

طراحی کنترل کننده شبکه عصبی برای ربات عمودپرواز با استفاده از آموزش دهنده مشتق گیر-تناسبی و بازگشت به عقب

محمد بناکار^۱
کارشناسی ارشد

حامد شهبازی^۲
استادیار

در این مقاله، به طراحی یک کنترل کننده شبکه عصبی هوشمند که بتواند ربات عمودپرواز را در حالت تعادل در میان انواع اغتشاشات قرار دهد، پرداخته شده است. پس از مقایسه دو نوع کنترل کننده مستقل بازگشت به عقب و PID در محیط شبیه سازی اختلاف آن ها به صورت داده هایی در نرم افزار ذخیره شده است. با مشخص کردن داده های ورودی کنترل کننده و داده های هدف و با استفاده از معماری شبکه عصبی پیش خور و نارس کنترل کننده ای هوشمند طراحی می شود و نتایج به دست آمده در نمودارهای تشخیص پایداری متعددی نشان داده می شود و نتایج به دست آمده نشان می دهد که تعادل و کنترل عمودپرواز کاملاً قابل قبول بوده است. در آخر نتایج به دست آمده در مدلی عملی بر روی عمودپرواز واقعی امتحان شده است.

واژه های راهنما: ربات پرنده عمود پرواز، شبکه عصبی مصنوعی، کنترل کننده بازگشت به عقب، کنترل پرواز، دینامیک کوادموتور

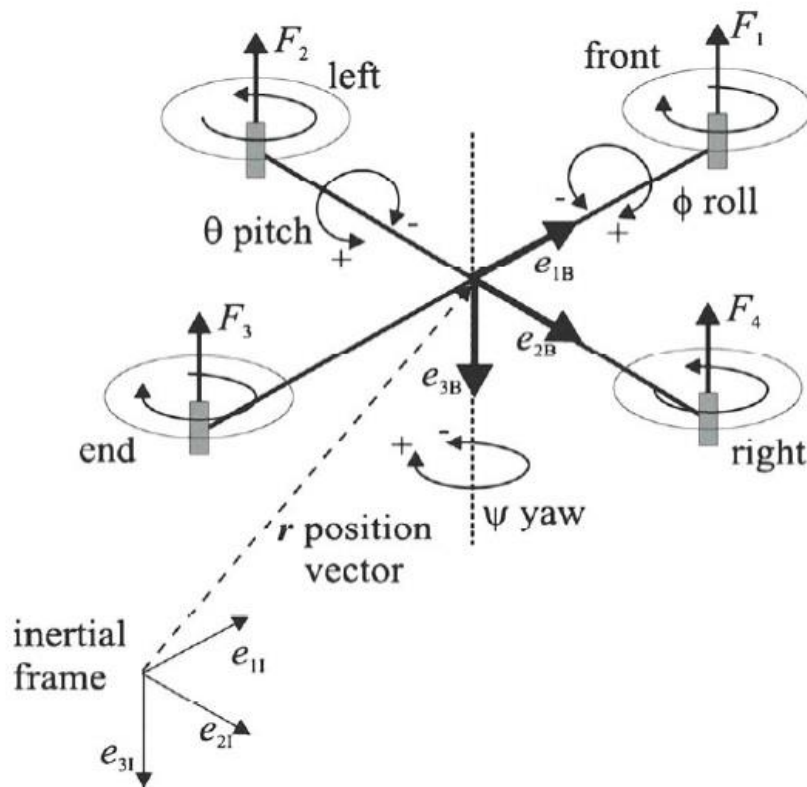
۱- مقدمه

علاقه بشر برای پرواز در سرتاسر تاریخ بشری کاملاً مشهود بوده است. رویای پرواز از گذشته های دور تا به امروز از بزرگ ترین چالش های بشری بوده است و منجر به انجام تلاش های زیادی جهت تحقق آن و ساخت وسایل پروازی شده است. وسایل هوایی را می توان به دو نوع وسایل سرنشین دار و بدون سرنشین تقسیم نمود که هر کدام از آن ها با توجه به قابلیت های پروازی مخصوص به خودشان در کاربردهای متفاوتی مورد استفاده قرار می گیرند. یکی از جذاب ترین انواع وسایل پرنده بدون سرنشین کوادروتور است. ساختار اصلی کوادموتور که با نام های دیگری همچون کوادروکوپتر یا هلی کوپتر چهار موتوره شناخته می شود، از یک قاب اصلی X شکل و چهار موتور که در چهار گوشه قاب اصلی قرار دارند و به هریک از آن ها یک پروانه

^۱ کارشناسی ارشد، دانشکده مهندسی مکانیک-دانشگاه آزاد اسلامی واحد خمینی شهر- اصفهان- ایران

^۲ نویسنده مسئول، استادیار، گروه مهندسی مکترونیک، دانشکده فنی مهندسی- دانشگاه اصفهان- اصفهان- ایران
shahbazi@eng.ui.ac.ir

متصل می‌باشد، تشکیل شده است. در شکل (۱) تصویری از این وسیله قابل مشاهده است. در این وسیله زاویه حمله پروانه‌ها تغییر نکرده و تغییر نیروی بالابر صرفاً با تغییر سرعت گردش موتورها انجام می‌گیرد. همچنین راستای چرخش هر چهارموتور ثابت و عمود بر قاب اصلی می‌باشد. به‌منظور خنثی شدن گشتاور اصطکاکی، جهت گردش موتورهای جلو و عقب نسبت به موتورهای چپ و راست در جهت عکس است. در سال ۲۰۰۴ بوعبدالله با ارائه یک مدل دینامیکی اولیه از کوادروتور به کنترل وضعیت این وسیله و حرکت ساده معلق پرداخت. در این پژوهش جهت پایداری و وضعیت چرخشی پرنده از تئوری پایداری لیاپانوف و برای کنترل حرکت معلق از کنترل کننده مشتق‌گیر-تناسبی استفاده شده است اما مدل دینامیکی ارائه شده ناقص بوده و در آن رابطه بین نرخ زوایای اوایلر و بردار سرعت زاویه‌ای نادیده گرفته شده است. علاوه بر آن در طراحی کنترل کننده، معادلات وضعیت چرخشی با حذف اثرات غیرخطی، ساده‌سازی و به‌صورت کلی شده است. لازم به ذکر است که این اثرات بعضاً قابل تغییر در نتایج هستند و همین‌طور از اثر وابستگی تغییرات زوایا به همدیگر و اثر ژيروسکوپی ۲ صرف‌نظر شده است. کنترل کننده طراحی شده هم بر روی نمونه واقعی این وسیله و هم بر روی مدل دینامیکی شبیه‌سازی شده این پرنده در فضای مجازی آزمایش شده است. در آزمایش بر روی نمونه واقعی، مرکز جرم پرنده به یک میز متصل شده است. به‌طوری‌که از حرکت انتقالی آن جلوگیری شده و فقط امکان تغییر در زوایای چرخشی وجود داشته باشد. کنترل کننده طراحی شده باهدف نگهداری زوایا در مقدار صفر بر روی پرنده پیاده‌سازی شده است. نتایج آزمایش نشان می‌دهد که زوایای پرنده با اختلافی حدود ۲ درجه حول مقدار مطلوب نوسان می‌کند و در شبیه‌سازی مجازی هدف آن بوده که کنترل کننده پرنده علاوه بر حفظ زوایا در مقدار صفر، پرنده را در دو متری سطح زمین در حالت معلق نگاه دارد.



