



طراحی و پیاده‌سازی عملی کنترلرهای هوشمند عصبی و PD ترکیبی با بهره‌گیری از شبکه عصبی برای کوادروتور

روح‌اله نوروزی¹، حامد شهبازی^{2*}، کمال جمشیدی³

1- دانشجوی کارشناسی ارشد، مهندسی مکترونیک، دانشگاه اصفهان، اصفهان

2- استادیار، مهندسی مکترونیک، دانشگاه اصفهان، اصفهان

3- دانشیار، مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان

* اصفهان، صندوق پستی 84591-33553، shahbazi@eng.ui.ac.ir

اطلاعات مقاله

مقاله پژوهشی کامل

دریافت: 01 اردیبهشت 1396

پذیرش: 05 شهریور 1396

ارائه در سایت: 28 مهر 1396

کلید واژگان:

کوادروتور

کنترلر کلاسیک PD

کنترلر PD- عصبی تنظیم‌گر

کنترلر شبکه عصبی

سیستم شبیه‌ساز کوادروتور

چکیده

کوادروتور یکی از انواع ربات‌های پرنده می‌باشد که به دلیل ساختار ساده و قابلیت عمود پروازی مورد توجه بسیاری از محققان قرار گرفته است. پایداری این سیستم به دلیل دینامیک شدیداً غیرخطی از اهمیت ویژه‌ای برخوردار است. برای پایداری این سیستم در این مقاله کنترلرهای ترکیبی، کنترلر کلاسیک PD و شبکه‌های عصبی طراحی و شبیه‌سازی شده‌اند. در این مقاله ابتدا به طراحی یک شبکه عصبی با توانایی آموزش برخط پرداخته شده است. این شبکه بدون نیاز به داده آموزشی خود را با سیستم تطبیق داده و ضرایب شبکه را تنظیم می‌کند. سپس یک کنترلر کلاسیک PD با استفاده از روش زیگلر نیکولز بیان شده و سپس با استفاده از این کنترلرها، شبیه‌سازی در محیط شبیه‌سازی متلب انجام گرفت و تنظیم پارامترهای کنترلی این کنترلر پرداخته می‌شود. برای تعیین کارایی این کنترلرها، شبیه‌سازی در محیط شبیه‌سازی متلب انجام گرفت و نتایج به شکل عملی آورده شده است. برای پیاده‌سازی این کنترلرها روی سیستم واقعی، یک مدل شبیه‌ساز برای کوادروتور ارائه شده که با پیاده‌سازی کنترلرهای طراحی شده روی سیستم کارایی آن‌ها بر روی سیستم واقعی مشخص گشته است.

Design and Implementation of a Neural Intelligent Controller and a Hybrid PD Controller for Quadrotor

Rouhollah Norouzi¹, Hamed Shahbazi^{2*}, Kamal Jamshidi²

1- Department of Mechatronic Engineering, University of Isfahan, Isfahan, Iran

2- Department of Computer Engineering, University of Isfahan, Isfahan, Iran

* P.O.B. 84591-33553, Isfahan, Iran, shahbazi@eng.ui.ac.ir

ARTICLE INFORMATION

Original Research Paper
Received 21 March 2017
Accepted 27 August 2017
Available Online 20 October 2017

Keywords:

Quadrotor
neural controller
PD control
Ziegler Nichols
Practical Implementation

ABSTRACT

Quadrotor is a type of flying robot which can fly vertically and has a simple structure. This robot is one of the best models of flying robot that is considered by many researchers recently. Because of nonlinear dynamics of the system, Stability of the control process has an important role in this robot. In this paper we design hybrid attitude controllers for quadrotor to track the desired input and robustness efficiency in front of disturbances. First, a Neural Network controller is designed to track the desired state and stabilize the quadrotor. The presented neural network controller is used to stabilize the attitude of the quadrotor that it does not need to training data for learning and adapt its weights and biases. Then we designed a PD controller using Ziegler Nichols method, and then an online learner neural controller is trained for tuning the parameters of this PD controller. Finally, to verify these controllers, first a simulation performed in the Simulink environment of the Matlab. In addition to simulation we have practically implemented these control methods on a quadrotor test bench. Practical implementation results demonstrate the effectiveness of the presented method.

1- مقدمه

تحقیقات طراحی کنترلرهای جدید و بررسی و مقایسه کارایی آن‌ها با روش‌های دیگر کنترلی و بررسی کارایی آن‌ها می‌باشد. امروزه کنترلرهایی که بتواند اهدافی همچون کم کردن هزینه کنترلی را دنبال کند مورد توجه بسیاری از محققان قرار گرفته است. در دهه‌های اخیر ربات‌های پرنده بدلیل توانایی نشست و برخاست عمودی، مانورپذیری بالا، ساختار ساده، هزینه تعمیر و نگهداری پایین، انجام ماموریت‌های سخت و خطرناک مورد توجه اهداف صنعتی و اکادمیک قرار گرفته است. تنها جنبه‌ای که کوادروتور را

امروزه کنترل سیستم‌ها یکی از نیازات ضروری در جهت اتوماسیون و خودکارسازی سیستم‌ها به شمار می‌رود. این کنترلرها براساس مدل‌های خطی و غیرخطی سیستم و برخی از آن‌ها فارغ از مدل سیستم طراحی می‌شوند. در این کنترلرها هدف پایداری سیستم، مقاوم نمودن سیستم در مقابل اغتشاشات خارجی، تعقیب مسیر و همچنین در نظر گرفتن عدم قطعیت‌ها و رفع وابستگی کنترلر به پارامترها می‌باشد. یکی از اهداف

Please cite this article using:

R. Norouzi, H. Shahbazi, K. Jamshidi, Design and Implementation of a Neural Intelligent Controller and a Hybrid PD Controller for Quadrotor, *Modares Mechanical Engineering*, Vol. 17, No. 10, pp. 194-204, 2017 (in Persian)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

است. در [13] یک کنترلر PID بهینه‌سازی شده با الگوریتم ازدحام ذرات را پیشنهاد داده‌اند و با مقایسه نتایج کنترلر گام به عقب و کنترلر PID بهینه‌سازی شده، کارایی کنترلر PID بهینه‌سازی شده مشخص شده است. در [14] برای یک ربات دو درجه آزادی دو کنترلر فازی-PID و PID با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات بهینه‌سازی شده و نشان داده شده است که کارایی کنترلر فازی-PID بهینه‌سازی شده بهتر و مناسب‌تر می‌باشد. در [15] یک کنترلر هوشمند براساس شبکه عصبی برای کنترلر نشست و برخاست ربات عمود پرواز طراحی و ارائه شده است. در [16] با استفاده از روش‌های فیدبک خطی و شبکه عصبی تطبیقی، کنترلر برای کنترل وضعیت کوادروتور ارائه شده است و با شبیه‌سازی این کنترلر با استفاده از پارامترهای کوادروتور 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 کنترلر ترکیبی اسلایدینگ-مود-شبکه عصبی ارائه شده است. در این مقاله برای مقابله با عدم قطعیت‌ها و بهبود کارایی کنترلر اسلایدینگ مود یک شبکه عصبی به این کنترلر اضافه شده است. در [19] یک کنترلر عصبی-فازی با بهینه‌سازی پارامترهای کنترلی توسط الگوریتم‌های بهینه‌سازی ژنتیک ارائه و کارایی این کنترلر با مقایسه با رفتار کنترلر PID کلاسیک مشخص شده است. در [20] یک کنترلر شبکه عصبی ارائه شده است که در آن این کنترلر با استفاده از کنترلر PID آموزش داده شده است.

هدف این مقاله طراحی الگوریتم‌های کنترلی ترکیبی عصبی و کلاسیک PID برای کنترل وضعیت کوادروتور و پیاده‌سازی این الگوریتم‌ها بر روی مدل شبیه‌ساز کوادروتور برای تعیین ارزش هریک از این الگوریتم‌های کنترلی و مقایسه کارایی این کنترلرها و مشخص نمودن توانایی شبکه عصبی در تنظیم کنترل کننده کلاسیک می‌باشد. نتایج حاصل از کارایی هر کنترلر با استفاده از نرم افزار متلب شبیه‌سازی شده و روی نمودار آورده شده‌اند. برای پیاده‌سازی این کنترلرها بر روی مدل واقعی، در این مقاله دو مدل شبیه‌ساز یک درجه آزادی و سه درجه آزادی برای بررسی کارایی کنترلرها ارائه شده است. کارایی کنترلر براساس معیارهایی همچون زمان نشست، بیشینه فراجهش، و میزان نوسان پاسخ سیستم مشخص می‌شود. ارزش کنترلرهای شبکه عصبی طراحی شده با شبیه‌سازی و مقایسه با کنترلر PID کلاسیک مشخص شده است.

نواوری مقاله به‌طور مشخص در یادگیری و بازسازی عملکرد یک کنترل کننده کلاسیک نا مشخص با کمک یک کنترل کننده عصبی است که با کمک طراحی عملی یک بستر تست چهارپره برای پیاده‌سازی شبکه‌های عصبی مصنوعی انجام شده است.

هدف اصلی در این جا مهندسی معکوس کنترل کننده‌های تجاری آماده و موجود در بازار به کمک روش یادگیری ماشین و شبکه‌های عصبی مصنوعی است تا بتوان این کنترل کننده‌ها را در تعداد زیاد تولید کرد.

