

طراحی و پیاده‌سازی عملی مشاهده‌گر به کمک شبکه عصبی پیش‌خور جهت کنترل پهباد چهارپره

روح اله نوروزی^۱

دانشجوی کارشناسی ارشد

حامد شهبازی^۲

استادیار

پایداری چهارروتور به دلیل دینامیک شدیداً غیرخطی از مهم‌ترین مسائل پیش روی در طراحی این ربات بوده است. این مقاله ابتدا به طراحی یک کنترلر کلاسیک تناسبی- مشتق‌گیر به روش تنظیم با الگوریتم تکاملی ازدحام ذرات برای کنترل وضعیت پرنده پرداخته شده است. سپس به طراحی یک مشاهده‌گر غیرخطی برای تخمین حالت‌های سرعت زاویه‌ای چهارروتور پرداخته شده است. در ادامه یک مشاهده‌گر شبکه عصبی با استفاده از داده‌های آموزشی استخراج شده از مشاهده‌گر غیرخطی آموزش داده شده است. با شبیه‌سازی کنترلر و مشاهده‌گر طراحی شده در محیط متلب، کارایی آن‌ها مشخص شده است. برای تعیین کارایی این کنترلر و مشاهده‌گر بر روی سیستم واقعی، یک مدل شبیه‌ساز برای چهارروتور ارائه شده که با پیاده‌سازی کنترلر و مشاهده‌گر طراحی شده روی سیستم کارایی آن‌ها بر روی سیستم واقعی مشخص شده است.

واژه‌های راهنما: چهارپره، کنترلر کلاسیک PD، مشاهده‌گر شبکه عصبی، سیستم شبیه‌ساز چهارپره

۱- مقدمه

کنترلرها از مهم‌ترین اجزا یک سیستم مکترونیک می‌باشند. از وظایف این کنترلرها می‌توان به ردگیری ورودی مطلوب، مقاوم بودن در برابر اغتشاش و نویز خارجی به سیستم و توانایی مقابله با آن می‌باشد. کنترلرها در دو نوع حلقه باز و حلقه بسته می‌باشند. در این مقاله از کنترلر حلقه بسته با پس‌خورد استفاده شده است. در این کنترلرها نیاز به مشخص بودن تمامی حالات موردنیاز برای کنترل سیستم می‌باشد. چهارپره یک وسیله پرنده با شش درجه آزادی حرکت می‌باشد که قابلیت پرواز عمودی و انجام مانورهای پیچیده را دارا می‌باشد. این سازه با ساختاری شبیه صلیب، و چهار ملخ در گوشه‌های آن می‌باشد که با استفاده از تغییر سرعت ملخ‌ها می‌تواند حرکات و مانورهای مختلف را انجام بدهد.

^۱ دانشجوی کارشناسی ارشد مکترونیک، دانشکده فنی و مهندسی، دانشگاه اصفهان، اصفهان

^۲ نویسنده مسئول، استادیار، دانشکده فنی و مهندسی، دانشگاه اصفهان، اصفهان shahbazi@eng.ui.ac.ir

تاریخ دریافت: ۹۶/۰۶/۱۶، تاریخ پذیرش: ۹۷/۰۲/۰۷

مشکلاتی که بر سر راه طراحی چنین پرنده‌هایی وجود دارد مسئله پایداری و کنترل این نوع پرنده‌ها می‌باشد که کار بسیار پیچیده است. از این‌رو در این مقاله از یک کنترلر حلقه بسته برای تنظیم دور چرخش موتورها و ردیابی ورودی موردنظر استفاده شده است.

چهارپره‌ها عامل‌های هوشمندی می‌باشند که محیط برای آن‌ها به‌طور کامل مشاهده‌پذیر نمی‌باشد [۱]. در طراحی سیستم‌های کنترل حلقه بسته با پس‌خورد و با فرض کنترل‌پذیری کامل متغیرهای حالت و انتخاب مناسب بهره فیدبک، قطب‌های حلقه بسته سیستم در مکان‌های موردنظر جایابی می‌شوند. در حالت کلی لازم است که کلیه متغیرهای حالت برای فیدبک در دسترس باشند در بسیاری از کاربردهای واقعی اندازه‌گیری تمامی متغیرهای حالت امکان‌پذیر و یا از نظر اقتصادی مقرون‌به‌صرفه نیست و همچنین به دلیل وجود نویز و اغتشاش، به دست آوردن مقادیر دقیق تمامی متغیرهای حالت امکان‌پذیر نمی‌باشد. در مواردی از این قبیل نمی‌توان به تمام متغیرهای حالت دسترسی پیدا کرد. در این موارد، به‌کارگیری مشاهده‌گر حالت الزامی است. طراحی مشاهده‌گر جهت تخمین بخشی یا همه متغیرهای حالت سیستم و کنترل سیستم به‌کاربرده می‌شود. هدف اصلی این مقاله طراحی مشاهده‌گر مناسب با کمک شبکه عصبی هوشمند می‌باشد که محیط را برای چهارپره مشاهده‌پذیر نماید و متغیرهای حالتی که در دسترس نیستند را تخمین بزند. پرنده هدایت‌پذیر از دور یا پهنای یکی از مهم‌ترین اختراعات سال‌های اخیر محسوب می‌شود که کاربردهای عملی بسیار زیادی پیدا کرده است. از کاربردهای این وسیله می‌توان به بازرسی و کنترل مرزها، مأموریت‌های شناسایی، عکس‌برداری، عملیات جستجو، بازدید و بازرسی از خطوط انتقال نفت و خطوط فشارقوی، کشف آتش‌سوزی‌ها، استفاده در محیط‌های خطرناک و غیرقابل‌دسترس و حمل بار اشاره کرد. در موارد مذکور نیاز به پایداری پرنده در هر لحظه با توجه به شرایط محیطی می‌باشد.

در Lendek و همکاران [۱] مشاهده‌گر تاکاگی-سوگنو برای تخمین سرعت‌های زاویه‌ای ارائه شده است. در این مقاله مشاهده‌گر تاکاگی-سوگنو با مشاهده‌گر خطی مورد مقایسه قرار گرفته‌اند که نتایج نشان‌دهنده رفتار بهتر مشاهده‌گر طراحی شده می‌باشد. در Mokhtari و همکاران [۲] یک مشاهده‌گر لیونبرگر و کنترلر خطی سازی فیدبک برای سیستم چهارپره طراحی شده است که کارایی آن با شبیه‌سازی مشخص شده است. در این مقاله تخمین پارامترهای باد نیز صورت گرفته است. در Bouadi و Tadjine [۳] یک کنترلر مود لغزشی برای تعقیب مسیر و پایداری سیستم و سپس یک مشاهده‌گر غیرخطی برای تخمین حالت‌های غیرقابل اندازه‌گیری و تأثیرات اغتشاشات خارجی از قبیل باد و اغتشاش اراده شده است.

در Dierks و Jagannathan [۴] یک مشاهده‌گر شبکه عصبی برای تخمین حالت‌های سرعت چرخشی و سرعت انتقالی ارائه شده است. در Benzemrane و همکاران [۵] یک مشاهده‌گر سرعت بر اساس اینکه شتاب، زاویه و سرعت‌های زاویه‌ای قابل‌اندازه‌گیری باشد ارائه شده است. در Benallegue و همکاران [۶] یک کنترلر خطی سازی فیدبک با یک مشاهده‌گر مود لغزشی برای سیستم چهارپره طراحی شده است. مشاهده‌گر مود لغزشی به‌عنوان یک تخمینگر اثرات اغتشاشات خارجی از جمله باد و نویز به کار گرفته شده است. مشاهده‌گر حالت می‌تواند در طرح کنترلی مثل Boudjedir و Bouhali [۷] که در آن یک مشاهده‌گر عصبی - مود لغزش که برای کاهش اندازه نویز توسعه داده شده است، به‌کاربرده شود.