شکل ۱- تصویر کلی عمودپرواز [۱]

نتایج حاصل از شبیه‌سازی نشان می‌دهد که پرنده با افزایش و کاهش ارتفاع حول موقعیت مطلوب نوسان کرده و علاوه بر آن زوایای چرخشی با اختلافی حدود شش درجه حول نقطه صفر دائماً در حال نوسان هستند که این خود نشان‌دهنده حرکت پرنده در جهات طولی و عرضی هم می‌باشد. [۲] در سال ۲۰۰۶ تایبی یک روش کنترلی بر پایه بردار کواترنین جهت پایداری و وضعیت چرخشی کوادروتور پیشنهاد کرد. این روش کنترل بر مبنای مدل دینامیک بو عبدالله بود و بر روی مدل عملی امتحان نشد [۳]. در سال ۲۰۰۷ ارگینر با طراحی چهار عدد کنترل‌کننده مشتق‌گیر-تناسبی جداگانه به کنترل‌کننده حرکت انتقالی و چرخشی این وسیله پرداخته است. معادلات حرکت انتقالی و چرخشی در این پژوهش به‌طور کامل استخراج شده ولی در طراحی کنترل‌کننده معادلات ساده‌سازی شده و از کلیه اثرات غیرخطی صرف‌نظر شده است. در این پژوهش با فرض اینکه حرکت در جهت طولی فقط وابسته به زاویه پیچ و حرکت در جهت عرضی فقط وابسته به زاویه رول پرنده می‌باشد، برای هر یک از این زوایا یک کنترل‌کننده مشتق‌گیر-تناسبی طراحی شده و با استفاده از ساده‌سازی معادلات حرکت انتقالی مقادیر مطلوب این زوایا به‌صورت وابسته به موقعیت مکانی پرنده محاسبه شده است. همچنین برای کنترل زاویه گردش یاو و کنترل حرکت عمودی پرنده از دو عدد مشتق‌گیر-تناسبی کنترل‌کننده جداگانه استفاده شده است [۴]. در سال ۲۰۰۸ سانکا یک مدل دینامیکی کامل از معادلات حرکت انتقالی و چرخشی کوادروتور ارائه کرد. در این پژوهش بیشتر اثرات غیرخطی و فیزیکی حاکم بر پرنده در پرواز در محیط واقعی به‌صورت معادلات ریاضی استخراج شده است که از آن جمله می‌توان به استخراج معادلات آئرو دینامیکی حاکم بر چرخش پروانه‌های پرنده، معادلات دینامیکی موتورهای DC و موج کنترلی به گردش در آورنده آن‌ها با ورودی تحریک مدولاسیون پهنای پالس، استخراج معادلات دینامیکی حرکت انتقالی و چرخشی پرنده، نحوه محاسبه ماتریس ممان اینرسی پرنده و اثرات استهلاک آئرو دینامیکی حین پرواز پرنده به‌طور کامل بررسی شد [۵]. پس از آن به‌مرور کنترل‌های غیرخطی برای عمودپروازها مورد استفاده قرار گرفت و هرکدام نقاط قوت و ضعف مخصوص به خود را داشتند. کنترل‌گر بازگشت به عقب که ساختار کنترلی بسیار پیچیده فعال‌سازی را دارد، ولی درعین حال بسیار مهم و دقیق عملکرد پایداری عمودپرواز را انجام می‌دهد [۶،۷] و ساختار کنترل‌های فازی که دقت و صحت کنترل‌گرهای مدل قبلی را ندارد ولی پیاده‌سازی ساده‌تری در سیستم دارد که با قانون‌های بهتر و بهینه‌تر توانسته است که عملکرد قابل قبولی را ارائه دهد [۸،۹]. روش‌های کنترل‌کننده غیرخطی مثل خطی‌سازی فیدبک که در برابر اغتشاشات عملکرد خوبی را دارد ولی در مورد پایداری عملکرد ضعیف‌تری را نشان می‌دهد [۱۰،۱۱]. کنترل‌کننده‌های مد لغزشی که توسط بنارد ارائه شد ولی در مورد حرکت طولی و عرضی عمودپرواز نظری نداد و فقط در مورد پایداری سیستم کارایی خود را نشان می‌دهد [۱۲،۱۳]. همان‌طور که قبلاً گفته شد حس‌گرها و کنترل پرواز عمودپرواز نویز و خطای زیادی را می‌تواند در سیستم القا کند. با استفاده از فیلتر کالمن این اغتشاشات را به طرز مطلوبی کم می‌کند [۱۴]. به تدریج عمودپروازهایی مورد توجه قرار گرفت که بتوان به‌صورت هوشمند به‌طور کامل مأموریت‌های خود را انجام دهند برای انجام این مهم نیازمند درک بهتر نسبت به آئرو دینامیک سیستم می‌باشیم که چه در مورد حرکت پرنده و چه در مورد جریان‌ات هوا داشته باشیم [۱۵-۱۸]. در مرجع شماره [۱۹،۲۰] مدل کنترلی بر مبنای شبکه عصبی برای عمودپروازهای از نوع کوچک و میکرو طراحی شده است. ساختار کنترلی این عمودپروازها فرقی با مدل‌های معمول ندارد ولی این پرنده‌های به‌خاطر فیزیک مشخصی که دارند