2- مدل‌سازی دینامیکی کوادروتور

حرکت کوادروتور با استفاده از تغییر سرعت چرخش روتورها انجام می‌گیرد. پیکربندی کوادروتور در "شکل 1" مشاهده می‌شود، این ساختار متشکل از دو بازوی عمود برهم می‌باشد و هرکدام از این بازوها دارای موتورهایی در انتهای خود می‌باشند. موتور شماره 1 و 3 در راستای ساعت‌گرد و موتور شماره 2 و 4 در راستای پادساعت‌گرد می‌چرخند.

نسبت به بقیه ماشین‌های پرنده ضعیف می‌کند، مداومت پروازی کم این ربات پرنده می‌باشد. کوادروتورها به دلیل نیاز به مصرف نیروی الکتریکی زیاد برای به حرکت درآوردن چهار روتور و محدود بودن منبع تغذیه الکتریکی، مدت زمان پروازی کوتاهی دارند. از مهمترین مزیت‌های کوادروتور می‌توان به انجام ماموریت‌های سخت و پیچیده بدون نیاز به انسان، نقشه‌برداری هوایی، عکس‌برداری هوایی، امداد و کنترل ترافیک اشاره کرد. سیستم کواد یک سیستم کاملا ناپایدار می‌باشد و کنترل وضعیت نقش بسیار مهمی در کنترل این پرنده ایفا می‌کند. این کنترلر به پایداری سیستم، جلوگیری از سقوط پرنده، حفظ تعادل سیستم، مقابله با اغتشاشات خارجی و همچنین کنترل وضعیت پرنده در طول پرواز می‌پردازد. در کنترل وضعیت به تنظیم سرعت موتورها برای رسیدن به زوایای مطلوب (رول، پیچ و یاوو) و ارتفاع مطلوب پرداخته می‌شود. پایداری کوادروتور برای انجام یک مانور موفق بسیار حائز اهمیت می‌باشد. مانور پروازی و همچنین پایداری سیستم با کنترل سرعت موتورهای ربات، امکان‌پذیر می‌باشد. این عمل می‌تواند با استفاده از یک الگوریتم که توانایی محاسبه سرعت موردنیاز هر روتور را دارد انجام شود. با توجه به کارهای گوناگونی که بر روی کنترل این سیستم انجام شده سعی بر آن شده تا تعدادی از این مقالات بررسی شوند.

در [1] یک کنترلر فازی براساس کنترلر کلاسیک PID برای کنترل وضعیت کوادروتور طراحی کرده‌اند که در آن ضرایب کنترلی توسط کنترلر فازی تغییر می‌کند و پایداری بهتری به سیستم در طول پرواز می‌دهد. این کار کنترلر کلاسیک PID را به یک کنترلر پویا تبدیل کرده است. نتایج این تحقیق سرعت خوب پایدارکنندگی این کنترلر را نشان می‌دهد. در [2] نیز کنترلر فازی-PID طراحی کرده‌اند که نتایج حاصل نشان‌دهنده کارایی مناسب این کنترلر می‌باشد [2]. در [3] یک کنترلر اسلایدینگ مود و مشاهده‌گر غیرخطی طراحی شده است. در [4] به بررسی کنترل وضعیت پرداخته‌اند. در این تحقیق به مقایسه‌های روش‌های کنترلی PID، بک استپینگ، مد لغزشی و کنترلر معکوس برای کنترل وضعیت پرداختند و نتایج خود را با شبیه‌سازی کامپیوتری انجام دادند. در این تحقیق نشان داده شد که کنترلر اسلایدینگ مود کارایی بهتری نسبت به دیگر کنترلرها دارد. در مقاله [5] کارایی کنترلرهای PID کلاسیک با کنترلر بهینه‌ساز خطی روی یک سیستم کوادروتور برای کنترل ارتفاع سیستم بررسی شده است. در این مقاله یک مدل دینامیکی دقیق و مناسب برای سیستم آمده و توصیف‌هایی در مورد عملگرها و حسگرهای سیستم آورده شده است. در مقالات [6,7] مدل دینامیکی براساس مدل لاگرانژ به‌دست آمده و رفتار کنترلر گام به عقب برای کنترل وضعیت زوایا و کنترل موقعیت کوادروتور بررسی شده است. در این مقالات کنترلر پیشنهاد شده بر روی سیستم واقعی نیز پیاده‌سازی شده است. یکی از روش‌هایی که می‌تواند نامعینی‌های غیرخطی سیستم را تشخیص دهد شبکه عصبی وفق یافته با روش‌های کنترلی تطبیقی می‌باشد. در مقاله [8,9] کنترلرهای تطبیقی برای تعیین عدم قطعیت‌ها ارائه شده است. در [10] دینامیک معکوس با استفاده از شبکه عصبی برای محاسبه نامعینی‌های دینامیکی و خطاهای دینامیک معکوس برای مدل بالگرد استفاده شده است. در [11] یک کنترلر جدید با نام کنترلر پوسودو هدینگ¹ با استفاده از شبکه عصبی یک‌لایه، برای محاسبه نامعینی‌های سیستم به‌کار گرفته شده است. در [12] یک کنترلر ترکیبی گام به عقب با شبکه عصبی برای مواجه شدن با خطاهای مدل و دینامیک مدل نشده سیستم پیاده شده

¹ pseudo-control hedging

$$U_1 = \sum_{i=1}^4 T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (3-الف)$$

$$U_2 = (-T_2 + T_4) = bl(-\Omega_2^2 + \Omega_4^2) \quad (3-ب)$$

$$U_3 = (T_1 + T_3) = bl(\Omega_1^2 - \Omega_3^2) \quad (3-پ)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (3-ت)$$

که در این معادلات $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ سرعت‌های زاویه‌ای روتورها می‌باشند و b ضریب نیروی بالابرنده ملخ و d ضریب نیروی درگ می‌باشد. طبق معادلات سیستم و ارتباط بین آن‌ها مشاهده می‌شود که حرکت چرخشی کوادروتور به حرکت انتقالی آن بستگی ندارد اما حرکت انتقالی این ربات به حرکت چرخشی آن شدیداً وابسته است؛ بنابراین سیستم کلی کوادروتور می‌تواند به دو بخش حلقه وضعیت و حلقه موقعیت تقسیم شود. "شکل 2" ساختار این وضعیت را مشخص می‌کند.

که در "شکل 2"، مقادیر مطلوب می‌باشند و $(x_d, \dot{x}_d, y_d, \dot{y}_d, z_d, \dot{z}_d)$ مقادیر مطلوب زاویه‌ای می‌باشند که توسط کنترلر موقعیت تعیین می‌شوند. ورودی U_1 که ورودی کنترلی ارتفاع را مشخص می‌کند توسط کنترلر موقعیت و سه ورودی کنترلی U_2, U_3, U_4 توسط کنترلر وضعیت محاسبه می‌شوند.

3- مدل‌سازی سیستم در محیط کتیا

برای به دست آوردن اندازه پارامترهای دینامیکی سیستم، برای مدل‌سازی سیستم در محیط سیمولینک این پارامترها با استفاده از محیط شبیه‌سازی کتیا، محاسبه شده و در جدول 1 نشان داده شده‌اند. پیاده‌سازی سیستم در محیط کتیا نیز در "شکل 3" نشان داده شده است.

4- طراحی کنترلرهای وضعیت برای کوادروتور

در این بخش برای پایدارسازی کوادروتور به طراحی کنترلرهای هوشمند شبکه عصبی، کنترلر کلاسیک تناسبی- مشتق‌گیر و طراحی کنترلر PD با

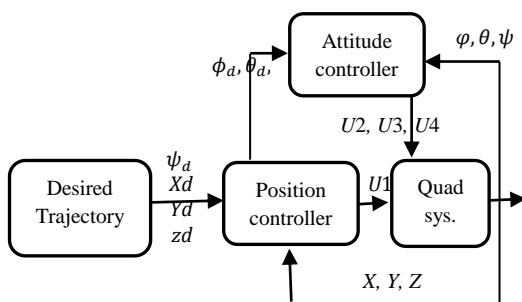


Fig. 2 Diagram of quad rotor position controller

شکل 2 دیاگرام کنترلی موقعیت کوادروتور

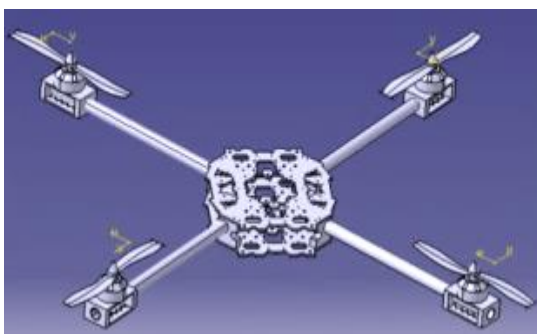


Fig. 3 Modeling of quad rotor in Catia

شکل 3 مدل‌سازی سیستم در محیط کتیا

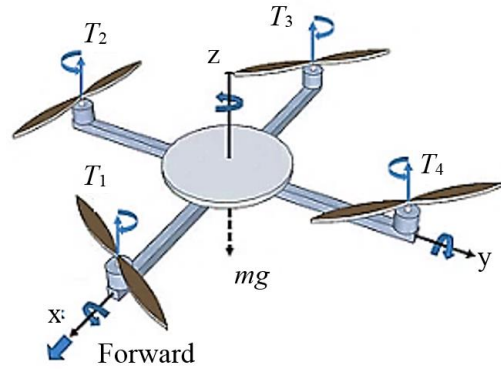


Fig. 1 Structure of quad rotor

شکل 1 پیکربندی پرنده کوادروتور

کنترل حرکت کوادروتور 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 کوادروتور می‌شود. با کم و زیاد کردن سرعت همه‌ی موتورها به یک اندازه می‌تواند حرکت عمودی را برای سیستم تولید کرد. تغییر در سرعت هر یک از موتورهای 2 و 4 باعث حرکت چرخشی حول محور x و تغییر در سرعت هر یک از موتورهای 1 و 3 باعث حرکت چرخشی حول محور y می‌شود. حرکت چرخشی حول محور z ناشی از اختلاف گشتاور تولیدی بین هر جفت روتور می‌باشد. مدل ریاضی کوادروتور با استفاده از معادلات (1) که برگرفته از روش نیوتن-اویلر می‌باشد، از [1] گرفته شده است. در این معادلات، معادلات (1-الف) تا (1-پ)، سیستم را در فضای کارترین توصیف می‌کند. سه معادله (1-ت) تا (1-ه) نیز دینامیک سیستم را براساس زوایای اویلر معرفی می‌کنند. این معادلات رفتار ساده کوادروتور با صرف نظر کردن از نیروهای آئرو دینامیک وارد بر سیستم و تأثیرات ژيروسکوپی تولیدشده از چرخش روتورها را مدل می‌کند.