در مرجع Khalil [۸] یک مشاهده‌گر برای تخمین سرعت استفاده شده است.

در این مقاله خلاصه‌ای از مشاهده‌گرهای مرتبه بالا آورده شده است و نقش اشباع عملگرها در کنترل بررسی شده است. در مرجع Boudjedir و همکاران [۹] یک مدل مقاوم تطبیقی که در آن نیازی به دانستن مدل دینامیکی مسئله و اندازه حالت‌ها نیست ارائه شده است که در آن متغیرهای حالت با استفاده از مشاهده‌گر تطبیقی تولید می‌شوند. در Ahmed و همکاران [۱۰] یک کنترلر فازی بر اساس کنترلر کلاسیک PID برای کنترل وضعیت چهارپره 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 Seidabad و همکاران [۱۱] نیز کنترلر فازی-PID طراحی کرده‌اند که نتایج حاصل نشان‌دهنده کارایی مناسب این کنترلر می‌باشد. در Dikmen و همکاران [۱۲] به بررسی کنترل وضعیت پرداخته‌اند.

در این تحقیق به مقایسه‌های روش‌های کنترلی PID، گام به عقب، 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 مود لغزش کارایی بهتری نسبت به دیگر کنترلرها دارد. در Farrell و همکاران [۱۳] یک کنترلر ترکیبی گام به عقب با شبکه عصبی برای مواجه شدن با خطاهای مدل و دینامیک مدل نشده سیستم پیاده شده است. در Bolandi و همکاران [۱۴] 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 Zareb و همکاران [۱۵] 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 Dunfied و همکاران [۱۶] یک کنترلر هوشمند بر اساس شبکه عصبی برای کنترلر نشست و برخاست ربات عمود پرواز طراحی و ارائه شده است. در Lee و همکاران [۱۷] با استفاده از روش‌های فیدبک خطی و شبکه عصبی تطبیقی، کنترلی برای کنترل وضعیت چهارپره ارائه شده است و با شبیه‌سازی این کنترلر با استفاده از پارامترهای چهارپره از پیش ساخته شده، کارایی این کنترلر مشخص شده است.

در Frye و Provence [۱۸] یک روش کنترلی بر اساس روش کنترل معکوس، با استفاده از شبکه عصبی آموزش داده شده است. در Rezazadeh و همکاران [۱۹] 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 Lower و Tamawski [۲۰] یک کنترلر شبکه عصبی ارائه شده است که در آن این کنترلر با استفاده از کنترلر PID آموزش داده شده است. در Boudjedir و همکاران [۲۱] یک کنترلر شبکه عصبی با استفاده از مشاهده‌گر عصبی برای متغیرهای حالت طراحی شده است. از این مشاهده‌گر برای رفع حساسیت به اغتشاشات استفاده شده است.

۲- مدل سازی دینامیکی چهارپره

حرکت چهارپره با استفاده از تغییر سرعت چرخش روتورها انجام می‌گیرد. پیکربندی چهارپره در شکل شماره (۱) 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. چهارپره سیستمی تشکیل شده از چهار روتور است. دو روتور (۳ و ۱) که در مقابل یکدیگرند در یک سمت می‌چرخند و دو روتور (۴ و ۲) در جهت خلاف عقربه‌های ساعت می‌چرخند. با تغییر سرعت همه روتورها به یک اندازه، نیروی لیفت روتورها تغییر خواهد کرد که تأثیر این تغییر، کنترل ارتفاع و قابلیت نشست و برخاست عمودی را برای سیستم فراهم می‌کند. زاویه مطلوب یاوو با تغییر سرعت روتورهایی که روبه روی یکدیگر قرار دارند فراهم می‌شود. این حرکت با افزایش (کاهش) سرعت موتورهای (۳ و ۱) در مقایسه با سرعت موتورهای (۴ و ۲) به دست می‌آید. زاویه پیچ که حرکت چرخشی حول محور Y می‌باشد به چهارپره اجازه می‌دهد تا در راستای محور X حرکت کند. زاویه رول که حرکت چرخشی حول محور X می‌باشد به چهارپره اجازه می‌دهد تا در راستای محور Y حرکت کند. روتورها کنترل‌کننده حرکت چهارپره می‌باشند. زوایای اوایلر، نیرو و مومنت‌های موردنیاز سیستم را برای کنترل زوایا و موقعیت سیستم را محاسبه می‌کنند.

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

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

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

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

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

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

که در آن x, y, z موقعیت مرکز جرم در سازه و ϕ, θ, ψ زوایای اوایلر هستند. همچنین m, I_x, I_y, I_z جرم و ممان اینرسی چهارپره هستند و l نیز فاصله بین روتورها و مرکز جرم است و J_r, Ω_d 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 مومنت‌های چرخش حول محور Y ، چرخش حول محور X و چرخش حول محور z تولید شده توسط تیغه‌های پروازی می‌باشند و Ω_d مجموع سرعت‌های زاویه‌ای تیغه‌های پروازی می‌باشد که توسط رابطه (۲) تعریف می‌شود.

$$\Omega_d = \sum_{i=1}^4 (-1)^{i+1} \Omega_i \quad (۲)$$

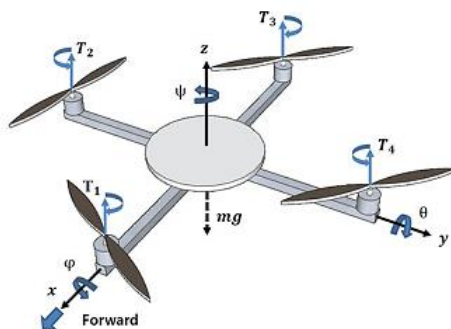
رابطه ورودی‌های کنترلی با سرعت روتورها به صورت (۳) بیان می‌شوند.

$$U_1 = \sum_{i=1}^4 T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (۳-الف)$$

$$U_2 = (-T_2 + T_4) = bl(-\Omega_2^2 + \Omega_4^2) \quad (۳-ب)$$

$$U_3 = (T_1 + T_3) = bl(\Omega_1^2 - \Omega_3^2) \quad (۳-پ)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (۳-ت)$$



شکل ۱- پیکربندی پرنده چهارپره

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. ورودی U_1 که ورودی کنترلی مناسب برای تنظیم ارتفاع را مشخص می‌کند توسط کنترلر موقعیت و سه ورودی کنترلی U_2, U_3, U_4 که توسط کنترلر وضعیت محاسبه می‌شوند، ورودی کنترلی مناسب را برای تنظیم زوایای سیستم مشخص می‌کنند.

