6)$$

که در آن U_1 نشان‌دهنده نیروی پیشرانه کل تولید شده توسط روتورها است و U_2 ، U_3 و U_4 به ترتیب نیروی پیشرانه کل ایجاد شده برای دستیابی به حرکات رول، پیچ، و یاو را نشان می‌دهند. متغیرهای ω_1 ، ω_2 ، ω_3 و ω_4 به ترتیب بیانگر سرعت‌های زاویه‌ای روتورهای ۱ تا ۴ هستند. همچنین، b و d به ترتیب نشان‌دهنده ضرایب نیروی پیشرانه و نیروی پسا هستند.

ورودی‌های کنترلی به صورت ماتریسی در رابطه ۷ توصیف شده‌اند:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (7)$$

سرعت زاویه‌ای باقی‌مانده کلی ω_r هنگامی که کوادروتور حرکات رول یا پیچ را انجام می‌دهد، به صورت رابطه ۸ تعریف می‌شود:

$$\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \quad (8)$$

۲-۲ معادلات حرکت کوادروتور

مدل دینامیکی کوادروتور در قالب معادلات تعیین کننده شتاب‌های خطی و زاویه‌ای ارائه می‌شود که به کمک معادله حرکت نیوتن و اویلر به دست آمده‌اند. معادلات حرکت خطی به صورت زیر روابط ۹ تا ۱۰ بیان می‌شوند:

$$\ddot{x} = \frac{U_1}{m} [\cos\phi \cdot \sin\theta \cdot \cos\psi - \sin\phi \cdot \sin\psi] \quad (9)$$

$$\ddot{y} = \frac{U_1}{m} [\cos\phi \cdot \sin\theta \cdot \sin\psi - \sin\phi \cdot \cos\psi] \quad (10)$$

$$\ddot{z} = \frac{U_1}{m} [\cos\phi \cdot \cos\theta] - g \quad (11)$$

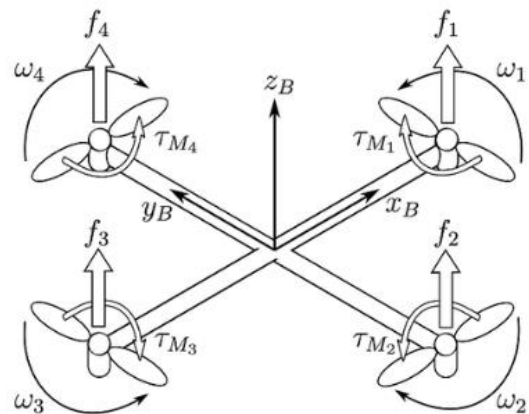
همچنین، معادلات تعیین کننده زوایای اویلر که شتاب‌های زاویه‌ای کوادروتور را تعریف می‌کنند، به صورت روابط ۱۲ تا ۱۴ بیان می‌شوند:

هدایت ربات طراحی شده است. ویژگی بارز VFH+ توانایی آن در فیلترکردن نویزهای محیطی و ایجاد مسیرهای نرم و پیوسته است که برای حرکت روان‌تر و جلوگیری از توقف‌های ناگهانی یا حرکات غیرطبیعی در ربات بسیار مؤثر است [۱۵]-[۱۷]. الگوریتم VFH+ در نسخه‌های دوبعدی و سه‌بعدی برای ناوبری و اجتناب از موانع طراحی شده است. این دو نسخه از نظر کاربرد و نحوه پردازش داده‌ها تفاوت‌های قابل توجهی دارند. الگوریتم VFH+ دوبعدی برای ربات‌هایی که در محیط‌های ساده و مسطح حرکت می‌کنند مناسب است. در مقابل، الگوریتم VFH+ سه‌بعدی برای وسایل نقلیه و ربات‌هایی که در محیط‌های پیچیده و سه‌بعدی فعالیت دارند، طراحی شده است. نسخه سه‌بعدی از لحاظ پردازش و مدل‌سازی پیچیده‌تر است، اما برای کاربردهای پیشرفته‌تر، مانند مسیریابی ربات‌های هوایی و زیردریایی، ضروری است [۱۷].

در پژوهش حاضر به مسیریابی خودکار ربات هوایی مبتنی بر اجتناب از موانع برای کاربردهای فضای باز پرداخته شده است. بدین منظور، ابتدا مدل دینامیکی کوادروتور به عنوان یک ربات هوایی استخراج شده است و عملکرد آن در حالت گذرا و حالت ماندگار مورد بررسی قرار داده شده است. در ادامه، دو کنترل کننده خطی از نوع «تتاسبی، مشتقی، انتگرالی» برای کنترل ارتفاع و وضعیت کوادروتور طراحی شده است. به منظور مسیریابی خودکار کوادروتور یک الگوریتم اجتناب از موانع با استفاده از روش هیستوگرام میدان برداری سه بعدی بهبود یافته، توسعه داده شده است. همچنین به منظور تشخیص موانع از مدل حسگر لیدار استفاده شده است.

۲ مدل‌سازی کوادروتور

کوادروتور از چهار موتور به عنوان عملگر تشکیل شده است که با فاصله‌های مساوی و زاویه ۹۰ درجه از یکدیگر قرار گرفته‌اند. این عملگرها به صورت جداگانه کنترل می‌شوند تا نیروی پیشرانه نسبی تولید کنند. شکل ۱ ساختار کوادروتور را به همراه چارچوب ثابت بدنه، سرعت زاویه‌ای روتورها و نیروهای پیشرانه‌ای که بر آن اعمال می‌شوند نشان می‌دهد.



شکل ۱- ساختار کوادروتور.

Fig. 1. Quadrotor structure.

کوادروتور در قالب زوایای رول، پیچ و یاو به صورت روابط ۱۷ تا ۱۹ تعیین می‌شوند:

$$U_2 = K_p(\phi_d - \phi_a) + K_i \int_0^t (\phi_d - \phi_a) dt + K_d(\dot{\phi}_d - \dot{\phi}_a) \quad (17)$$

$$U_3 = K_p(\theta_d - \theta_a) + K_i \int_0^t (\theta_d - \theta_a) dt + K_d(\dot{\theta}_d - \dot{\theta}_a) \quad (18)$$

$$U_4 = K_p(\psi_d - \psi_a) + K_i \int_0^t (\psi_d - \psi_a) dt + K_d(\dot{\psi}_d - \dot{\psi}_a) \quad (19)$$

همچنین، قوانین کنترلی برای کنترل‌کننده‌های موقعیت x و y به صورت روابط ۲۰ و ۲۱ تعریف می‌شوند:

$$U_x = K_p(x_d - x_a) + K_i \int_0^t (x_d - x_a) dt + K_d(\dot{x}_d - \dot{x}_a) \quad (20)$$

$$U_y = K_p(y_d - y_a) + K_i \int_0^t (y_d - y_a) dt + K_d(\dot{y}_d - \dot{y}_a) \quad (21)$$

مقادیر K_p ، K_i و K_d عملکرد کنترل‌کننده را تعیین می‌کنند و می‌توان آن‌ها را با استفاده از روش‌های مختلف تنظیم به دست آورد. برای تعیین این مقادیر می‌توان از معیارهای انتگرال مربع خطا (ISE) و انتگرال زمانی مطلق خطا (ITAE) ارائه شده در روابط ۲۲ و ۲۳ استفاده کرد:

$$ITAE = \int_0^t |e(t)| dt \quad (22)$$

$$ISE = \int_0^t e(t)^2 dt \quad (23)$$

که در آن t زمان شبیه‌سازی و $e(t)$ خطایی است که بین خروجی مطلوب و خروجی کنترل‌شده واقعی محاسبه می‌شود.

