

طراحی سامانه یادگیری لایه ای و شبیه سازی لایه به لایه کنترلگر بر روی ربات پرنده چهارروتور بر اساس الگوریتم بهینه سازی ازدحام افزایشی ذرات

دانشجو، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران
استادیار، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران

روح اله نوروزی
حامد شهبازی*

چکیده

در این مقاله به طراحی یک سامانه خلبان خودکار با معماری لایه ای و مکانیسم پیاده سازی لایه به لایه کنترلگر بر روی ربات پرنده چهار روتور بر اساس الگوریتم بهینه سازی ازدحام افزایشی ذرات ارائه می شود. در این سامانه هدایت پرنده به جای آنکه در انحصار یک کنترلگر واحد باشد توسط یک سیستم چندعاملی که حداقل از ۴ عامل کنترلگر ساخته شده است، انجام می گیرد. با توزیع وظایف کنترلی و قراردادن فصول مشترک برای لایه های کنترل کننده، پهباد قادر به جابجاکرد ناختمیاریات کنترلی بین لایه ها می باشد و این امر موجب جلوگیری از عملکرد نامناسب پهباد خواهد شد. علاوه بر آن لایه ها قادر خواهند بود از مکانیسم کنترلی لایه دیگر رفتار مورد نظر خود را یاد بگیرند. در فرآیند یادگیری لایه ای، یک مکانیسم بهینه سازی جدید برای کنترلگرهای شبکه عصبی ارائه می شود که بهینه سازی ازدحام افزایشی ذرات نامیده می شود. در یک لایه آموزشی کنترلگر غیرخطی مد لغزشی و گام به عقب پیاده سازی شده است و رفتار و کارایی این کنترلگرها با شبیه سازی مورد مقایسه قرار گرفته اند در نهایت در این مقاله یک کنترلگر موفق شبکه عصبی در لایه، برای سیستم کاملاً غیرخطی با تنظیم وزن های شبکه توسط الگوریتم ازدحام افزایشی ذرات طراحی شده است که قابلیت یادگیری رفتار کنترلگرهای لایه های دیگر را دارا می باشد.

واژه های کلیدی: کنترل لایه ای، چهارروتور، کنترلگر گام به عقب، کنترلگر مد لغزشی، شبکه عصبی مصنوعی، ازدحام افزایشی ذرات.

Designing a Layered Learning System and Layer by Layer Implementation of a Controller on Quad-rotor Based on Growing Particle Swarm Optimization

R. Norouzi

Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

H. Shahbazi

Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

Abstract

In the paper an auto-pilot with layered control architecture has been designed for an autonomous quad-rotor and an optimized proportional-derivative controller has simulated on the quad-rotor. In the auto-pilot with layered control architecture, quad-rotor will be controlled with a multi agent system which has at least 4 agents to control the system instead of just having a single controller. Each agent has placed in an independent layer with its own CPU, memory and processing equipment that perform individually. Nonlinear controllers such as back stepping and sliding mode are presented. In the layered learning process, a new optimization method for neural networks is presented in the paper which is named Growing Particle Swarm Optimization. In one of the layers some classic controllers such as back stepping and sliding mode is used to train the neural controller in the other layer. Optimization results show that the neural network controller can successfully learn the controlling task using Growing Particle Swarm Optimization.

Keywords: Quad-rotor, layered learning system, back stepping controller, sliding mode controller, artificial neural network, growing particle swarm.

۱- مقدمه

می دهد از این مدل سازی ریاضی برای شبیه سازی و طراحی کنترلگر استفاده شده است. آنچه که این ربات را نسبت به سایر ربات های پرنده متفاوت می کند ساختار منحصر به فرد آن می باشد. به دلیل ساختار این ربات، دینامیک وضعیت ربات به شدت با دینامیک موقعیت ربات وابسته است. برای سیستم های پیچیده غیرخطی شبکه های عصبی مصنوعی با توانایی تقریب غیرخطی، توانایی پردازش موازی و توانایی یادگیری تطبیقی به صورت گسترده توسعه داده شده اند. در این مقاله اهداف زیر دنبال شده است:

- ۱- طراحی و پیاده سازی سامانه خلبان خودکار با معماری لایه ای
- ۲- طراحی کنترلگر غیرخطی گام به عقب
- ۳- طراحی کنترلگر غیرخطی مد لغزشی
- ۴- مقایسه رفتار دو کنترلگر غیرخطی
- ۵- بهینه سازی کنترلگرهای غیرخطی با الگوریتم ازدحام افزایشی

چهارروتوریک وسیله پرنده با شش درجه آزادی حرکت می باشد که قابلیت پرواز عمودی و انجام مانورهای پیچیده را دارا است. این سازه دارای ساختاری شبیه صلیب است با دو بازوی عمود برهم که چهار ملخ در گوشه های آن قرار دارد و با استفاده از تغییر سرعت ملخ های تواند حرکات و مانورهای مختلف را انجام بدهد. این وسیله به سبب قابلیت نشست و برخاست عمودی در دسته عمودپروازها قرار می گیرد که به جهت برخی مزایا و ویژگی ها مورد توجه قرار گرفته است. توانایی حمل بار، سادگی ساختار وسیله، قابلیت مانور پذیری بالا، داشتن قیود حرکتی کم و هزینه پایین تعمیر و نگهداری از جمله این ویژگی هاست. مدل سازی ریاضی چهارروتور، حالت این پرنده را با پارامترهای فیزیکی موجود از قبیل سرعت، نیرو و نیروی رانش نتیجه

* نویسنده مکاتبه کننده، آدرس پست الکترونیکی: hamedshahbazi@gmail.com

ذرات

۶- آموزش شبکه عصبی با استفاده از کنترلگر گام به عقب بهینه سازی شده

۷- طراحی کنترلگر عصبی بر اساس الگوریتم ازدحام افزایشی ذرات در اولین هدف به منظور مقابله با اخلاص در عملکرد پهباد در اثر نفوذ سایبری یا عوامل جوی-محیطی یک سیستم خلبان خودکار ارائه شده است که دارای معماری لایه ای می باشد، و به شکل هوشمندی کنترل و یادگیری پهباد را در هر لایه توزیع کرده است. هدف آخر در این مقاله طراحی یک کنترلگر شبکه عصبی، با تنظیم وزن های شبکه توسط الگوریتم ازدحام افزایشی ذرات است. به منظور آموزش بین لایه ای روی شبکه عصبی مصنوعی در این خلبان خودکار از تعدادی روش کلاسیک استفاده می شود.