۳- طراحی کنترلر وضعیت PD برای چهارپره

پایدارسازی چهارپره توسط یک الگوریتم کنترلی که به تنظیم سرعت‌های موتورها می‌پردازد، انجام می‌شود. در این مقاله یک کنترلر کلاسیک تناسبی-مشتق‌گیر برای کنترل ربات پرنده چهارپره طراحی شده و کارایی و رفتار آن مورد بررسی قرار می‌گیرد. در این مقاله از الگوریتم هوشمند ازدحام ذرات برای تنظیم ضرایب کنترلی این سیستم استفاده شده است. با وجود پیشرفت بسیاری که در طراحی کنترلرهای خطی، غیرخطی و هوشمند به وجود آمده، کنترلرهای PID از رایج‌ترین کنترلهایی می‌باشد که در بسیاری از کارهای صنعتی و دانشگاهی برای سیستم‌های خطی و برخی سیستم‌های غیرخطی مورد استفاده قرار گرفته است. از ویژگی‌های این نوع کنترلر می‌توان به سادگی ساختار و قابل‌درک بودن نحوه کارکرد اشاره کرد. کنترلر PD طراحی شده در این مقاله از ترکیب دو بخش تناسبی و مشتق‌گیر تشکیل شده است. این کنترلر بر اساس خطا و مشتق خطای مابین ورودی مطلوب و خروجی سیستم، ورودی کنترلی مطلوب سیستم را جهت انجام عمل کنترلی مورد نظر تولید می‌کند.

در کنترلرهای تناسبی-مشتق‌گیر برای گرفتن جواب بهینه و مطلوب مورد نظر از سیستم، باید ضرایب کنترلی در آن به گونه‌ای تنظیم شوند که سیستم، ورودی مطلوب را تعقیب کند. ضرایب مورد نیاز برای کنترلر تناسبی-مشتق‌گیر شامل ضرایب تناسبی (K_p) و مشتق‌گیر (K_d) می‌باشد. در طراحی کنترلر به روش تناسبی-مشتق‌گیر ورودی‌های کنترلی از رابطه (۴) پیروی می‌کنند.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (4)$$

که در آن ضرایب K_p, K_d ضرایب کنترلی طراحی شده برای کنترلر تناسبی-مشتق‌گیر می‌باشند و $e(t)$ خطای مقدار 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 تتا^۱ به صورت (۵) تعریف می‌شود.

$$U_3 = K_{pth} (\theta_d - \theta) + K_{dth} (\dot{\theta}_d - \dot{\theta}) \quad (5)$$

ورودی‌های کنترلی برای کنترل زاویه فی^۲ و سای^۳ و کنترل ارتفاع به صورت معادلات (۶) تا (۸) تعریف می‌شود.

$$U_2 = K_{pphi} (\varphi_d - \varphi) + K_{dphi} (\dot{\varphi}_d - \dot{\varphi}) \quad (6)$$

$$U_4 = K_{ppsi} (\psi_d - \psi) + K_{dpsi} (\dot{\psi}_d - \dot{\psi}) \quad (7)$$

$$U_1 = \frac{m(g + k_{pz}(z_d - z) + k_{dz}(\dot{z}_d - \dot{z}))}{\cos \theta \cos \varphi} \quad (8)$$

برای تنظیم ضرایب کنترلی کنترلر کلاسیک می‌توان به روش زیگلر و نیکولز، روش سعی و خطا، روش هوشمند فازی، روش هوشمند شبکه عصبی و استفاده از الگوریتم‌های تکاملی هوشمند اشاره کرد. پاسخ مطلوب سیستم باید با کمترین زمان نشست، کمترین بیشینه فرا جهش و کمترین نوسان ممکن همراه باشد. در این مقاله از روش الگوریتم هوشمند ازدحام ذرات برای تعیین ضرایب کنترلر کلاسیک تناسبی- مشتق گیر استفاده شده است. در این روش، پروسه یافتن ضرایب کنترلی، یک پروسه تکراری است که از یک مجموعه n تایی مقادیر اولیه تصادفی شروع به کار می‌کند و پارامترهای بهینه فضای حالت را جستجو می‌کند. روش الگوریتم ازدحام ذرات بر اساس رفتار گروهی جمعی از کنشگران تشکیل شده که این کنشگران به‌طور محلی با یکدیگر و با پیرامون خود در ارتباط هستند. در این روش یک سرعت ابتدایی به ذرات اختصاص داده می‌شود، همچنین کانال‌های ارتباطی بین ذرات در نظر گرفته می‌شود. این ذرات در فضای پاسخ حرکت می‌کنند و نتایج بر مبنای یک ملاک شایستگی پس از هر بازه‌ی 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. شایستگی ذرات توسط تابع هدف در نظر گرفته شده بررسی می‌شود. این چرخه تا زمانی که الگوریتم به کمینه تابع هدف نرسیده باشد تکرار می‌شود و ذرات را به سمت جواب بهینه سوق می‌دهد. تابع هدف در نظر گرفته شده در این مقاله طبق معادله (۹) می‌باشد.

$$Cost = e^{-\beta} (M_p + e_{ss} + t_r) + (1 - e^{-\beta}) t_s \quad (9)$$

که در تابع هدف، t_s 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 است. روند بهینه‌سازی ضرایب کنترلی و کمینه کردن تابع هدف طبق شکل (۲) آورده شده است. با توجه به پاسخ آورده شده، تابع هدف در هر سیکل تنظیم ضرایب کم شده و به مقدار کمینه خود رسیده است. پارامترهای سیستم برای شبیه‌سازی در جدول (۲) و ضرایب تنظیم شده به روش الگوریتم تکاملی ازدحام ذرات در جدول (۱) آورده شده‌اند. با شبیه‌سازی کنترلر PD تنظیم شده با الگوریتم ازدحام ذرات، پاسخ سیستم به صورت شکل (۳) می‌باشد.

¹ Theta

² Phi

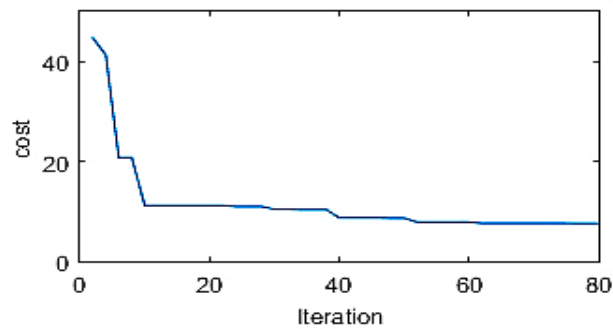
³ Psi

جدول ۱- ضرایب تنظیم‌شده به روش الگوریتم تکاملی ازدحام ذرات

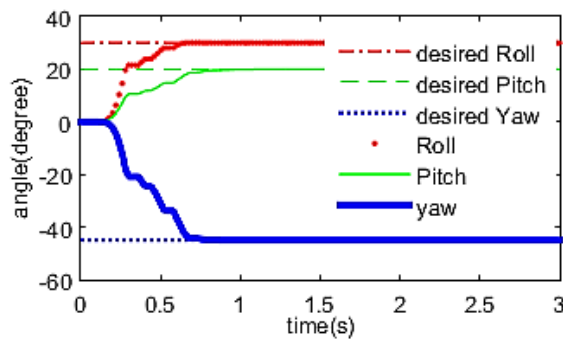
ضرایب	K_{pphi}	K_{dphi}	K_{proll}	K_{droll}	K_{ppsi}	K_{dpsi}
مقدار	۲/۲	۱/۴	۲/۳	۱/۴	۱/۲	۰/۴

جدول ۲- پارامترهای سیستم برای شبیه‌سازی

پارامترهای سیستم	مقدار	واحد
m	۰/۶۵	kg
Ix	۰/۰۰۷۵	kg.m ²
Iy	۰/۰۰۷۵	kg.m ²
Iz	۰/۰۱۳	kg.m ²
Jr	۰/۰۰۰۰۶۵	kg.m ²
l	۰/۲۳	m
b	۰/۰۰۰۰۲۳	Ns ²
d	۰/۰۰۰۰۰۷۵	Nms ²



شکل ۲- مقدار تابع هدف در هر سیکل از تنظیم ضرایب



شکل ۳- پاسخ کنترلر PD به ورودی مطلوب

۴- طراحی مشاهده گر

برای تهیه داده‌های آموزشی برای طراحی مشاهده گر شبکه عصبی، یک مشاهده گر غیرخطی برای سیستم چهارپره طبق مقاله [۳] با در نظر گرفتن مدل سیستم (۱) و \hat{X} به عنوان تخمینی از بردار حالت (۱۰) تعریف شده و شبیه‌سازی شده است.

$$x^T = [\varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}] \quad (10)$$

۴-۱- طراحی مشاهده گر غیرخطی

مدل مشاهده گر یک کپی از سیستم اصلی (معادله (۱۱)) می‌باشد که بهره‌های تصحیح کننده از خطاهای پیش‌بینی شده دارند.