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

$$\ddot{z} = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{U_1}{m} \quad (1-ب)$$

$$\ddot{z} = -g + (\cos(\phi) \cos(\theta)) \frac{U_1}{m} \quad (1-پ)$$

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

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

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

که در آن x, y, z موقعیت مرکز جرم در سازه و ϕ, θ, ψ زوایای اویلر هستند. همچنین m, I_x, I_y, I_z جرم و ممان اینرسی کوادروتور هستند و l نیز فاصله بین روتورها و مرکز جرم است و Ω_d, J_r به ترتیب ممان اینرسی و سرعت زاویه‌ای 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 پروازی می‌باشند و مجموع سرعت‌های زاویه‌ای تیغه‌های پروازی می‌باشد که توسط رابطه (2) تعریف می‌شود.

$$\Omega_d = \Omega_1 + \Omega_2 + \Omega_3 + \Omega_4 \quad (2)$$

رابطه ورودی‌های کنترلی با سرعت روتورها به صورت (3) بیان می‌شوند.

جدول 1 پارامترهای سیستم برای شبیه‌سازی

Table 1 System parameters for simulation

واحد	مقدار	پارامترهای سیستم
kg	0.65	m
kgm^2	7.5×10^{-3}	I_x
kgm^2	7.5×10^{-3}	I_y
kgm^2	1.3×10^{-2}	I_z
kgm^2	6.5×10^{-6}	J_r
m	0.23	l
Ns^2	3.13×10^{-6}	b
Nms^2	0.75×10^{-6}	d

استفاده از شبکه عصبی برای تنظیم بهره‌های کنترلی پرداخته می‌شود و کارایی و رفتار آن‌ها مورد بررسی قرار می‌گیرند.

1-4- کنترلر شبکه عصبی

شبکه‌های عصبی با قابلیت استنتاج معانی از داده‌های پیچیده و مبهم، می‌تواند برای استخراج الگوها و تشخیص یک‌روند که برای انسان یا روش‌های دیگر کامپیوتری بسیار پیچیده است، به‌کاربرده شود. یک شبکه عصبی آموزش‌دیده می‌تواند به‌عنوان یک متخصص در رده اطلاعاتی که برای آنالیز به آن داده شده است به‌کار گرفته شود. این شبکه‌ها برای تخمین و تقریب کارایی بسیار بالایی از خود نشان داده‌اند. از این رو به‌دلیل دشوار بودن تنظیم ضرایب کنترلی به روش‌های زیگلر نیکولز و سعی و خطا، در این مقاله از شبکه‌های عصبی به‌عنوان تنظیم‌کننده و تطبیق‌دهنده کنترلر کلاسیک PID استفاده شده است. در این‌جا از شبکه‌های عصبی رایج با پرسپترون‌های چندلایه برای طراحی کنترلر استفاده شده است. سپس با استفاده از شبکه‌های عصبی به طراحی یک کنترلر تطبیقی PID-عصبی پرداخته می‌شود که توانایی تنظیم ضرایب کنترلی کنترلر کلاسیک را بدون نیاز به داده‌های آموزشی دارد. به‌دلیل این‌که کنترلرهای کلاسیک جواب‌گوی دینامیک‌های ناشناخته سیستم تحت کنترل نیست و شبکه عصبی با توانایی اصلاح‌پذیری تا حد زیادی این توانایی را دارد از کنترلر شبکه عصبی به‌عنوان اصلاح‌کننده کنترلر کلاسیک استفاده شده است. برای شبکه عصبی استفاده شده در این مقاله برای هر جزء کنترلی، دو ورودی خطای پاسخ و مشتق خطای پاسخ و یک خروجی کنترلی برای سیستم در نظر گرفته شده است. در این کنترلر سه لایه پنهان در نظر گرفته شده است. در لایه اول 10 نورون، در لایه دوم 6 نورون و در لایه آخر یک نورون به‌کار گرفته شده است. این ساختار با آزمایش ساختارهای متفاوت شبکه و کارایی آن‌ها انتخاب شده است. این شبکه در "شکل 4" آورده شده است.

برای تنظیم پارامترهای شبکه عصبی با روش‌های معمول، نیاز به تهیه داده‌های آموزشی توسط یک متخصص می‌باشد تا بتوان با استفاده از این

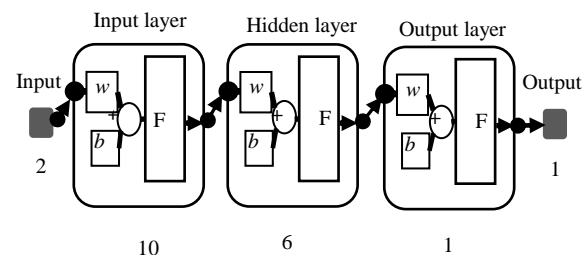


Fig. 4 Structure of neural network in this paper

شکل 4 ساختار شبکه عصبی استفاده‌شده در این مقاله

داده‌ها وزن‌های شبکه را مشخص نمود. در اغلب موارد تهیه این داده‌ها منجر به هزینه‌های اضافی و صرف زمان زیادی می‌باشد که در این مقاله فرض بر آن شده که داده‌های آموزشی برای آموزش شبکه وجود ندارد و برای مقداردهی وزن‌ها از الگوریتم هوشمند ازدحام ذرات استفاده شده است. بدین صورت که طبق "شکل 5" وزن‌دهی ابتدا به‌صورت تصادفی مقداردهی می‌شود. سپس با اجرای این شبکه و محاسبه خطای بین خروجی مطلوب تعریف شده (که در این مقاله صفر شدن خطای حالت به‌عنوان خروجی مطلوب در نظر گرفته شده است) و خروجی شبکه، خطای حاصل از پاسخ سیستم و ورودی مطلوب به الگوریتم ازدحام ذرات وارد شده و به تنظیم وزن‌های مناسب‌تر شبکه عصبی می‌پردازد. با استفاده از این روش در حین انجام کنترل سیستم شبکه عصبی آموزش داده می‌شود و شبکه عصبی خبره‌تری خواهیم داشت. این الگوریتم در فضای پاسخ برای تنظیم وزن‌های نرون به‌دنبال جواب بهینه می‌گردد تا خطای معرفی شده را کمینه کند و وزن‌های شبکه را به قسمی معین کند که پاسخ سیستم در بهینه‌ترین حالت خود باشد. این روند در "شکل 5" آورده شده است. خطای تعریف شده بین خروجی شبکه عصبی و خروجی هدف برای تنظیم وزن‌های شبکه برای الگوریتم ازدحام ذرات به صورت معادله (4) آورده شده است:

$$E(\vec{w}) = \frac{1}{2} \sum_{t=0}^{\text{time}} (\text{target}_t - \text{Output}_t)^2 \quad (4)$$

که در آن مقصود از Output_t خروجی‌های مجموعه واحدهای لایه خروجی و target_t مقدار هدف موردنظر و خروجی متناظر واحد خروجی در زمان t می‌باشد.

در هر سیکل از این آموزش، وزن‌دهی توسط الگوریتم ازدحام ذرات انجام می‌شود. در روش آموزشی در این بخش، الگوریتم ازدحام ذرات در ابتدا با انتخاب یک جمعیت تصادفی به‌عنوان وزن‌های شبکه، مقدار تابع هزینه را برای پاسخ سیستم محاسبه می‌کند و متناسب با کارایی آن، وزن‌های شبکه را تغییر می‌دهد تا کارایی شبکه بهبود پیدا کند و مقدار تابع هزینه را کمینه کند. در این‌جا روند این روش آموزشی، روش خود اصلاح‌گر نامیده شده است. در این روش وزن‌های شبکه به‌صورت بخش‌های جداگانه‌ای مقداردهی می‌شود که با تغییر هرکدام از آن‌ها کارایی شبکه تغییر می‌کند و مقدار اثرگذاری هر وزن در کارایی شبکه مشخص می‌شود که با به‌کارگیری الگوریتم هوشمند ازدحام ذرات این تغییرات به سمت بهینه کردن کارایی شبکه عصبی جهت‌گیری می‌کند که در نهایت با بهینه شدن کارایی سیستم ضرایب وزن‌های شبکه تنظیم می‌شوند. این روند در "شکل 6" آورده شده است.