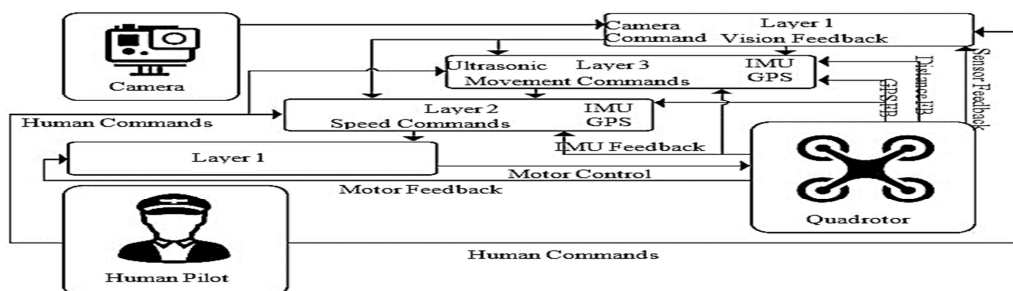
در ادامه این بخش به بررسی تعدادی از روش های کلاسیک ارائه شده برای کنترل ربات پرنده پرداخته می شود. در سال 2005 بو عبدالله و همکارانش دو کنترلگر غیرخطی گام به عقب و مد لغزشی را برای سیستم چهارروتور طراحی کرده و رفتار این دو کنترلگر را با یکدیگر مقایسه کردند. این مقایسه نشان می دهد کنترلگر گام به عقب پاسخ بهتری دارد [1]. در [2] یک روش کنترلی جدید گام به عقب ارائه شده است و برای بررسی کارایی کنترلگر طراحی شده، رفتار این کنترلگر با دو کنترلگر مد لغزشی و خطی سازی فیدبک مقایسه شده است. در مقاله [3] کارایی کنترلگرهای PID کلاسیک با کنترلگر بهینه ساز خطی روی یک سیستم چهارروتور برای کنترلگر ارتفاع سیستم بررسی شده است. در این مقاله یک مدل دینامیکی دقیق و مناسب برای سیستم آمده و توصیف هایی در مورد عملگرها و حسگرهای سیستم آورده شده است. در مقالات [4، 5] مدل دینامیکی بر اساس مدل لاگرانژ به دست آمده و رفتار کنترلگر گام به عقب برای کنترل وضعیت زوایا و کنترل موقعیت چهارروتور بررسی شده است. در این مقالات کنترلگر پیشنهاد شده بر روی سیستم واقعی نیز پیاده سازی شده است. در [6] یک کنترلگر گام به عقب برای مدل دینامیکی به دست آمده به روش لاگرانژ برای بهبود پایداری سیستم طراحی شده که قابلیت بالایی در کنترل سازه دارد. در این مقالات کنترلگر وضعیت می تواند در برابر عدم قطعیت های مدل پاسخ خوبی از خود نشان بدهد. یکی از روش هایی که می تواند نامعینی های غیرخطی سیستم را تشخیص دهد شبکه عصبی وفق یافته با روش های کنترلی تطبیقی می باشد. در مقاله [7، 8] کنترلگرهای تطبیقی برای تعیین عدم قطعیت ها ارائه شده است. در [9] دینامیک معکوس با استفاده از شبکه عصبی برای محاسبه نامعینی های دینامیکی و خطاهای دینامیک معکوس برای مدل بالگرد استفاده شده می شود. در [10] یک کنترلگر جدید که بانام کنترلگر پسودو هدینگ¹ با استفاده از شبکه عصبی یک لایه، برای محاسبه نامعینی های سیستم به کار گرفته شده است. در [11] یک کنترلگر ترکیبی گام به عقب با شبکه عصبی برای مواجه شدن با خطاهای مدل و دینامیک مدل نشده سیستم پیاده شده است. در [12] یک کنترلگر PID بهینه سازی شده با الگوریتم ازدحام ذرات را پیشنهاد داده اند و با مقایسه نتایج کنترلگر گام به عقب و

کنترلگر PID بهینه سازی شده، کارایی کنترلگر PID بهینه سازی شده مشخص شده است. در [13] نیز یک کنترلگر نا وابسته به مدل از نوع کنترلگر فازی-PID طراحی کرده اند که نتایج حاصل نشان دهنده کارایی مناسب این کنترلگر می باشد. در [14] مدل دینامیکی چهارروتور را بر اساس معادلات نیوتن اوپلر به دست آورند و یک کنترلگر گام به عقب طراحی و شبیه سازی کردند که کارایی خوبی را از خود نشان داده است. در [15] یک کنترلگر هوشمند براساس شبکه عصبی برای کنترلگر نشست و برخاست ربات عمود پرواز طراحی و ارائه شده است. در [16] با استفاده از روش های باز خورد خطی و شبکه عصبی تطبیقی، کنترلگری برای کنترل وضعیت چهارروتور ارائه شده است و با شبیه سازی این کنترلگر با استفاده از پارامترهای چهارروتور از پیش ساخته شده، کارایی این کنترلر مشخص شده است. در [17] یک روش کنترلی براساس روش کنترل معکوس، با استفاده از شبکه عصبی آموزش داده شده است. در [18] یک کنترلگر ترکیبی مد لغزشی-شبکه عصبی ارائه شده است. در این مقاله برای مقابله با عدم قطعیتها و بهبود کارایی کنترلگر مد لغزشی مد یک شبکه عصبی به این کنترلگر اضافه شده است. در [19] یک کنترلگر عصبی-فازی با بهینه سازی پارامترهای کنترلی توسط الگوریتم های بهینه سازی ژنتیک ارائه و کارایی این کنترلگر با مقایسه با رفتار کنترلگر PID کلاسیک مشخص شده است. در [20] یک کنترلگر شبکه عصبی ارائه شده است که در ان این کنترلگر با استفاده از کنترلگر PID آموزش داده شده است.

۲- طراحی خلبان خودکار لایه ای

در این بخش مقاله به ارائه یک معماری جدید برای چهارروتور پرداخته می شود. اساس این سامانه بر مبنای معماری یادگیری لایه ای است، که در طراحی خلبان خودکار به کار برده شده است. و این امر سطح هوشمندی پرنده را به بیش از ۴ سطح افزایش داده است. در اینجا هر لایه وظایف خاصی را برعهده دارد و برخی از وظایف لایه ها با یکدیگر دارای هم پوشانی است که لایه ها می توانند این وظایف مشترک را به یکدیگر یاد دهند. شکل ۱ معماری کلی این خلبان خودکار و ارتباطات بین لایه ها را نمایش می دهد. لایه اول این سامانه از تعدادی ترانزیستور الکترونیکی ماسفت ساخته شده است که وظیفه کنترل عملگرهای پهباد را در پایین ترین لایه بر عهده می گیرد و از شدت جریان ۱ تا ۵۰ آمپر را برای موتورهای پهباد تغییر می دهد. در لایه دوم سامانه مجموعه ای از حسگرها و مدارات مجتمع الکترونیکی قرار داده شده است که وظیفه کنترل لایه اول و مجموعه ای از وظایف کنترلی دیگر از جمله پایداری پهباد و تعقیب نقاط تنظیم را بر عهده دارد. لایه سوم سامانه وظیفه اصلی ناوبری خودکار و یادگیری مسیر را برعهده دارد. این لایه با کمک حسگرهای گوناگون جهت یابی و مکان یابی و فاصله سنجی حالت محیط پیرامون پرنده را کشف کرده و باداشتن پردازنده مستقل و پر سرعت الگوریتم های هوش مصنوعی را اجرا می نماید. لایه های بالاتر از لایه سوم در این خلبان خودکار مجموعه ای از کنترلگرهای هوشمند نرم افزاری هستند که درون رایانه اصلی پهباد قرار گرفته اند و قادر به تصمیم گیری در مورد عملیات گروهی پهباد و ارتباط با سایر لایه های بالاتر در سایر پهبادها خواهند بود. استفاده از

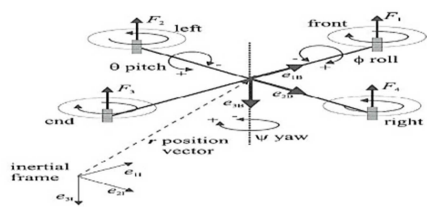
¹ pseudo-control hedging



شکل ۱- نمای کلی از خلبان خودکار با معماری لایه‌ای

چندعاملی در هوشمند سازی پهپادها امکان ساخت دستگاه‌های چندعاملی همکار، بین عامل‌ها را فراهم می‌سازد و از این طریق می‌توان مأموریت‌های دسته‌جمعی را برای پهپادها تعریف و پیاده‌سازی نمود.

$$\ddot{\psi} = \dot{\theta} \phi \left[\frac{I_{xx} - I_{yy}}{I_{zz}} \right] + \frac{1}{I_{yy}} U_4 \quad (۱-۵)$$



شکل ۲- پیکربندی پرنده چهارروتور

که در آن x, y, z موقعیت مرکز جرم در سازه و ϕ, θ, ψ زوایای اولیه هستند. همچنین m, I_x, I_y, I_z جرم و ممان اینرسی چهارروتور هستند و l نیز فاصله بین روتورها و مرکز جرم است و Ω, Ω_R به ترتیب ممان اینرسی و سرعت زوایای تیغه‌های پرواز هستند. U_1, U_2, U_3, U_4 ورودی‌های سیستم و به ترتیب نیروی مجموع در راستای z و مومنتم‌های چرخش حول محور y ، چرخش حول محور x و چرخش حول محور z تولیدشده توسط تیغه‌های پرواز می‌باشند. رابطه ورودی‌های کنترلی با سرعت روتورها به صورت زیر بیان می‌شوند:

$$U_1 = \sum_{i=1}^4 T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (۲-الف)$$

$$U_2 = (-T_2 + T_4) = bl(-\Omega_2^2 + \Omega_4^2) \quad (۲-ب)$$

$$U_3 = (T_1 + T_3) = bl(\Omega_1^2 - \Omega_3^2) \quad (۲-پ)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (۲-ت)$$

که در این معادلات $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ سرعت‌های زوایای روتورها می‌باشند و b ضریب نیروی بالابرنده ملخ و d ضریب نیروی پسا می‌باشد. طبق معادلات سیستم و ارتباط بین آن‌ها مشاهده می‌شود که حرکت چرخشی چهارروتور به حرکت انتقالی آن بستگی ندارد اما حرکت انتقالی این ربات به حرکت چرخشی آن شدیداً وابسته است؛ بنابراین سیستم کلی چهارروتور می‌تواند به دو بخش حلقه وضعیت و حلقه موقعیت تقسیم شود. شکل ۳ ساختار این وضعیت را مشخص می‌کند.