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \bar{\Omega} x_4 + b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega} x_2 + b_1 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_4 x_2 + a_8 x_6^2 + b_3 U_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + U_x \frac{U_1}{m} \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_1}{m} \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + U_1 \frac{\cos x_1 \cos x_3}{m} - g \end{array} \right. \quad (11)$$

بنابراین طبق تعریف \hat{X} به عنوان تخمینی از بردار حالت (۱۰) معادلات (۱۲) را داریم:

$$\left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 + A_1(z_1) \\ \dot{\hat{x}}_2 = a_1 \hat{x}_4 \hat{x}_6 + a_2 \hat{x}_2^2 + a_3 \bar{\Omega} \hat{x}_4 + b_1 U_2 + A_2(z_2) \\ \dot{\hat{x}}_3 = \hat{x}_4 + A_3(z_3) \\ \dot{\hat{x}}_4 = a_4 \hat{x}_2 \hat{x}_6 + a_5 \hat{x}_4^2 + a_6 \bar{\Omega} \hat{x}_2 + b_2 U_3 + A_4(z_4) \\ \dot{\hat{x}}_5 = \hat{x}_6 + A_5(z_5) \\ \dot{\hat{x}}_6 = a_7 \hat{x}_4 \hat{x}_2 + a_8 \hat{x}_6^2 + b_3 U_4 + A_6(z_6) \\ \dot{\hat{x}}_7 = \hat{x}_8 + A_7(z_7) \\ \dot{\hat{x}}_8 = a_9 \hat{x}_8 + U_x \frac{U_1}{m} + A_8(z_8) \\ \dot{\hat{x}}_9 = \hat{x}_{10} + A_9(z_9) \\ \dot{\hat{x}}_{10} = a_{10} \hat{x}_{10} + U_y \frac{U_1}{m} + A_{10}(z_{10}) \\ \dot{\hat{x}}_{11} = \hat{x}_{12} + A_{11}(z_{11}) \\ \dot{\hat{x}}_{12} = a_{11} \hat{x}_{12} + U_1 \frac{\cos x_1 \cos x_3}{m} - g + A_{12}(z_{12}) \end{array} \right. \quad (12)$$

با تعریف خطاهای دینامیکی پیش‌بینی‌شده $\Lambda_i(z_i)$ ، معادلات (۱۳) را داریم:

$$\left\{ \begin{array}{l} \dot{Z}_1 = Z_2 - A_1 \\ \dot{Z}_2 = a_1 \Delta_{x_4 x_6} + a_2 \Delta_{x_2^2} + a_3 \Delta_{x_4} \bar{Q} - A_2 \\ \dot{Z}_3 = Z_4 - A_3 \\ \dot{Z}_4 = a_4 \Delta_{x_2 x_6} + a_5 \Delta_{x_4^2} + a_6 \Delta_{x_2} \bar{Q} - A_4 \\ \dot{Z}_5 = Z_6 - A_5 \\ \dot{Z}_6 = a_7 \Delta_{x_2 x_4} + a_8 \Delta_{x_6^2} - A_6 \\ \dot{Z}_7 = Z_8 - A_7 \\ \dot{Z}_8 = a_9 Z_8 - A_8 \\ \dot{Z}_9 = Z_{10} - A_{10} \\ \dot{Z}_{10} = a_{10} Z_{10} - A_{10} \\ \dot{Z}_{11} = Z_{12} - A_{11} \\ \dot{Z}_{12} = a_{11} Z_{12} - A_{12} \end{array} \right. \quad (13)$$

که در آن معادلات (۱۴) صادق می‌باشند.

$$\left\{ \begin{array}{l} \Delta_{x_i x_j} = x_i x_j - \hat{x}_i \hat{x}_j \\ \Delta_{x_i^2} = x_i^2 - \hat{x}_i^2 \\ \Delta_{x_i} = x_i - \hat{x}_i \\ Z_i = y_i - \hat{y}_i \quad \text{if } i \text{ impair} \\ Z_i = x_i - \hat{x}_i \quad \text{if } i \text{ pair} \end{array} \right. \quad (14)$$

خروجی‌های در نظر گرفته‌شده برای این مشاهده‌گر طبق بردار (۱۵) تعریف شده‌اند.

$$Y = [x_1, x_3, x_5, x_7, x_9, x_{11}]^T \quad (15)$$

برای به دست آوردن و محاسبه بهره‌های تصحیح‌کننده، ضروری است تا خطاهای دینامیکی، تخمینی پایدار باشند بنابراین طبق قانون لیاپانوف، معادله لیاپانوف طبق معادله (۱۶) انتخاب می‌شود.

$$V(z_1, z_2) = \frac{1}{2} (z_1^2 + z_2^2) \quad (16)$$

بنابراین داریم:

$$\dot{V}(z_1, z_2) = z_1 \dot{z}_1 + z_2 \dot{z}_2 = z_1 (z_2 - A_1) + z_2 (a_1 \Delta_{x_4 x_6} + a_2 \Delta_{x_2^2} + a_3 \Delta_{x_4} \bar{Q} - A_2) \quad (17)$$

شرط لازم برای برقراری پایداری لیاپانوف $\dot{V}(z_1, z_2) \leq 0$ می‌باشد. به همین منظور طبق معادله (۱۸) داریم:

$$\left\{ \begin{array}{l} A_1(z_1) = z_2 + k_1 z_1 \\ A_2(z_2) = a_1 \Delta_{x_4 x_6} + a_2 \Delta_{x_2^2} + a_3 \Delta_{x_4} \bar{Q} + k_2 z_2 \end{array} \right. \quad (18)$$