تحت اغتشاشات هوایی زیادی قرار می‌گیرند. در این مدل از شبکه‌های عصبی مرتبه‌ای استفاده شده است. به‌گونه‌ای که اولین هدف ثابت نگه‌داشتن عمودپرواز بود و در مرحله دوم حرکت آن در راستای محورهای مختصات بدون خطا بود. در نهایت تستی که بر روی این مدل انجام شد ثابت شد که نسبت به کنترل کننده PID ۵ برابر در مقابل اغتشاشات هوایی مقاوم‌تر است. همچنین خطای عملگرها ۸ برابر کاهش یافته است. در مرجع شماره [۲۱] تلاش شده تا برای یک مدل عمودپرواز مشخص کنترل پروازی طراحی شود که به کمک گونه‌ای از شبکه عصبی مصنوعی بتوان عامل‌های کنترلی را در آن اجرا کرد. باور نویسندگان این بود که برای رسیدن به یک عمودپرواز به‌طور کامل خودکار راه بسیار زیادی وجود دارد و دلیل این امر یکی ناپایداری ذاتی سیستم و پیچیده بودن کنترل آن و دیگری اغتشاشات مکرر هوایی بر روی سیستم است. شبکه عصبی درون سیستم بر اساس مدل مشخصی طراحی نشد و نوع آموزش شبکه عصبی تابع‌های آموزشی است و نه بر اساس حدس و پیش‌بینی درون الگوریتم‌ها. هدف کنترل در این سیستم علاوه بر عامل‌های کنترلی رول، پیچ و یاو این قصد را دارد که میزان ارتفاع پرنده را هم مورد تحلیل قرار داده و به‌گونه‌ای آموزش داده است که پس از دادن فرمان ارتفاع ثابت دیگر پرنده به‌صورت افقی پرواز می‌کند. آموزش دادن عمودپرواز و خودکار کردن عمودپرواز بر ۴ اصل گرفتن اطلاعات، آموزش دادن این اطلاعات در رایانه، سوار بر روی عمودپرواز و انتقال این داده‌ها در درون شبکه عصبی استوار است. کنترل‌گرهای متعددی به‌منظور پایدارسازی عمودپروازها ارائه شده است که کنترل‌های تلفیقی بهترین عملکرد را از خود تابه‌حال نشان داده است در این پژوهش این نوع کنترل‌گر بر مبنای طرحی جدید که مزایای قابل توجه آن نسبت به مدل‌های قبلی سادگی پیاده‌سازی و ارزان بودن کنترل پرواز ارائه شده نسبت به مدل خارجی که بر مبنای آن آموزش دیده، دارد [۲۴]. کنترل عمودپرواز از دیرباز به دلیل پیچیدگی ذاتی این سیستم بسیار مهم بوده است. و همین‌طور ارائه کنترلی که بتواند به‌طور کامل این پرنده را متعادل سازد، امری بسیار مهم و قابل تأمل به شمار می‌رود. مهم‌ترین نوآوری این پژوهش ساخت کامل یک کنترل کننده پرواز کوادروتور که به روش کنترل تلفیقی بر مبنای شبکه عصبی و کنترل کننده PD کار می‌کند و توانایی مشابه با مدل کنترل پرواز نازا، پرفرودارترین کنترل پرواز موجود در بازار، را دارد. اعمال شبیه‌سازی کامپیوتری به‌طور کامل برای سیستم کوادروتور با مدل سیستم کنترل‌های تلفیقی با تابع هدف بازگشت به عقب می‌باشد که قبلاً به این شکل انجام نشده و به‌طور کامل با موفقیت همراه بود. عموم کنترل‌های موجود یا ساختار کنترلی بسیار پیچیده‌ای دارند نظیر کنترل‌های بازگشت به عقب و یا مد لغزشی دارند که کنترل کننده‌های خوبی برای تعادل کوادروتورها هستند ولی برای هوشمند سازی و یا ماموریت های خودکار مناسب نیستند اما کنترل حاضر که تلفیقی از دو کنترل کننده می باشد ساختار دستوری ساده‌تری دارد و البته پیاده‌سازی راحت‌تری دارد.

ساختار پژوهش بدین گونه است که به چگونگی تحلیل دینامیکی عمودپرواز در بخش دوم می‌پردازیم و پس از آن شبیه‌سازی کامپیوتری عمودپرواز را نشان خواهیم داد. در بخش چهارم کنترل کننده بازگشت به عقب طراحی شده و آن را با مشتق‌گیر- تناسبی مقایسه می‌کنیم و پس از آن شبکه عصبی را با این داده‌ها مورد آموزش قرار می‌دهیم و نتایج در قالب نمودارهایی تحلیل می‌گردند و در انتها شبکه عصبی در درون یک مدل عمودپرواز واقعی دوباره طراحی شده و شبکه عصبی آن با مدل کنترل پرواز نازا که تابع آن با برد طراحی شده به سیستم داده شد، آموزش داده می‌شود و کنترل پرواز نهایی به‌طور کامل بر ربات پیاده می‌شود.

۲- ارائه مدل دینامیکی عمودپرواز

تعداد عملگرهای عمودپروازها ۴ عدد می‌باشد ولی درجات آزادی سیستم ۶ درجه است. پس عمودپروازهای ۴ موتور به جزو سیستم‌های کم عملگر تقسیم‌بندی می‌شوند. در سال‌های اخیر عمودپروازهای با ۶ موتور هم بسیار مورد توجه قرار گرفته است به گونه‌ای که بسیاری از شرکت‌ها استفاده از آن را در دستورکار خود قرار داده اند. سیستم‌های کم عملگر کنترل‌کننده‌های پیچیده‌ای دارند و کنترل سیستم در آن‌ها مشکل است، اما از طرف دیگر عامل‌های کنترل که باعث تغییر در سیستم می‌شود کمتر شده و در شرایط معین کنترل سیستم‌ها با تعداد کمتری عملگر انجام می‌شود.

به منظور به دست آوردن مدل دینامیکی سیستم به صورت مدل فضای حالت متغیرهای حالت را به صورت $\dot{X} = f(X, U)$ منظور می‌کنیم که U بردار ورودی و X بردار متغیرهای حالت است که X و Z وق مرکز جرم در سازه و ϕ و θ و ψ زوایای اوایلر هستند و به صورت زیر تعریف می‌شوند [۲۲]:

$$X = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}] \quad (1)$$

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4]$$

که ورودی‌های کنترلی به صورت زیر تعریف می‌شوند:

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

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

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

$$U_4 = (-1)^i \sum_{i=1}^4 M_{D_i} = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$$

در معادلات بالا b ضریب رانشی و ضریب پسا است. Γ آن تراست موتورها است، Ω سرعت چرخش موتور و M گشتاور حاصله می‌باشد.

معادلات حرکتی برگرفته شده از مدل دینامیکی کوادروتور به صورت زیر هستند:

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

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

$$\ddot{z} = g - (\cos \varphi \cos \theta) \frac{U_1}{m}$$

$$\dot{\varphi} = \dot{\theta} \dot{\psi} \left[\frac{I_{yy} - I_{zz}}{I_{xx}} \right] + \frac{J_r}{I_{xx}} \dot{\theta} \Omega_r + \frac{l}{I_{xx}} U_2$$

$$\ddot{\theta} = \dot{\theta} \dot{\psi} \left[\frac{I_{zz} - I_{xx}}{I_{yy}} \right] + \frac{J_r}{I_{yy}} \dot{\varphi} \Omega_r + \frac{l}{I_{yy}} U_3$$

$$\ddot{\psi} = \dot{\theta} \dot{\varphi} \left[\frac{I_{xx} - I_{yy}}{I_{zz}} \right] + \frac{1}{I_{yy}} U_4$$

متغیرهای mI_x و I_y و I_z جرم و مان اینرسی کوادروتور است. L ف ل این موتورها و مرکز جرم است و J_R و Ω_R مان این و سرعت زاویه‌ای تیغه‌های پرواز هستند. U_1 و U_2 و U_3 و U_4 ورودی‌های کنترلی سیستم و به ترتیب نیروی مجموع و ممانهای چرخش حول محور Y ، چرخش حول محور X و چرخش حول محور Z تولید شده توسط تیغه‌های پروازی می‌باشند.