یادگیری لایه‌ای و هوشمندی چند عامله در طراحی این خلبان خودکار سبب می‌گردد تا احتمال بروز اختلال و خرابی در سیستم هوشمند پهپاد تا حدود زیادی کاهش یابد. استفاده از معماری

۳- مدل‌سازی دینامیکی چهارروتور

حرکت چهارروتور با استفاده از تغییر سرعت چرخش روتورها انجام می‌گیرد. پیکربندی چهارروتور در شکل ۲ مشاهده می‌شود، این ساختار متشکل از دو بازوی عمود برهم می‌باشد و هرکدام از این بازوها دارای موتورهایی در انتهای خود می‌باشند. موتور شماره ۱ و ۳ در راستای ساعت‌گرد و موتور شماره ۲ و ۴ در راستای پادساعت‌گرد می‌چرخند. کنترل حرکت چهارروتور در کنترل و تنظیم سرعت موتورهای این ربات خلاصه می‌شود. کنترلرها سعی بر آن دارند تا با استفاده از تغییر سرعت روتورها به یک نیروی تعادل برای یک پرواز مناسب دست پیدا کنند. با تغییر سرعت هر یک از روتورها نیروی بالابرنده همان موتور افزایش یا کاهش پیدا می‌کند و باعث حرکت چهارروتور می‌شود. باکم و زیاد کردن سرعت همه‌ی موتورها به یک اندازه می‌تواند حرکت عمودی را برای سیستم تولید کرد. تغییر در سرعت هریک از موتورهای ۲ و ۴ باعث حرکت چرخشی حول محور x و تغییر در سرعت هریک از موتورهای ۱ و ۳ باعث حرکت چرخشی حول محور y می‌شود. حرکت چرخشی حول محور z ناشی از اختلاف گشتاور تولیدی بین هر جفت روتور می‌باشد. مدل ریاضی چهارروتور با استفاده از معادلات زیر که برگرفته از روش نیون-اولر می‌باشد، از [1] گرفته شده است. در این معادلات، معادلات (۱-الف) تا (۱-پ)، سیستم را در فضای دکارتی توصیف می‌کند. سه معادله (۱-ت) تا (۱-ه) نیز دینامیک سیستم را بر اساس زوایای اولیه و نیون-اولر معرفی می‌کنند. این معادلات رفتار ساده چهارروتور با صرف‌نظر کردن از نیروهای آئرو دینامیک وارد بر سیستم و تأثیرات ژيروسکوپی تولیدشده از چرخش روتورها را مدل می‌کند.

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \quad (۱-الف)$$

$$\ddot{y} = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{U_1}{m} \quad (۱-ب)$$

$$\ddot{z} = -g + (\cos(\phi) \cos(\theta)) \frac{U_1}{m} \quad (۱-پ)$$

$$\ddot{\phi} = \dot{\theta} \psi \left[\frac{I_{yy} - I_{zz}}{I_{xx}} \right] + \frac{J_r}{I_{xx}} \dot{\theta} \Omega_d + \frac{1}{I_{xx}} U_2 \quad (۱-ت)$$

$$\ddot{\theta} = \dot{\phi} \psi \left[\frac{I_{zz} - I_{xx}}{I_{yy}} \right] - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_d + \frac{1}{I_{yy}} U_3 \quad (۱-ج)$$

که یک کنترل کننده بازگشتی با در نظر گرفتن برخی حالت های سیستم به عنوان ورودی مجازی، طراحی کرده و نهایتاً ورودی های کنترلی واقعی برای پایداری سازی کل سیستم مورد استفاده قرار می گیرد. با تعریف متغیرهای حالت به صورت بردار $x^T = [\varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}]$ فضای حالت به صورت (۳) بازنویسی می شود:

$$f(x,u) = \begin{pmatrix} x_2 \\ x_4 a_1 + x_4 a_2 \Omega + b_1 U_2 \\ x_4 \\ x_2 x_6 a_3 + x_2 x_4 \Omega + b_2 U_3 \\ x_6 \\ x_4 x_2 a_5 + b_3 U_4 \\ x_8 \\ -g + (\cos x_1 \cos x_3) \frac{U_1}{m} \\ x_{10} \\ U_x \frac{U_1}{m} \\ x_{12} \\ U_y \frac{U_1}{m} \end{pmatrix} \quad (3)$$

که روابط (۴) در فرم فضای حالت (۳) برقرار می باشد:

$$a_1 = \frac{I_y - I_z}{I_x} \quad (4-الف)$$

$$a_2 = -\frac{J_R}{I_x} \quad (4-ب)$$

$$a_3 = \frac{I_z - I_x}{I_y} \quad (4-پ)$$

$$a_4 = \frac{J_R}{I_y} \quad (4-ت)$$

$$a_5 = \frac{I_x - I_y}{I_z} \quad (4-ج)$$

$$b_1 = \frac{1}{I_x} \quad (4-ه)$$

$$b_2 = \frac{1}{I_y} \quad (4-و)$$

$$b_3 = \frac{1}{I_z} \quad (4-ن)$$

$$U_x = (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \quad (4-م)$$

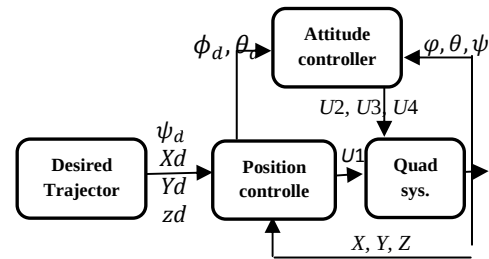
$$U_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \quad (4-ی)$$

معادله (۳) بادر نظر گرفتن ورودی های جدید، کنترل پذیر بوده که در ادامه به روش گام به عقب ورودی های کنترلی طراحی می گردند. در اولین گام با تعریف خطای ردیابی به صورت $z_1 = x_{1d} - x_1$ و انتخاب تابع لیاپانوف $v(z_1) = \frac{1}{2} z_1^2$ مشتق زمانی آن به صورت (۵) حاصل می گردد:

$$\dot{v}(z_1) = z_1(\dot{x}_{1d} - \dot{x}_2) \quad (5)$$

با در نظر گرفتن x_2 به عنوان ورودی مجازی و طراحی آن به صورت $\alpha_2 = \dot{x}_{1d} + \alpha_1 z_1$ رابطه لیاپانوف با انتخاب مقادیر مثبت پارامتر α_1 به صورت (۶) منفی معین شده و پایداری z_1 تضمین می شود.

$$\dot{v}(z_1) = -\alpha_1 z_1^2 \quad (6)$$

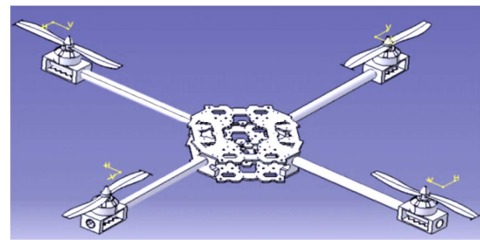


شکل ۳- دیاگرام کنترلی موقعیت چهارروتور

که $(x_d, \dot{x}_d, y_d, \dot{y}_d, z_d, \dot{z}_d)$ مقادیر مطلوب می باشند و $(\phi_d, \dot{\phi}_d, \theta_d, \dot{\theta}_d, \psi_d, \dot{\psi}_d)$ مقادیر مطلوب زاویه ای می باشند که توسط کنترلر موقعیت تعیین می شوند. ورودی U_1 که ورودی کنترلی ارتفاع را مشخص می کند توسط کنترلر موقعیت و سه ورودی کنترلی U_2, U_3, U_4 توسط کنترلر وضعیت محاسبه می شوند.