With $(k_1, k_2) \in \mathbb{R}^{+2}$

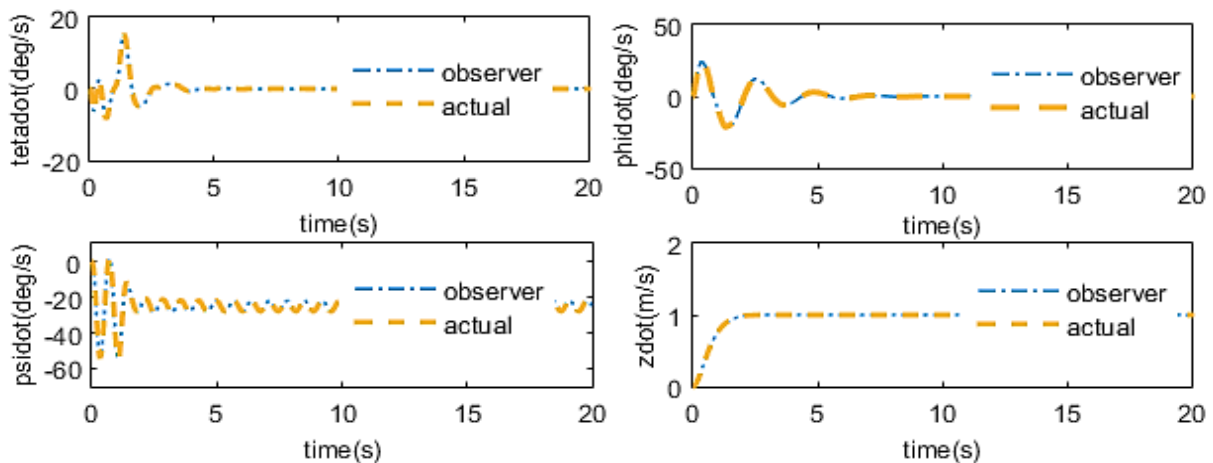
با همین روش، دیگر بهره‌های تصحیح‌کننده حصول می‌شود. طبق معادله (۱۸)، معادلات (۱۹) را داریم:

$$\left\{ \begin{array}{l} \Lambda_3(z_3) = z_4 + k_3 z_3 \\ \Lambda_4(z_4) = a_4 \Delta_{x_2 x_6} + a_5 \Delta_{x_4^2} + a_6 \Delta_{x_2} \bar{\Omega} + k_4 z_4 \\ \Lambda_5(z_5) = z_6 + k_5 z_5 \\ \Lambda_6(z_6) = a_7 \Delta_{x_2 x_4} + a_8 \Delta_{x_6^2} + k_6 z_6 \\ \Lambda_7(z_7) = z_8 + k_7 z_7 \\ \Lambda_8(z_8) = a_9 z_8 + k_8 z_8 \\ \Lambda_9(z_9) = z_9 + k_{10} z_{10} \\ \Lambda_{10}(z_{10}) = a_{10} z_{10} + k_{10} z_{10} \\ \Lambda_{11}(z_{11}) = z_{12} + k_{11} z_{11} \\ \Lambda_{12}(z_{12}) = a_{11} z_{12} + k_{12} z_{12} \end{array} \right. \quad (19)$$

با شبیه‌سازی این مشاهده‌گر بر روی سیستم چهارپره در محیط شبیه‌سازی متلب و با استفاده از حالت‌های مشاهده‌شده در مشاهده‌گر در پیاده‌سازی کنترلر، نتایج خروجی شکل (۴) حاصل شده است.

۴-۲- طراحی مشاهده‌گر شبکه عصبی

شبکه‌های عصبی برگرفته از ساختار ذهن انسان می‌باشند. این ساختار دارای قابلیت‌هایی همچون یادگیری، تصمیم‌گیری، پردازش موازی، تخمین تابع‌های غیرخطی و . . . می‌باشد. در این پژوهش سعی شده تا از قابلیت یادگیری و تصمیم‌گیری شبکه عصبی استفاده شود و با استفاده از حالت‌هایی که در ربات چهارپره قابل اندازه‌گیری می‌باشد بتوان حالت‌هایی که قابل اندازه‌گیری نیستند را مشاهده نمود و بتوان در هزینه‌هایی از قبیل خرید حس‌گرهایی که برای محاسبه تمامی حالت‌ها استفاده می‌شود و خرید آن‌ها مقرون به صرفه نیست صرفه‌جویی کرد. برای طراحی شبکه عصبی نیاز به یک سری داده برای آموزش سیستم می‌باشد تا بتوان شبکه عصبی توانمندی را طراحی کرد. در این پژوهش این داده‌ها از شبیه‌سازی سیستم با استفاده از کنترلر گام به عقب که در بخش قبل به طراحی آن پرداخته شد در محیط شبیه‌سازی متلب تهیه شده است. در این پژوهش فرض بران شده که تمام حالت‌های سیستم به جز سرعت‌های زاویه‌ای و نرخ تغییرات ارتفاع قابل اندازه‌گیری می‌باشد.

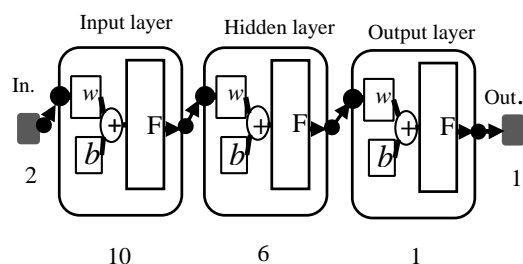


شکل ۴- مقایسه خروجی سیستم با استفاده از مشاهده‌گر غیرخطی در مقایسه با مقادیر واقعی

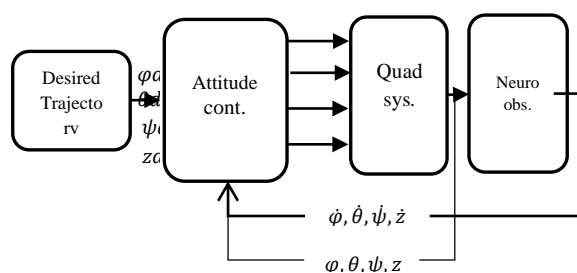
با این فرض به طراحی مشاهده‌گری پرداخته شده است که بتواند حالت‌های نامعلوم سیستم که به آن نیاز است را تخمین بزند. شکل (۵) ساختار مشاهده‌گر شبکه عصبی طراحی شده را نشان می‌دهد.

در این ساختار، لایه ورودی شبکه از ۱۰ نورون، لایه پنهان شبکه از ۶ نورون و لایه خروجی شبکه از یک نورون تشکیل شده است. ورودی‌های این شبکه را حالت‌های معین سیستم $(x_d, \dot{x}_d, y_d, \dot{y}_d, z_d, \phi_d, \theta_d, \psi_d)$ و خروجی‌های شبکه را حالت‌های نامشخص سیستم $(\dot{\phi}_d, \dot{\theta}_d, \dot{\psi}_d, \dot{z}_d)$ مشخص می‌کنند. در اینجا در ابتدا فرض شده که تمامی حالت‌های سیستم با استفاده از کنترلر کلاسیک PD و مشاهده‌گر طراحی شده مشخص می‌باشد و با استفاده از این داده‌های معین به آموزش شبکه پرداخته شده و شبکه عصبی آموزش دیده شده، جایگزین شده تا کارایی این مشاهده‌گر شبکه عصبی مشخص شود. شکل (۶) نحوه پیاده‌سازی این مشاهده‌گر را نشان می‌دهد. با شبیه‌سازی مشاهده‌گر طراحی شده روی سیستم طبق ساختار شکل (۵) و مقایسه با داده‌های اصلی، نتایج حاصل نشان از رفتار خوب این مشاهده‌گر می‌باشد.