۳- طراحی و شبیه‌سازی کنترل کلاسیک مشتق‌گیر-تناسبی

یکی از پرترفدارترین کنترل‌کننده‌های موجود در صنعت کنترل‌کننده مشتق‌گیر تناسبی است که دلیل این پراستفاده بودن سادگی محاسبه و پیاده‌سازی رابطه کنترل‌کننده است. ورودی‌های کنترلی در این نوع کنترل‌کننده به صورت روابط زیر است. در این روابط ثوابت با k نشان داده شده است.

$$U_1 = \frac{m(g + k_{pz} E_z + k_{dz} (-z))}{\cos \varphi \cos \theta}$$

$$U_2 = (k_{pp} E_{\varphi} + k_{dp} (-\dot{\varphi}))$$

(۳)

$$U_3 = (k_{pt} E_{\theta} + k_{dt} (-\dot{\theta}))$$

$$U_4 = (k_{ps} E_{\psi} + k_{ds} (-\dot{\psi}))$$

در معادلات بالا E همان خطای متغیر است. عامل اول معادلات بالا تغییراتی را در ساختار سنتی کنترل‌کننده‌های مشتق‌گیر تناسبی دارد که در مرجع [۲۳] چگونگی افزایش چشم‌گیر دقت کنترل‌کننده، فقط برای عمودپروازها ذکر شده است. ضرایب کنترل تناسبی مشتق‌گیر با اعدادی که به صورت تجربی بهترین عملکرد را در سیستم داشته‌اند جایگزین شده‌اند. که برای $kpp, kdp, kpt, kdt, kps, kds, kpz, kdz$ به ترتیب $۰/۴, ۱/۲, ۰/۴, ۰/۸, ۰/۸, ۰/۴, ۰/۴, ۰/۴$ و ۱ مقادیر می‌باشند.

در مورد تداخل کانال‌ها لازم به ذکر است که زاویه چرخش حول محور عمودی مستقل از بقیه زاویه‌ها عمل نمی‌کند و برای پایداری هر پالسی با تغییری که در بقیه متغیرها هم انجام می‌شود کاملاً مورد تحلیل قرار گرفته است.

با توجه به مقاله‌های گذشته و آزمون‌وخطا و همین‌طور بلوک بهینه‌ساز متلب ضرایب کنترل‌کننده مشتق‌گیر تناسبی انتخاب شد و با توجه به اینکه پارامترهای کوادروتور ساخته‌شده توسط مؤلف خصوصیات مثل وزن و زاویه حمله پروانه و... منحصر به فردی نسبت به بقیه کوادروتورها دارد بنابراین نمی‌توانیم از داده‌های پژوهش‌های پیشین به طور کامل استفاده کنیم ولی عامل‌هایی مثل وزن و... اختلاف کمی با کوادروتورهای معمول دارد به همین خاطر با آزمون‌وخطا سعی بر این شد که بهترین ضرایب را استفاده کنیم. اما برای اینکه بتوانیم تحلیل و طراحی به کمک شبکه عصبی به خوبی انجام شود نیازمند یک نوع کنترل‌کننده دیگر هستیم. که در اینجا کنترل بازگشت به عقب انتخاب می‌شود.

۴- کنترل‌کننده بازگشت به عقب

به منظور طراحی کنترل‌کننده بازگشت به عقب ۳ با کمک گرفتن از معادلات به دست آمده از معادلات تحلیل دینامیکی به طراحی کنترل‌کننده پرداخته شد. مرحله نخست در طراحی کنترل‌کننده بازگشت به عقب در کوادروتور محاسبه خطای تعقیب و معادلات آن به صورت زیر است:

$$z_1 = x_{1d} - x_1 \quad (۴)$$

اکنون با استفاده از تئوری لیاپانوف مقدار z_2 را که مقدار مثبت دارد را در تابع وابسته به زمان v محاسبه می‌کنیم.

$$v(z_1) = \frac{1}{2} z_1^2 \quad (۵)$$

بر حسب زمان از تابع بالا مشتق می‌گیریم:

$$\dot{v}(z_1) = z_1(\dot{x}_{1d} - \dot{x}_2) \quad (۶)$$

پایداری متغیر z_1 وابسته به معرفی متغیر مجازی x_2 که این متغیر به شرح زیر به دست می‌آید:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \quad (7)$$

معادله تبدیل خواهد شد به:

$$\dot{v}(z_1) = -\alpha_1 z_1^2 \quad (8)$$

متغیر را طبق روش بازگشت به عقب تغییر می‌دهیم:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \quad (9)$$

دوباره تابع لیاپانوف را برای گام دوم محاسبه می‌کنیم و مشتق می‌گیریم:

$$\dot{v}(z_1, z_2) = z_2(a_1 \dot{\theta}\dot{\psi} + a_2 \dot{\theta}\dot{\Omega}_r + b_1 u_2) - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2 \quad (10)$$

اکنون با توجه به منفی بودن تابع بالا و کران دار بودن تابع تمام شروط تابع لیاپانوف انجام شده است و عامل‌های نیروی درون سیستم شبیه‌سازی به شرح زیر به دست می‌آید.

$$U_2 = \frac{1}{b_1} (z_1 - a_1 \dot{\theta}\dot{\psi} - a_2 \dot{\theta}\dot{\Omega}_r - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \quad (11)$$

به همین ترتیب کنترل برای پیچ و یابو به صورت زیر محاسبه می‌شوند.

$$U_3 = \frac{1}{b_2} [z_3 - a_3 \dot{\phi}\dot{\psi} - a_4 \dot{\phi}\dot{\Omega}_r - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4] \quad (12)$$

$$U_4 = \frac{1}{b_3} [z_5 - a_5 \dot{\phi}\dot{\theta} - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6] \quad (13)$$