۴- مدل سازی سیستم در محیط Catia

برای به دست آوردن اندازه پارامترهای دینامیکی سیستم، برای مدل سازی سیستم در محیط Simulink این پارامترها با استفاده از محیط شبیه سازی Catia، محاسبه شده و در جدول ۱ نشان داده شده اند. پیاده سازی سیستم در محیط کتیا نیز در شکل ۴ نشان داده شده است.



شکل ۴- مدل سازی سیستم در محیط Catia

جدول ۱- پارامترهای سیستم برای شبیه سازی

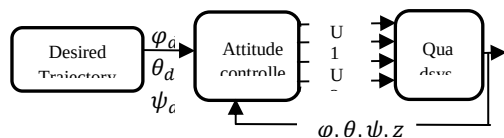
پارامترهای سیستم	مقدار	واحد
m	۱	kg
Ix	۰/۰۰۷۵	kg.m ²
Iy	۰/۰۰۷۵	kg.m ²
Iz	۰/۰۱۳	kg.m ²
Jr	۰/۰۰۰۰۰۶۵	kg.m ²
l	۰/۲۳	m
b	۰/۰۰۰۰۳۱۳	Ns ²
d	۰/۰۰۰۰۰۷۵	Nms ²

۵- طراحی کنترلرهای غیرخطی در لایه پایین

در این بخش به طراحی کنترلرهای غیرخطی گام به عقب و مود لغزشی در لایه دوم پرداخته می شود. این کنترلر ها وظیفه تعقیب دستورات ارسالی از لایه سوم را بر عهده می گیرند و علاوه بر آن جهت آموزش شبکه عصبی لایه سوم مورد استفاده قرار می گیرند.

۵-۱- طراحی کنترلر گام به عقب

در این بخش یک روش کنترلی گام به عقب برای کنترل چهارروتور ارائه شده است. رویکرد روش گام به عقب بر این اساس است



شکل ۵- دیاگرام کنترلی وضعیت سیستم

بهره های کنترلی کنترلر گام به عقب برای شبیه سازی به صورت

جدول ۲ در نظر گرفته شده اند.
 $\dot{v}(z_1, z_2) = z_2(a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2) - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2$
 جدول ۲- بهره های کنترلر گام به عقب برای شبیه سازی

پارامتر	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
مقدار	۵	۵	۵	۱۵	۸	۱۵	۷	۱۲

۵-۲- طراحی کنترلر مد لغزشی

در این روش با استفاده از معادله فرم فضای حالت (۳) به طراحی کنترلر مد لغزشی پرداخته می شود. در این روش در گام اول به یک سطح لغزش برای تعقیب خطا انتخاب می شود و سپس با استفاده از معادله لیاپانوف به طراحی ورودی های کنترلی پرداخته می شود. سطوح لغزش بر اساس حالت های سیستم به صورت معادلات (۱۶) تعریف شده است:

$$S_0 = \dot{e}_0 + \lambda_0 e_0 \quad (۱۶-الف)$$

$$S_\theta = \dot{e}_\theta + \lambda_\theta e_\theta \quad (۱۶-ب)$$

$$S_\psi = \dot{e}_\psi + \lambda_\psi e_\psi \quad (۱۶-پ)$$

$$S_z = \dot{e}_z + \lambda_z e_z \quad (۱۶-ت)$$

$$e_0 = \theta_d - \theta \quad (۱۶-ج)$$

$$e_\theta = \theta_d - \theta \quad (۱۶-ه)$$

$$e_\psi = \psi_d - \psi \quad (۱۶-و)$$

$$e_z = z_d - z \quad (۱۶-ن)$$

که در آن رابطه $\lambda_i > 0$ برقرار است. در گام دوم با در نظر گرفتن تابع لیاپانوف به صورت (۱۷) داریم:

$$v(S_0) = \frac{1}{2} S_0^2 \quad (۱۷)$$

اگر رابطه $\dot{v}(S_0) = S_0 \dot{S}_0 < 0$ برقرار باشد پایداری لیاپانوف تضمین می شود و سپس با استفاده از همین رابطه قوانین کنترلی را می تواند استخراج کرد. بنابراین از معادله پایداری لیاپانوف داریم:

$$\dot{S}_0 = -K_1 \text{sign}(S_0) = \ddot{x}_{1d} - \dot{x}_2 + \lambda_1 e_1 = -a_1 x_4 x_6 + a_2 x_4 \Omega + \ddot{\theta}_d + \lambda_\theta \dot{e}_\theta \quad (۱۸-الف)$$

$$U_2 = \frac{1}{b_1} (-K_1 \text{sign}(S_0) - a_1 x_4 x_6 + a_2 x_4 \Omega + \ddot{\theta}_d + \lambda_\theta \dot{e}_\theta) \quad (۱۸-ب)$$

به همین ترتیب می توان قوانین کنترلی U_1, U_3, U_4 را به صورت معادلات (۱۹) استخراج کرد.

$$U_3 = \frac{1}{b_2} (-K_2 \text{sign}(S_\theta) - a_3 x_2 x_6 + a_4 x_2 \Omega + \ddot{\theta}_d + \lambda_\theta \dot{e}_\theta) \quad (۱۹-الف)$$

$$U_4 = \frac{1}{b_3} (-K_3 \text{sign}(S_\psi) - a_5 x_2 x_4 + \ddot{\psi}_d + \lambda_\psi \dot{e}_\psi) \quad (۱۹-ب)$$

$$U_1 = \frac{m}{\cos x_1 \cos x_3} (g - K_4 \text{sign}(S_z) + \ddot{z}_d + \lambda_z \dot{e}_z) \quad (۱۹-پ)$$

x_2 در رابطه (۵) در واقع یک مقدار مطلوب x_{2d} می باشد که چنانچه متغیر حالت واقعی x_2 با آن برابر شود آنگاه x_1 به مقدار مطلوب x_{1d} همگرا می شود. بنابراین متغیر خطای جدید (۷) معرفی می گردند.

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \quad (۷)$$

در گام دوم با در نظر گرفتن تابع لیاپانوف جدید به صورت $v(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2)$ مشتق زمانی آن از (۸) بدست می آید:

$$\dot{v}(z_1, z_2) = z_2(a_1 x_4 x_6 + a_2 x_4 \Omega + b_1 U_2) - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) \quad (۸)$$

$$-z_1 z_2 - \alpha_1 z_1^2$$

ورودی کنترلی U_2 از رابطه بالا و با در نظر گرفتن $\dot{x}_{1,2,3,d} = 0$ ارضا شدن رابطه $\dot{v}(z_1, z_2) < 0$ به صورت (۹) استخراج می شود:

$$U_2 = \frac{1}{b_1} (z_1 - a_1 x_4 x_6 + a_2 x_4 \Omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \quad (۹)$$

ترم $\alpha_2 z_2$ با $\alpha_2 > 0$ برای پایداری z_1 اضافه شده است. با طراحی ورودی کنترلی به صورت زیر، رابطه بالا با انتخاب $\alpha_2 > \alpha_1$ منفی معین شده و بنابراین تضمین می گردند حالت های جدید (z_1, z_2) به صفر میل کرده و حالت x_2 به مقدار مطلوب همگرا شده که در نتیجه، حالت x_1 به طور مجانبی به x_{1d} همگرا می شود.

$$\dot{v}(z_1, z_2) = -\alpha_1 z_1^2 - (\alpha_2 - \alpha_1) z_2^2 \geq 0 \quad (۱۰)$$

با دنبال کردن همین روند سایر ورودی های کنترلی به صورت (۱۱) و (۱۲) به دست می آیند:

$$U_3 = \frac{1}{b_2} (z_3 - a_3 x_2 x_6 + a_4 x_2 \Omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4) \quad (۱۱)$$

$$U_4 = \frac{1}{b_3} (z_5 - a_5 x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6) \quad (۱۲)$$