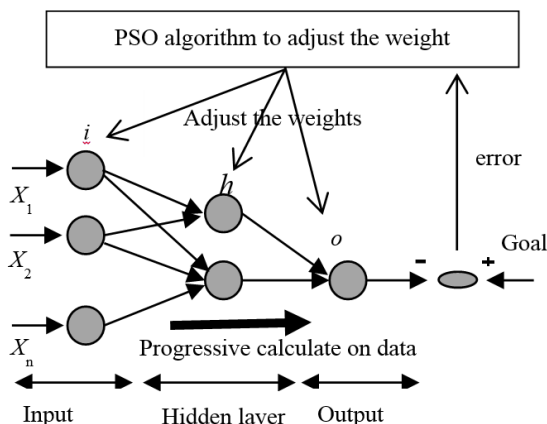


Fig. 5 over view of learning algorithm

شکل 5 دیدگاه کلی الگوریتم آموزش شبکه عصبی به روش ازدحام ذرات

می‌باشد. از روش‌های به دست آوردن این ضرایب می‌توان به روش زیگلر و نیکولز و روش سعی و خطا اشاره کرد. از روش‌های دیگر کنترلی همچون کنترلرهای منطق فازی، کنترلرهای شبکه عصبی و الگوریتم‌های هوشمند می‌توان برای بهینه کردن ضرایب کنترلی تناسبی- مشتق‌گیر استفاده کرد. پاسخ مطلوب سیستم باید با کمترین زمان نشست، کمترین بیشینه فرا جهش و کمترین نوسان همراه باشد. در این مقاله از روش زیگلر و نیکولز برای تعیین ضرایب کنترلر کلاسیک تناسبی- مشتق‌گیر استفاده شده است. در این روش در ابتدا با صفر در نظر گرفتن ضرایب کنترلی مشتق‌گیر، ضرایب کنترلی تناسبی (K_{cr}) به قسمی افزایش داده شد که پاسخ سیستم به صورت نوسانی باشد. سپس با اندازه‌گیری زمان یک سیکل از پاسخ سیستم (T_{cr}) و با استفاده از روابط (5) ضرایب کنترلی استخراج شده‌اند. در "شکل 7" شمای کلی از این کنترلر نشان داده شده است.

$$G_c(s) = 0.6K_{cr}(1 + 0.125T_{cr}s) \quad (5)$$

در طراحی کنترلر به روش تناسبی- مشتق‌گیر ورودی‌های کنترلی از رابطه (6) پیروی می‌کنند.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (6)$$

که در آن ضرایب K_p, K_d ضرایب کنترلی طراحی شده برای کنترلر تناسبی- مشتق‌گیر می‌باشند و $e(t)$ مقدار خطای اندازه‌گیری شده از حسگرها و مقدار مورد انتظار در لحظه t می‌باشد و $de(t)$ نرخ خطای مقدار اندازه‌گیری شده از حسگرها و مقدار مورد انتظار می‌باشد. ورودی کنترلی برای کنترل زاویه تنا¹ به صورت (7) تعریف می‌شود.

$$U_3 = K_{pth}(\theta_d - \theta) + K_{dth}(\dot{\theta}_d - \dot{\theta}) \quad (7)$$

ورودی‌های کنترلی برای کنترل زاویه فی² و سای³ و کنترل ارتفاع به صورت معادلات (8) تا (10) تعریف می‌شود.

$$U_2 = K_{pphi}(\varphi_d - \varphi) + K_{dphi}(\dot{\varphi}_d - \dot{\varphi}) \quad (8)$$

$$U_4 = K_{ppsi}(\psi_d - \psi) + K_{dpsi}(\dot{\psi}_d - \dot{\psi}) \quad (9)$$

$$U_1 = \frac{m(g + k_{pz}(z_d - z) + k_{dz}(\dot{z}_d - \dot{z}))}{\cos \theta \cos \varphi} \quad (10)$$

ضرایب تنظیم شده به روش زیگلر نیکولز در جدول 2 آورده شده‌اند.

3-4- طراحی کنترل کننده خود تنظیم PD مبتنی بر شبکه عصبی

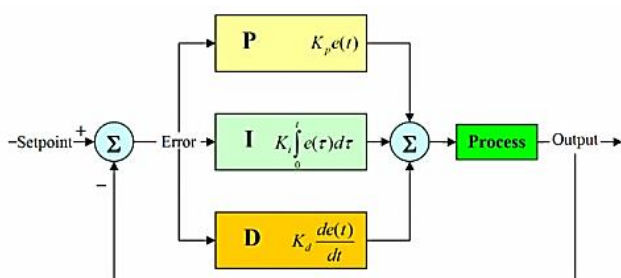


Fig. 7 PID algorithm

شکل 7 دیاگرام کلی الگوریتم کنترلی PID

جدول 2 ضرایب تنظیم شده به روش زیگلر نیکولز

Table 2 Kp and Kd reached with zigler method for simulation

K_d	K_p	ضرایب کنترلی
1.4	2.2	مقدار

¹ Theta

² Phi

³ Psi

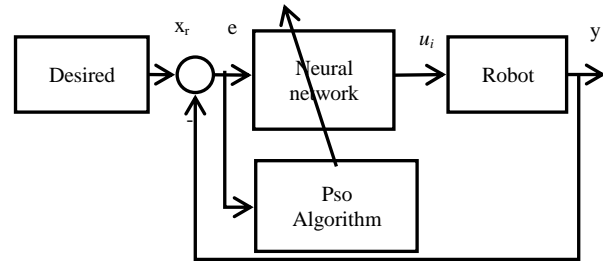


Fig. 6 training algorithm

شکل 6 روند آموزش شبکه به صورت برخط

شبکه عصبی استفاده شده در این پژوهش یک شبکه عصبی چندلایه است، که اتصال میان واحدهای تشکیل دهنده آن، یک چرخه را تشکیل نمی‌دهند. در واقع این شبکه متفاوت از شبکه‌های عصبی بازگشتی می‌باشد. در این شبکه اطلاعات تنها از یک مسیر رو به جلو حرکت می‌کنند تا خروجی مناسب را تولید کنند. در واقع اطلاعات و ورودی‌ها با شروع از گره (نورون)‌های ورودی و گذر از لایه‌های پنهان به سمت گره‌های خروجی می‌روند. همان‌طور که گفته شد در این شبکه حلقه یا دوری وجود ندارد و اتصالات مختلف سبب رفتارهای متفاوت شبکه می‌شود. بطور کلی می‌توان گفت شبکه‌های جلوسو، شبکه‌های استاتیک هستند، به این معنی که از ورودی داده شده تنها یک‌دسته مقدار خروجی تولید می‌کنند. شبکه‌های جلوسو بی‌حافظه هستند و پاسخ آن‌ها به یک ورودی، مستقل از وضعیت قبلی شبکه است. از طرف دیگر شبکه‌های بازگشتی، سیستم‌های دینامیک هستند و زمانی که یک دنباله‌ی ورودی جدید به آن‌ها داده شود، خروجی نورون‌ها محاسبه می‌شود و در اثر وجود مسیرهای فیدبک، ورودی هر نورون اصلاح می‌شود و این باعث می‌شود که شبکه وارد یک وضعیت جدید شود. با توجه به مطالب گفته شده، شبکه عصبی چندلایه معرفی شده به دلیل عدم وجود سیگنال بازگشتی در شبکه، باعث انباشت انرژی در خود نمی‌شود و نسبت به هر ورودی یک خروجی ارائه می‌دهد، از این رو پایداری آن تضمین شده می‌باشد.

2-4- طراحی کنترلر کلاسیک

کنترلر PID یکی از رایج‌ترین کنترلرهایی می‌باشد که در بسیاری از کارهای صنعتی و دانشگاهی برای سیستم‌های خطی و برخی سیستم‌های غیرخطی مورد استفاده قرار گرفته است. با وجود پیشرفت بسیاری که در طراحی کنترلرهای خطی، غیرخطی و هوشمند به وجود آمده، این کنترلر به دلیل سادگی ساختار و قابل درک بودن نحوه کارکرد این کنترلر توانسته جایگاه خود را حفظ کند و همیشه مورد استفاده بوده است. کنترلر PD از ترکیب سه بخش تناسبی- مشتق‌گیر-انترگالی به وجود آمده است. وظیفه کنترلر تناسبی کم کردن زمان نشست سیستم می‌باشد. این کنترلر براساس خطای سیستم ورودی کنترلی موردنیاز را تولید می‌کند. برای کاهش خطای ماندگار سیستم می‌توان کنترلر انترگالی را به کنترلر تناسبی اضافه کرد. با اضافه شدن این کنترلر به سیستم پاسخ سیستم نوسانی می‌شود برای کاهش نوسانات پاسخ می‌توان یک کنترلر مشتق‌گیر به سیستم اضافه کرد. اضافه کردن این بخش کنترلی به سیستم باعث می‌شود تا کنترلر رفتار سیستم را پیش‌بینی کند و قبل از این که نرخ خطا افزایش پیدا کند ورودی کنترلی تصحیح شود. در این مقاله از کنترلر کلاسیک تناسبی- مشتق‌گیر استفاده شده است. در کنترلرهای تناسبی- مشتق‌گیر برای گرفتن جواب بهینه و مطلوب از سیستم، باید ضرایب کنترلی در آن تنظیم شوند. ضرایب موردنیاز برای کنترلر تناسبی- مشتق‌گیر شامل ضرایب تناسبی (K_p) و مشتق‌گیر

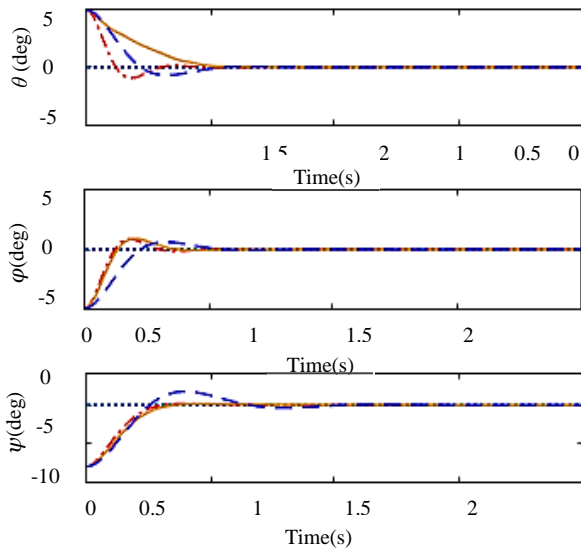


Fig. 9 System response per time (PD controller (- - -), PD-Neuro tuning (- . - .), Neuro-controller (—), Set point(.....))