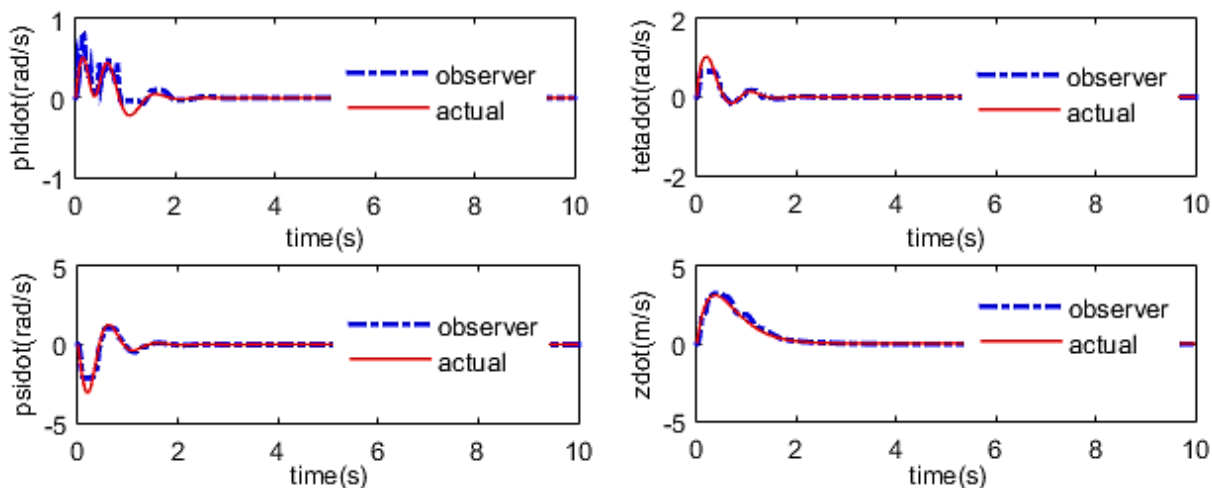
این نتایج برای دو حالت ورودی‌های مطلوب سیستم مقایسه شده‌اند که نتایج آن در شکل (۷) آورده شده است. با استفاده از این مشاهده‌گر، یک شبکه عصبی طراحی شده و با جایگزینی آن در سیستم و مقایسه رفتار خروجی سیستم برای ورودی‌های مطلوب، نتایج آن طبق شکل (۸) آورده شده است. همان‌طور که مشاهده می‌شود کنترلر توانسته باحالت‌هایی که مشاهده‌گر برای آن تخمین زده است به خوبی سیستم را کنترل کند و رفتار خوبی را از خود نشان دهد که این نشان‌دهنده توانمندی مشاهده‌گر طراحی شده می‌باشد. همان‌گونه که مشخص می‌باشد رفتار سیستم بدون مشاهده‌گر به دلیل وجود مقادیر دقیق خروجی‌های متغیر حالت، بهتر از سیستم در حضور مشاهده‌گر می‌باشد زیرا مشاهده‌گر سعی بر آن دارد تا با استفاده از متغیرهای حالت موجود متغیرهای نامعلوم را محاسبه نماید.



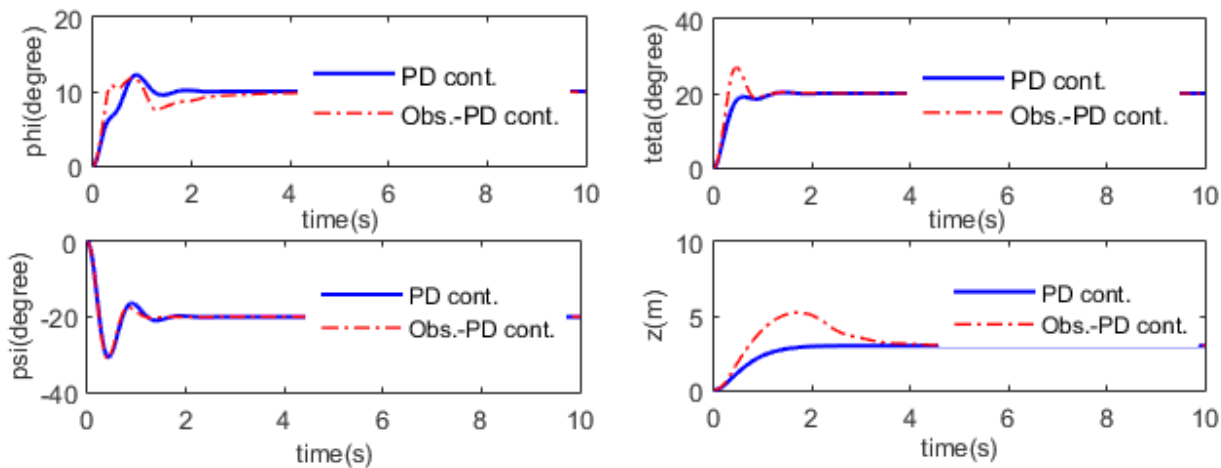
شکل ۵- ساختار شبکه عصبی استفاده شده در این مقاله



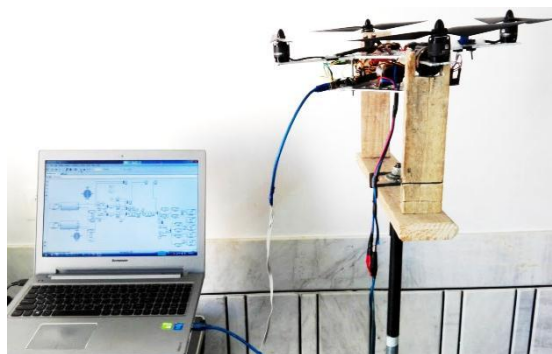
شکل ۶- دیاگرام کنترلی سیستم با مشاهده‌گر



شکل ۷- مقایسه خروجی سیستم با استفاده از مشاهده گر شبکه عصبی با مقادیر واقعی



شکل ۸- نمودار مقایسه خروجی های حالت چهارپره با جایگزینی مشاهده گر در سیستم کنترلی و حالت کنترلر بدون مشاهده گر به ازای شرایط یکسان



شکل ۹- ساختار مدل شبیه ساز یک درجه آزادی و ارتباط با محیط متلب

۵- پیاده‌سازی روی سیستم شبیه‌ساز

برای تعیین میزان کارایی کنترلر و مشاهده‌گر طراحی‌شده، نیاز به بستری می‌باشد تا بتوان این کنترلرها را بر روی سیستم واقعی پیاده‌سازی کرد و نتایج حاصل را بررسی نمود. در این بخش به‌منظور پیاده‌سازی و بررسی نتایج کنترلرهای طراحی‌شده، مدلی برای چهارپره ارائه شده تا بر این اساس بتوان کنترلرهای وضعیت و مشاهده‌گرهای طراحی‌شده را روی سیستم واقعی مدل‌سازی و نتایج حاصل و توانمندی و ناتوانی این کنترلرها را تعیین کرد.

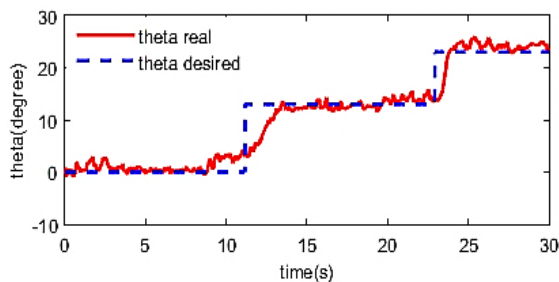
در این مقاله سیستم شبیه‌ساز به‌صورت مدل یک درجه آزادی برای پیاده‌سازی کنترلرهای وضعیت در نظر گرفته شده است. این ساختار با قابلیت ارتباط با محیط نرم‌افزاری متلب تهیه‌شده و قابلیت اجرای هرگونه کنترلر طراحی‌شده برای این سیستم در این محیط شبیه‌سازی را بر روی بردهای الکترونیکی آردیونو دارا می‌باشد. سیستم شبیه‌ساز طبق شکل (۹) (برای آزادی بخشیدن به سیستم در یک درجه آزادی و قابلیت انجام حرکت چرخشی در راستای زاویه تتا) ارائه شده است.