۴ الگوریتم هیستوگرام میدان برداری سه بعدی بهبود یافته

الگوریتم VFH+ یکی از روش‌های پیشرفته و مؤثر در ارائه مسیری بهینه و بدون برخورد با موانع می‌باشد. این الگوریتم نسخه بهبود یافته‌ای از الگوریتم VFH اصلی است و بر پایه ساخت هیستوگرام موانع و استفاده از یک رویکرد قطبی برای هدایت ربات طراحی شده است [۱۶].

در الگوریتم VFH+ محیط سه‌بعدی ربات به صورت یک حجم سلولی مدل‌سازی می‌شود. این الگوریتم با ترکیب داده‌های دریافتی از حسگرها (مانند لیدار، رادار، یا دوربین‌های سه‌بعدی)، مسیریابی ممکن را براساس فاصله از موانع و هدف مشخص می‌کند. در هر

$$\ddot{\phi} = \frac{1}{I_x} [\dot{\theta}\dot{\psi}(I_z - I_y) - J_r\dot{\theta}\Omega + lU_2] \quad (12)$$

$$\ddot{\theta} = \frac{1}{I_y} [\dot{\theta}\dot{\psi}(I_x - I_z) - J_r\dot{\phi}\Omega + lU_3] \quad (13)$$

$$\ddot{\psi} = \frac{1}{I_z} [\dot{\theta}\dot{\psi}(I_y - I_x) + lU_4] \quad (14)$$

که در آن (x, y, z) و $(\dot{x}, \dot{y}, \dot{z})$ به ترتیب موقعیت‌ها و سرعت‌های خطی در امتداد محورهای مختصات را نشان می‌دهند. همچنین، متغیرهای (ϕ, θ, ψ) و $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ به ترتیب موقعیت و سرعت زاویه‌ای کوادروتور در راستای زوایای رول، پیچ و یاو را مشخص می‌کنند.

در روابط بالا، متغیر g شتاب ناشی از جاذبه، J_r گشتاور اینرسی موتورها، m جرم کوادروتور، و l فاصله از مرکز جرم تا هر روتور را نشان می‌دهند. همچنین I_x ، I_y و I_z به ترتیب نشان‌دهنده ممان‌های اینرسی حول محورهای x ، y و z هستند.

زوایای اوایلر کوادروتور به $-\pi/2 \leq \phi \leq \pi/2$ ، $-\pi/2 \leq \theta \leq \pi/2$ و $-\pi \leq \psi \leq \pi$ محدود شده‌اند تا از تکیگی‌ها و چرخش‌های بیش از حد جلوگیری شود. با توجه به ورودی‌های کنترلی و معادلات حرکت می‌توان نتیجه گرفت که مدل کوادروتور دارای شش خروجی $(x, y, z, \phi, \theta, \psi)$ و چهار ورودی مستقل است. بنابراین، این سیستم به‌عنوان یک سیستم کم‌عملگر در نظر گرفته می‌شود.

۳ کنترل‌کننده PID

قانون کنترلی برای کنترل‌کننده PID به صورت رابطه تعریف می‌شود:

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (15)$$

که در آن $U(t)$ سیگنال کنترلی، $e(t)$ خطای بین خروجی مطلوب و خروجی واقعی، K_p ضریب بهره تناسبی، K_i ضریب بهره انتگرالی، و K_d ضریب بهره مشتقی است.

کنترل‌کننده ارتفاع، با استفاده از سیگنال خطای بین ارتفاع مطلوب و واقعی، یک سیگنال کنترلی تولید می‌کند تا به کمک آن رفتار سیستم را بر اساس قانون کنترلی ارائه شده در رابطه ۱۶ تغییر دهد.

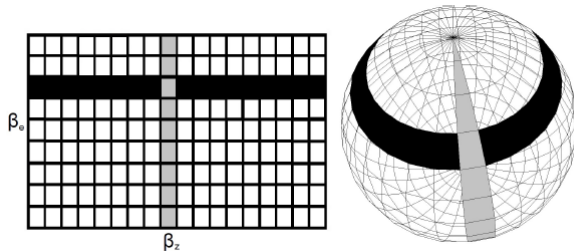
$$U_1 = K_p(z_d - z_a) + K_i \int_0^t (z_d - z_a) dt + K_d(\dot{z}_d - \dot{z}_a) \quad (16)$$

به‌طور مشابه، کنترل‌کننده وضعیت با استفاده از خطای بین موقعیت زاویه‌ای مطلوب و واقعی کوادروتور (ϕ, θ, ψ) ، سیستم را به موقعیت هدف مطلوب که به صورت مختصات x و y تعریف شده است، هدایت می‌کند. قوانین کنترلی مورد استفاده برای تغییر وضعیت

ارتفاع و عمق) است و نقطه مرکزی وسیله (Vehicle Center Point) به عنوان مرکز این جعبه در نظر گرفته می‌شود. VCP نقطه مرکزی ربات است و کل ربات توسط این نقطه نمایش داده می‌شود. اگر یک حجم سلولی در داخل جعبه مرزی قرار داشته باشد، آن حجم به عنوان یک سلول فعال ($C_{i,j,k}$) با مختصات i, j, k در منطقه فعال (C_a) شناخته می‌شود. در مرحله کاوش تنها نیاز است که حجم‌های سلولی پیدا شوند که در مرزهای جعبه مرزی قرار دارند. پس از آن که یک حجم سلولی در مرحله اول پیدا شد، در مرحله دوم از اطلاعات آن حجم برای ایجاد هیستوگرام قطبی اولیه دوبعدی استفاده خواهد شد.

۴-۲ مرحله دوم: هیستوگرام قطبی اولیه دوبعدی

در مرحله دوم، اطلاعات مربوط به حجم سلولی (که در مرحله اول شناسایی شده است) به هیستوگرام قطبی اولیه دوبعدی اضافه می‌شود. این هیستوگرام، همان‌طور که در شکل ۳ نشان داده شده است، یک هیستوگرام قطبی است که موقعیت یک حجم سلولی فعال را بر اساس دو زاویه تعیین می‌کند. این زوایا بر اساس موقعیت حجم سلولی و VCP محاسبه می‌شوند. با استفاده از این دو زاویه به عنوان مختصات، یک وزن که بر اساس ویژگی‌های حجم سلولی محاسبه شده است، در هیستوگرام قطبی اولیه دوبعدی قرار می‌گیرد.



شکل ۳- هیستوگرام قطبی دوبعدی.

Fig. 3. 2D Polar Histogram.

ابتدا، لیست سلول‌های فعال با ایجاد یک کره مرزی در داخل جعبه مرزی کاهش می‌یابد. این کار با محاسبه فاصله اقلیدسی $d_{i,j,k}$ بین VCP و هر حجم سلولی انجام می‌شود. اگر فاصله اقلیدسی بیشتر از شعاع کره مرزی باشد ($d_{i,j,k} > w_s/2$)، سلول فعال نادیده گرفته می‌شود. ایجاد کره مرزی برای ساخت یک هیستوگرام قطبی دوبعدی مستقل از دوران ضروری است.

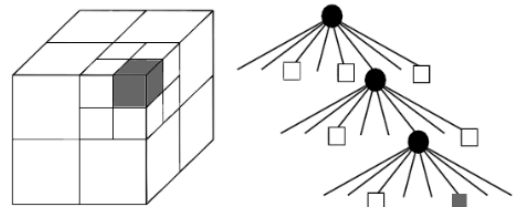
در گام بعدی، الگوریتم دو زاویه قطبی را برای تعیین مختصات در هیستوگرام قطبی (H^p) اولیه دوبعدی محاسبه می‌کند. این زوایا عبارتند از:

۱- زاویه آزیموت β_z که محور x هیستوگرام قطبی دوبعدی را مشخص می‌کند.