که در آن Z برای هر یک از متغیرهای مطرح شده می‌باشد که در زیر به آن اشاره می‌کنیم:

$$z_3 = \theta_d - \theta$$

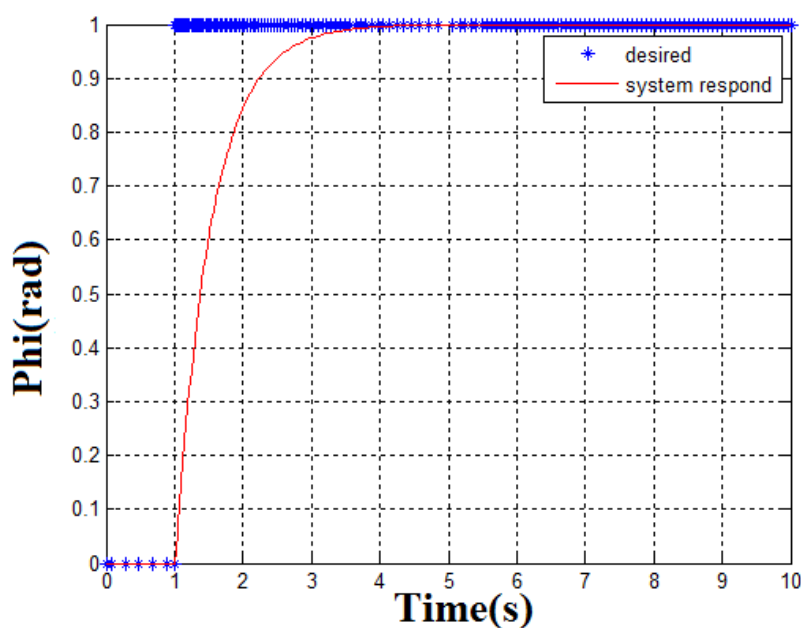
$$z_4 = \dot{\theta} - \dot{\theta}_d - \alpha_3 z_3$$

$$z_5 = \psi_d - \psi$$

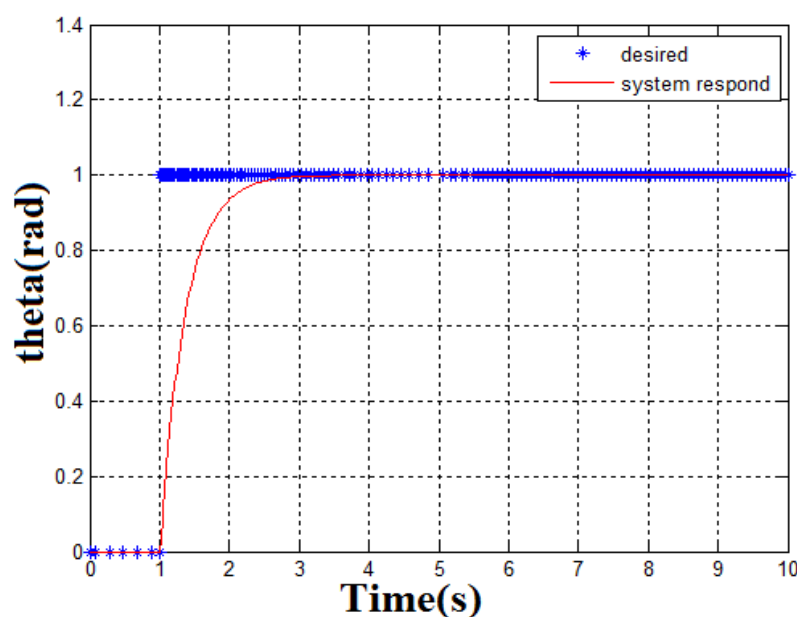
$$z_6 = \dot{\psi} - \dot{\psi}_d - \alpha_5 z_5$$

ثوابت مورد استفاده در معادلات بالا که با توجه به اینکه این ثوابت به خصوصیات ذاتی کوادروتور وابسته هستند و نمی‌توان به صورت عمومی این ثوابت را مقدار مشخصی داد ولی برای بهینه کردن این ضرایب به وسیله بلوک بهینه‌ساز متلب به صورت کلی مقدار مشخص می‌گیریم و پس از آن با آزمون و خطا به صورت دستی آن را بهبود می‌دهیم. این مقادیر برای الفا ۱ تا ۶ به ترتیب ۱۱-۳/۵-۹/۵-۴-۲/۵-۲/۱ می‌باشند.

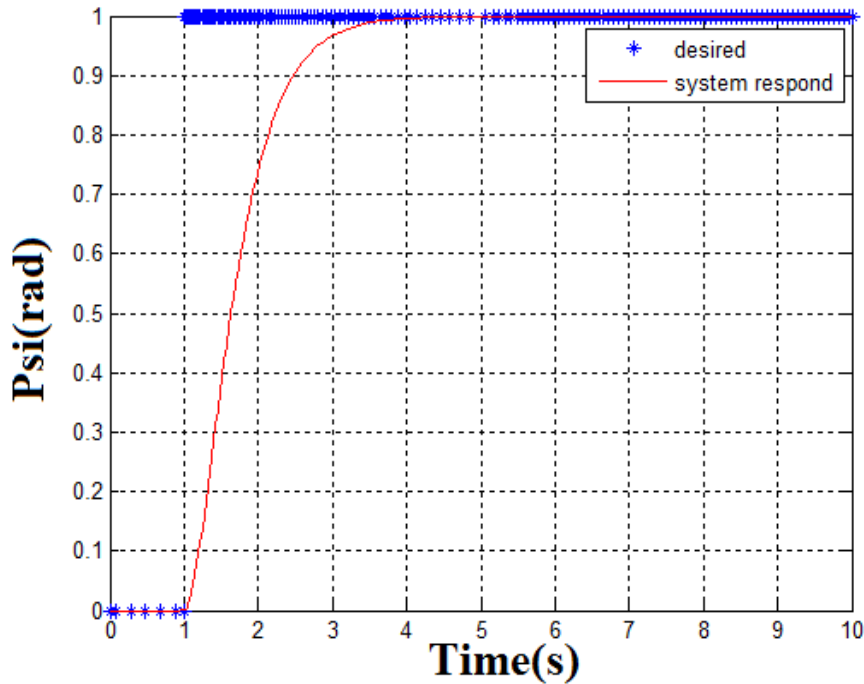
شبیه‌سازی این کنترل کننده در محیط نرم‌افزاری MATLAB انجام شده و همان‌طور که مشخص است کاملاً در کنترل وضعیت عمودپرواز موفق بوده است. نتایج این شبیه‌سازی کنترل کننده در شکل‌های (۲) تا (۴) برای هر زاویه به تصویر درآمده است.



شکل ۲- شبیه‌سازی کنترل کننده در محیط نرم‌افزاری برای زاویه ϕ



شکل ۳- شبیه‌سازی کنترل کننده در محیط نرم‌افزاری برای زاویه θ



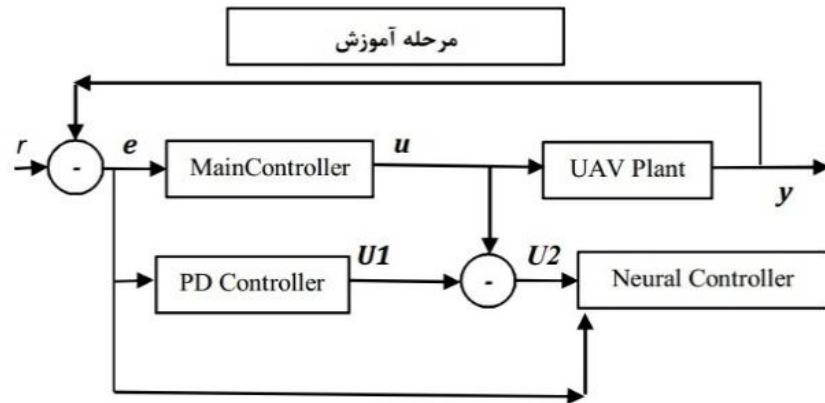
شکل ۴- شبیه‌سازی کنترل کننده در محیط نرم‌افزاری برای زاویه ψ

میزان خطای حالت ماندگار سیستم که بعد از ۱۰ مرحله محاسبه شده مقدار بسیار ناچیزی هست، به طوری که در انتها به چند صدم درصد حالت ایده آل در آمده است. اما ثابت زمانی که سیستم از خود نشان داد و در واقع همان سرعت پاسخ است، در ثانیه ۴,۵ از ابتدای شبیه‌سازی به ۹۰ درصد مقدار مطلوب رسید که مشخصاً نسبت به کنترل کننده مشتق گیر-تناسبی از هر دو نظر بهتر عمل کرده است و اختلاف چشم‌گیری بین دو مقدار ایده آل و مقدار از کنترل حاصل شده دیده نمی‌شود.