۶- نتایج تجربی پیاده‌سازی الگوریتم کنترلی و مشاهده‌گر طراحی‌شده روی سیستم شبیه‌ساز

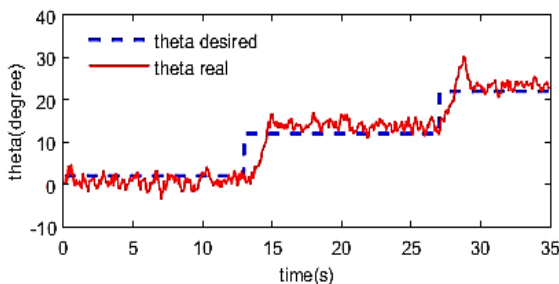
برای تعیین کارایی و رفتار کنترلرهای طراحی‌شده و اهمیت پایداری سیستم چهارپره، در این بخش به پیاده‌سازی کنترلر و مشاهده‌گر طراحی‌شده بر روی سیستم شبیه‌ساز پرداخته شده است. برای بررسی عملکرد کنترلر و مشاهده‌گر طراحی‌شده (کنترلر PD بدون استفاده از مشاهده‌گر و کنترلر PD با استفاده از مشاهده‌گر شبکه عصبی) در کنترل زاویه تتا، 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.

برای تعیین کارایی این کنترلر و مشاهده‌گر طراحی‌شده و توانایی آن‌ها ورودی‌های مطلوب به‌صورت پله واحد در نظر گرفته شده‌اند و کارایی کنترلرهای طراحی‌شده مشخص شده است. همچنین برای بررسی کارایی مشاهده‌گر طراحی، خروجی حالت (نرخ زاویه) در حالت واقعی و با استفاده از مشاهده‌گر آورده شده‌اند. با مقایسه رفتار مشاهده‌گر نسبت به حالت واقعی مشخص می‌شود که مشاهده‌گر شبکه عصبی توانسته به خوبی رفتار حالت خروجی موردنظر را تخمین بزند.

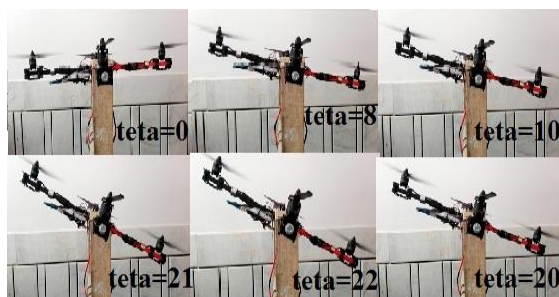
رفتار سیستم با مشاهده‌گر در مقابل با سیستم بدون مشاهده‌گر (کاملاً مشاهده‌پذیر) دارای نوساناتی می‌باشد که این نشان از برتری سیستم مشاهده‌پذیر می‌باشد و همچنین نشان می‌دهد که مشاهده‌گر طراحی‌شده از توانایی خوبی برای تخمین متغیرهای حالت دارد. نتایج پیاده‌سازی در شکل‌های (۱۰) تا (۱۳) آورده شده است. بخشی از پاسخ سیستم با استفاده از کنترلر کلاسیک PD بدون استفاده از مشاهده‌گر در شکل (۱۰) و با استفاده از مشاهده‌گر شبکه عصبی طبق شکل (۱۱) آورده شده‌اند.



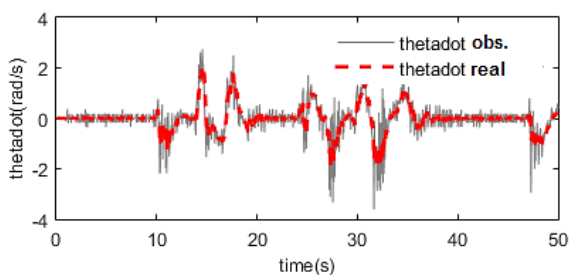
شکل ۱۰- پاسخ کنترلر PD بدون استفاده از مشاهده گر روی مدل شبیه ساز



شکل ۱۱- پاسخ کنترلر PD با استفاده از مشاهده گر روی مدل شبیه ساز



شکل ۱۲- پاسخ سیستم با استفاده از کنترلر کلاسیک PD و مشاهده گر



شکل ۱۳- مقایسه خروجی مشاهده گر شبکه عصبی طراحی شده با مقدار واقعی بر روی مدل شبیه ساز

۷- نتیجه گیری

در این مقاله به طراحی و شبیه سازی کنترلر کلاسیک PD و مشاهده گر شبکه عصبی پرداخته شده است. در ابتدا کنترلر کلاسیک PD با روش زیگلر و نیکولز تنظیم شده و یک شبکه عصبی مشاهده گر، بر اساس داده های خروجی مشاهده گر غیرخطی طراحی شده، آموزش داده شده است و برای بررسی عملکرد روی سیستم، در محیط متلب شبیه سازی شده اند.

همان‌گونه که از پاسخ سیستم حاصل شد، مشاهده‌گر شبکه عصبی به‌خوبی توانسته مقادیر موردنظر را تخمین بزند. در ادامه یک سیستم شبیه‌ساز برای پیاده‌سازی کنترلرها و مشاهده‌گرهای طراحی‌شده در محیط متلب برای سیستم چهارپره، ارائه‌شده، که قابلیت اجرای انواع کنترلرها بر روی این سیستم را دارد. درنهایت کنترلر و مشاهده‌گر شبکه عصبی طراحی‌شده روی مدل واقعی شبیه‌ساز، پیاده‌سازی شده‌اند و کارایی آن‌ها مشخص شده است. پاسخ کنترلر و مشاهده‌گر شبکه عصبی روی مدل شبیه‌ساز، کارایی خوب مشاهده‌گر را نشان می‌دهد. به‌منظور توسعه مدل شبیه‌ساز می‌توان با افزودن حسگرها، دوربین‌ها و ماژول‌های التراسونیک و توسعه پایه نگه‌دارنده‌ی این سیستم می‌توان الگوریتم‌های کنترلی موقعیت را علاوه بر الگوریتم‌های کنترلی وضعیت روی این سیستم پیاده‌سازی کرد. همچنین به دلیل ایمنی این سیستم می‌توان با استفاده از این مدل آزمایشگاهی، اعمال نویزها و اغتشاشات به سیستم را بررسی کرد و کنترلرهای مناسب‌تری را تهیه نمود.

همان‌گونه که در این مقاله نشان داده شد شبکه‌های عصبی می‌توانند به‌خوبی رفتار کنترلرها و مشاهده‌گرها را یاد بگیرند و جایگزین آن‌ها شوند. از این‌رو توسعه کنترلرها و مشاهده‌گرهای شبکه عصبی پیشنهاد می‌شود. از سوی دیگر می‌توان به توسعه شبکه‌های عصبی با قابلیت آموزش برخط اشاره کرد. با افزایش توانایی یادگیری برخط می‌توان نیاز به مدل سیستم را برای طراحی کنترلرها از بین برد و کنترلرهای دقیق‌تری طراحی کرد.