مرحله، ربات نه تنها مسیریابی با کمترین احتمال برخورد با موانع را انتخاب می‌کند، بلکه با استفاده از معیارهای دیگری مانند فاصله تا هدف، پیچیدگی مسیر، و دینامیک سیستم، مسیری بهینه را ارائه می‌دهد.

چارچوب نقشه اکتالی یک چارچوب نقشه‌برداری شبکه اشغالی (Occupancy Grid) سه‌بعدی است که بر اساس ساختار درخت اکتالی طراحی شده است. ساختار داده‌ای درخت اکتالی یک ساختار سلسله‌مراتبی شامل چندین حجم سلولی است. این حجم‌های سلولی (که به آنها گره نیز گفته می‌شود) حجم‌های مکعبی در فضا هستند. هر حجم سلولی می‌تواند شامل ۸ یا ۰ حجم سلولی فرزند باشد. هنگامی که یک حجم سلولی شامل ۸ حجم فرزند اشغال شده باشد، نقشه اکتالی این ۸ فرزند را به یک حجم سلولی والد اشغال شده، کاهش می‌دهد. اگر هیچ حجم سلولی فرزندی اشغال نشده باشد، حجم‌های سلولی به یک حجم والد که آزاد است، کاهش پیدا می‌کنند. تمام ترکیب‌های میان این دو حالت با استفاده از ۸ حجم سلولی فرزند ذخیره می‌شود.

تعداد لایه‌ها در این ساختار سلسله‌مراتبی (که به آن وضوح درخت اکتالی نیز گفته می‌شود) دقت و اندازه درخت اکتالی را تعیین می‌کند. بدین ترتیب، کل محیط با استفاده از حجم‌های سلولی نمایش داده می‌شود. این موضوع در شکل ۲ نشان داده شده است.



شکل ۲- ساختار درخت اکتالی.

Fig. 2. Octree Structure.

الگوریتم VFH+ سه‌بعدی شامل پنج مرحله است که برای محاسبه یک حرکت جدید برای ربات به کار می‌رود. این مراحل در ادامه به تفصیل شرح داده می‌شوند [۱۷].

۴-۱ مرحله اول: کاوش در نقشه اکتالی

ساختار داده‌ای نقشه اکتالی از ساختار درخت اکتالی استفاده می‌کند. هنگامی که ربات در یک محیط بزرگ حرکت می‌کند، به دلیل محدودیت‌های محاسباتی امکان کاوش تمامی حجم‌های سلولی وجود ندارد. بنابراین، مرحله کاوش در درخت اکتالی تنها حجم‌های سلولی را بررسی می‌کند که در داخل یک جعبه مرزی در اطراف ربات قرار دارند. این جعبه مرزی دارای ابعاد $w_s \times w_s \times w_s$ (عرض،

زاویه الیوشن و مقدار p با استفاده از روابط ۲۵ و ۲۶ محاسبه می‌شوند:

$$\beta_e = \text{floor} \left(\frac{1}{\alpha} \arctan \frac{z_i - z_0}{p} \right) \quad (25)$$

$$p = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad (26)$$

در نتیجه:

$$\beta_e = \text{floor} \left(\frac{1}{\alpha} \arctan \frac{z_i - z_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right) \quad (27)$$

در محاسبات قبلی از نقطه VCP و مرکز حجم سلولی برای محاسبه دو زاویه استفاده شد. برای جبران ابعاد ربات، اندازه ربات با بزرگ‌تر کردن تمام حجم‌های سلولی فعال در هیستوگرام قطبی اولیه دوبعدی به مقدار r_{r+s+v} در نظر گرفته می‌شود. مقدار عامل بزرگ‌سازی با جمع شعاع ربات r_r ، شعاع ایمنی r_s ، و اندازه حجم سلولی r_v محاسبه می‌شود. بنابراین، حجم‌های سلولی بزرگ‌تر می‌شوند تا الگوریتم بتواند زوایای بیشینه و کمینه‌ای را که حجم سلولی در آن قرار می‌گیرد، مطابق رابطه ۲۸ محاسبه کند:

$$\lambda_{i,j,k} = \text{floor} \left(\frac{1}{\alpha} \arctan \frac{r_{r+s+v}}{d_{i,j,k}} \right) \quad (28)$$

حداقل فاصله بین حجم سلولی و نقطه دید مجازی VCP را می‌توان با کسر مقدار بزرگ‌سازی از فاصله اقلیدسی طبق رابطه ۲۹ محاسبه کرد:

$$l_{i,j,k} = d_{i,j,k} - r_{r+s+v} \quad (29)$$

پس از تعیین این که کدام سلول‌ها در هیستوگرام قطبی اولیه دوبعدی تحت تأثیر حجم سلولی هستند، الگوریتم باید وزن حجم سلولی را که به هیستوگرام دوبعدی اضافه می‌شود محاسبه کند. وزن یک حجم سلولی بر اساس فاصله اقلیدسی $l_{i,j,k}$ و میزان قطعیت اشغال (Occupancy Certainty) آن $o_{i,j,k}$ طبق رابطه ۳۰ محاسبه می‌شود:

$$\text{if } e \in \left[\beta_e - \frac{\lambda}{\alpha}, \beta_e + \frac{\lambda}{\alpha} \right] \text{ and } z \in \left[\beta_z - \frac{\lambda}{\alpha}, \beta_z + \frac{\lambda}{\alpha} \right] \quad (30)$$

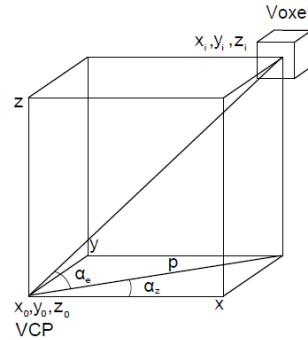
$$H_{z,e}^p = \sum_{i,j,k \in C_a} (o_{i,j,k})^2 (a - b l_{i,j,k})$$

در غیر این صورت صفر می‌باشد.

دو ثابت a و b با استفاده از رابطه ۳۱ محاسبه می‌شوند تا نسبت متوازن a و b به دست آید. لازم به ذکر است که مقدار عددی هر کدام از این ثوابت مهم نیست، بلکه اندازه نسبی یا کسری آن‌ها اهمیت دارد:

$$a - b \left(\frac{w_s - 1}{2} \right)^2 = 1 \quad (31)$$

۲- زاویه الیوشن β_e که محور y هیستوگرام قطبی دوبعدی را مشخص می‌کند.
این زوایا در شکل ۴ با α_e و α_z نشان داده شده‌اند.



شکل ۴- چگونگی محاسبه زوایای آزیموت و الیوشن.

Fig. 4. An image that clarifies the Azimuth and elevation angle calculations.

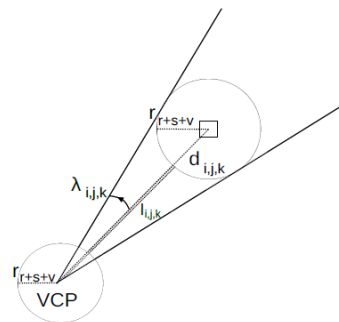
زاویه آزیموت با استفاده از رابطه ۲۴ محاسبه می‌شود که در الگوریتم VFH+ نیز به کار رفته است:

$$\beta_z = \text{floor} \left(\frac{1}{\alpha} \arctan \frac{x_i - x_0}{y_i - y_0} \right) \quad (24)$$

$$\beta_z = \text{floor} \left(\frac{1}{\alpha} \arctan \frac{x_i - x_0}{y_i - y_0} \right) \quad (24)$$

ترکیب تابع Floor و تقسیم‌بندی بر اساس وضوح هیستوگرام قطبی دوبعدی α یک عدد طبیعی ایجاد می‌کند که یک پیش‌نیاز برای هیستوگرام قطبی دوبعدی اولیه است. وضوح این هیستوگرام براساس اختلاف بین بزرگ‌ترین زاویه و کوچک‌ترین زاویه تعیین می‌شود که به همان سلول در هیستوگرام قطبی دوبعدی منجر می‌شود.

