

طراحی کنترل گام به عقب انتگرالی هوشمند و طراحی مسیر بهینه برای کوادروتور با استفاده از الگوریتم ازدحام افزایشی ذرات

چکیده: در این مقاله یک کنترل کننده گام به عقب انتگرالی برای کنترل موقعیت سیستم کوادروتور ارائه شده که با استفاده از الگوریتم هوشمند بهینه سازی ازدحام افزایشی ذرات، جهت بهینه سازی پارامترهای کنترلر طراحی شده مورد بررسی قرار گرفته است که در نهایت پارامترهای طراحی به منظور حصول پاسخ مطلوب بهینه بر روی کنترل کننده منظور شده اند. در ادامه به طراحی مسیر بهینه برای پرنده کوادروتور به کمک روش الگوریتمی تکاملی ازدحام ذرات و با در نظر گرفتن موانع از پیش تعیین شده در محیط سه بعدی پرداخته شده است، که امنیت پرواز کوادروتور را در محیط های سه بعدی از پیش شناخته شده تأمین می کند، همچنین با در نظر گرفتن موانع محیطی، کوتاه ترین مسیر بین مبدأ و مقصد به عنوان مسیر مطلوب کوادروتور مشخص می شود. در این مقاله علاوه بر بهینه سازی ضرایب کنترلر گام به عقب انتگرالی در جهت کاهش تلاش کنترلی ربات، به بهینه سازی مسیر حرکت کوادروتور پرداخته شده که هدف از این بهینه سازی، کاهش مصرف انرژی در حرکت ربات از مبدأ به مقصد می باشد که در نهایت مسیر تولید شده توسط الگوریتم به عنوان مسیر مطلوب پرنده به کنترل کننده گام به عقب انتگرالی بهینه شده داده شده و نتایج در محیط متلب شبیه سازی شده است.

واژه های راهنما: کوادروتور، کنترل گام به عقب انتگرالی بهینه، تولید مسیر بهینه، الگوریتم ازدحام ذرات

وحید تیکنی
دانشجوی دکتری

حامد شهبازی*
استادیار

پیمان نوروزی
دانشجوی دکتری،
گروه مهندسی مکانیک،
دانشکده فنی و مهندسی،
دانشگاه اصفهان، اصفهان

مقاله علمی پژوهشی
دریافت: ۱۴۰۱/۰۴/۱۲
پذیرش: ۱۴۰۱/۰۶/۲۹

Vahid Tikani
Ph.D. Candidate

Hamed Shahbazi*
Assistant Professor

Peyman Norouzi
Ph.D. Candidate,
Department of
Mechanical
Engineering, Faculty of
Engineering, University
of Isfahan, Isfahan

Design intelligent integral Back-step controller and path planning for Quadrotor by Swarm optimization algorithm

Abstract: In this paper, an intelligent integral Back-step controller is used to control the position of the quadrotor. Using a new strong particle swarm optimization algorithm, the optimal system response is Achieved by changing constant design values in the controller. Then, the optimal path for the quadrotor is designed using the algorithmic method of particle swarm and considering the predetermined obstacles in the three-dimensional environment, which ensures safety flight. Also, the shortest path between the origin and destination as the optimal path is determined. Finally, the path generated by the algorithm as the optimal path of planning. Thus, in addition to optimizing the control parameters in the controller, the quadrotor path is also optimized using the algorithm to minimize energy consumption from source to destination. Finally, by simulating the system in MATLAB environment, the system response to the desired input in a limited time was plotted and investigated.

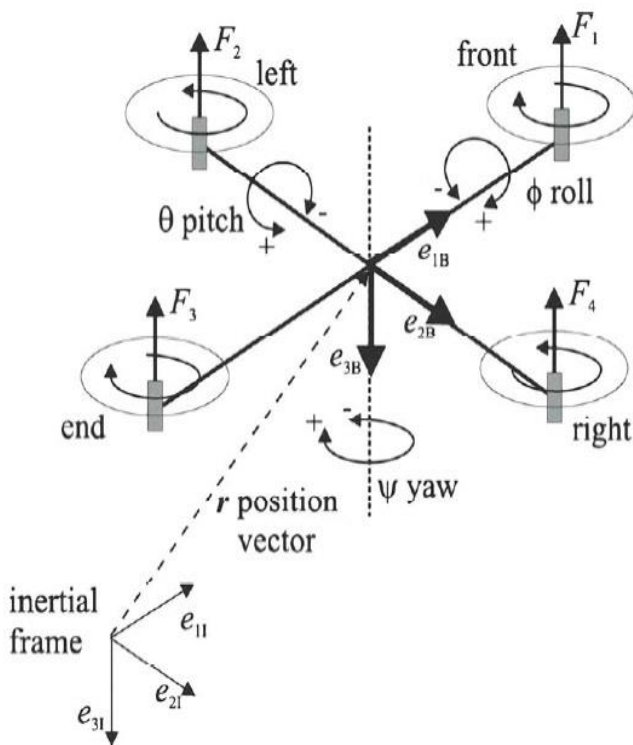
Keywords: Quadrotor, Integral back-step controller, Path-planing, Particle swarm optimization algorithm

۱- مقدمه

تعقیب مسیر در پرند کوادروتور از آن جهت مورد اهمیت است که برای تعریف عملیات برای ربات نیاز به کنترل ربات برای عبور در راستای مسیر مطلوب بدون برخورد با موانع می‌باشد. تولید مسیر به روش کمترین خطای برخورد از جمله روش پیاده شده برای تولید مسیر بهینه می‌باشد [۲۵]، همچنین تولید مسیر با استفاده از برنامه‌نویسی دودویی از روش‌های دیگر ارائه شده برای تولید مسیر برای ربات در فضای سه‌بعدی می‌باشد [۲۶].

در این مقاله شبیه‌سازی کنترل‌کننده گام به عقب انتگرالی برای کوادروتور و بهینه‌سازی پارامترهای کنترلی با استفاده از الگوریتم ازدحام افزایشی ذرات مورد توجه قرار گرفته است. در قسمت دوم این مقاله مدل ریاضی کوادروتور بررسی شده و در قسمت سوم طراحی کنترل‌کننده گام به عقب انتگرالی بیان شده است. در ادامه با توجه به توانایی الگوریتم‌های تکاملی در بهینه‌سازی سیستم‌ها، از الگوریتم ازدحام ذرات به عنوان الگوریتم تکاملی استفاده شده که به محاسبه ضرایب کنترلی با استفاده از این الگوریتم پرداخته شده است. در قسمت چهارم پیاده‌سازی و بررسی نتایج کنترل‌کننده برای عبور از مسیر مطلوب در فضای شبیه‌سازی متلب صورت گرفته است.

۲- مدل‌سازی ریاضی کوادروتور



شکل ۱ پیکربندی کوادروتور

با توجه به کاربرد فراوان پهپادها در دهه‌های اخیر برای کاربردهای نظامی و غیرنظامی از جمله: آتش‌نشانی، عملیات جستجو، نقشه‌برداری و... موجب شده این پرنده‌ها بسیار مورد توجه محققان قرار بگیرد. طراحی پرنده‌های چندملخه مانند کوادروتور از آغاز قرن بیستم مورد توجه بسیاری از محققان قرار گرفته است. اولین پرنده چهارملخه در سال ۱۹۲۱ توسط دوپوسزات^۱ ساخته شد [۱]. از آن پس پرنده‌های چهارملخه متعددی از جمله هلیکوپتر برگوت^۲ و یا هلیکوپتر امنیچ^۳ ساخته شد [۳ و ۲]. با توجه به اینکه در آن دوره ساخت و توسعه سنسورها و طراحی سیستم کنترلی چندان امکان‌پذیر نبود، طراحی پرنده‌های خودکار عملاً غیرممکن بود. پیشرفت‌های اخیر در حوزه کنترل و الکترونیک که منجر به ساخت سنسورهای با دقت بالا شده است، این امکان را برای طراحان به وجود آورده که به طراحی پهپادهای کنترل‌پذیر روی آوردند. از این رو پهپادها در اندازه‌های مختلف و با امکانات گسترده ساخت و توسعه یافته‌اند [۴-۷].