مراجع

- [1] Lendek, Z., Berna, A., and Guzm, j., "Application of Takagi-Sugeno Observers for State Estimation in a Quadrotor", 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, Florida, pp. 7530-7535, (2011).
- [2] Mokhtari, A., M'Sirdi, N. K., Meghriche, K., and Belaidi, A., "Feedback Linearization and Linear Observer for a Quadrotor Unmanned Aerial Vehicle", *Advanced Robotics*, Vol. 20, pp. 71-91, (2006).
- [3] Bouadi, H., and Tadjine, M., "Nonlinear Observer Design and Sliding Mode Control of Four Rotors Helicopter", *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol. 25, pp. 225-230, (2007).
- [4] Dierks, T., and Jagannathan, S., "Output Feedback Control of a Quadrotor UAV using Neural Networks", in *IEEE Transactions on Neural Networks*, Vol. 21, pp. 50-66, (2010).
- [5] Benzemrane, K., Santosuosso, G. L., and Damm, G., "Unmanned Aerial Vehicle Speed Estimation via Nonlinear Adaptive Observers", *American Control Conference Philadelphia, U.S.A.*, pp. 985-990, (2007).
- [6] Benallegue, A., Mokhtari, A., and Fridman, L., "High-order Sliding-mode Observer for a Quadrotor UAV", *International Journal of Robust and Nonlinear Control*, Vol. 18, pp. 427-440, (2008).

- [7] Bouhali, O., and Boudjedir, H., "Neural Network Control with Neuro-sliding Mode Observer Applied to Quadrotor Helicopter", International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, pp. 24-28, (2011).
- [8] Khalil, H. K., "High-gain Observers in Nonlinear Feedback Control", IEEE International Conference on Control and Automation, Christchurch, Newzealand, pp. 1527-1528, (2009).
- [9] Boudjedir, H., Bouhali, O., and Rizoug, N., "Neural Network Control Based on Adaptive Observer for Quadrotor Helicopter", International Journal of Information Technology, Control and Automation (IJITCA), Vol. 2, No. 3, pp. 39-54, (2012).
- [10] Ahmed, S. F., Kushsairy, K., Bakar, M. I. A., D., and Joyo, M. K., "Attitude Stabilization of Quad-rotor (UAV) System using Fuzzy PID Controller (an Experimental Test)", Second International Conference on Computing Technology and Information Management (ICCTIM), At Johar, Kalaysia, pp. 99-104, (2015).
- [11] Seidabad, E. A., Vandaki, S., and Kamyad, A. V., "Designing Fuzzy PID Controller for Quadrotor", International Journal of Advanced Research in Computer Science & Technology (IJARCST), Vol. 2, pp. 221-227, (2014).
- [12] Dikmen, I. C., Arisoy, A., and Temeltas, H., "Attitude Control of a Quadrotor", 4th International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, pp. 722-727, (2009).
- [13] Farrell, J., Sharma, M., and Polycarpou, M., "Backstepping-based Flight Control with Adaptive Function Approximation", Journal of Guidance, Control and Dynamics, Vol. 28, pp. 1089-1102, (2005).
- [14] Bolandi, H., Rezaei, M., Mohsenipour, R., Nemati, H., and Smailzadeh, S. M., "Attitude Control of a Quadrotor with Optimized PID Controller", Journal of Intelligent Control and Automation, Vol. 4, pp. 342-349, (2013).
- [15] Zareb, M., Ayad, R., and Nouibat, W., "Fuzzy-PID Hybrid Control System to Navigate an Autonomous Mini-quadrotor", 3rd International Conference on Systems and Control (ICSC), Algiers, Algeria, pp. 906-913, (2013).
- [16] Dunfied, J., Tarbouchi, M., and Labonte, G., "Neural Network Based Control of a Four Rotor Helicopter", IEEE International Conference on Industrial Technology, Hammamet, Tunisia, Vol. 3, pp. 1543-1548, (2004).
- [17] Lee, B. Y., Lee, H. I., and Tahk, M. J., "Analysis of Adaptive Control using on-line Neural Networks for a Quadrotor UAV", 13th International Conference on Automation and Systems (ICCAS), Gwangju, Korea, pp. 1840-1844, (2013).
- [18] Frye, M. T., and Provence, R. S., "Direct Inverse Control using an Artificial Neural Network for the Autonomous Hover of a Helicopter", IEEE International Conference on Systems, Man and Cybernetics (SMC), Manchester, United Kingdom, pp. 4121-4122, (2014).
- [19] Rezazadeh, S., Ardestani, M. A., and Sadeghi, P. S., "Optimal Attitude Control of a Quadrotor UAV using Adaptive Neuro-Fuzzy Inference System (ANFIS)", 3rd

International Conference on Control, Instrumentation, and Automation (ICCIA), Amirkabir University of Technology (Tehran), Iran, pp. 219-223, (2013).

- [20] Lower, M., and Tarnawski, W., "Quadrotor Navigation using the PID and Neural Network Controller", *Advances in Intelligent System and Computing*, Springer, Cham, Vol. 365, pp. 265-274, (2015).
- [21] Boudjedir, H., Bouhali, O., and Rizoug, N., "Adaptive Neural Network Control Based on Neural Observer for Quadrotor Unmanned Aerial Vehicle", *Advanced Robotics*, Vol. 28, pp. 1151-1164, (2014).

فهرست نمادهای انگلیسی

b	ضریب نیروی بالابرنده ملخ
d	ضریب نیروی درگ
$e(t)$	خطای مقدار اندازه‌گیری شده از حسگرها و مقدار مورد انتظار در لحظه t
e_{ss}	خطای ماندگار
I_x	ممان اینرسی در راستای X
I_y	ممان اینرسی در راستای y
I_z	ممان اینرسی در راستای z
J_R	ممان اینرسی
K_p	ضریب کنترلی تناسبی
K_d	ضریب کنترلی مشتق‌گیر
l	فاصله بین روتورها و مرکز جرم
M	جرم
M_p	بیشینه فرا جهش پاسخ
U_1	ورودی کنترلی نیروی مجموع در راستای z
U_2	ورودی کنترلی مومنت‌های چرخش حول محور y
U_3	ورودی کنترلی مومنت‌های چرخش حول محور x
U_4	ورودی کنترلی مومنت‌های چرخش حول محور z
t_s	زمان نشست سیستم
t_r	زمان فراز
X	موقعیت مرکز جرم در سازه
x_i	متغیر فضای حالت
\hat{X}	تخمینی از بردار حالت

Y	موقعیت مرکز جرم در سازه
Z	موقعیت مرکز جرم در سازه
z_d	ارتفاع مطلوب

نمادهای یونانی

φ	فی
ϕ_d	فی مطلوب
θ	تتا
θ_d	تتا مطلوب
ψ	سای
ψ_d	سای مطلوب
Ω	سرعت زاویه‌ای
Ω_1	سرعت‌های زاویه‌ای روتورها
Ω_2	زاویه‌ای روتورها
Ω_3	سرعت‌های زاویه‌ای روتورها
Ω_4	سرعت‌های زاویه‌ای روتورها
Ω_d	ورودی کنترلی مجموع سرعت‌های زاویه‌ای
$\Lambda_i(z_i)$	خطاهای دینامیکی پیش‌بینی شده

Abstract

Quadrotor stability is the most important concept because of its nonlinear dynamic. In this paper a PID controller is designed based on PSO algorithm to control attitude of quadrotor. Then a nonlinear observer is designed to estimate the angular velocity states. After that a neural observer is designed based on training data. Finally, to verify these controllers and observers, a simulation performed in the Simulink environment of the Matlab. In addition, we implemented these control and observer methods on a quadrotor test bench. Practical implementation results demonstrate the effectiveness of this presented method.