از آن‌جاکه زاویه الیوشن باید مستقل از زاویه آزیموت باشد، زاویه الیوشن به روش متفاوتی محاسبه می‌شود. این امر می‌تواند با محاسبه p بر اساس مختصات x و y نقطه VCP و حجم سلولی مطابق شکل ۵ انجام شود.



شکل ۵- بزرگ‌سازی حجم‌های سلولی.

Fig. 5. Enlargement of the voxels

$$d_r < (r_r + r_{r+s+v}), \quad (36)$$

$$d_l < (r_l + r_{r+s+v}), \quad (37)$$

تابع $\Delta x(i)$ فاصله x بین حجم سلولی و نقطه VCP را محاسبه می‌کند. در گام بعدی، الگوریتم باید حرکت صعودی ربات را نیز در نظر بگیرد. برای این منظور، الگوریتم سلول‌هایی را که در هیستوگرام قطبی ماسک شده دوبعدی توسط حجم سلولی مسدود شده‌اند، محاسبه می‌کند. این کار با محاسبه تمام زوایای α_e که زوایای α_z آن‌ها توسط حجم سلولی مسدود شده‌اند، انجام می‌شود. ابتدا، باید ثابت حرکت صعودی f محاسبه شود. این کار با محاسبه فاصله چرخش t تا مانع با استفاده از رابطه ۳۸ انجام می‌شود:

$$t = \frac{2\pi r(2\alpha_z)}{360} \quad (38)$$

سپس، ثابت حرکت صعودی با استفاده از تفاوت ارتفاع $Z_i - Z_0$ و فاصله چرخش طبق رابطه ۳۹ محاسبه می‌شود:

$$f = \frac{Z_i - Z_0}{p} \quad (39)$$

پس از آن، باید یک ارتفاع غیرقابل دسترس جدید $Z_{\alpha z}$ برای هر زاویه α_z بعد از حجم سلولی محاسبه شود. برای تعیین ارتفاع غیرقابل دسترس، ابتدا فاصله چرخش $t_{\alpha z}$ تا زاویه α_z بعدی و سپس $Z_{\alpha z}$ به کمک روابط ۴۰ و ۴۱ محاسبه می‌شوند:

$$t_{\alpha z} = \frac{2\pi r(2\beta_z \alpha)}{360} \quad (40)$$

$$Z_{\alpha z} = f t_{\alpha z} \quad (41)$$

در نهایت، زاویه الویشن غیرقابل دسترس β_e با استفاده از تفاوت ارتفاع و نیز فاصله اقلیدسی $l_{\alpha z}$ و بر اساس روابط زیر محاسبه می‌شود:

$$l_{\alpha z} = \sqrt{r^4 + 2r^2 \cos(270 - 2\alpha_z)} \quad (42)$$

$$\beta_e = \frac{1}{\alpha} \arctan\left(\frac{Z_{\alpha z}}{l_{\alpha z}}\right) \quad (43)$$

این محاسبات می‌توانند به گونه‌ای گسترش یابند که الگوریتم سلول‌های غیرقابل دسترسی را با توجه به سرعت حرکت رو به جلو و حداکثر سرعت صعود محاسبه کند.

۴-۴ مرحله چهارم: هیستوگرام قطبی باینری دوبعدی

مراحل قبلی یک هیستوگرام قطبی اولیه دوبعدی را بر اساس سلول‌های حجمی نقشه اکتالی تولید می‌کنند. در مرحله چهارم اطلاعات با ایجاد یک هیستوگرام قطبی باینری دوبعدی بر اساس هیستوگرام قطبی اولیه دوبعدی مجدداً کاهش پیدا می‌کنند. این کار با مقایسه هر سلول در هیستوگرام قطبی اولیه دوبعدی با دو آستانه τ_{low} و τ_{high} انجام می‌شود. زمانی که مقدار یک نقطه بیشتر از τ_{high} باشد، آن نقطه در هیستوگرام قطبی باینری دوبعدی برابر با ۱

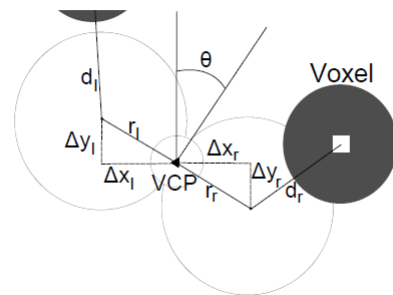
وزن یک حجم سلولی در مرحله چهارم و برای به دست آوردن هیستوگرام قطبی باینری دوبعدی استفاده خواهد شد.

۴-۳ مرحله سوم: ویژگی‌های فیزیکی

پس از مرحله دوم، مرحله سوم اطلاعات جدیدی را بر اساس ویژگی‌های فیزیکی ربات و موقعیت حجم سلولی محاسبه و به هیستوگرام قطبی اولیه دوبعدی اضافه می‌کند. زمانی که ربات در یک جهت مشخص θ که توسط حرکت می‌کند و دارای سرعت است، نمی‌تواند به طور ناگهانی تغییر جهت دهد، بنابراین باید بچرخد. هنگامی که ربات به آرامی حرکت می‌کند، این مرحله نادیده گرفته می‌شود زیرا می‌تواند به طور ناگهانی تغییر جهت دهد. مسیر چرخش ربات به سرعت حرکت رو به جلو، سرعت چرخش و سرعت صعود ربات بستگی دارد. سرعت چرخش و سرعت ربات در الگوریتم VFH+ محاسبه شده‌اند، بنابراین می‌توان از همان روش استفاده کرد. شتاب صعود تأثیری در این بخش از محاسبات ندارد. برای انجام محاسبات این قسمت، ابتدا نیاز است که دو نقطه مرکزی دایره چرخش، مطابق شکل ۶ و روابط ۳۲ و ۳۳ محاسبه شوند.

$$\Delta x_r = r_r \cdot \sin\theta, \quad \Delta y_r = r_r \cdot \cos\theta \quad (32)$$

$$\Delta x_l = -r_l \cdot \sin\theta, \quad \Delta y_l = -r_l \cdot \cos\theta \quad (33)$$



شکل ۶- دایره‌های چرخش در مرحله ویژگی‌های فیزیکی.