تحقیق و توسعه در مدل‌سازی و طراحی کنترل‌کننده برای پهپاد کوادروتور در دهه‌های اخیر بسیار مورد توجه قرار گرفته است. تعدادی از این پژوهش‌ها به صورت خلاصه در ادامه آمده است. التوگ و همکارانش کوادروتوری بر اساس روش اولیتر- نیوتون برای تمرکز بر پایدارسازی و کنترل سیستم با استفاده از تصاویر مدل‌سازی کرد [۸]. ساتر و همکارانش نیز به بررسی طراحی کنترل‌کننده با استفاده از سرو کنترلهای مبتنی بر پردازش تصویر پرداخته‌اند [۹]. دانفید و همکارانش یک کنترل‌کننده مبنی بر شبکه عصبی برای کوادروتور طراحی نمود [۱۰]. ارل و همکارانش برای تخمین موقعیت کوادروتور از یک فیلتر کالمن استفاده نمود [۱۱]. لی و همکارانش نیز به طراحی کنترل‌کننده مود لغزشی برای کوادروتور پرداخته‌اند [۱۲]. بورا و همکارانش یک کنترل‌کننده فازی هیبریدی برای کنترل کوادروتور ارائه کردند [۱۳]. اشتا و همکارانش در [۱۴] به مقایسه کنترل‌کننده PID و فازی پرداخته است. طراحی کنترل‌کننده فازی برای سیستم‌های دینامیکی بسیار مورد استفاده قرار گرفته که از جمله آن می‌توان به طراحی کنترل‌کننده فازی-منطقی برای کوادروتور اشاره داشت [۲۲-۱۵]. در مقالات اخیر محققان اقدام به طراحی کنترل‌کننده فازی برای کنترل ارتفاع و پایدارسازی آن در وضعیت ایستا نموده‌اند [۲۴ و ۲۳]. تولید و

³ Oemnichien¹ Debothezat² Berguet

$$X=[\phi \dot{\phi} \theta \dot{\theta} \psi \dot{\psi} z \dot{z} x \dot{x} y \dot{y}] \quad (7)$$

$$x_1=\phi \quad x_7=z$$

$$x_2=\dot{\phi} \quad x_8=\dot{x}_7=\dot{z}$$

$$x_3=\theta \quad x_9=x$$

$$x_4=\dot{\theta} \quad x_{10}=\dot{x}_9=\dot{x} \quad (8)$$

$$x_5=\psi \quad x_{11}=y$$

$$x_6=\dot{\psi} \quad x_{12}=\dot{x}_{11}=\dot{y}$$

$$U=[U_1 \ U_2 \ U_3 \ U_4]^T \quad (9)$$

که ورودی‌های کنترلی به صورت معادلات (۱۰) تا (۱۳) می‌باشند:

$$U_1 = \sum_{i=1}^4 T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (10)$$

$$U_2 = (-T_2 + T_4) = b(-\Omega_2^2 + \Omega_4^2) \quad (11)$$

$$U_3 = (T_1 + T_3) = b(\Omega_1^2 - \Omega_3^2) \quad (12)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (13)$$

در معادلات بالا b ضریب رانشی و d ضریب درگ است. تابع انتقال بین نرخ تغییرات سرعت‌های زاویه‌ای $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ و سرعت‌های خطی (p, q, r) می‌تواند به صورت یک ماتریس واحد منظور شود، بنابراین در صورتی که انحراف از حالت پرواز ایستا کم باشد می‌توان نوشت:

$$(\dot{\phi}, \dot{\theta}, \dot{\psi}) \approx (p, q, r) \quad (14)$$

که با ساده‌سازی معادلات (۱) تا (۶) داریم:

$$f(X, U) = \begin{pmatrix} \dot{\phi} \\ \dot{\theta}\psi a_1 - \dot{\theta} a_2 \Omega_r + b_1 U_2 \\ \dot{\theta} \\ \dot{\phi}\psi a_3 - \dot{\phi} a_4 \Omega_r + b_2 U_3 \\ \dot{\psi} \\ \dot{\theta}\dot{\phi} a_5 + b_3 U_4 \\ \dot{z} \\ g - (\cos\phi \cos\theta) \frac{1}{m} U_1 \\ \dot{x} \\ u_x \frac{1}{m} U_1 \\ \dot{y} \\ u_y \frac{1}{m} U_1 \end{pmatrix} \quad (15)$$

ساختار کوادروتور متشکل از دو بال عمود برهم می‌باشد و هر کدام دارای موتورهایی در انتهای خود هستند. دو موتور مقابل با شماره ۱ و ۳ در راستای ساعت گرد می‌چرخند و موتورهای مقابل با شماره ۲ و ۴ در راستای پادساعت گرد می‌چرخند. هر کدام از این موتورها یک نیروی تراست در جهت عمودی رو به بالا و یک گشتاور در جهت مثبت دوران ایجاد می‌کند. نیروی تراست تولیدشده در هر موتور برای انجام حرکات عمودی و گشتاور تولیدشده در هر روتور برای انجام حرکت چرخشی حول محور Z استفاده می‌شود. الگوریتم‌های کنترلی طراحی شده سعی دارند تا با تنظیم سرعت هر روتور بتوانند مانورهای مختلف را انجام دهند و سیستم را به سمت ورودی مطلوب هدایت کنند. پیکربندی کوادروتور در شکل ۱ نشان داده شده است.

به منظور طراحی الگوریتم‌های کنترلی نیاز به داشتن مدل دینامیکی سیستم می‌باشد که در اینجا مدل دینامیکی سیستم به صورت $\dot{X}=f(X, U)$ منظور می‌شود که U بردار ورودی کنترلی سیستم و X بردار متغیرهای حالت سیستم است و به صورت (۱) تا (۶) تعریف می‌شوند [۲۷]:

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \frac{U_1}{m} \quad (1)$$

$$\ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \frac{U_1}{m} \quad (2)$$

$$\ddot{z} = -g + (\cos(\phi)\cos(\theta)) \frac{U_1}{m} \quad (3)$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi} \left[\frac{I_{yy} - I_{zz}}{I_{xx}} \right] + \frac{J_r}{I_{xx}} \dot{\theta} \Omega_d + \frac{1}{I_{xx}} U_2 \quad (4)$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi} \left[\frac{I_{zz} - I_{xx}}{I_{yy}} \right] - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_d + \frac{1}{I_{yy}} U_3 \quad (5)$$

$$\ddot{\psi} = \dot{\theta}\dot{\phi} \left[\frac{I_{xx} - I_{yy}}{I_{zz}} \right] + \frac{1}{I_{yy}} U_4 \quad (6)$$