که در آن روابط (۱۳) برقرارند:

$$z_3 = x_{3d} - x_3 \quad (۱۳-الف)$$

$$z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \quad (۱۳-ب)$$

$$z_5 = x_{5d} - x_5 \quad (۱۳-پ)$$

$$z_6 = x_6 - \dot{x}_{5d} - \alpha_5 z_5 \quad (۱۳-ت)$$

به همین روش برای ورودی کنترلی ارتفاع U_1 داریم:

$$U_1 = \frac{m}{\cos x_1 \cos x_3} (z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \quad (۱۴)$$

که در آن روابط (۱۵) برقرارند می باشد:

$$z_7 = x_{7d} - x_7 \quad (۱۵-الف)$$

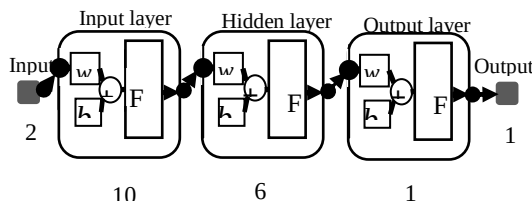
$$z_8 = x_8 - \dot{x}_{7d} - \alpha_7 z_7 \quad (۱۵-ب)$$

حال با توجه به ورودی های کنترلی به دست آمده به روش گام به عقب، می تواند به شبیه سازی این کنترلر بر روی مدل دینامیکی چهار روتور پرداخت. این کنترلر طبق شکل ۵ پیاده سازی شده است.

بهره های کنترلی کنترلر مد لغزشی برای شبیه سازی به صورت جدول ۳ در نظر گرفته شده اند.

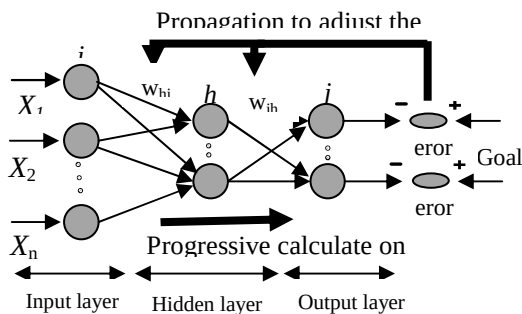
جدول ۳- بهره های کنترلگر مد لغزشی برای شبیه سازی

پارامتر	λ_θ	λ_ψ	λ_z	K_1	K_2	K_3	K_4
مقدار	0.5	0.3	4	0.1	0.1	0.1	0.1



شکل ۶- ساختار شبکه عصبی استفاده شده در لایه سوم

در لایه سوم برای آموزش شبکه عصبی از روش پس انتشار خطا برای یادگیری وزن های شبکه استفاده شده است. در این روش سعی می شود تا با وزندهی مناسب به بایاسها و وزنه های شبکه عصبی، خطای بین خروجی های تولید شده با این وزن ها توسط شبکه عصبی و تابع هدف در نظر گرفته شده برای شبکه عصبی کمینه شود. فضای پاسخ در این روش شامل فضای بزرگی است که توسط همه مقادیر ممکن برای وزنها تعریف می شود. در این پژوهش شرط خاتمه آموزش برای این سیستم نیز، کم شدن خطابه اندازه ۰/۰۱ در نظر گرفته شده است. ساختار این الگوریتم در شکل ۷ آورده شده است.



شکل ۷- دیاگرام کلی الگوریتم آموزش شبکه عصبی به روش پس انتشار خطا

در الگوریتم پس انتشار خطا، خطای بین خروجی شبکه عصبی و خروجی هدف به صورت معادله (۲۰) تعریف می شود:

$$E(\bar{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{output}} (t_{kd} - O_{kd})^2 \quad (20)$$

که در آن مقصود از outputs خروجی های مجموعه واحدهای لایه خروجی و t_{kd} مقدار هدف و خروجی متناظر با k امین واحد خروجی و مثال آموزشی d و D نیز مجموعه نمونه های آموزشی است. به ازای هر واحد خروجی مقدار خطا از رابطه (۲۱) به دست می آید:

$$\delta_k = O_k(1 - O_k)(t_k - O_k) \quad (21)$$

برای هر لایه مخفی رابطه خطا به صورت (۲۲) می باشد:

$$\delta_k = O_h(1 - O_h) \sum_j W_{jh} \delta_k \quad (22)$$

که O_h مقدار خروجی نورون های لایه پنهان و W_{jh} وزن های لایه پنهان هستند. مقادیر وزنی در هر مرحله به صورت (۲۳) تغییر می کند:

$$W_{ij} = W_{ij} + \Delta W_{ij} \quad (23 \text{ الف})$$

$$\Delta W_{ij} = \eta \delta_i X_{ij} \quad (23 \text{ ب})$$

که در آن η نرخ یادگیری سیستم می باشد. در این مقاله خروجی نورون ها توسط تابع غیرخطی (۲۴) محاسبه می شود.

۶- کنترلگر شبکه عصبی در لایه سوم

یکی از مزایای کنترلر لایه ای ایجاد مقاومت کنترلی در مجموعه کلی ربات پرنده است. به دلیل وجود عدم قطعیت در مدل سازی کنترلگر یک لایه ای مقاومت کافی از خود نشان نمی دهد. برای رفع این مشکل و رفع نیاز کنترلگر به پارامترهای چهاررور و برای طراحی کنترلر سعی شده است تا کنترلگری در لایه بالاتر مستقل از دینامیک پرنده طراحی شود. برای غیر وابسته نمودن کنترلگر به مدل سیستم می توان از روش های هوشمند که توانایی تصمیم گیری را دارند استفاده کرد. در این مقاله با استفاده از شبکه عصبی مصنوعی، کنترلگری در لایه سوم آموزش داده می شود تا با استفاده از خطای موجود بین حالت ایده ال که سیستم می خواهد در آن موقعیت قرار گیرد و حالت فعلی که سیستم در آن قرار دارد، ورودی کنترلی مناسبی برای سیستم ایجاد شود. برای آموزش دادن شبکه عصبی نیاز به یک سری داده ورودی و خروجی برای آموزشی باشد. برای به دست آوردن این داده های آموزشی در این مقاله، یک کنترلگر گام به عقب برای کنترلر سیستم طراحی و پیاده سازی شده است. کنترلر گام به عقب به دلیل سرعت پاسخ خوب از جمله کنترلگرهای مناسب برای این سیستم می باشد که مورد توجه قرار گرفته است. این شبکه عصبی مصنوعی از سه بخش لایه ورودی، لایه پنهان و لایه خروجی تشکیل شده است. در هر لایه از یک سری نورون برای پردازش اطلاعات استفاده شده است. خروجی نورون ها توسط یک تابع غیرخطی و با اعمال این تابع روی ورودی ها حاصل می شود. برای شبکه عصبی استفاده شده در این مقاله برای هر جز کنترلی، دو ورودی خطای کنترلی و مشتق خطای کنترلی و یک خروجی ورودی کنترلی در نظر گرفته شده است. در این کنترلگر سه لایه پنهان در نظر گرفته شده است. در لایه اول ۱۰ نورون، در لایه دوم ۶ نورون و در لایه آخر یک نورون بکار گرفته شده است. ساختار این شبکه در شکل ۶ نشان داده شده است. برای آموزش این شبکه عصبی نیاز به داده های آموزشی می باشد به همین دلیل ابتدا شبیه سازی کنترلگر گام به عقب صورت گرفته که نتایج آن به صورت زیر آورده شده است. در این پژوهش از ۳۰۰۰ داده ورودی، خروجی برای آموزش شبکه عصبی مصنوعی استفاده شده است. پس از آموزش کنترلگر شبکه عصبی طراحی شده در لایه سوم، می توان وظایف کنترلی لایه دوم را به لایه سوم واگذار نمود. ساختار شبکه عصبی استفاده شده در لایه سوم در شکل ۶ آورده شده است.