Fig. 6. Turning circles of the physical characteristics stage

پس از آن، الگوریتم نیاز دارد تا هر سلول فعال $C_{i,j,k}$ را بررسی کند تا ببیند آیا در دایره چرخش قرار دارد یا خیر. این کار با محاسبه فاصله $d_{l,r}$ بین مرکز دایره چرخش و حجم سلولی طبق روابط ۳۴ و ۳۵ انجام می‌شود:

$$d_r = \sqrt{(\Delta x_r - \Delta x(i))^2 + (\Delta y_r - \Delta y(j))^2} \quad (34)$$

$$d_l = \sqrt{(\Delta x_l - \Delta x(i))^2 + (\Delta y_l - \Delta y(j))^2} \quad (35)$$

برای شناسایی اینکه آیا حجم سلولی در دایره چرخش قرار دارد یا خیر، الگوریتم این فاصله را با محدوده ایمنی r_{r+s+v} و شعاع دایره چرخش $r_{l,r}$ با استفاده از روابط ۳۶ و ۳۷ مقایسه می‌کند:

پنجره از مرز چپ یا راست عبور کند، پنجره سلول‌های سمت دیگر هیستوگرام را بررسی می‌کند. پس از آن، سه وزن مسیر محاسبه و در یک وزن مسیر واحد ترکیب می‌شوند تا جهت کاندید تعیین شود:

$$k_i = \mu_1 \Delta(v, k_i) + \mu_2 \Delta\left(v, \frac{\theta}{\alpha}\right) + \mu_3 \Delta(v, k_{i-1}) \quad (45)$$

وزن مسیر اول بر اساس تفاوت بین زاویه هدف k_i و جهت کاندید v محاسبه می‌شود. وزن مسیر دوم تفاوت بین چرخش ربات θ و جهت کاندید v است. آخرین وزن مسیر تفاوت بین جهت انتخاب شده قبلی k_{i-1} و جهت کاندید v است. متغیر μ برای انتخاب اینکه کدام یک از سه وزن مسیر تأثیر بیشتری بر وزن نهایی مسیر دارد، استفاده می‌شود. تابع وزن مسیر می‌تواند به گونه‌ای تغییر یابد که رفتار متفاوتی را فعال کند. به عنوان مثال، تابع وزن مسیر می‌تواند به حرکات چرخشی به جای حرکات صعودی اولویت بدهد.

تابع $\Delta(v_1, v_2)$ با محاسبه تفاوت بین دو زاویه الیوشن و دو زاویه آزیموت، تابع وزن را بر اساس دو تفاوت زاویه‌ای محاسبه شده تولید می‌کند. هنگامی که وزن مسیر تمام جهت‌های کاندید محاسبه شد، الگوریتم می‌تواند جهتی را که کمترین وزن را دارد انتخاب کند. جهت انتخاب شده به یک حرکت برای ربات تبدیل می‌شود. این تبدیل می‌تواند به راحتی و با ایجاد یک درخت تصمیم که حرکت ربات را بر اساس مختصات جهت محاسبه شده تولید می‌کند، پیاده‌سازی شود.

۵ شبیه سازی

ابتدا مدل دینامیکی حرکت یک کوادروتور بر اساس معادلات ۹ تا ۱۴ سیمولینک شبیه سازی می‌شوند. مقادیر عددی پارامترهای فیزیکی مورد استفاده در شبیه‌سازی کوادروتور در جدول ۱ آورده شده است.

جدول ۱- مقادیر عددی پارامترهای فیزیکی کوادروتور

Table 1. Physical Parameters of Quadrotor.

Nu	Parameter	S	Value	Units
1	Thrust factor	b	6.317×10^{-4}	
2	Drag factor	d	1.61×10^{-4}	
3	Gravity force	g	9.81	m/s ²
4	Inertia around x-axis	Ix	1.453×10^{-2}	Kg/m ²
5	Inertia around y-axis	Iy	1.453×10^{-2}	Kg/m ²
6	Inertia around z-axis	Iz	2.884×10^{-2}	Kg/m ²
7	Motors' moment of inertia	Jr	2.82×10^{-7}	
8	Length from the rotor to CG	L	0.225	m
9	Quadrotor mass	m	1.888	kg

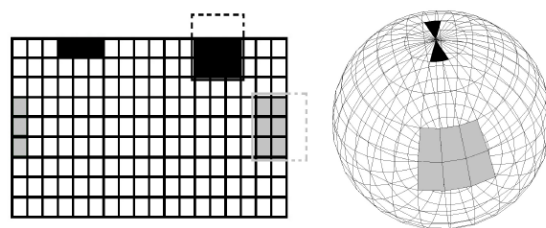
و هنگامی که مقدار یک نقطه کمتر از τ_{low} باشد، مقدار آن در هیستوگرام قطبی باینری دوبعدی برابر با ۰ در نظر گرفته می‌شود. اگر یک نقطه بین دو آستانه قرار گیرد، مقدار بعدی آن نقطه در هیستوگرام قطبی باینری دوبعدی مورد استفاده قرار می‌گیرد. این دو آستانه به الگوریتم اجازه می‌دهند که موانع واقعی و خطاهای اندازه‌گیری را تشخیص دهد.

از آنجاکه الگوریتم VFH+ سه بعدی از یک هیستوگرام قطبی دوبعدی استفاده می‌کند، آستانه‌های τ_{low} و τ_{high} باید هنگام استفاده از زاویه ارتفاع β_e متفاوت، تغییر کنند. همان‌طور که در شکل ۶ نشان داده شده است، سلول‌های هیستوگرام قطبی دوبعدی اندازه یکسانی ندارند. بنابراین، برای جبران این تفاوت‌ها، به آستانه‌های متفاوتی برای زوایای الیوشن مختلف نیاز است. اندازه این آستانه‌ها به ربات، سرعت ربات، اندازه پنجره مرحله پنج، دقت نقشه اکتالی و اندازه کره محدودکننده بستگی دارد:

$$H_{\beta_z, \beta_e}^b = \begin{cases} 1 & \text{if } H_{\beta_z, \beta_e}^p > \tau_{\beta_e} \text{ high} \\ 0 & \text{if } H_{\beta_z, \beta_e}^p < \tau_{\beta_e} \text{ low} \\ H_{\beta_{z-1}, \beta_p}^p & \text{otherwise} \end{cases} \quad (44)$$

۴-۵ مرحله پنجم: تشخیص و انتخاب مسیر

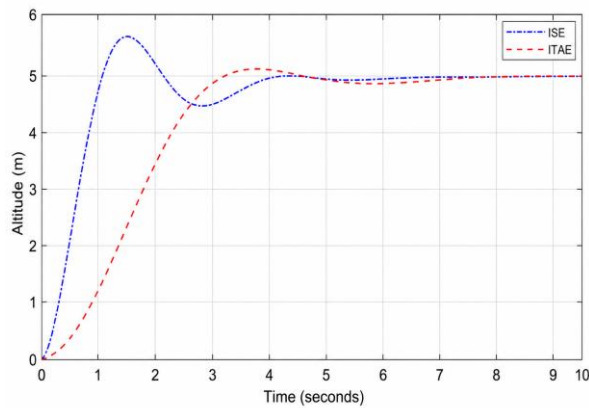
مرحله پنجم به جستجوی مسیرهای موجود در هیستوگرام قطبی باینری دوبعدی می‌پردازد و مسیری را با کمترین وزن انتخاب می‌کند. برای تعیین اینکه کدام مسیرها در دسترس هستند، الگوریتم با حرکت یک پنجره در اطراف هیستوگرام قطبی باینری دوبعدی، شکاف‌ها را شناسایی می‌کند. اگر تمام عناصر داخل پنجره برابر با ۰ باشند این پنجره مسیر را قابل عبور علامت‌گذاری می‌کند. به این ترتیب فقط شکاف‌های بزرگ به عنوان مسیرهای قابل دسترسی در نظر گرفته می‌شوند.



شکل ۷- پنجره متحرک در مرحله پنجم.