شکل 9 پاسخ سیستم در واحد زمان (کنترلر PD (- - -)، کنترلر Neuro- (—)، کنترلر PD-Neuro (- . - .))، ورودی مطلوب (.....) (کنترلر کلاسیک تنظیم شده با شبکه عصبی) (- - -)، کنترلر شبکه عصبی (—)، ورودی مطلوب (.....)

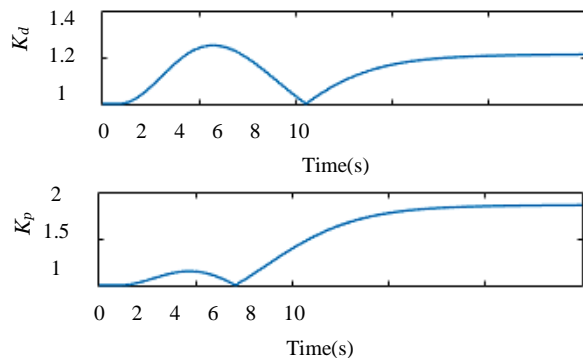


Fig. 10 Gain value for PD-Neuro algorithm (θ)

شکل 10 بهره‌های کنترلی کنترلر تا

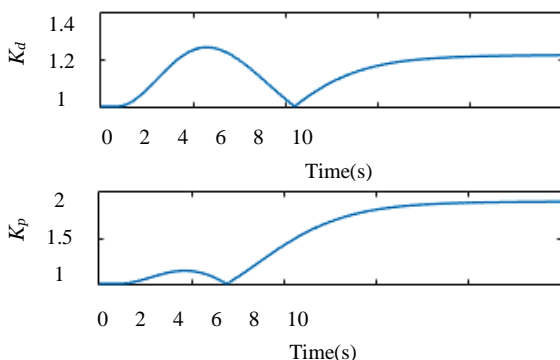


Fig. 11 Gain value for PD-Neuro algorithm (ϕ)

شکل 11 بهره‌های کنترلی کنترلر فی

جدول 3 و جدول 4 و جدول 5 مشخصات پاسخ به ورودی برای زاویه‌های تا و فی و سای شامل زمان نشست (t_s)، میزان فراجهش (M_p (%))، پاسخ ماندگار (e_{ss})، و زمان رسیدن به پاسخ مطلوب (t_r) را برای کنترلرهای طراحی شده مشخص می‌کند.

امروزه کنترل کننده‌های PD به دلیل سادگی کاربرد به طور گسترده‌ای در صنعت به کار برده می‌شود. این کنترل کننده اغلب توسط افراد خبیره و با روش آزمون و خطا و چند قاعده عملی تنظیم می‌شوند. زمانی که فرایند کند است و یا عملکرد آن غیرخطی و سیستم شامل تاخیر زمانی است، عمل تنظیم ضرایب کنترلی PD بسیار دشوار و زمان‌بر خواهد بود. به ویژه اگر سیستم پس از تنظیم ضرایب، در اثر عوامل غیرخطی سیستم یا عوامل دیگر در معرض اغتشاشات نیز قرار گرفته باشد، انگاه عملکرد سیستم تحت تاثیر قرار می‌گیرد که نیاز به تنظیم مجدد پارامترهای کنترلی می‌باشد. مسئله تنظیم کنترل کننده PD به نحوه انتخاب پارامترهای کنترل کننده به منظور دستیابی به پاسخ مناسب حلقه بسته مربوط می‌شود. پیچیدگی عمل تنظیم ضرایب کنترلی با بالا رفتن تعداد پارامترهایی که باید تنظیم شوند، افزایش می‌یابد. از روش‌های تنظیم میتوان به روش زیگلر نیکولز و روش سعی و خطا اشاره کرد. در این‌جا از روش هوشمند شبکه عصبی و برای بهبود عملکرد سیستم و رسیدن به خروجی مطلوب و برای تنظیم ضرایب کنترلی PD طراحی شده است. به منظور تنظیم پارامترهای کنترلر کلاسیک طبق "شکل 8" یک شبکه عصبی قرار گرفته و از خطای سیستم، ورودی مطلوب و خروجی سیستم به عنوان ورودی شبکه، برای تنظیم پارامترهای شبکه استفاده می‌کند. این شبکه عصبی به تعداد ضرایب کنترل کننده PD خروجی تولید می‌کند. برای آموزش این شبکه و با توجه به نبود داده‌های آموزشی مناسب برای آموزش شبکه عصبی از الگوریتم ازدحام ذرات و به شکل برخط استفاده شده است.

5- نتایج شبیه‌سازی

در این بخش کنترلرهای طراحی شده (کنترلرهای کلاسیک تناسبی- مشتق‌گیر، کنترلر کلاسیک تناسبی- مشتق‌گیر با بهره‌گیری از شبکه‌های عصبی برای تنظیم بهره‌های کنترلی و کنترلر هوشمند عصبی آموزش داده شده به روش برخط در نرم‌افزار متلب شبیه سازی شده‌اند تا عملکرد این کنترلرها با همدیگر مقایسه و ارزیابی شود. "شکل 9" نتایج مقایسه شبیه‌سازی برای کنترل وضعیت سیستم را نشان می‌دهد. پاسخ کنترلرها به ازای ورودی مطلوب صفر و برای پایدار کردن سیستم رسم شده‌اند. همان گونه که مشخص است پاسخ کنترلر هوشمند عصبی نسبت به PD تنظیم شده با شبکه عصبی و PD کلاسیک پاسخ مطلوب‌تری دارد. روند تنظیم ضرایب کنترلی کنترلر PID توسط شبکه عصبی در "شکل- های 10 تا 12" آورده شده است.

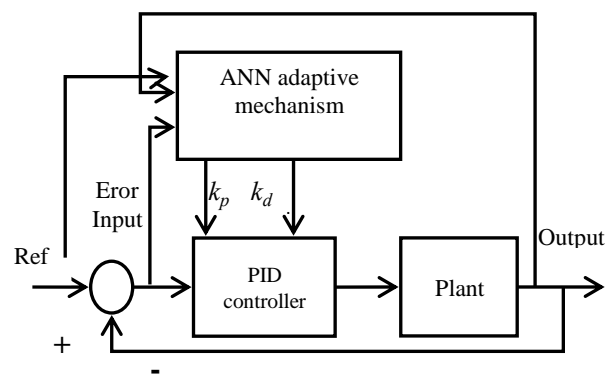


Fig. 8 PD algorithm tuned with Neural Network (PD-Neuro)

شکل 8 دیاگرام کلی الگوریتم تنظیم بهره‌های کنترلی PD با شبکه عصبی

6- پیاده‌سازی روی سیستم شبیه‌ساز

برای تعیین میزان کارایی کنترلرهای طراحی شده، نیاز به بستری می‌باشد تا بتوان این کنترلرها را بر روی سیستم واقعی پیاده‌سازی کرد و نتایج حاصل را بررسی نمود. در این بخش به منظور پیاده‌سازی و بررسی نتایج کنترلرهای طراحی شده، مدلی برای کوادروتور ساخته و فراهم شده تا بر این اساس بتوان کنترلرهای وضعیت را روی سیستم واقعی مدل‌سازی و نتایج حاصل و توانمندی و ناتوانی این کنترلرها را تعیین کرد. با تفکیک کنترلرهای ربات پرنده کوادروتور به کنترل وضعیت و کنترل موقعیت و به دلیل اهمیت ویژه‌ای که کنترل وضعیت در این سیستم دارد و وظیفه اصلی کنترل و پایداری این سیستم برعهده کنترل وضعیت می‌باشد، سیستم شبیه‌ساز به صورت یک مکانیسم سه درجه آزادی و یک درجه آزادی برای پیاده‌سازی کنترلرهای وضعیت در نظر گرفته شده است. این ساختار با قابلیت ارتباط با محیط نرم‌افزاری متلب تهیه شده است و قابلیت اجرای هر گونه کنترلر طراحی شده برای این سیستم در این محیط شبیه‌سازی را دارا می‌باشد.

برای پیاده‌سازی کنترلرهای وضعیت کوادروتور به یک ساختار با حداکثر سه و حداقل یک درجه آزادی نیاز می‌باشد. برای این منظور دو پایه طبق "شکل 13" (برای آزادی بخشی به سیستم در سه درجه آزادی و قابلیت انجام حرکات چرخشی در راستای تتا و فی و سای) و طبق "شکل 14" (برای آزادی بخشیدن به سیستم در یک درجه آزادی و قابلیت انجام حرکت چرخشی در راستای زاویه تتا) ارائه شده‌اند.