که در آن x و y و z موقعیت مرکز جرم در سازه و ϕ و θ و ψ زوایای اوپلر هستند. همچنین m و I_x و I_y و I_z جرم و ممان اینرسی سیستم حول محور مختصات سیستم می‌باشد. L فاصله بین روتورها و مرکز جرم سیستم است و J_r و Ω_R ممان اینرسی و سرعت زاویه‌ای تیغه‌های پرواز هستند. U_1 و U_2 و U_3 و U_4 ورودی‌های کنترلی سیستم و به ترتیب نیروی تراست مجموع و مومنتم تولیدشده چرخشی حول محور y ، مومنتم چرخشی حول محور x و مومنتم چرخشی حول محور z توسط تیغه‌های پروازی می‌باشند. مدل (۱) برای نوشتن معادلات به فرم $\dot{X}=f(X, U)$ با ورودی U به عنوان متغیر حالت و X به عنوان بردار ورودی کنترلی به صورت معادلات (۷) تا (۹) نوشته می‌شود:

در این الگوریتم بهینه‌سازی ابتدا پروسه بهینه‌سازی را برای اولین گام زمان ($k=k_0$) شرح می‌دهیم تا مقدار بهینه W^{k_0} را به دست آوریم. پروسه بهینه‌سازی ترکیبی پارامترهای سیستم کنترلی، یک پروسه تکراری است که از یک مجموعه Q تایی مقادیر اولیه تصادفی شروع به کار می‌کند و پارامترهای بهینه فضای حالت را جستجو می‌کند. هر مقدار اولیه داخل یک بردار M تایی ($Pos(j)$) قرار می‌گیرد که بردار پارامتر W را مقداردهی می‌کند. این بردار Pos در طول اجرای الگوریتم مرتباً بروز رسانی می‌شود. در حلقه اصلی الگوریتم، تکرار تا زمانی که خطای برآورد زیاد است و یا تعداد مشخصی تکرار به پایان نرسیده است ادامه می‌یابد. در هر دور هر بردار $Pos(j)$ با بردار جدید $Pos(j)+V(j)$ جایگزین می‌شود که در اینجا $V(j)$ بردار سرعت کلی $Pos(j)$ می‌باشد. بردار سرعت کلی $Pos(j)$ در دور i ام از رابطه زیر محاسبه می‌شود:

$$V(j)^{it} = \gamma \cdot V(j)^{it-1} + \alpha \cdot (BPos(it) - Pos(i)^{i-1}) + \beta \cdot (GPos(it) - Pos(i)^{i-1}) \quad (26)$$

در این روش همانگونه که در معادله بالا آورده شده است ضرایب α و β و γ ضرایب وزن دهی متغیری می‌باشند که برای بهبود یافتن جواب بهینه به این الگوریتم اضافه شده‌اند. α و β دو ضریب وزن دهی هستند که مقدار بیشینه ثابت اولیه دارند و در طول اجرا به طور تصادفی از مقدار بیشینه تولید می‌شوند. مقدار بیشینه α با افزایش سیکل‌های الگوریتم به تدریج کاهش می‌یابد تا الگوریتم بتواند به صورت بهینه همگرا گردد. $BPos(i)$ بهترین بردار در مجموعه بردارهای دور i ام است که کمترین مقدار خطا را تولید کرده و $GPos(i)$ مقدار بهترین بردار محاسبه شده تا دور $i-1$ ام می‌باشد.

الگوریتم پس از محاسبه $V(i)$ و بروز رسانی هر بردار $Pos(j)$ ، آن‌ها را ارزیابی کرده و خطای برآورد را با محاسبه تابع هزینه از معادله (۶) بدست می‌آورد. پس از آن بهترین بردار آن دور که کمترین خطا را تولید کرده است، استخراج شده و در $GPos$ قرار می‌گیرد. پس از پایان یک دور، مقدار بیشینه α بروز رسانی می‌شود و بهترین بردار پارامتر W انتخاب می‌شود. در پایان حلقه اصلی الگوریتم، بهترین بردار پارامتر به خروجی تابع ارسال می‌شود.

پس از کشف W^{k_0} روند افزایشی الگوریتم ازدحام ذرات برای محاسبه W^{k_l} $l=1, 2, 3, \dots$ شروع می‌شود. در این مرحله یک پروسه تکراری شروع شده که از یک مجموعه P تایی مقادیر اولیه تصادفی شروع به کار می‌کند و پارامترهای جدید فضای حالت را جستجو می‌کند. جهت کاهش هزینه محاسباتی سیستم و امکان

ضرایب عبارت ۱۵ به صورت معادلات (۱۶) تا (۲۵) می‌باشند:

$$a_1 = \frac{I_y - I_z}{I_x} \quad (16)$$

$$a_2 = -\frac{I_R}{I_x} \quad (17)$$

$$a_3 = \frac{I_z - I_x}{I_y} \quad (18)$$

$$a_4 = \frac{I_R}{I_y} \quad (19)$$

$$a_5 = \frac{I_x - I_y}{I_z} \quad (20)$$

$$b_1 = \frac{l}{I_x} \quad (21)$$

$$b_2 = \frac{l}{I_y} \quad (22)$$

$$b_3 = \frac{l}{I_z} \quad (23)$$

$$U_x = (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \quad (24)$$

$$U_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \quad (25)$$

۳- الگوریتم بهینه‌سازی افزایشی ازدحام ذرات

روش بهینه‌سازی ازدحام ذرات، یک روش سراسری کمینه‌سازی است که با استفاده از آن می‌توان به حل مسائلی که جواب آن‌ها یک نقطه یا سطح در فضای چندبعدی می‌باشد، پرداخت. در این الگوریتم مجموعه ذراتی در فضای پاسخ فرض می‌شوند و یک سرعت ابتدایی به آن‌ها اختصاص داده می‌شود. همچنین کانال‌های ارتباطی بین ذرات در نظر گرفته می‌شود. سپس این ذرات در فضای پاسخ حرکت می‌کنند، و نتایج حاصله بر مبنای یک ملاک شایستگی پس از هر بازه زمانی محاسبه می‌شود. با گذشت زمان، ذرات به سمت ذراتی که دارای ملاک شایستگی بالاتری هستند و در گروه ارتباطی یکسانی قرار دارند، شتاب می‌گیرند. علی‌رغم اینکه هر روش در محدوده‌ای از مسائل به‌خوبی کار می‌کند، این روش در حل مسائل بهینه‌سازی بلادرنگ در کنترل‌کننده کارکرد خوبی از خود نشان نمی‌دهد.

در اینجا یک روش جدید برای بهینه‌سازی کنترلی که ایده‌های تکاملی ازدحام ذرات و سیستم‌های بلادرنگ را با هم ترکیب کرده است تا راهکار کامل‌تری از تکامل سیستم ارائه نماید. این روش بسیاری از مشکلات محاسباتی موجود در طراحی کنترل بهینه را حل نموده است و مزایای روش‌های گذشته را نیز داراست.