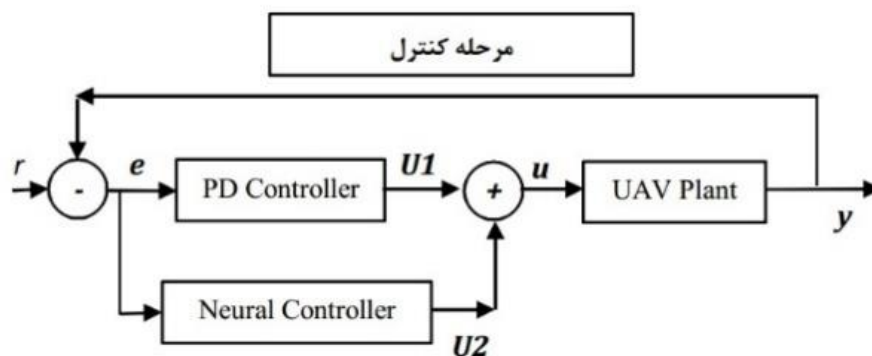
با توجه به نمودارهای به دست آمده و دقت قابل قبولی که کنترل کننده بازگشت به عقب برای سیستم ما ارائه داده به معرفی سیستمی با شبکه عصبی به همراه کنترل کننده مشتق گیر-تناسبی می‌پردازیم [۲۱].

۵- مقایسه و طراحی شبکه عصبی

در این قسمت به نحوه طراحی و مقایسه شبکه عصبی کنترلی طراحی شده می‌پردازیم. اساس طراحی شبکه عصبی در مرحله آموزش در شکل (۵) نمایش داده شده است. در مرحله آموزش به کمک یک کنترل کننده اصلی همانند بازگشت به عقب و یک کنترل کننده مشتق گیر تناسبی همزمان سیستم ربات پرنده را کنترل می‌کنیم. در هر لحظه از زمان، خطای بازخورد پرنده و اختلاف فرمان کنترلی بین دو کنترل کننده را به عنوان داده آموزشی به شبکه عصبی می‌دهیم تا با این دو آموزش بگیرد. شبکه عصبی در قالب دو تابع که هر کدام حاوی ماتریس‌های استخراج شده از شبیه‌سازی کامپیوتری هستند آموزش می‌بیند که این ماتریس‌ها مقادیر مورد نظر را بر حسب زمان به ما نشان می‌دهند یکی از این ماتریس‌ها تابع ورودی شبکه عصبی و یکی دیگر ماتریس هدف شبکه عصبی می‌شوند بدین صورت با توابع پیش فرض متلب که در زیر این توابع را مشاهده می‌کنید، آموزش می‌بینند و در نهایت شبکه عصبی با الگو قرار دادن کنترل کننده قبلی سعی به نزدیکی به تابع هدف دارد.



شکل ۵- شیوه آموزشی کنترل کننده عصبی به کمک کنترل کننده مشتق گیر تناسبی

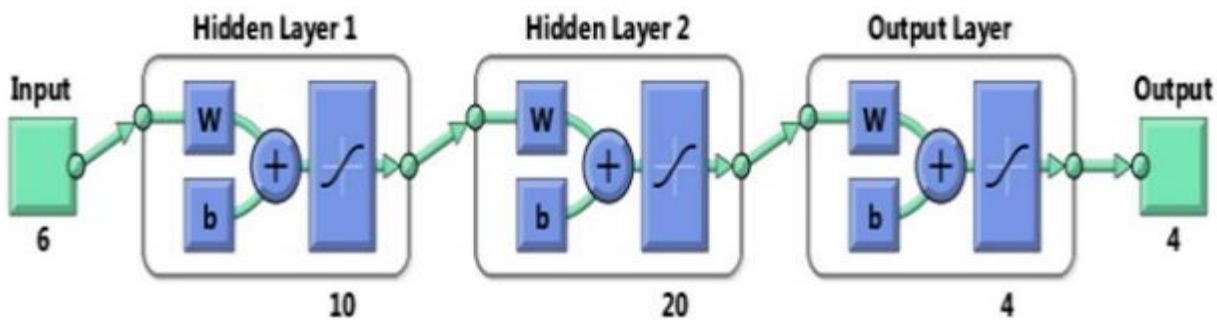


شکل ۶- شیوه کنترل ترکیبی با کنترل کننده عصبی و کننده مشتق گیر تناسبی

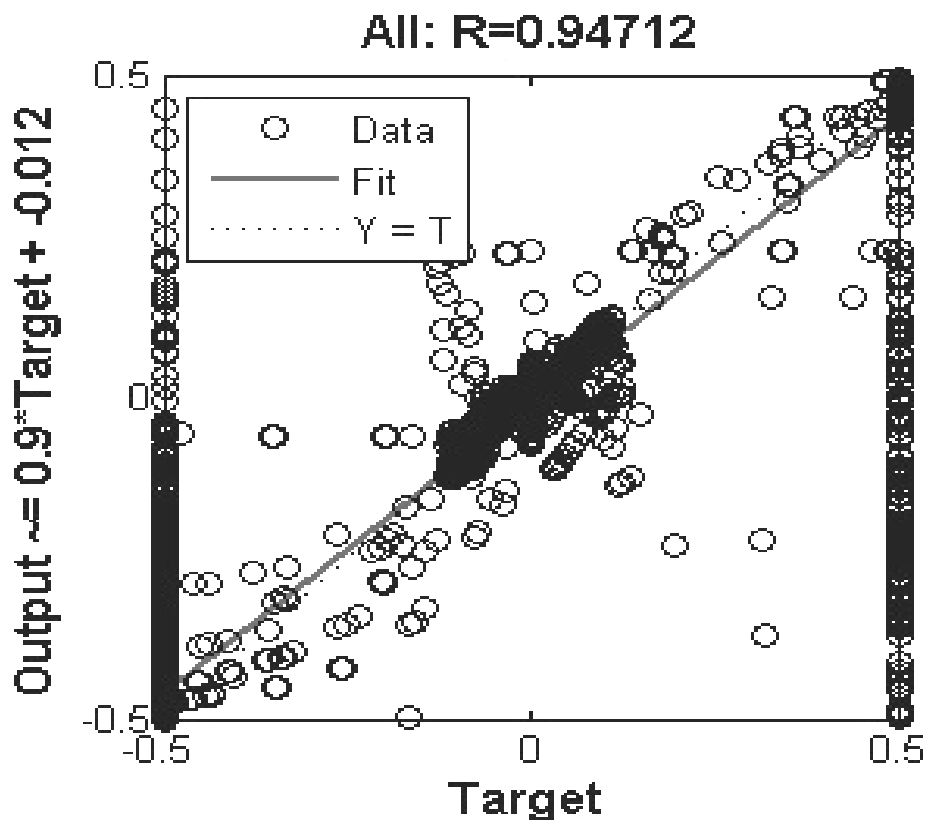
در مرحله کنترل با شبکه عصبی خروجی شبکه عصبی به عنوان اصلاح کننده کنترلی با خروجی روش مشتق گیر تناسبی جمع شده و به ربات پرنده به عنوان فرمان کنترلی ارسال می شود (شکل ۶). به کمک این روش کنترلی مزایای کنترل کننده اصلی را در شبکه عصبی به دست خواهیم آورد.