6-1- مدل موتور-ملخ

کوادروتور یک سیستم شش درجه آزادی با ساختار صلیب شکل می‌باشد که از چهار روتور مجزا به‌عنوان پیشران استفاده می‌کند. در این مقاله روتورها از نوع براسلس ایر 350¹ می‌باشند که ولتاژ و جریان موردنیاز آن‌ها از طریق یک

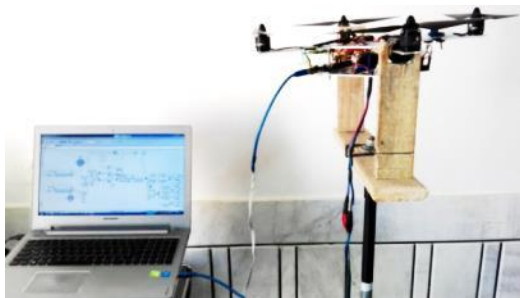


Fig. 13 Simulation model for quadrotor with one degree of freedom
شکل 13 ساختار مدل شبیه‌ساز یک درجه آزادی و ارتباط با محیط متلب



Fig. 14 Simulation model for quadrotor with three degree of freedom
شکل 14 ساختار مدل شبیه‌ساز سه درجه آزادی

¹ AirGear350

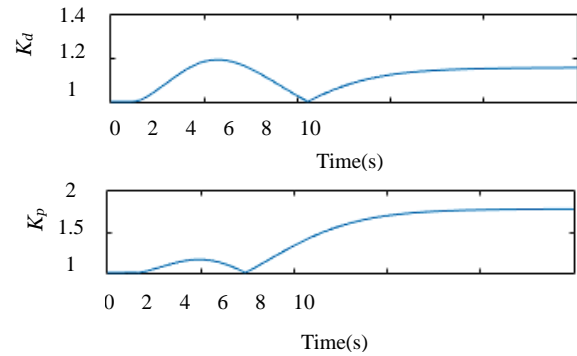


Fig. 12 Gain value for PD-Neuro algorithm (ψ)

شکل 12 بهره‌های کنترلی کنترلر فی

جدول 3 مقایسه پاسخ کنترلر تتا در کنترلرهای طراحی شده

Table 3 Comparison of θ controlling

کنترلر	t_r	e_{ss}	$M_p(\%)$	t_s
PD	0.5	0	0.93	0.12
Neuro-PD	0.56	0	0.69	0.22
Neuro	0.52	0	0	0.52

جدول 4 مقایسه پاسخ کنترلر فی در کنترلرهای طراحی شده

Table 4 Comparison of ϕ controlling

کنترلر	t_r	e_{ss}	$M_p(\%)$	t_s
PD	0.38	0	0.91	0.13
Neuro-PD	0.56	0	0.63	0.22
Neuro	0.38	0	0.9	0.14

جدول 5 مقایسه پاسخ کنترلر سای در کنترلرهای طراحی شده

Table 5 Comparison of ψ controlling

کنترلر	t_r	e_{ss}	$M_p(\%)$	t_s
PD	0.36	0	0.93	0.35
Neuro-PD	0.63	0	1.76	0.25
Neuro	0.34	0	0	0.34

با بررسی پاسخ سیستم و کارایی آن مشخص می‌شود که کنترلر هوشمند شبکه عصبی بهتر از کنترلر PD-عصبی تنظیم شده با شبکه عصبی و رفتار این کنترلر بهتر از کنترلر کلاسیک PD می‌باشد.

در جدول 6 یک مقایسه کلی بین کنترلرهای طراحی شده در این مقاله آورده شده است. با مقایسه رفتار این کنترلرها مشاهده می‌شود که کنترلر عصبی آموزش داده شده براساس الگوریتم تکاملی ازدحام ذرات توانسته به خوبی رفتار کنترلی را یاد بگیرد و پاسخ خوبی را از خود نشان داده است. همچنین این نتایج نشان می‌دهد که کنترلر هوشمند PID که با استفاده از شبکه عصبی تنظیم شده نیز پاسخ بهتری نسبت به کنترلر کلاسیک PID از خود نشان داده است. بنابراین می‌توان نتیجه گرفت که شبکه‌های عصبی می‌توانند به خوبی در کار کنترل سیستم‌ها به کار گرفته شوند و رفتار سیستم را بهبود دهند.

جدول 6 مقایسه کلی پاسخ کنترلرهای طراحی شده

Table 6 Comparison of controllers

کنترلر	t_r	e_{ss}	$M_p(\%)$	t_s
PD	0.5	2	0	0.5
Neuro-PD	0.5	0.6	0.2	0.75
Neuro	0.85	0	0.53	1.8

برای بررسی عملکرد کنترلرهای طراحی شده (کنترلرهای PD، کنترلر شبکه عصبی و کنترلر PD تنظیم شده با شبکه عصبی) در کنترل زاویه تنا، این کنترلرها روی مدل آزمایشگاهی یک درجه آزادی پیاده‌سازی شده است و نتایج پیاده‌سازی آورده شده است. برای محاسبه موقعیت زاویه‌ای سیستم در این مقاله از حسگر GY80 استفاده شده است. خروجی‌های این حسگر دارای نویز و اغتشاش می‌باشند که برای استفاده از این داده‌ها در مدل طراحی شده از یک فیلتر پایین‌گذر برای فیلتر کردن نویزهای سیستم استفاده شده است. در نتایج ارائه شده، پاسخ سیستم بدون استفاده از فیلتر و با استفاده از فیلتر ارائه شده، آورده شده‌اند.

برای تعیین کارایی این کنترلرها و توانایی آن‌ها ورودی‌های مطلوب به صورت پله واحد در نظر گرفته شده‌اند و کارایی کنترلرهای طراحی شده مشخص شده است. همچنین برای بررسی این کنترلرها در پایدار کردن سیستم در مقابل اغتشاشات تصادفی، ورودی مطلوب به صورت زاویه صفر در نظر گرفته شده و چندین اغتشاش تصادفی به سیستم وارد شده تا کارایی این کنترلرها مشخص شود. نتایج پیاده‌سازی در "شکل‌های 18 تا 22" زیر آورده شده است. بخشی از پاسخ سیستم با استفاده از کنترلر کلاسیک PD در "شکل 16" و با استفاده از کنترلر شبکه عصبی طبق "شکل 17" آورده شده‌اند.

برای پیاده‌سازی کنترلر PD تنظیم شده با شبکه عصبی در ابتدا سیستم در حالت مطلوب صفر قرار داده شده است و سپس با وارد کردن خطا به سیستم، شبکه عصبی به تنظیم پارامترهای کنترلر PD پرداخته تا آن‌که بتواند سیستم را به سمت پاسخ مطلوب هدایت کند. همان‌گونه که مشخص می‌باشد سیستم پس از 50 ثانیه، بعد از اعمال خطا توانسته ضرایب صحیح کنترلی را بیابد و سیستم را به سمت پاسخ مطلوب هدایت کند. ضرایب مطلوب برای این سیستم که توسط شبکه عصبی ارائه شده در جدول 7 آورده شده است. با مقایسه ضرایب به‌دست آمده با روش شبکه عصبی و ضرایب روش زیگلر نیکولز مشاهده می‌شود که شبکه عصبی به دقت توانسته بدون



Fig. 16 PD controller response on real system

شکل 16 پاسخ سیستم با استفاده از کنترلر کلاسیک PD



Fig. 17 Neuro controller response on real system

شکل 17 پاسخ سیستم با استفاده از کنترلر شبکه عصبی

منبع تغذیه دائمی با ولتاژ 12 ولت و جریان 60 امپر تامین می‌شود. برای تغییر سرعت این روتورها از سیگنال مدولاسیون پهنای پالس (PWM) استفاده می‌شود. دینامیک کوادروتور و پارامترهای کنترلی آن، تابعی از نیروها و گشتاورهای تولید شده توسط موتور و ملخ تعبیه شده روی سیستم می‌باشند. تولید گشتاور مورد نیاز برای کنترل سیستم با سرعت موتورها رابطه مستقیم دارد. از این رو برای اجرای ورودی‌های کنترلی U_i تولید شده توسط کنترلرها در معادلات بالا بر روی سیستم، رابطه زیر بین ورودی‌های کنترلی و نیروی تراست لازم هر روتور به صورت (11) به‌دست می‌آید.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0 & -\frac{1}{2l} & 0 & \frac{1}{2l} \\ -\frac{1}{2l} & 0 & \frac{1}{2l} & 0 \\ -\frac{b}{4d} & \frac{b}{4d} & -\frac{b}{4d} & \frac{b}{4d} \end{bmatrix}^T \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (11)$$

با توجه به ورودی روتورها که به صورت PWM می‌باشند برای اعمال این تراست‌ها به هر روتور، رابطه بین تراست و سیگنال PWM ورودی موتورها با توجه به ویژگی و اطلاعات موتورهای مورد استفاده به صورت معادله (12) استخراج شده است.

$$T_1=T_2=T_3=T_4=0.25(\text{signal}_{\text{PWM}} - 490) + 165 \quad (12)$$

2-6- الگوریتم کنترلی سیستم شبیه‌ساز

برای پیاده‌سازی کنترلرها و مشاهده‌گرها بر روی سیستم در ابتدا حالت‌های مورد نیاز و معین سیستم توسط برد اردیونو که یک سیستم متن باز و ساده برای استفاده است، از حسگرها خوانده می‌شود و به‌عنوان مقادیر خروجی سیستم و به‌عنوان ورودی نرم‌افزار متلب به داخل این نرم‌افزار فرستاده می‌شود. سپس در این نرم‌افزار با اجرای کنترلرها و مشاهده‌گرهای طراحی شده می‌توان ورودی‌های کنترلی سیستم را به‌عنوان ورودی موتورها تولید کرده و به برد اردیونو برای انجام فرمان‌های کنترلی ارسال کرد.