کوادرورتور، حالات سرعت زاویه‌ای به‌عنوان ورودی مجازی کنترلی در کنترل‌کننده گام به عقب انتگرالی منظور می‌گردد. مرحله نخست در طراحی کنترل‌کننده طراحی‌شده‌ی گام به عقب انتگرالی در کوادرورتور، محاسبه خطای تعقیب می‌باشد که معادلات آن به‌صورت (۲۷) و (۲۸) است [۲۷]:

$$e_1 = \phi_d - \phi \quad (27)$$

$$\frac{de_1}{dt} = \dot{\phi}_d - \omega_x \quad (28)$$

e_1 خطای مقدار زاویه رول است و معادله ۲۸ مشتق زمانی خطا را تعیین می‌کند. بنابراین برای تولید رفتار مطلوب برای سیستم کوادرورتور، حالات سرعت زاویه‌ای به‌عنوان ورودی مجازی کنترلی در کنترل‌کننده گام به عقب انتگرالی به‌صورت (۲۹) منظور شده است.

$$\omega_{xd} = c_1 e_1 + \dot{\phi}_d + \lambda_1 \chi_1$$

$$x_1 = \int_0^t e_1 dt \quad (29)$$

که λ_1 و c_1 ضرایب مثبت کنترلی هستند و x_1 انتگرال خطای تعقیب زاویه رول است. با توجه به اینکه ω_x خطای منحصر به فرد خود (e_2) را دارد، بنابراین آن را به‌صورت (۳۰) تعریف می‌کنیم:

$$\frac{de_2}{dt} = c_1 (\dot{\phi}_d - \omega_x) + \ddot{\phi}_d + \lambda_1 e_1 - \ddot{\phi} \quad (30)$$

که در آن خطای تعقیب سرعت زاویه‌ای است و به‌صورت (۳۱) تعریف می‌شود:

$$e_2 = \omega_{xd} - \omega_x \quad (31)$$

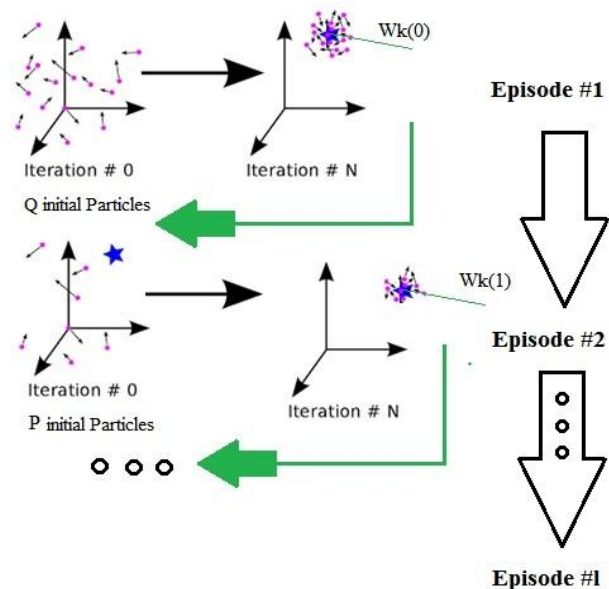
با استفاده از معادلات ۱۸ و ۱۹ خطای تعقیب رول را به‌صورت (۳۲) بازنویسی می‌کنیم:

$$\frac{de_1}{dt} = -c_1 e_1 - \lambda_1 \chi_1 + e_2 \quad (32)$$

با جابجا کردن $\ddot{\phi}$ در معادله ۱۸ با معادله متناظر آن در رابطه (۶) ورودی کنترلی U_2 در معادله پدیدار می‌شود:

$$\frac{de_2}{dt} = c_1 (\dot{\phi}_d - \omega_x) + \ddot{\phi}_d + \lambda_1 e_1 - \dot{\theta} \psi a_1 - \dot{\theta} a_2 \Omega_T - b_1 U_2 \quad (33)$$

اجرای بلادرنگ کنترل‌کننده فرض می‌کنیم رابطه $P \ll Q$ برقرار باشد. به جمعیت اولیه تصادفی الگوریتم ذره جدید W^{k0} را اضافه می‌کنیم و الگوریتم را مشابه قبل اجرا می‌کنیم تا W^{k1} بدست آید. این روند در هر لحظه زمانی تکرار می‌شود تا مقدار W^{k1} با داشتن W^{k-1} محاسبه گردد. این روند در شکل ۲ به تصویر درآمده است.



شکل ۲ الگوریتم ازدحام افزایشی ذرات

۴- پیاده‌سازی کنترل‌کننده گام به عقب انتگرالی موقعیت با استفاده از الگوریتم بهینه سازی ازدحام ذرات

در ادامه روند پیاده‌سازی کنترل‌کننده گام به عقب انتگرالی را برای کنترل موقعیت کوادرورتور بررسی می‌شود. با توجه به اینکه کنترل موقعیت کوادرورتور با کنترل زاویه‌ای آن همراه است، از این رو ابتدا روند پیاده‌سازی کنترل وضعیت زاویه‌ای سیستم بررسی می‌شود.

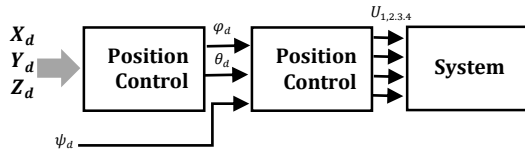
۴-۱- کنترل وضعیت کوادرورتور

کنترل وضعیت یکی از مهم‌ترین قسمت‌های طراحی کنترل‌کننده برای کوادرورتور است. این کنترل‌کننده، وضعیت سه‌بعدی سیستم را در موقعیت‌های مطلوب حفظ می‌کند. در کنترل‌کننده گام به عقب انتگرالی با در نظر گرفتن برخی از حالت‌های سیستم به‌عنوان ورودی کنترلی مجازی و با بررسی پایداری لیاپانوف سیستم، ورودی‌های کنترلی برای سیستم تعیین می‌شوند. بنابراین برای تولید رفتار مطلوب برای سیستم

۴-۳- کنترل موقعیت کوادروتور

کنترل کننده موقعیت، کوادروتور را به سمت موقعیت افقی مطلوب (x, y) هدایت می کند. حرکت افقی با جهت گیری بردار وضعیت (ϕ, θ) به سمت مسیر حرکت مطلوب صورت می گیرد. همان گونه که در شکل شماره ۳ نشان داده شده است، در عمل، کنترل کننده موقعیت با تغییر زوایای رول و پیچ مطلوب و بامنظور کردن موقعیت مطلوب x_d و y_d از موقعیت (x, y) انجام می شود.

همانگونه که در شکل ۳ نشان داده شده است کنترل کننده موقعیت با توجه به مقادیر مطلوب ورودی زوایای رول و پیچ مطلوب برای رسیدن به موقعیت مطلوب را تعیین می کند. کنترل کننده وضعیت در هر لحظه با توجه به مقادیر مطلوب زوایای رول، پیچ و یاو مقادیر ورودی های کنترلی را مشخص می کند.



شکل ۳ کنترل موقعیت کوادروتور

بنابراین کنترل کننده موقعیت، زوایای مطلوب را به عنوان خروجی تولید می کند که توسط کنترل کننده زوایا همان گونه که در شکل ۳ نشان داده شده در هر لحظه دنبال می شود. انتقال بردار پیشران به محورهای مختصات زمینی با ماتریس R انجام می شود. با منظور کردن تغییر زاویه کم در ماتریس (۴۱) داریم:

$$R = \begin{bmatrix} I & \psi & \theta \\ \psi & I & -\phi \\ -\theta & \phi & I \end{bmatrix} \quad (41)$$

از معادله ۱۵ و ۴۱، دینامیک حرکت افقی پس از ساده سازی به صورت زیر می باشد:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} -\theta U_1 \\ \phi U_2 \end{bmatrix} \quad (42)$$

خطای تعقیب موقعیت های x و y به روش گام به عقب انتگرالی به صورت (۴۳) می باشند:

$$\begin{aligned} e_9 &= x_d - x \\ e_{11} &= y_d - y \end{aligned} \quad (43)$$

بنابراین خطای تعقیب سرعت نیز این چنین تعریف می شود:

بنابراین معادله مطلوب برای خطای تعقیب سرعت زاویه ای به صورت (۳۴) است:

$$\frac{de_2}{dt} = -c_2 e_2 - e_1 \quad (34)$$

بنابراین با ترکیب معادلات بالا ورودی کنترلی U_2 به صورت (۳۵) تعریف می شود:

$$U_2 = \frac{I}{b_1} \left[\begin{aligned} &(1-c_1^2+\lambda_1)e_1 + (c_1+c_2)e_2 - c_1\lambda_1\chi_1 \\ &+ \ddot{\phi}_d - \dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r \end{aligned} \right] \quad (35)$$