نمونه اول معماری شبکه عصبی مورد استفاده یک شبکه سه لایه است که نرون های هر لایه باهم در ارتباط هستند. این شبکه از نوع پیش خور ۴ است که در مورد شبکه هایی با پیچیدگی زیاد و پیش بینی ارتباطات کنترلی عملکرد بهتری را نسبت به بقیه انواع هم رده خود در شبکه های عصبی نشان داده است. در هر شبکه یک نرون می تواند با بیش از یک لایه ارتباط برقرار کند. میزان تعداد لایه های هر شبکه عصبی می تواند در ارتباط با دقت شبکه باشد اما اگر تعداد لایه ها را به شکل افراطی زیاد کنیم با مشکل بیش تمرینی شبکه و حجم زیاد آن مواجه می شویم. مقدار خروجی از هر یک از کنترل کننده ها در محیط شبیه سازی باهم مقایسه می شود و بعد از گذشتن از یک فیلتر حالت اشباع در یک متغیر ذخیره می شود. همین متغیر به همراه ورودی به کنترل کننده ها تابع های هدف و ورودی در شبکه عصبی اعمال می شوند.

معماری شبکه عصبی همان طور که در شکل (۷) دیده می شود از یک لایه به لایه بعدی انتقال پیدا می کند. که با ۳ لایه مجزا طراحی شده است که هر کدام از این لایه ها به ترتیب ۴ و ۲۰ و ۱۰ نرون مجزا دارند چگونگی انتخاب این لایه ها بر این منوال است که لایه اول اگر n نرون داشته باشد لایه بعدی را معمولاً با دو برابر آن معماری می کنند.



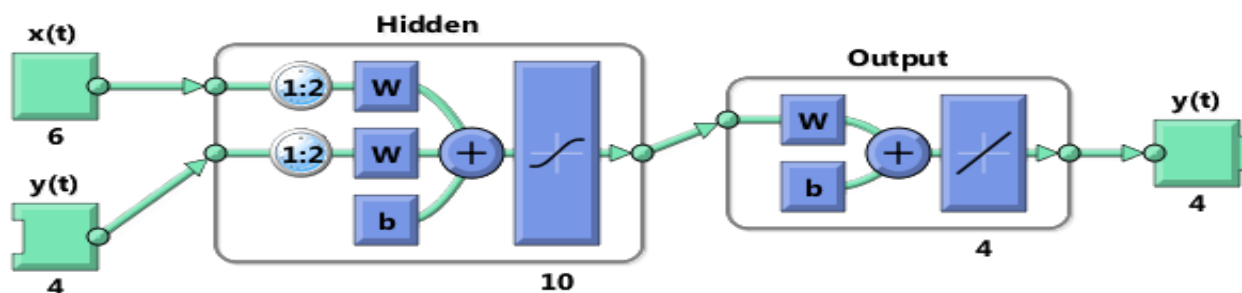
شکل ۷- لایه‌های شبکه عصبی



شکل ۸- پراکندگی داده‌ها در شبکه عصبی

معماری شبکه عصبی مورد استفاده پیش‌خور است. شکل (۸) میزان پراکندگی داده‌ها را نسبت به کنترل‌کننده‌ای که طراحی شده به خوبی نشان می‌دهد.

یکی از مهم‌ترین قسمت‌های هر شبکه عصبی وزن‌های آن به همراه مقادیری که به آن تحت عنوان بایاس اضافه می‌شوند. مقدار وزن‌ها و بایاس‌ها توسط نرم‌افزار متلب استخراج شده و وارد تابع فعال‌ساز سیگموئید می‌شود. و اما گونه‌های دیگر از معماری شبکه‌های عصبی در اینجا استفاده شده است که با نام نارس ۵ شناخته می‌شود. مزیت این نوع شبکه نسبت به نوع قبلی رسیدن بهتر به پایداری در خصوص سیستم‌های دینامیک محور می‌باشد. این نوع شبکه از نوع شبکه‌های بازگشتی است. در شکل (۹) معماری این شبکه مورد استفاده نمایش داده شده است.



شکل ۹- ساختار شبکه عصبی نارکس مورد استفاده

نتایج اعتبار سنجی شبکه نارکس نشان می‌دهد عملکرد کنترل‌کننده بر مبنای نارکس بهتر از شبکه‌های پیش‌خور است. این نکته حاکی از درستی پیش‌بینی در مورد رابطه شبکه عصبی نارکس و سیستم‌های دینامیکی است. میزان نزدیکی شبکه عصبی به پراکندگی داده‌ها در سیستم، به‌طور تقریبی در هر دو روش یک نوع نتیجه را داشتند. و البته این نزدیکی با داده‌های هدف بسیار مهم و در صحت کنترل‌کننده اهمیت دارد.