این سیکل از ابتدا تا انتهای شبیه‌سازی انجام می‌شود تا میزان کارایی سیستم با استفاده از کنترلرهای طراحی شده بر روی مدل واقعی مشخص شود. با حلقه کنترلی ایجاد شده بین کامپیوتر (نرم‌افزار متلب)، حسگرها و عملگرها می‌توان برنامه‌های کنترلی طراحی شده در برنامه متلب را به راحتی بر روی این سیستم پیاده‌سازی کرد. حلقه کنترلی ایجاد شده در "شکل 15" نشان داده شده است.

3-6- نتایج تجربی پیاده‌سازی الگوریتم‌های کنترلی روی سیستم

شبیه‌ساز

برای تعیین کارایی و رفتار کنترلرهای طراحی شده و اهمیت پایداری سیستم کوادروتور، در این بخش به پیاده‌سازی این کنترلرها بر روی سیستم شبیه‌ساز پرداخته شده است.

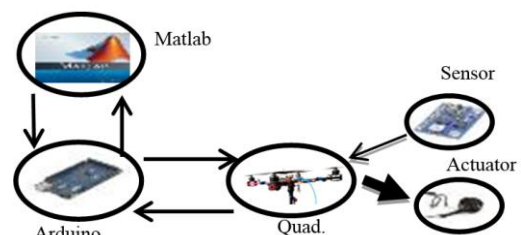


Fig. 15 Simulation loop for quadrotor

شکل 15 حلقه کنترلی شبیه‌ساز برای کوادروتور

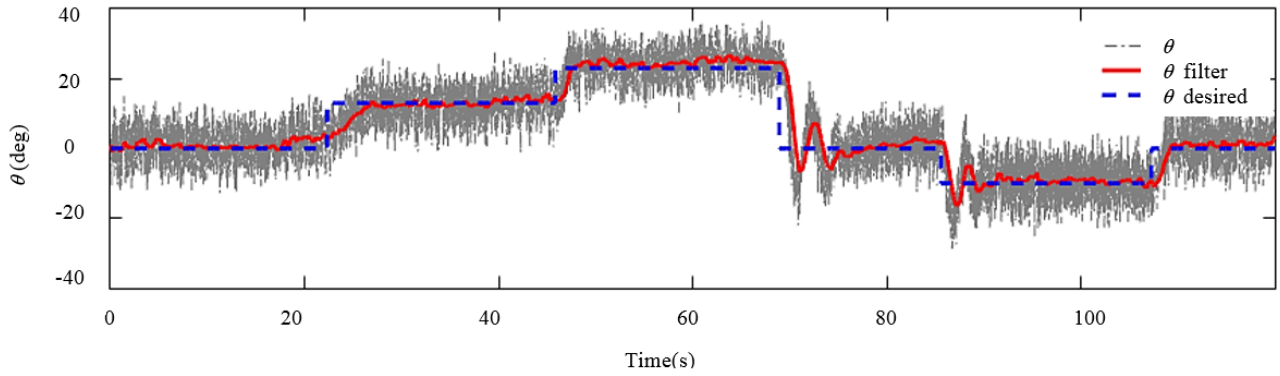


Fig. 18 PD controller response with disturbance per time

شکل 18 پاسخ کنترلر PD به ورودی مطلوب با اعمال اغتشاش در واحد زمان

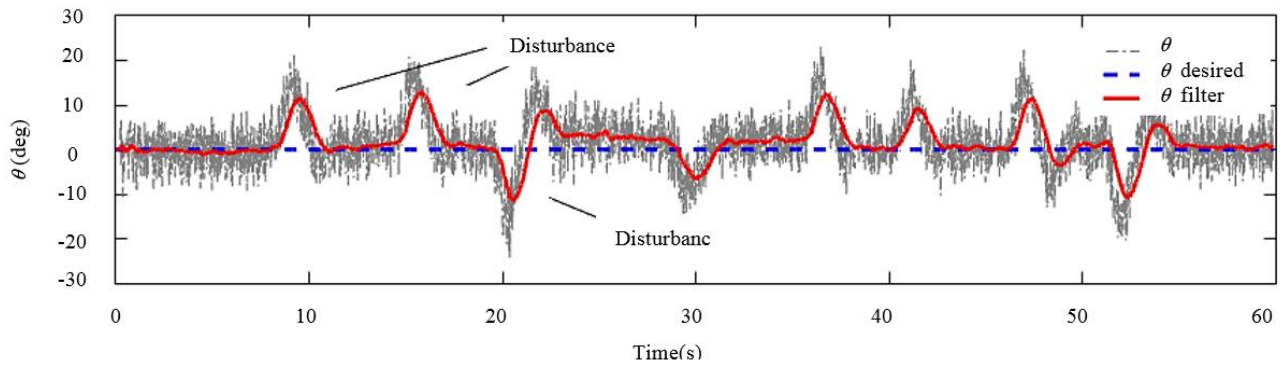


Fig. 19 PD controller response with disturbance per time

شکل 19 پاسخ کنترلر PD به ورودی مطلوب با اعمال اغتشاش در واحد زمان

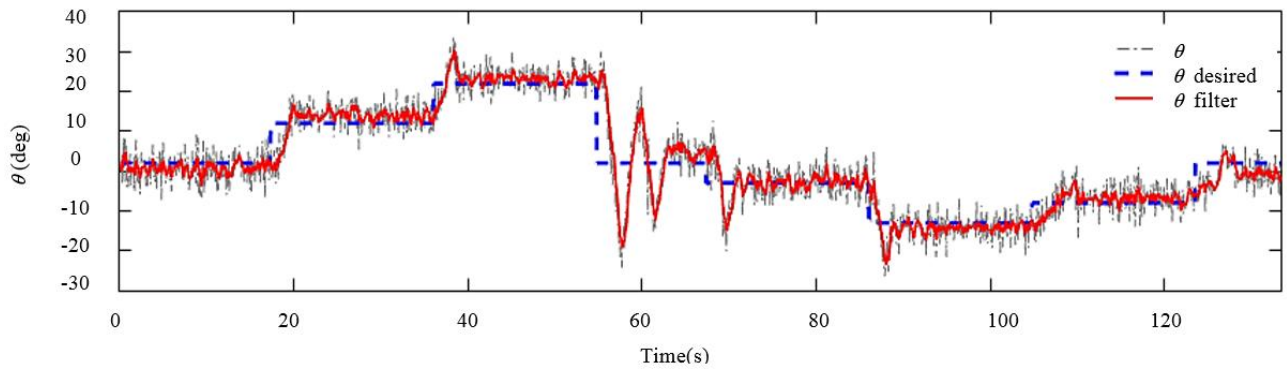


Fig. 20 Neuro controller response with disturbance per time

شکل 20 پاسخ کنترلر شبکه عصبی به ورودی مطلوب با اعمال اغتشاش در واحد زمان

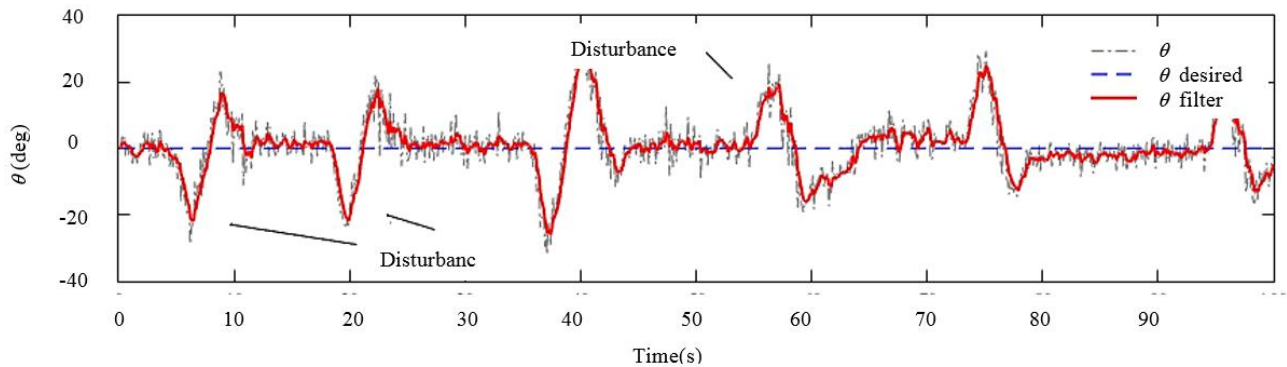


Fig. 21 Neuro controller response with disturbance per time

شکل 21 پاسخ کنترلر شبکه عصبی به ورودی مطلوب با اعمال اغتشاش در واحد زمان

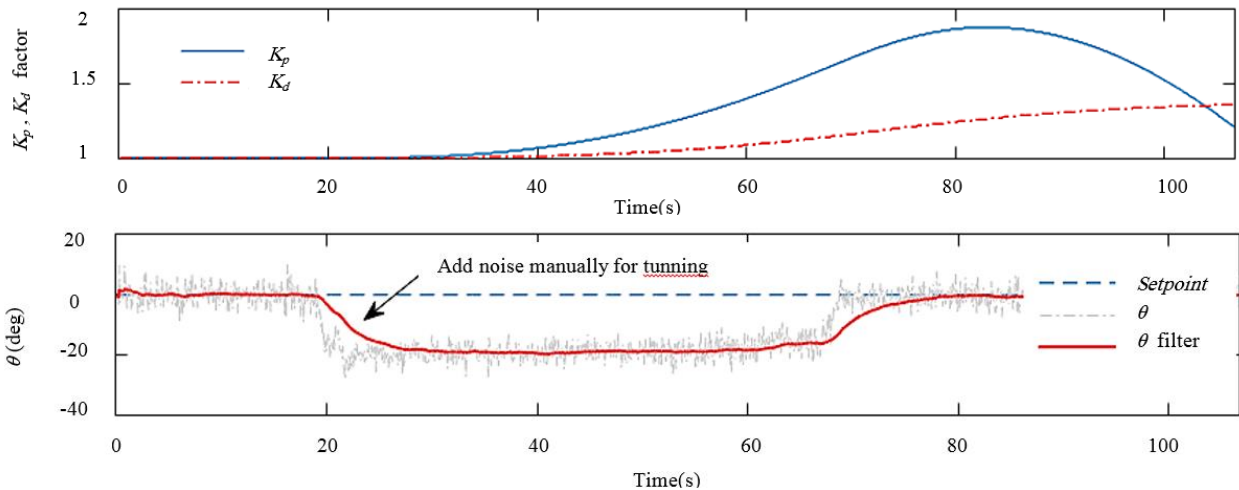


Fig. 22 Neuro-PD controller response with disturbance per time

شکل 22 پاسخ کنترلر PD تنظیم شده با شبکه عصبی به ورودی مطلوب با اعمال اغتشاش در واحد زمان و بهره‌های کنترلر PD تنظیم شده با شبکه عصبی

جدول 8 مقایسه کلی پاسخ کنترلرهای طراحی شده

Table 8 Comparison of different controllers desined in the paper

t_r	e_{ss}	t_s	کنترلر
0.5	2	0.5	PID
0.5	0.6	0.75	PID-Neuro
1.6	0	2	Back stepping
1.6	0	1.6	Back stepping-Opt
0.85	0	1.8	Neural network
1.1	0	1.6	Neuro-Opt

روی مدل واقعی سیستم شبیه‌ساز، پیاده‌سازی شدند و کارایی آن‌ها مشخص شد.

در جدول 8 یک مقایسه کلی بین کنترلرهای طراحی شده در این پژوهش آورده شده است. با مقایسه رفتار این کنترلرها مشاهده می‌شود که کنترلر عصبی آموزش داده شده براساس الگوریتم تکاملی ازدحام ذرات توانسته به خوبی رفتار کنترلر اصلی (کنترلر گام به عقب) را یاد بگیرد و پاسخ خوبی را از خود نشان داده است. همچنین این نتایج نشان می‌دهد که کنترلر هوشمند PID که با استفاده از شبکه عصبی تنظیم شده نیز پاسخ بهتری نسبت به کنترلر کلاسیک PID از خود نشان داده است. بنابراین می‌توان نتیجه گرفت که شبکه‌های عصبی می‌توانند به خوبی در کار کنترل سیستم‌ها به کار گرفته شوند و رفتار سیستم را بهبود دهند.

8- مراجع

- [1] S. F. Ahmed, K. Kushsairy, M. I. A. Bakar, D. Hazry, M. K. Joyo, Attitude stabilization of Quad-rotor (UAV) system using Fuzzy PID controller (an experimental test), 2015 Second International Conference on Proceeding of Computing Technology and Information Management (ICCTIM), pp. 99-104, 2015.
- [2] E. A. Seidabad, S. Vandaki, A. V. Kamyad, Designing fuzzy PID controller for quadrotor, International Journal of Advanced Research in Computer Science & Technology 2014, 2014.
- [3] H. Bouadi, M. Tadjine, Nonlinear observer design and sliding mode control of four rotors helicopter, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, Vol. 1, No. 7, 2007.
- [4] I. C. Dikmen, A. Arisoy, H. Temeltas, Attitude control of a quadrotor, Proceeding of 4th International Conference on Recent Advances in Space Technologies, pp. 722-727, 2009.
- [5] S. Bouabdallah, A. Noth, R. Siegwart, PID vs LQ control techniques applied to an indoor micro quadrotor, IEEE/RSI International Conference on Intelligent Robots and Systems, 2004, (IROS 2004), vol.3, pp. 2451-2456, 2004.

نیاز به هیچ داده‌ای از پیش تعیین شده‌ای، این عمل را انجام دهد و با انتخاب ضرایب صحیح کنترلی، پاسخ سیستم را به سمت ورودی مطلوب سوق دهد. روند تنظیم ضرایب در سیستم شبیه ساز در "شکل 23" آورده شده است.

7- نتیجه گیری