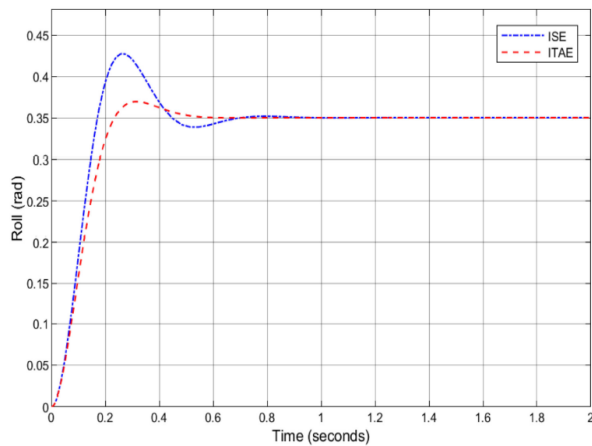
Fig. 7. Moving window of the fifth stage

هنگام پیاده‌سازی این پنجره، الگوریتم باید ویژگی‌های قطبی هیستوگرام را مد نظر قرار دهد. هنگامی که پنجره از مرزهای هیستوگرام عبور می‌کند، پنجره از عناصری استفاده می‌کند که با قواعد هیستوگرام قطبی دوبعدی به هم متصل هستند. وقتی پنجره از مرز بالایی یا پایینی هیستوگرام عبور کند (علامت‌های سیاه در شکل ۷ سلول‌هایی که در همان زاویه الیوشن قرار دارند اما ۱۸۰ درجه حول زاویه آزیموت چرخانده شده‌اند، بررسی می‌شوند. وقتی



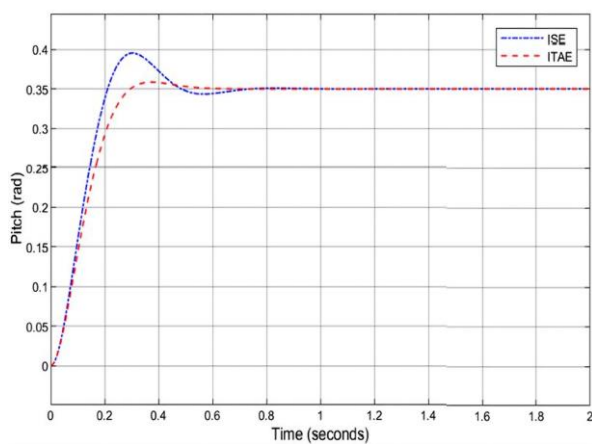
شکل ۹- پاسخ گذرای کنترل ارتفاع کوادروتور.

Fig. 9. Transient Response of the Quadrotor Control System, Altitude Control



شکل ۱۰- پاسخ گذرای کنترل رول کوادروتور.

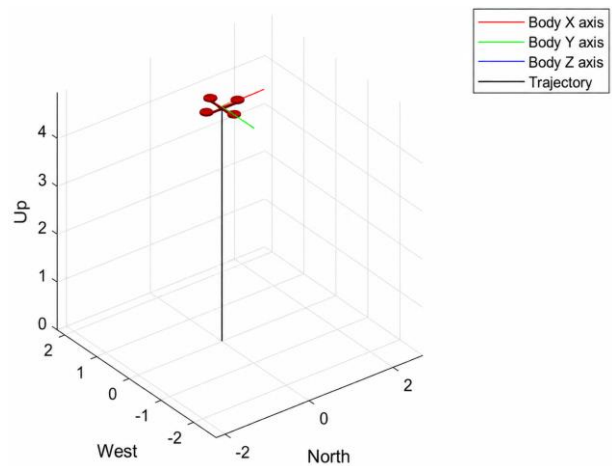
Fig. 10. Transient Response of the Quadrotor Control System, Roll Control



شکل ۱۱- پاسخ گذرای کنترل پیچ کوادروتور.

Fig. 11. Transient Response of the Quadrotor Control System, Pitch Control

مدل کوادروتور برای مدت ۱ ثانیه و با اعمال نیروی مساوی بر روی روتورهای سیستم شبیه‌سازی شده است تا درستی مدل تأیید شود. نیروی مساوی ۵ نیوتن به هر موتور اعمال می‌شود و در نتیجه، روتورها با سرعت زاویه‌ای یکسان شروع به چرخش می‌کنند. شکل ۸ پاسخ سیستم کوادروتور را به شرایط مذکور نشان می‌دهد. این وضعیت منجر به بلند شدن عمودی کوادروتور تا ارتفاع ۴/۲ متر می‌شود. با توجه به پاسخ سیستم مشخص است که هیچ حرکتی در راستای محورهای x و y انجام نمی‌شود، زیرا زوایای رول، پیچ و یاو برابر با صفر هستند. این موضوع صحت مدل را با توجه به پاسخ آن به اعمال نیروهای مساوی به تمامی موتورها تأیید می‌کند.



شکل ۸- پاسخ کوادروتور به نیروی ۵ نیوتن به هر روتور.

Fig. 8. Response of the Quadrotor when Applying 5N Per Motor.

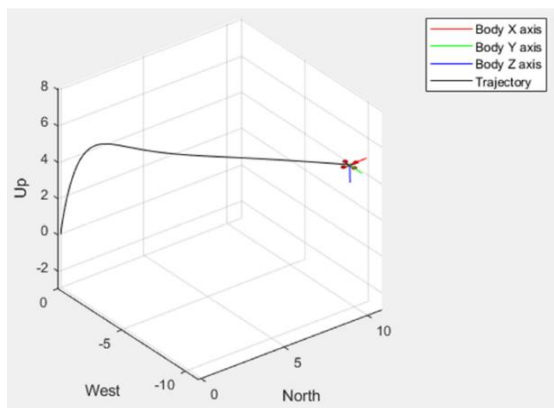
ضرایب کنترل‌کننده PID، که بر اساس دو معیار عملکرد ISE و ITAE به دست آمده‌اند، در جدول ۲ ارائه شده‌اند.

جدول ۲- ضرایب کنترل‌کننده PID

Table 2. PID Controller Parameters.

Gains	ITAE		ISE	
	Altitude Controller	Attitude Controller	Altitude Controller	Attitude Controller
Kp	5.000	9.999	12.875	9.999
Ki	2.572	6.4267E-5	6.684	0.025
Kd	3.648	1.091	4.546	0.754

هم‌چنین، پاسخ گذرای سیستم کنترل کوادروتور با اعمال کنترل‌کننده ارتفاع و وضعیت در شکل ۹ تا ۱۲ نشان داده شده است.



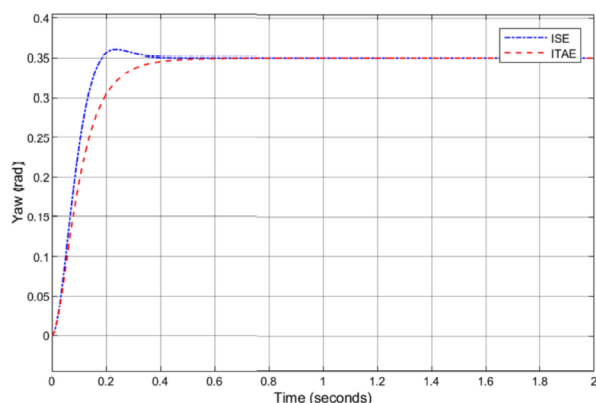
شکل ۱۳- ناوبری کوادروتور به موقعیت هدف (۱۰،۱۰).

Fig. 13. Quadrotor Navigation to a Goal Location of (10,10).

شکل ۱۳ نشان می‌دهد که کوادروتور در یک محیط بدون مانع با موقعیت مسیر خود را به سمت مقصد مورد نظر طی می‌کند. در مرحله بعدی، یک مانع به عرض ۱ متر و ارتفاع ۱۰ متر در مختصات (۵، ۵) در محیط شبیه‌سازی قرار داده می‌شود. در شبیه‌سازی‌های انجام شده در سیمولینک از یک بلوک حسگر لیدار استفاده شده است تا موانع را تشخیص دهد و آن‌ها را بر اساس مش‌های موجود در سناریو شبیه‌سازی کند و داده‌های ابر نقطه‌ای را به‌عنوان خروجی ارائه دهد. حسگر لیدار از پالس‌های لیزری بازتاب شده برای محاسبه فاصله تا موانع استفاده می‌کند. نتایج به دست آمده از اعمال الگوریتم تشخیص مانع با آستانه‌های ۰/۵ متر و ۱ متر نشان داده می‌شوند. عرض مانع برابر ۱ متر در نظر گرفته می‌شود، به این معنی که مانع در صفحه در موقعیت ± 0.5 متر در امتداد محورهای x و y قرار می‌گیرد. برای موانع استفاده شده، آستانه‌ای که فاصله‌ای ۱ متری از مانع دارد، این اطمینان را ایجاد می‌کند که کوادروتور - حتی در صورت حرکت ناگهانی مانع - به‌طور کامل از برخورد با مانع اجتناب می‌کند.