به همین ترتیب ورودی کنترلی برای پیچ و یاو به صورت (۳۶) و (۳۷) محاسبه می شوند:

$$U_3 = \frac{I}{b_2} \left[\begin{aligned} &(1-c_3^2+\lambda_2)e_3 + (c_3+c_4)e_4 - c_3\lambda_2\chi_2 \\ &+ \ddot{\theta}_d - \dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega_r \end{aligned} \right] \quad (36)$$

$$U_4 = \frac{I}{b_3} \left[\begin{aligned} &(1-c_5^2+\lambda_3)e_5 \\ &+ (c_5+c_6)e_6 - c_5\lambda_3\chi_3 \end{aligned} \right] \quad (37)$$

که $(c_3, c_4, c_5, c_6, \lambda_2, \lambda_3) > 0$ ضرایب مثبت کنترلی و χ_1, χ_2 و χ_3 به ترتیب انتگرال خطای تعقیب رول و پیچ و یاو هستند.

۴-۲- کنترل ارتفاع کوادروتور

کنترل کننده ارتفاع، فاصله پرنده از زمین را به اندازه مطلوب تنظیم و کنترل می کند. خطای تعقیب ارتفاع به صورت (۳۸) تعریف می شود:

$$e_7 = z_d - z \quad (38)$$

خطای سرعت تعقیب نیز به صورت (۳۹) تعریف می شود:

$$e_8 = c_7 e_7 + \dot{z}_d + \lambda_4 \chi_4 - \dot{z} \quad (39)$$

پس ورودی کنترلی برای کنترل ارتفاع به صورت (۴۰) خواهد بود:

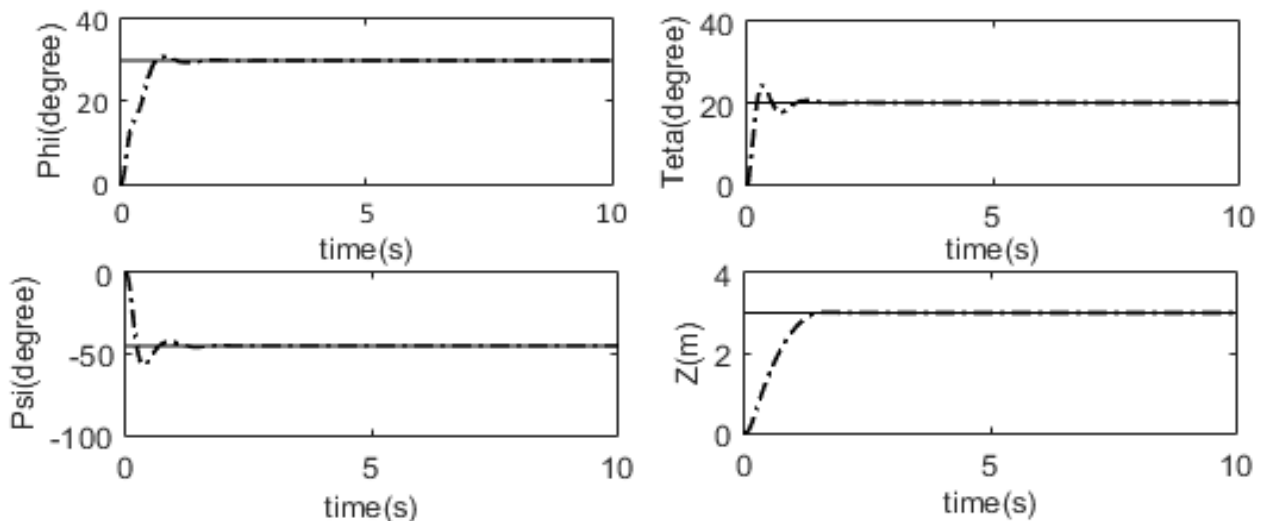
$$U_1 = \frac{m}{\cos \phi \cos \theta} \left[\begin{aligned} &g + (1-c_7^2+\lambda_4)e_7 + (c_7+c_8)e_8 \\ &-c_7\lambda_4\chi_4 \end{aligned} \right] \quad (40)$$

که ضرایب c_7, c_8, λ_4 ضرایب مثبت کنترلی هستند.

در تابع هزینه t_s بیانگر زمان نشست سیستم، t_r زمان فراز، e_{ss} خطای ماندگار و M_p بیشینه فرا جهش پاسخ می‌باشد. مقدار β برابر ۱/۴ در نظر گرفته شده که در نتیجه به هر عبارت، میزان اهمیت یکسانی در تابع هزینه اختصاص داده شده است. الگوریتم‌های بهینه‌ساز با تنظیم ضرایب مثبت کنترل کننده و کاهش تابع هزینه معرفی شده، به بهینه شدن رفتار کنترلی سیستم کمک می‌کنند. مقادیر اولیه ضرایب با استفاده از الگوریتم به صورت تصادفی در بازه‌ی مشخص شده برای الگوریتم تعیین می‌شود و در ادامه این ضرایب با هدف کاهش تابع هزینه تنظیم می‌شوند.

۵- شبیه‌سازی کنترلر انتگرالی بهینه

در این بخش در ابتدا الگوریتم ازدحام افزایشی ذرات برای تنظیم ضرایب کنترلی کنترل کننده گام به عقب انتگرالی پیاده‌سازی شده تا به بررسی و بهبود عملکرد کنترل کننده پس از تنظیم بهره‌های کنترلی توسط این روش فرا ابتکاری پرداخته شود. در شکل شماره ۵ روند کاهش تابع هزینه برای یافتن مقادیر مثبت کنترل کننده در طی ۵۰ سیکل نشان داده شده است. پس از ۵۰ سیکل انجام روند الگوریتم، تغییرات ضرایب کنترلی در شکل‌های ۶ و ۷ نشان داده شده‌اند. سپس به شبیه‌سازی کنترل کننده طراحی شده در محیط متلب برای تعیین کارایی آن، پرداخته می‌شود. پاسخ کنترل کننده گام به عقب انتگرالی وضعیت نسبت به ورودی پله بررسی شده است. پاسخ این شبیه‌سازی در شکل ۴ آورده شده است همانگونه که در شکل ۴ نشان داده شده است در روند کنترل وضعیت کوادروتور زوایای رول، پیچ و یاو از موقعیت زاویه‌ای صفر به ترتیب به موقعیت‌های ۳۰، ۲۰ و ۴۵- درجه تغییر یافته‌اند و موقعیت عمودی کوادروتور از ۰ به ۳ متر تغییر ارتفاع داده است.