در این مقاله به طراحی و شبیه‌سازی کنترلرهای ترکیبی کلاسیک PD و شبکه‌های عصبی پرداخته شد. در این مقاله کنترلرهای کلاسیک PD، تنظیم شده با شبکه عصبی و کنترلر شبکه عصبی که به صورت بر خط آموزش داده شده است، طراحی، و برای بررسی عملکرد در پایدارسازی سیستم در محیط متلب شبیه‌سازی شدند. همان‌گونه که از پاسخ سیستم بررسی شد پاسخ کنترلر شبکه عصبی بهتر از کنترلر PD تنظیم شده با شبکه عصبی و این کنترلر بهتر از کنترلر کلاسیک PD می‌باشد. در ادامه یک سیستم شبیه‌ساز برای پیاده‌سازی کنترلرها و مشاهده‌گرهای طراحی شده در محیط متلب برای سیستم کوادروتور، طراحی و ساخته شده که قابلیت اجرای انواع کنترلرها بر روی این سیستم را دارد. در نهایت کنترلرهای طراحی شده

جدول 7 ضرایب تنظیم شده به روش شبکه عصبی

Table 7 K_p and K_d factor that reached with Neural Network method for real system

K_d	K_p	ضرایب کنترلی روش محاسبه
1.3	1.8	شبکه عصبی
1.4	2.2	زیگلر و نیکولز



Fig. 23 Neuro-PD controller response on real system

شکل 23 پاسخ سیستم با استفاده از کنترلر PD تنظیم شده با شبکه عصبی

- [15] J. Dunfied, M. Tarbouchi, G. Labonte, Neural network based control of a four rotor helicopter, *Proceeding of IEEE International Conference in Industrial Technology*, 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, *Proceeding of 13th International Conference on Control, Automation and Systems (ICCAS) 2013*, pp. 906-913, 2013.
- [17] M. T. Frye, R. S. Provence, Direct inverse control using an artificial neural network for the autonomous hover of a helicopter, *Proceeding of IEEE International Conference on Systems, Man and Cybernetics (SMC) 2014*, 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, *Proceeding of The 2012 International Joint Conference on Neural Networks (IJCNN)*, 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), *Proceeding of 3rd International Conference on Control, Instrumentation, and Automation (ICCIA) 2013*, pp. 219-223, 2013.
- [20] M. Lower, W. Tarnawski, *Quadrotor Navigation Using the PID and Neural Network Controller*, in: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, J. Kacprzyk, *Theory and Engineering of Complex Systems and Dependability: Proceedings of the Tenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, June 29 – July 3 2015, Brunów, Poland*, Eds., pp. 265-274, Cham: Springer International Publishing, 2015.
- [6] P. Castillo, R. Lozano, A. Dzul, Stabilization of a mini-rotorcraft having four rotors, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol.3, pp. 2693-2698, 2004.
- [7] P. Castillo, A. Dzul, R. Lozano, Real-time stabilization and tracking of a four-rotor mini rotorcraft, *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 4, pp. 510-516, 2004.
- [8] Y. Ciann-Dong, L. Wen-Hsiung, Nonlinear H_∞ decoupling hover control of helicopter with parameter uncertainties, *Proceeding of American Control Conference 2003*, vol.4, pp. 3454-3459, 2003.
- [9] R. Enns, J. Si, Helicopter flight control design using a learning control approach, *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 2, pp. 1754-1759, 2000.
- [10] A. J. Calise, B. S. Kim, J. Leitner, J. V. R. Prasad, Helicopter adaptive flight control using neural networks, *Proceeding of the 33rd IEEE Conference on Decision and Control*, vol.4, pp. 3336-3341, 1994.
- [11] S. K. Kannan, E. N. Johnson, Adaptive trajectory based control for autonomous helicopters, *Proceeding of The 21st Digital Avionics Systems Conference*, Vol. 2, pp. 81-92, 2002.
- [12] J. Farrell, M. Sharma, M. Polycarpou, Backstepping-Based Flight Control with Adaptive Function Approximation, *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 6, pp. 1089-1102, 2005.
- [13] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, S. M. Smailzadeh, Attitude control of a quadrotor with optimized PID controller, published by Intelligent Control and Automation, Vol. 4, 2013.
- [14] M. Zareb, R. Ayad, W. Nouibat, Fuzzy-PID hybrid control system to navigate an autonomous mini-Quadrotor, *Proceeding of 3rd International Conference on Systems and Control (ICSC)*, pp. 906-913, 2013.