حرکات دوبعدی و سه‌بعدی کوادروتور زمانی که الگوریتم اجتناب از موانع با آستانه فاصله ۰/۵ متر پیاده‌سازی شود، در شکل‌های ۱۴ و ۱۵ نشان داده شده است.

با توجه به شکل ۱۴ مشخص است که کوادروتور حتی با پیاده‌سازی الگوریتم اجتناب از موانع، خیلی نزدیک به مانع است، زیرا رنگ‌های آبی و زرد بر روی مانع نشان دهنده ابر نقطه‌ای حسگر لیدار هستند. شکل ۱۵ نشان می‌دهد که کوادروتور قبل از اجتناب از مانع با آن برخورد می‌کند. این امر به این دلیل رخ می‌دهد که زاویه یابو محاسبه شده بر اساس مختصات تولید شده توسط الگوریتم اجتناب از موانع کوچک است. بنابراین، با توجه به شکل ۱۵ واضح است که جهت‌گیری کوادروتور در راستای محور z در اطراف مختصات (۵/۲، ۵/۲) تغییر می‌کند.



شکل ۱۲- پاسخ گذرای کنترل یابو کوادروتور.

Fig. 12. Transient Response of the Quadrotor Control System, Yaw Control

شکل‌های ۹ تا ۱۲ ارتفاع سیستم کوادروتور و وضعیت آن را شامل پاسخ رول، پیچ، و یابو با ارتفاع مطلوب ۵ متر و وضعیت مطلوب ۰/۳۵ رادیان برای هر سه متغیر رول، پیچ و یابو نشان می‌دهد. با توجه به این شکل مشخص است که کوادروتور بر اساس معیارهای عملکرد ISE و ITAE به ارتفاع و وضعیت مطلوب با ویژگی‌های گذرای مختلف دست یافته است. برای ارزیابی عملکرد کنترل‌کننده‌های طراحی شده، پارامترهای پاسخ گذرای به دست آمده برای کوادروتور بر اساس ISE و ITAE در جدول ۳ ثبت شده‌اند.

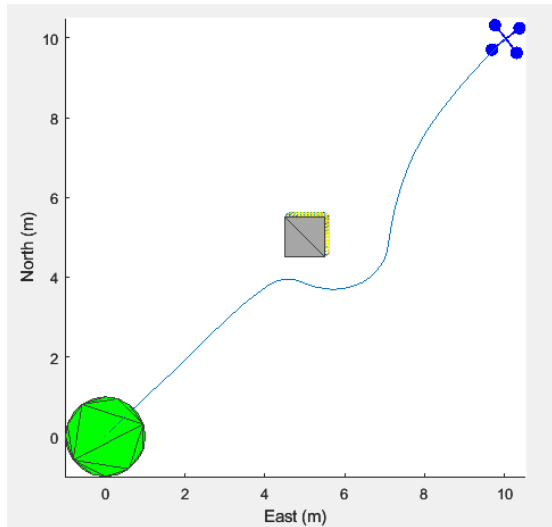
جدول ۳- پارامترهای پاسخ گذرای سیستم کنترلی کوادروتور

Table 3. Transient Parameters of the Quadrotor Control System.

Transient Parameters	ITAE		ISE	
	Altitude Controller	Attitude Controller	Altitude Controller	Attitude Controller
Overshoot(%)	2.653	2.502	14.296	12.969
Settling Time(s)	6.508	0.427	3.794	0.451
Rise Time(s)	2.011	0.185	0.734	0.144
Tracking Error	0.001	2.343E-10	0.002	5.657E-5

نتایج ارائه شده در جدول ۳ نشان می‌دهند که مقادیر پارامترهای پاسخ گذرای کنترل‌کننده مبتنی بر ITAE در کنترل ارتفاع و وضعیت کوادروتور نسبت به کنترل‌کننده مبتنی بر ISE منطقی‌تر هستند. بنابراین، کنترل‌کننده مبتنی بر ITAE برای کنترل حرکت سیستم کوادروتور انتخاب می‌شود. در شبیه‌سازی بعدی، سیستم کنترلی طراحی شده برای کوادروتور برای مسیریابی از موقعیت شروع (۰، ۰) به موقعیت هدف (۱۰، ۱۰) در ارتفاع ۵ متری آزمایش شده است. نتایج این شبیه‌سازی در شکل ۱۳ ارائه شده‌اند.

شکل ۱۷ تغییر در جهت‌گیری کوادروتور را در اطراف مختصات (۴/۸، ۵) نشان می‌دهد. با توجه به این شکل می‌توان نتیجه گرفت که آستانه فاصله ۱ متر برای اجتناب از برخورد کافی است.



شکل ۱۷ - اجتناب از مانع در فاصله ۱ متر از مانع-نمای سه بعدی.

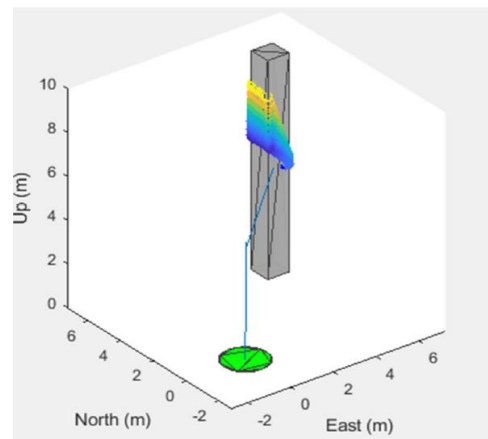
Fig. 17. Obstacle Avoidance at 1 m from the obstacle, 2D Motion

۶ نتیجه‌گیری

در این مقاله مدل‌سازی و شبیه‌سازی مسیریابی خودکار مبتنی بر اجتناب از موانع سیستم کوادروتور به کمک انجام شد. دو کنترل‌کننده PID برای کنترل ارتفاع و وضعیت کوادروتور با استفاده از دو معیار انتگرال مربع خطا (ISE) و انتگرال زمانی مطلق خطا (ITAE) توسعه داده شد. عملکرد کنترل‌کننده‌های طراحی شده با استفاده از پارامترهای پاسخ گذرا (مانند زمان نشست، زمان برخاست و درصد فراجهش) و ارزیابی شد. شبیه‌سازی‌های انجام شده نشان داد که استفاده از کنترل‌کننده PID مبتنی بر ITAE به نتایج بهتر و دقیق‌تری برای کنترل ارتفاع (با زمان نشست ۶/۵۱ ثانیه، مقدار فراجهش ۲/۶۵٪) و کنترل وضعیت (با زمان نشست ۰/۴۳ ثانیه، مقدار فراجهش ۲/۵۰٪) منجر شد. در ادامه برای جلوگیری از برخورد کوادروتور با موانع از الگوریتم هیستوگرام میدان برداری سه بعدی بهبود یافته استفاده شد. نتایج شبیه‌سازی الگوریتم اجتناب از موانع نشان داد که کوادروتور می‌تواند به‌طور مؤثری از برخورد با موانع اجتناب کند، مشروط بر اینکه فاصله بین کوادروتور و مانع کمتر یا برابر با عرض مانع باشد.