می‌کند، این روش در حل مسائل بهینه‌سازی بلادرنگ در کنترلگر 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. در ابتدا فرایند بهینه‌سازی را برای اولین گام زمان ($k = k_0$) شرح می‌دهیم تا مقدار بهینه W^{k_0} را به دست آوریم. فرایند بهینه‌سازی ترکیبی پارامترهای سیستم کنترلی، یک فرایند 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 (۲۶) محاسبه می‌شود:

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

در این معادله α و β و γ سه 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 دور های الگوریتم به تدریج کاهش می‌یابد تا الگوریتم بهتر همگرا گردد. γ یک ضریب شتاب دهی است که مقدار اینرسی ذرات را معین می‌سازد. $BPos(it)$ بهترین بردار در مجموعه بردار های دور i ام است که کمترین مقدار خطا را تولید کرده است و $GPos(it)$ 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 بر آورد را با محاسبه تابع هزینه 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 $W^{k_{pl}} = 1, 2, 3, \dots$ شروع می‌شود.

در این مرحله یک فرایند 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 اجرای بلادرنگ کنترلگر فرض می‌کنیم $P \ll Q$ باشد. به جمعیت اولیه تصادفی جدید ذره W^{k_0} را اضافه می‌کنیم و الگوریتم را مشابه قبل اجرا می‌کنیم تا W^{k_1} بدست آید. این روند در هر لحظه زمانی تکرار می‌شود تا مقدار W^{k_l} با داشتن $W^{k_{l-1}}$ محاسبه گردد. این روند در شکل ۸ به تصویر درآمده است. شکل اجرای این روش افزایشی است. بدین‌صورت که در حین اجرای مراحل الگوریتم برای دوره جمعیت می‌توانیم الگوریتم کنترلی را اجرا نماییم.

$$a = \text{tansig}(n) = \frac{2}{(1 + e^{-2n})} - 1 \quad (24)$$

۷- طراحی کنترلگر شبکه عصبی مصنوعی بر اساس الگوریتم ازدحام افزایشی ذرات

در سامانه یادگیری لایه ای، لایه چهارم سیستم یادگیری لایه ای وظیفه بهینه سازی پارامتر های لایه سوم را بر عهده گرفته است. در لایه چهارم از یک الگوریتم هوشمند بهینه‌سازی استفاده شده است. ازجمله الگوریتم‌های هوشمند بهینه‌سازی می‌توان به الگوریتم‌های ازدحام ذرات، الگوریتم مورچه، الگوریتم زنبور عسل و ... اشاره کرد. این الگوریتم‌ها یک روش هوشمند بر اساس رفتار گروهی جمعی از کنشگران تشکیل شده‌اند. این کنشگران به‌صورت محلی با یکدیگر و با محیط پیرامون خود در ارتباط هستند و سعی بران دارند تا در فضای پاسخ به دنبال پاسخی باشند تا تابع هزینه تعریف شده را کمینه کنند. در همه روش‌های بهینه‌سازی مذکور رفتار بهینه‌سازی به شکل برون خط و با صرف زمان محاسباتی زیادی شکل می‌گیرد.

الگوریتم ارائه شده در این مقاله که ازدحام افزایشی ذرات نامیده شده است این امکان را فراهم می‌سازد تا بهینه‌سازی کنترلگر شبکه عصبی را انجام دهد. در این الگوریتم ابتدا یک جمعیت تصادفی نه چندان بزرگ برای ذرات که بیانگر پاسخ سیستم می‌باشند تعریف می‌شود. سپس تابع هزینه هر ذره محاسبه شده و بر اساس مناسب بودن پاسخ نسبی هر ذره و پاسخ بهینه مطلق، سرعتی به هر ذره اختصاص داده می‌شود. مراحل بعدی بهینه‌سازی در حین اجرای کنترلر با تعریف جمعیت‌های کوچک تصادفی جدید صورت می‌گیرد. در بهینه‌سازی با الگوریتم ازدحام ذرات تابع هزینه به‌صورت (۲۵) معرفی شده است:

$$\text{Cost} = e^{-\beta} (M_p + e_{ss} + t_r) + (1 - e^{-\beta}) t_s \quad (25)$$

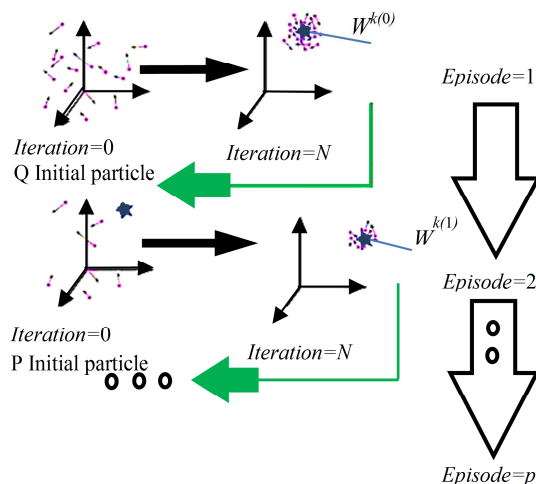
در تابع هزینه t_s زمان نشست سیستم، t_r زمان فراز، e_{ss} خطای ماندگار و M_p بیشینه فرا جهش پاسخ سیستم است. مقدار β مقدار ثابت و برابر ۱/۴ در نظر گرفته شده است. در این مقاله به بهینه‌سازی کنترلرهای غیرخطی و تنظیم وزن‌های کنترلر شبکه عصبی با در نظر گرفتن وزن‌های این شبکه به‌عنوان پارامترهای متغیرهای الگوریتم پرداخته شده است. نتایج بهینه‌سازی این کنترلرها در بخش شبیه‌سازی آورده شده است.

۸- بهینه‌سازی ازدحام افزایشی ذرات

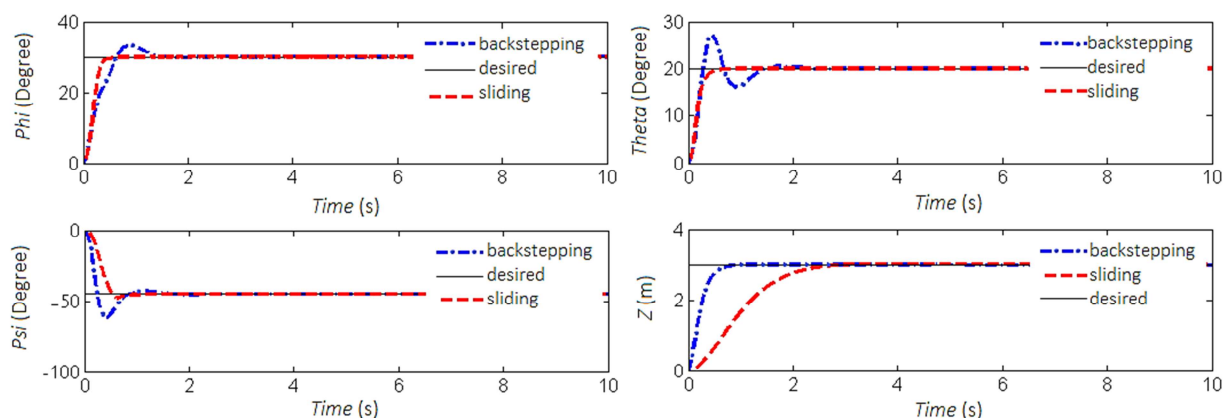
روش بهینه سازی ازدحام ذرات، یک روش سراسری کمینه‌سازی است که با استفاده از آن می‌توان به حل مسائلی که جواب آن‌ها یک نقطه یا سطح در فضای چند بعدی می‌باشد، پرداخت. در این الگوریتم مجموعه ذراتی در فضای پاسخ فرض می‌شوند و یک سرعت ابتدایی به آن‌ها اختصاص داده می‌شود، همچنین کانال‌های ارتباطی بین ذرات در نظر 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