شکل ۴ پاسخ کنترل کننده وضعیت و ارتفاع سیستم به ورودی پله

$$e_{10} = c_9 e_9 - \dot{x}_d - \lambda_5 \chi_5 - \dot{x} \quad (44)$$

$$e_{12} = c_{11} e_{11} - \dot{y}_d - \lambda_6 \chi_6 - \dot{y}$$

به این ترتیب ورودی‌های کنترلی به صورت (۴۵) و (۴۶) خواهند بود:

$$U_x = \frac{m}{U_1} [(1 - c_9^2 + \lambda_5) e_9 + (c_9 + c_{10}) e_{10} - c_9 \lambda_5 \chi_5] \quad (45)$$

$$U_y = -\frac{m}{U_1} \begin{bmatrix} (1 - c_{11}^2 + \lambda_6) e_{11} + (c_{11} + c_{12}) e_{12} \\ -c_{11} \lambda_6 \chi_6 \end{bmatrix} \quad (46)$$

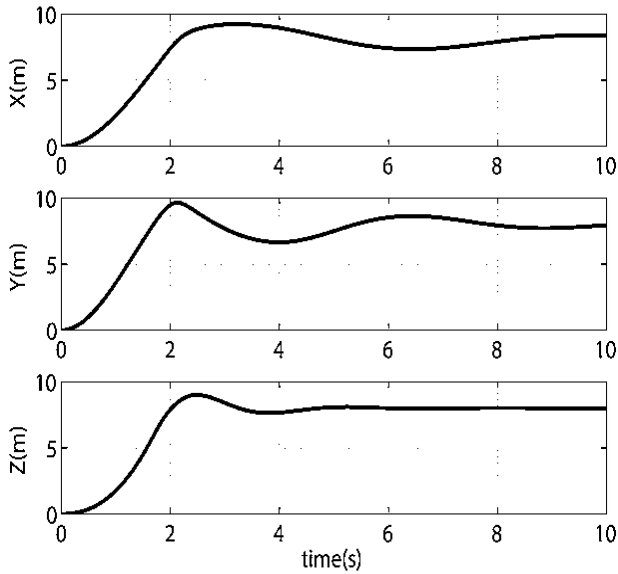
که در آن $c_9, c_{10}, c_{11}, c_{12}, \lambda_5, \lambda_6$ مقادیر مثبت طراحی هستند

۴-۴ تنظیم ضرایب کنترلی با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات

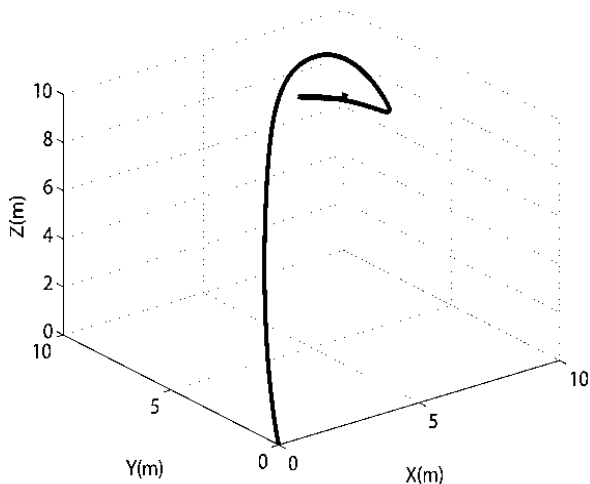
در پیاده‌سازی کنترل کننده گام به عقب ضرایب مثبت در معادلات با استفاده از الگوریتم بهینه‌ساز محاسبه و مقادیری می‌شوند. بدین منظور مشخصه‌های سیستم از جمله: زمان پاسخ‌دهی سیستم، خطای ماندگار سیستم، فرا جهش و زمان نشست سیستم در تابع هزینه برای بهینه‌سازی منظور شده‌اند لازم به ذکر است که ورودی‌های کنترلی در طراحی محدود شده‌اند. به این ترتیب تابع هزینه معرفی شده برای تنظیم ضرایب کنترلی توسط الگوریتم ذرات طبق زیر آورده شده است.

$$Cost = e^{-\beta} (M_p + e_{ss} + t_r) + (1 - e^{-\beta}) t_s \quad (47)$$

و مقصد از این جهت مورد توجه است که ربات کمترین انرژی را در طول مسیر مصرف کند. در این مقاله تولید مسیر بهینه با استفاده از الگوریتم بهینه‌سازی ازدحام افزایشی ذرات صورت گرفته که در آن وجود مانع‌های از پیش تعیین شده مسئله را چالش برانگیز می‌کند. تابع هزینه معرفی شده در این بخش طول مسیر تولید شده از مبدأ تا مقصد است که هدف یافتن کوتاه‌ترین مسیر، با وجود عدم برخورد با موانع محیط می‌باشد.



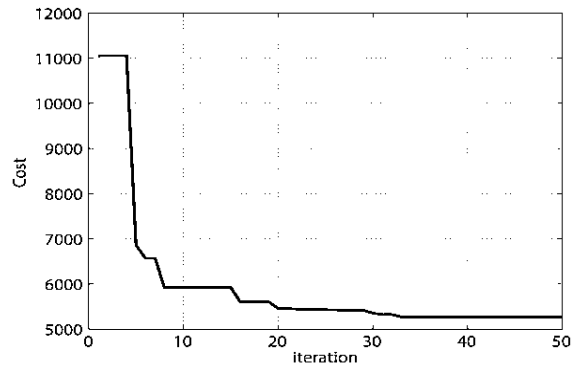
شکل ۸ پاسخ سیستم به ورودی پله



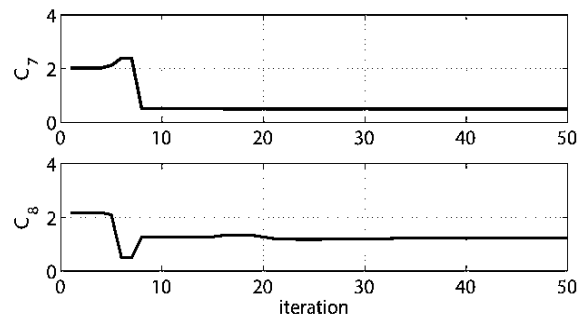
شکل ۹ حرکت در فضای سه‌بعدی در پاسخ به ورودی پله

شکل شماره ۱۰ نمای فرضی از محیط سه‌بعدی مورد نظر را با وجود موانع کروی نشان می‌دهد.

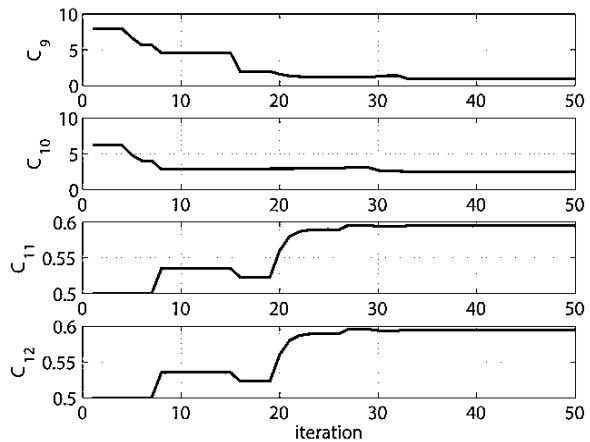
مسیر مطلوب حرکت کوادروتور برای شبیه‌سازی در این شبیه‌سازی با مبدأ حرکت به مختصات $[-5 -5 -5]$ و نقطه‌ای به مختصات $[10 10 10]$ به‌عنوان مقصد انتخاب شده است.