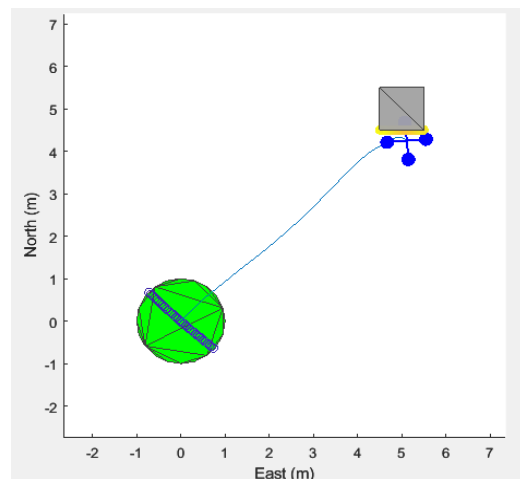
تعارض منافع

هیچ گونه تعارض منافع توسط نویسندگان بیان نشده است



شکل ۱۴ - اجتناب از مانع در فاصله ۰/۵ متر از مانع-نمای سه بعدی.

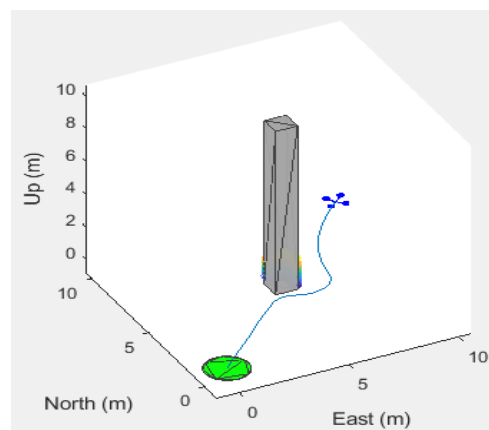
Fig. 14. Obstacle Avoidance at 0.5 m from the obstacle, 3D Motion



شکل ۱۵ - ناوبری کوادروتور به موقعیت هدف (۱۰، ۱۰).

Fig. 15. Obstacle Avoidance at 0.5 m from the obstacle; 2D Motion

حرکت سه‌بعدی کوادروتور زمانی که آستانه فاصله به ۱ متر تغییر می‌کند، در شکل ۱۶ نشان داده شده است.



شکل ۱۶ - اجتناب از مانع در فاصله ۱ متر از مانع-نمای سه بعدی.

Fig. 14. Obstacle Avoidance at 1 m from the obstacle, 3D Motion

مراجع

- [9] S. Huang, R. S. Huat Teo, and K. Kiong Tan, "Collision avoidance of multi unmanned aerial vehicles: A review." *Annual Reviews in Control*, vol. 48, 2019, pp. 147-164.
- [10] T. Shima and S. Rasmussen, *UAV Cooperative Decision and Control: Challenges and Practical Approaches*, [Bookshelf], 'Society for Industrial and Applied Mathematics,' *IEEE Control Systems Magazine*, vol. 30, Issue 2, 2010, <https://doi.org/10.1109/MCS.2010.935899>
- [11] J. Park and N. Cho, "Collision avoidance of hexacopter UAV based on LiDAR data in dynamic environment," *Remote Sens.*, vol. 12, no. 6, 2020, <https://doi.org/10.3390/rs12060975>.
- [12] A. Moffatt, E. Platt, B. Mondragon, A. Kwok, D. Uryeu, S. Bhandari, "Obstacle detection and avoidance system for small UAVs using a LiDAR," in *Proc. Int. Conf. Unmanned Aircr. . (ICUAS)*, Greece, 2020, <https://doi.org/10.1109/ICUAS48674.2020.9213897>.
- [13] A. H. Ahmed and H. Tomán. "A Hybrid Framework for UAV Obstacle Avoidance Integrating Reactive Sensing, LiDAR Planning, and Deep Reinforcement Learning in Real Time." *Results in Engineering*, 2026, <https://doi.org/10.1016/j.rineng.2026.110150>.
- [14] Q. Liang, Z. Wang, Y. Yin, W. Xiong, J. Zhang, Z. Yang, "Autonomous aerial obstacle avoidance using LIDAR sensor fusion," *PLoS ONE*, vol. 18, no. 6, 2023, <https://doi.org/10.1371/journal.pone.0287177>.
- [15] J. Borenstein and Y. Koren, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278-288, 1991, <https://doi.org/10.1109/70.88137>.
- [16] I. Ulrich and J. Borenstein, "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots," in *Proc. 1998 IEEE Int. Conf. Robotics and Automation*, Belgium, pp. 1572-1577, 1998, <https://doi.org/10.1109/ROBOT.1998.677362>
- [17] S. Vanneste, B. Bellekens, and M. Weyn, "3DVFH+: Real-Time Three-Dimensional Obstacle Avoidance Using an Octomap," in *MORSE@STAF 2014 Computer Science, Engineering*, vol. 1319, 2014.
- [1] J. Slingsby, B. E. Scott, L. Kregting, J. McIlvenny, J. Wilson, and B. J. Williamson, "A review of unmanned aerial vehicles usage as an environmental survey tool within tidal stream environments," *J. Mar. Sci. Eng.*, vol. 11, no. 12, pp. 2907-2924, 2023, <https://doi.org/10.3390/jmse11122298>.
- [2] P. Pounds, R. E. Mahony, and P. I. Corke, "Modelling and Control of a Quad-Rotor Robot," *Control Engineering Practice*, vol. 18, no. 7, 2010, <https://doi.org/10.1016/j.conengprac.2010.02.008>.
- [3] H. Fernando, ATA De Silva, M.D.C De Zoysa, K. Dilshan, S.R. Munasinghe, "Modelling, Simulation and Implementation of a Quadrotor UAV," in *IEEE 8th Int. Conf. Industrial and Information Systems*, Sri Lanka, 2013, <https://doi.org/10.1109/ICIInfS.2013.6731982>
- [4] Idrissi, Moad, Mohammad Salami, and Fawaz Annaz. "A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms." *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, Ar. no. 22, 2022, <https://doi.org/10.1007/s10846-021-01527-7>.
- [5] Sha'aban, Y.A. Automatic Tuning of MPC Using Genetic Algorithm with Historic Process Data. In *Proceedings of the 2022 IEEE*, <https://doi.org/10.1109/CSPA55076.2022.9782011>.
- [6] Ghanifar, Mana, Milad Kamzan, and Morteza Tayefi. "Intelligent tuning PID controller, simulation and comparison for a quadrotor." *Journal of Technology in Aerospace Engineering*, vol. 7. Issue 4, 2023, pp. 23-33, <https://doi.org/10.30699/jtae.2023.7.4.3>.
- [7] I. Shafieenejad, E. Dehghan Rouzi, J. Sardari, M. Siami Araghi, A. H. Esmaeili, and Sh. Zahedi "Fuzzy logic, neural-fuzzy network and honey bees algorithm to develop the swarm motion of aerial robots." *Evolving Systems*, vol. 13, no. 2, 2022, pp. 319-330, <https://doi.org/10.1007/s12530-021-09391-4>
- [8] I. Shafieenejad, "Design of swarm guidance for aerial spherical robots with social intelligence inspired by grasshopper'optimal motion and fuzzy logic." *Aerospace Knowledge and Technology Journal*, vol. 14, Issue 1, 2025, pp. 193-216, [20.1007.1.23221070.1404.14.1.11.4](https://doi.org/10.1007/s123221070.1404.14.1.11.4).