۹- شبیه سازی

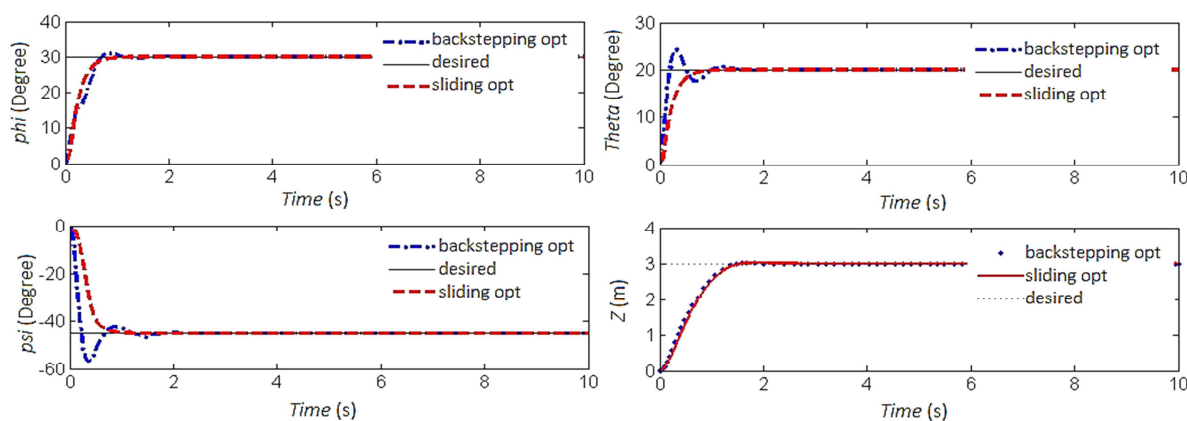
برای کنترل موقعیت چهارروتور به صورت دستی نیاز به کنترلگری است تا با گرفتن زوایای مطلوب از هدایت کننده، و با تغییر این زوایا به زوایای مطلوب، چهارروتور را به سمت موقعیت مورد نظر هدایت کند. در این بخش سعی بر آن شده است تا با تغییر این زوایای مطلوب به شبیه سازی کنترلگرهای مورد مطالعه و بررسی پاسخ های گرفته شده از کنترلگرها و بررسی کارایی کنترلگر جدید و هوشمند شبکه عصبی معرفی شده، پرداخته شود. برای شبیه سازی پارامترهای مورد نیاز از جدول ۱ استخراج شده اند. سپس یک سیستم حلقه بسته با استفاده از کنترلگرهای غیرخطی گام به عقب و مد لغزشی شبیه سازی شده است. با شبیه سازی سیستم برای پاسخ مطلوب $\varphi_d=30, \theta_d=20, \psi_d=-45, z_d=3$ برای این دو کنترلگر غیرخطی، پاسخ سیستم در شکل ۹ آورده شده است. در بخش بعدی به بهینه سازی عملکرد کنترلگرهای غیرخطی مد لغزشی و گام به عقب با استفاده از الگوریتم بهینه سازی ازدحام افزایشی ذرات می شود، عملکرد هر دو کنترلگر نسبت به حالت بهینه سازی نشده، بهبود یافته است.



شکل ۸- مراحل الگوریتم بهینه سازی ازدحام افزایشی ذرات



شکل ۹- پاسخ کنترلگرهای گام به عقب و مد لغزشی به ورودی مطلوب

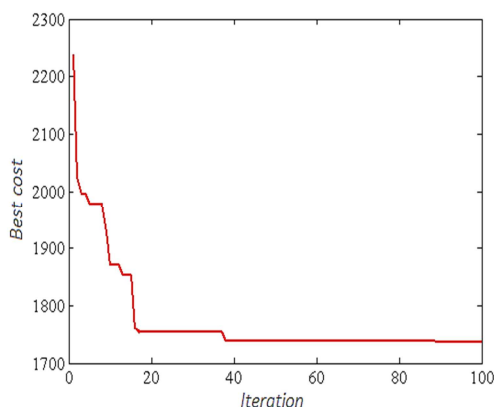


شکل ۱۰- پاسخ کنترلگرهای گام به عقب و مد لغزشی بهینه سازی شده

کنترلگر مد لغزشی توانایی لازم برای کنترل این سیستم را ندارد. به همین جهت از کنترلگر گام به عقب جهت آموزش شبکه عصبی بهره گرفته شده است. بهره های کنترلی کنترلگر گام به عقب پس از بهینه سازی برای شبیه سازی به صورت جدول ۴ تنظیم شده اند. با استفاده از

همان طور که از پاسخ هر دو کنترلگر مشاهده می شود پاسخ سیستم با استفاده از کنترلگر مد لغزشی بهتر می باشد ولی به دلیل تأخیر زمانی عملگرها در سیستم چهارروتور و اندازه ورودی های کنترلی هر دو کنترلگر غیرخطی نشان داده شده در شکل ۱۱ و شکل ۱۲،

کاهش زیادی داشته است که این کاهش بیانگر بهتر شدن رفتار کنترلر هست. در شکل ۱۵ مقایسه رفتار خروجی‌های حالت سیستم به ازای کنترلر شبکه عصبی آموزش داده شده با کنترلر گام به عقب و شبکه عصبی طراحی شده بر اساس الگوریتم هوشمند ازدحام افزایشی ذرات آورده شده‌اند. همان‌گونه که مشاهده می‌شود کنترلر شبکه عصبی طراحی شده بر اساس الگوریتم بهینه‌سازی عملکرد بهتری نسبت به کنترلر شبکه عصبی ساده از خود نشان داده است.



شکل ۱۳- نمودار بهترین هزینه کنترلی به ازای هر سیکل بهینه‌سازی

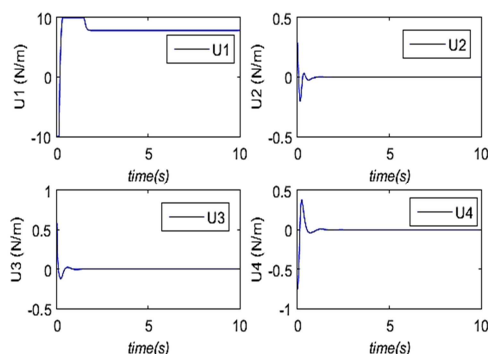
۱۰- نتیجه‌گیری

کنترل سیستم چهارروتور به دلیل غیرخطی بودن و نامعینی‌های زیادی که در آن وجود دارد، کار بسیار پیچیده‌ای می‌باشد. طراحی کنترلرهای همچون کنترلر بازخورد و رودی، کنترلر گام به عقب، کنترلر مد لغزشی و... به دلیل وابستگی به مدل، کار بسیار مشکلی می‌باشد. در این مقاله یک سیستم یادگیری لایه ای برای کنترل پهناد چهارروتور ارائه شد که بدون وابستگی به مدل و بر اساس ورودی-خروجی‌های سیستم در لایه هایش عمل کنترلی را یاد می‌گیرد. در سومین لایه سیستم، کنترلر جدید هوشمند شبکه عصبی با کمک بهینه‌سازی ازدحام افزایشی ذرات معرفی شد که توانایی آن با شبیه‌سازی و مقایسه با رفتار کنترلر گام عقب به‌خوبی مشخص شده است. برای بهینه‌سازی و تنظیم وزن‌های این کنترلر شبکه عصبی از الگوریتم هوشمند ازدحام افزایشی ذرات استفاده شد و بهبود رفتار کنترلر شبکه عصبی نسبت به کنترلر عصبی آموزش داده شده با کنترلر گام به عقب مورد توجه قرار گرفت. با کمک این سیستم یادگیرنده قادر هستیم تا رفتار کنترلرهای تجاری موفق را به یک سیستم کنترلی آموزش دهیم و از قابلیت‌های پیچیده آنها استفاده کنیم.

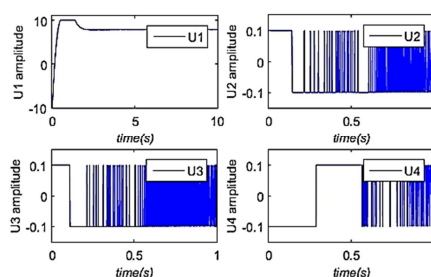
شبیه‌سازی کنترلر گام به عقب بهینه‌سازی داده‌های آموزشی برای طراحی کنترلر شبکه عصبی استفاده شده است.