شکل ۵ تغییرات تابع هزینه



شکل ۶ تغییرات ضرایب کنترلی ارتفاع

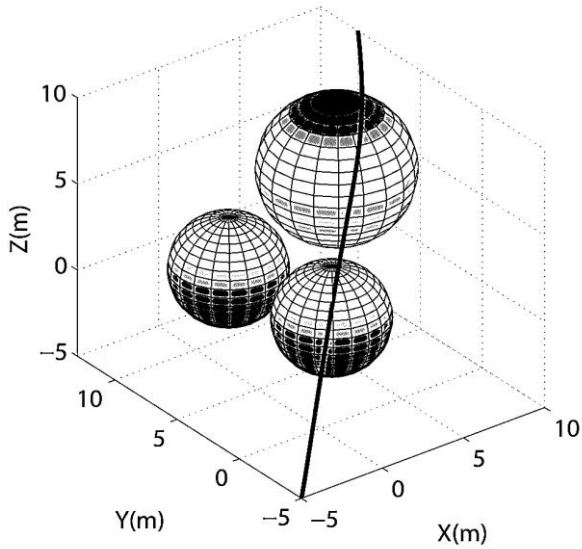


شکل ۷ تغییرات ضرایب کنترلی X و Y

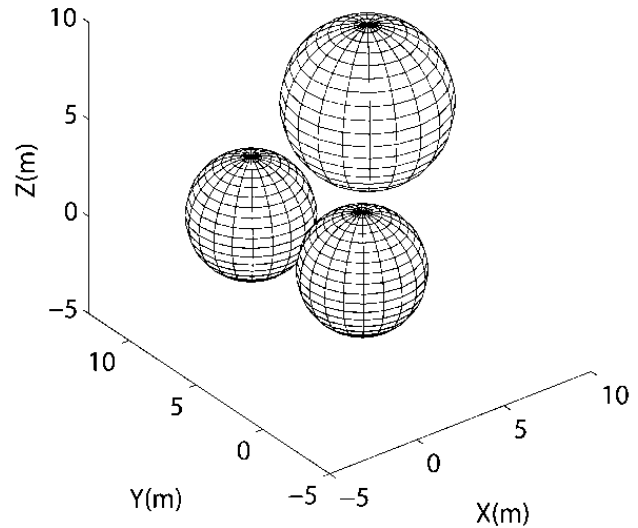
پاسخ سیستم برای ورودی به‌صورت پله و مقادیر مطلوب $x=10$ و $y=10$ و $z=10$ برای هر سه حالت موقعیت x ، y و z در شکل شماره ۸ نشان داده شده که در مجموع حرکت در سه بعد مانند شکل شماره ۹ برای سیستم کوادروتور حاصل شده است.

۶- تولید مسیر با استفاده از الگوریتم ازدحام افزایشی ذرات

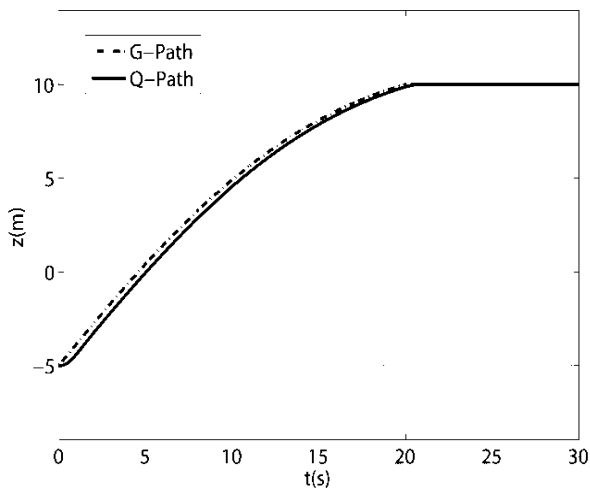
تاکنون روش‌های متعددی برای تولید مسیر در فضای سه‌بعدی برای ربات‌ها ارائه شده است. یافتن مسیر بهینه بین نقطه شروع



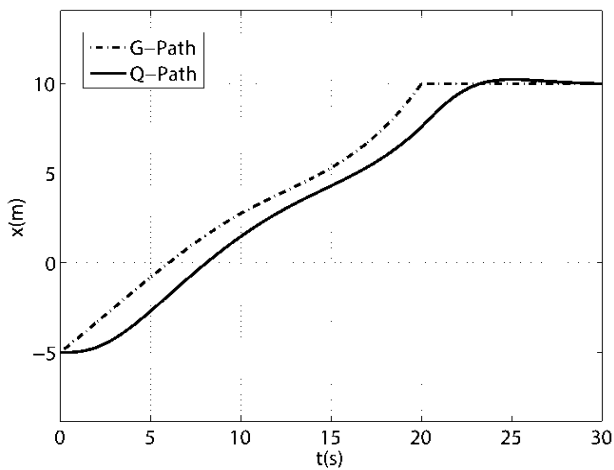
شکل ۱۲ مسیر بهینه بین مبدأ و مقصد



شکل ۱۰ نمای سه بعدی موانع کروی

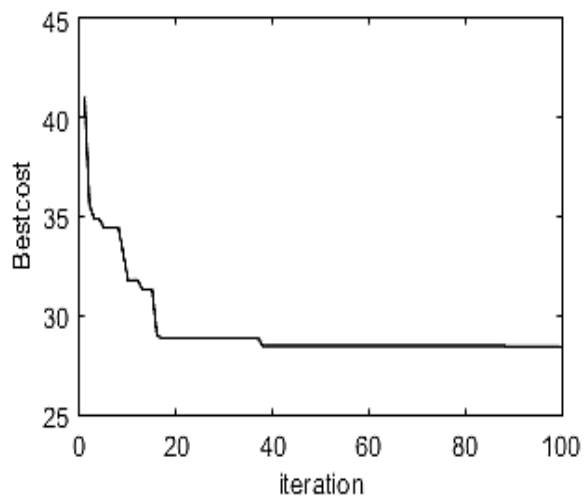


شکل ۱۳ حرکت کوادروتور در راستای z



شکل ۱۴ حرکت کوادروتور در راستای X

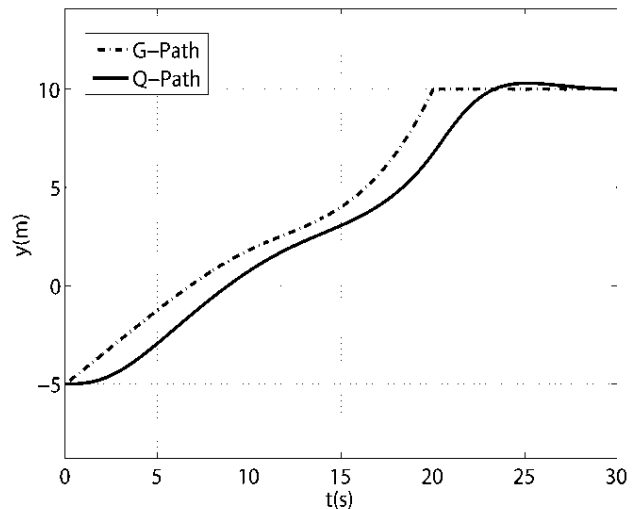
به این ترتیب پس از انجام ۱۰۰ سیکل در الگوریتم تکاملی ازدحام ذرات، روند کاهش تابع هزینه برای پیدا کردن مسیر بهینه در شکل ۱۱ نشان داده شده است. مسیر بهینه طراحی شده توسط الگوریتم تکاملی ازدحام ذرات برای عبور از این موانع و رسیدن به نقطه‌ی هدف نیز در شکل شماره ۱۰ نشان داده شده است. در ادامه این مسیر به عنوان ورودی کنترلی کنترل کننده گام به عقب انتگرالی به کوادروتور داده شده است. مسیر بهینه بین مبدأ و مقصد که توسط الگوریتم ازدحام افزایشی ذرات شناسایی می شود به عنوان ورودی مطلوب به کنترل کننده انتگرال گام به عقب انتگرالی موقعیت داده می شود. به این ترتیب پاسخ سیستم به ورودی مورد نظر در مدت زمان ۳۰ ثانیه در شکل های ۱۳ الی ۱۵ نشان داده شده است.



شکل ۱۱ تابع هزینه مسیریابی

controller and a hybrid PD controller for quadrotor, *Modares Mechanical Engineering*, Vol. 17(10), pp. 194-204, (2018).