جدول ۴- بهره‌های کنترلر گام به عقب بهینه‌سازی شده

پارامتر	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
مقدار	0.01	7	0.01	10	10	10	10	0.01

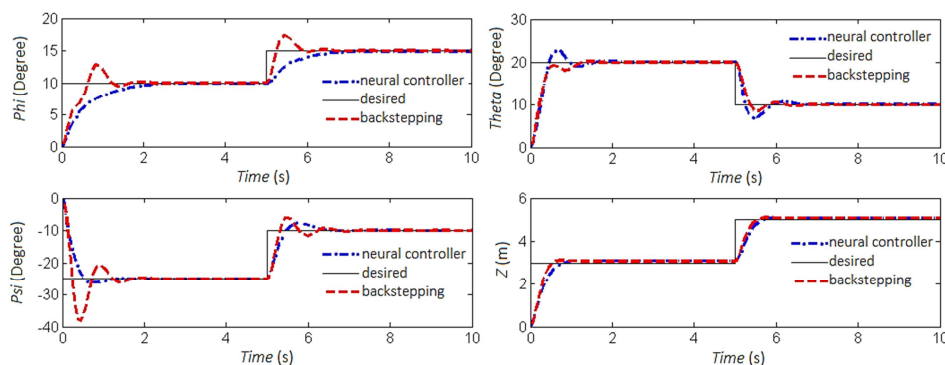


شکل ۱۱- نمودار اندازه خروجی‌های کنترلر گام به عقب

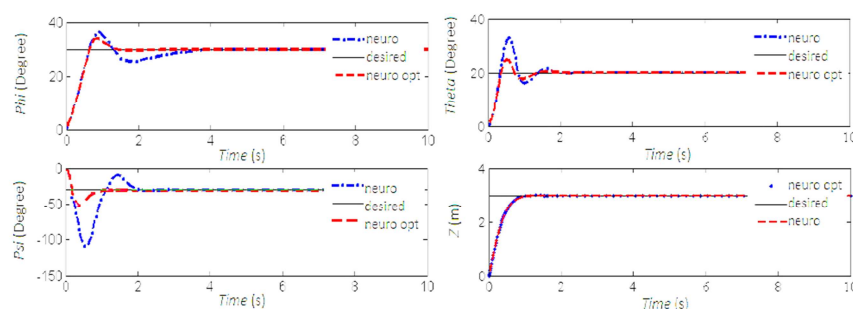


شکل ۱۲- نمودار اندازه خروجی‌های کنترلر مد لغزشی

همان‌طور که از پاسخ زمانی دو کنترلر برای کنترل حالت‌های زاویه‌ای در شکل ۱۴ مشخص می‌باشد با بررسی رفتار کنترلرهای گام عقب و کنترلر شبکه عصبی می‌توان دریافت که کنترلر عصبی طراحی شده قابلیت کنترل مناسب سیستم را داشته است. این نتایج و تفاوت رفتاری دو کنترلر نشان می‌دهد که کنترلر شبکه عصبی طراحی شده یک شبکه تصمیم‌گیر و یادگیرنده است و کنترلر مقلد نیست. حال به شبیه‌سازی کنترلر شبکه عصبی طراحی شده بر اساس استفاده از الگوریتم هوشمند ازدحام افزایشی ذرات پرداخته می‌شود. این نتایج در شکل ۱۵ آورده شده‌است. همان‌طور که در شکل ۱۳ مشاهده می‌شود مقدار تابع هزینه تعریف شده برای سیستم به ازای تعداد سیکل‌ها روند کاهشی داشته و این مقدار نسبت به حالت ابتدایی



شکل ۱۴- خروجی‌های حالت چهارروتور در دو کنترلگر عصبی و کنترلگر گام به عقب



شکل ۱۵- نمودار خروجی‌های حالت چهارروتور در دو کنترلگر شبکه عصبی و شبکه عصبی تنظیم‌شده با الگوریتم ازدحام افزایشی ذرات

۱۱-مراجع

- [11] Farrell J., M. Sharma, Polycarpu M, Backstepping-Based Flight Control with Adaptive Function Approximation, Journal of Guidance, Control, and Dynamics, vol. 28, pp. 1089-1102, 2005.
- [12] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemat, S. M. Smailzadeh, Attitude Control of a Quadrotor with Optimized PID Controller, vol. 4, ed: published by Intelligent Control and Automation, 2013.
- [13] M. Zareb, R. Ayad, W. Nouibat, Fuzzy-PID hybrid control system to navigate an autonomous mini-Quadrotor, in Systems and Control (ICSC), 2013 3rd International Conference on, pp. 906-913, 2013.
- [14] A. A. Mian, W. Daobo, Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor, Chinese Journal of Aeronautics, vol. 21, pp. 261-268, 2008.
- [15] J. Dunfield, M. Tarbouchi, G. Labonte, Neural network based control of a four rotor helicopter, in Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on, vol. 3, pp. 1543-1548, 2004.
- [16] B. Y. Lee, H. I. Lee, M. J. Tahk, Analysis of adaptive control using on-line neural networks for a quadrotor UAV, in Control, Automation and Systems (ICCAS), 2013 13th International Conference, pp. 1840-1844, 2013.
- [17] M. T. Frye, R. S. Provence, Direct Inverse Control using an Artificial Neural Network for the Autonomous Hover of a Helicopter, in Systems, Man and Cybernetics (SMC), IEEE International Conference, pp. 4121-4122, 2014.
- [18] C. L. Hwang, Hybrid neural network under-actuated sliding-mode control for trajectory tracking of quad-rotor unmanned aerial vehicle, in Neural Networks (IJCNN), The 2012 International Joint Conference on, pp. 1-8, 2012.
- [19] S. Rezazadeh, M. A. Ardestani, P. S. Sadeghi, Optimal attitude control of a quadrotor UAV using Adaptive Neuro-Fuzzy Inference System (ANFIS), in Control, Instrumentation, and Automation (ICCIA), 2013 3rd International Conference on, pp. 219-223, 2013.
- [20] M. Lower, W. Tarnawski, Quadrotor Navigation Using the PID and Neural Network Controller, in Theory and Engineering of Complex Systems and Dependability: Proceedings of the Tenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Poland, Springer International Publishing, pp. 265-274, 2015.
- [1] Bouabdallah S., Siegwart R., Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor, in Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference, pp. 2247-2252, 2005.
- [2] Adigbli p., Nonlinear Attitude and Position Control of a MicroQuadrotor using Sliding Mode and Backstepping Techniques, in 3rd US-European Competition and Workshop on Micro Air Vehicle Systems & European Micro Air Vehicle and Conference and Flight Competition (EMAV2007), Toulouse, France, 2007.
- [3] Bouabdallah S., Noth A., Siegwart R., PID vs LQ control techniques applied to an indoor micro quadrotor, in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol.3, pp. 2451-2456, 2004.
- [4] Castillo P., R. Lozano, Dzul A., Stabilization of a mini-rotorcraft having four rotors, in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. IEEE/RSJ International Conference on, vol.3, pp. 2693-2698, 2004.
- [5] Castillo P., A. Dzul A., R. Lozano, Real-time stabilization and tracking of a four-rotor mini rotorcraft, IEEE Transactions on Control Systems Technology, vol. 12, pp. 510-516, 2004.
- [6] Wang B., Wang F., Chen B. M., Lee T. H., Robust flight control system design for an indoor miniature coaxial helicopter, in Intelligent Control and Automation (WCICA), 10th World Congress on, pp. 2918-2924, 2012.
- [7] Y. Ciann-Dong, L. Wen-Hsiung, Nonlinear H_∞ & infinity decoupling hover control of helicopter with parameter uncertainties, in American Control Conference, 2003. Proceedings of the 2003, vol.4, pp. 3454-3459, 2003.
- [8] R. Enns, J. Si, Helicopter flight control design using a learning control approach, in Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, vol.2, pp. 1754-1759, 2000.
- [9] Calise A. J., Kim B. S., Leitner J., J. Prasad J. V. R., Helicopter adaptive flight control using neural networks, in Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, vol.4, pp. 3336-3341, 1994.
- [10] S. K. Kannan, E. N. Johnson, Adaptive trajectory based control for autonomous helicopters, in Digital Avionics Systems Conference, 2002. Proceedings. The 21st, vol.2, pp. 8D1-1-8D1-12, 2002.