- [2] Pulok, M., Uttam, K., A Study of the aerodynamics of a helicopter rotor blade, In *ASME International Mechanical Engineering Congress and Exposition*, Vol. 59360, pp. V001T03A025, *American Society of Mechanical Engineers*, (2019).
- [3] Uluocak, S., Mustafa, P., Oğuz, U., Experimental investigation of tip anhedral effects on the aerodynamics of a model helicopter rotor in hover, *Aerospace Science and Technology*, (2021).
- [4] Padfield, D., Helicopter flight dynamics: including a treatment of tiltrotor aircraft, *John Wiley & Sons*, (2018).
- [5] Offermann, A., Pedro, C., Jérôme M., Nonlinear model and control validation of a tilting quadcopter, In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pp. 50-55. IEEE, (2020).
- [6] Chao, H., YongCan, C., YangQuan, C., Autopilots for small unmanned aerial vehicles: a survey, *International Journal of Control, Automation and Systems*, Vol. 8.1, PP. 36-44, (2010).
- [7] Lee, D., Autonomous feature following for visual surveillance using a small unmanned aerial vehicle with gimbaled camera system, *International Journal of Control, Automation and Systems*, Vol. 8.5, PP. 957-966, (2010).
- [8] Han, D., Development of unmanned aerial vehicle (UAV) system with waypoint tracking and vision-based reconnaissance, *International Journal of Control, Automation and Systems* Vol. 8.5, PP. 1091-1099, (2010).
- [9] Zhang, X., Yuansen, D., Fang, C., Linlin, Q, Indoor position control of a quadrotor uav with monocular vision feedback, In *2018 37th Chinese Control Conference (CCC)*, pp. 9760-9765, IEEE, (2018).
- [10] Tognon, M., Antonio, F., Theory and applications for control of aerial robots in physical interaction through tethers, Vol. 140. Springer Nature, (2020).
- [11] Razmi, H., Afshinfar, S., Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerospace Science and Technology*, Vol. 91, PP. 12-27, (2019).
- [12] Noordin, A., Mohd, A., Zaharuddin, M., Sensor fusion for attitude estimation and PID control of quadrotor UAV, *International Journal of Electrical*



شکل ۱۵ حرکت کوادروتور در راستای y

شکل ۱۲ یک نمایش عمومی از مسیر بهینه محاسبه شده توسط الگوریتم در سه بعد است. روند طی کردن این مسیر توسط کنترل کننده در راستاهای Z و X و y به ترتیب در تصاویر ۱۳ الی ۱۵ نمایش داده شده است. برای هر سه راستا موقعیت اولیه کوادروتور در مختصات -5 و مختصات مطلوب 10 است. از ثانیه 20 الی 30 ورودی مطلوب روی نقطه پایانی مسیر ثابت می ماند تا به کوادروتور زمان لازم برای رسیدن به مکان نقطه هدف داده شود. همان گونه که از پاسخ سیستم مشخص می باشد در کنترل ارتفاع، خروجی کوادروتور با خطای کمتری بر ورودی مطلوب منطبق است اما کنترل در جهت x و y زمان بیشتری برای رسیدن به مسیر مطلوب نیاز دارد.

۷- نتیجه گیری

در این مقاله کنترل کننده گام به عقب انتگرالی طراحی شده توسط الگوریتم بهینه سازی ازدحام ذرات در جهت کاهش خطای سیستم بهینه سازی شد. در راستای بهینه سازی مسیر حرکت کوادروتور از الگوریتم ازدحام ذرات جهت کاهش مصرف انرژی ربات در طی مسیر بهره گیری شد. با استفاده از محیط متلب به شبیه سازی حرکت کوادروتور پرداخته شد که نتایج و نمودارها عملکرد خوب ربات در مسیر طراحی شده و عبور از موانع از پیش تعیین شده را نشان میدهد.

۸- مراجع

- [1] Norouzi, R., Shahbazi, H., Jamshidi, K., Design and implementation of a neural intelligent

- Automation and Systems*, Vol. 9.1, PP. 69-79, (2011).
- [21] Ferdaus, M., Sreenatha, G., Mahardhika, P., Matthew, A., Towards the use of fuzzy logic systems in rotary wing unmanned aerial vehicle: a review, *Artificial Intelligence Review*, Vol. 1, PP. 257-290, (2020).
- [22] Ma, H., Mou, C., Gang, F., Qingxian, W., Disturbance-observer-based adaptive fuzzy tracking control for unmanned autonomous helicopter with flight boundary constraints, *IEEE Transactions on Fuzzy Systems*, (2022).
- [23] Hu, Y., Yanping, Y., Shu, L., Yaoming, Z., Fuzzy controller design of micro-unmanned helicopter relying on improved genetic optimization algorithm, *Aerospace Science and Technology*, Vol. 98, (2020).
- [24] Vitzilaios, N., Nikos, C., An experimental test bed for small unmanned helicopters, *Journal of Intelligent and Robotic Systems*, Vol. 54.5, PP. 769-794, (2009).
- [25] Khalesi, M. H., Salarieh, H., Saadat Foumani, M., System identification and robust attitude control of an unmanned helicopter using novel low-cost flight control system, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 5, PP. 634-645, (2020).
- [26] Wu, Y., Development and Implementation of a control system for a quadrotor UAV, *Hochschule Ravensburg-Weingarten, Μάρτιος*, (2009).
- [27] Nguyen, K. D., Cheolkeun, H., Development of hardware-in-the-loop simulation based on gazebo and pixhawk for unmanned aerial vehicles, *International Journal of Aeronautical and Space Sciences*, Vol. 1, PP. 238-249, (2018).
- [28] Tomashevich, S., Rostislavich Andrievsky, B., Full control of a quadrotor with simple adaptive control, *Differential Equations & Control Processes*, (2020).
- and *Electronic Engineering and Telecommunications*, Vol. 4, PP. 183-189, (2018).
- [13] Lee, D., Jin Kim, H., Shankar, S., Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter, *International Journal of Control, Automation and Systems*, Vol. 3, PP. 419-428, (2009).
- [14] Erginer, B., Erdinç, A., Design and implementation of a hybrid fuzzy logic controller for a quadrotor VTOL vehicle, *International Journal of Control, Automation and Systems*, Vol. 10.1, PP. 61-70, (2012).
- [15] Sharma, A., Amol, B., Controlling of quad-rotor uav using pid controller and fuzzy logic controller, *International Journal of Electrical, Electronics and Computer Engineering*, Vol. 1.2, PP. 38-41, (2012).
- [16] Glida, E., Latifa, A., Abdelghani, C., Chouki, S., Gabriele, P., Optimal model-free fuzzy logic control for autonomous unmanned aerial vehicle, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 5, PP. 952-967, (2022).
- [17] Sarabakha, A., Erdal, K., Online deep fuzzy learning for control of nonlinear systems using expert knowledge, *IEEE Transactions on Fuzzy Systems*, Vol. 7, PP. 1492-1503, (2019).
- [18] Rusli, A., Mohd, A., Aripin, M., Azmi, M., Kasno, A., Khamis, A., Rizman, Z. I., Modelling of unmanned aerial vehicle (UAV) for altitude control using system identification technique, *J Fund Appl Scie*, Vol. 10, PP. 936-950, (2018).
- [19] Fung, E., Hoi, K., Smart hanger dynamic modeling and fuzzy controller design, *International Journal of Control, Automation and Systems*, Vol. 9.4, PP. 691-700, (2011).
- [20] Hafaiifa, A., Ferhat, L., Kouider, L., A Numerical structural approach to surge detection and isolation in compression systems using fuzzy logic controller, *International Journal of Control*,