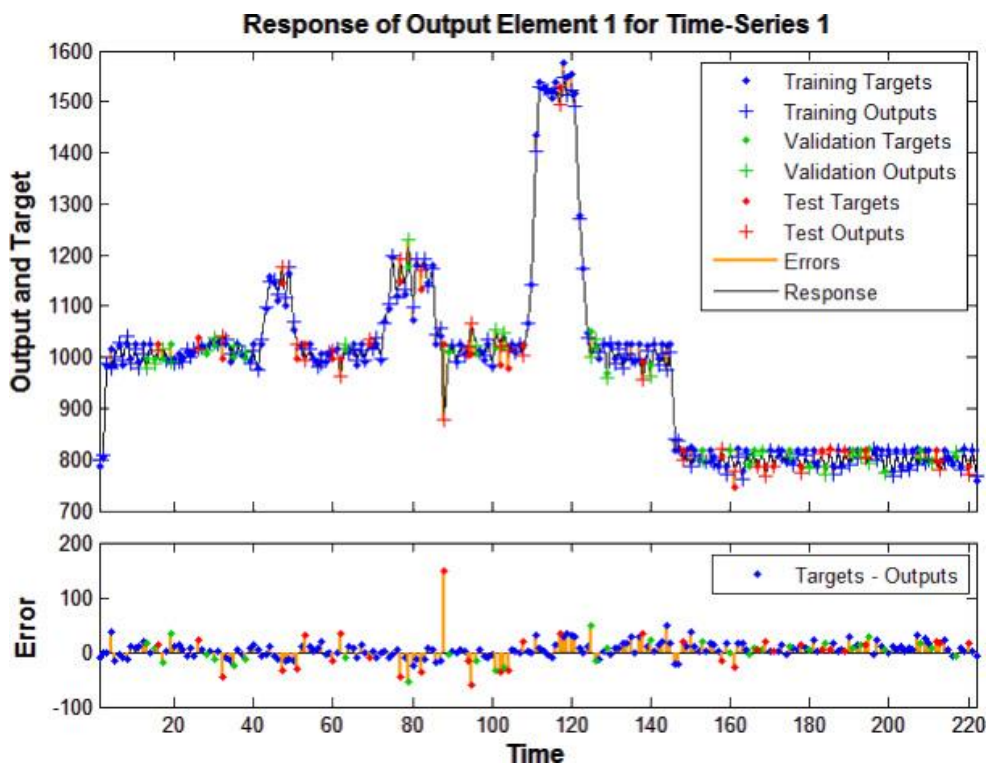
۶- نتایج به‌دست‌آمده برای شبکه عصبی و کنترل‌کننده مشتق‌گیر-تناسبی

در این قسمت مراحل تلفیق کلی دو نوع کنترل‌کننده مشتق‌گیر-تناسبی و شبکه عصبی شرح داده می‌شود. ساختار کلی بدین شکل است که میزان اختلافی که کنترل‌کننده مشتق‌گیر-تناسبی نسبت به بازگشت به عقب دارد محاسبه‌شده و مقدار تابع هدف را همین اختلاف تعیین می‌کنیم. تابع ورودی هم همانند حالت عادی، تمام ورودی‌ها به سیستم کنترل‌کننده است. قابل‌ذکر است که برای آموزش بهتر شبکه عصبی باید به‌طور کامل از تمامی توابع (مثل تابع سینوسی رمپ و یا سیگنال ساده ضربه) استفاده شود. پس از انجام این کار تابعی به بزرگی ۷۱۹۶ داده برای هر متغیر به دست آمد که برای آموزش شبکه عصبی مناسب است. تمام تفاضل بین کنترل‌کننده تناسبی اشتقاقی و بازگشت به عقب در محیط سیمولینک متلب محاسبه شده است. تعدادی تابع اشباع در انتهای هر تفاضل سیگنالی گذاشته شد که داده‌های با خطای زیاد را از سیستم حذف می‌نماید. همان‌طور که قبلاً گفته شد حاصل این عملیات تابع هدف خواهد بود و ورودی‌های متنوع به‌عنوان ورودی ثبت می‌شوند. نتایج به‌دست‌آمده در هر قالب در زیر به‌طور کامل مشخص شده است و حاکی از این دارد که کنترل‌کننده وظیفه تعادل عمودپرواز را به‌خوبی انجام می‌دهد. نتایج شبیه‌سازی در شکل‌های (۱۰) تا (۱۲) آورده شده است.

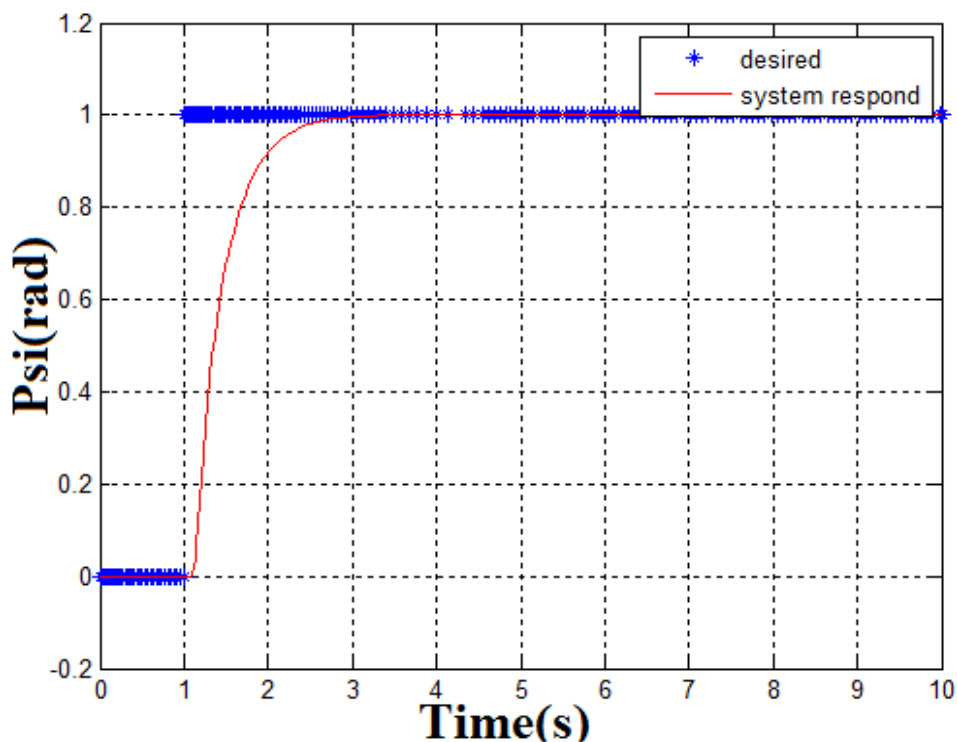
همان‌طور که در نمودارها دیده می‌شود، در چرخش‌ها حول سه محور خطای ماندگار حدود ۵ درصد تا مقدار مطلوب به چشم می‌خورد که نسبت به کنترل‌کننده بازگشت به عقب بیشتر است و اما سرعت رسیدن به ۹۰ درصد پاسخ کاملاً مشابه حالت بازگشت به عقب و در ثانیه ۴٫۵ شبیه‌سازی صورت گرفته است.

در ادامه به نتایجی که از آموزش سیستم عصبی خالص به دست آوردیم اشاره می‌کنیم. میزان خطایی که سیستم در این حالت نشان می‌دهد بسیار بالا است و همین‌طور برنامه متلب امکانی را در اختیار کاربر قرار می‌دهد که کنترل‌کننده طراحی‌شده را یک‌بار تست کند پس از انجام این عمل خطای سیستم تا مقدار حداکثر ۸۰۰ (در واحد PWM) رسید. که در حالت کنترل تلفیقی در نهایت به ۱۰۰ (PWM) رسیدیم (شکل ۱۰). این

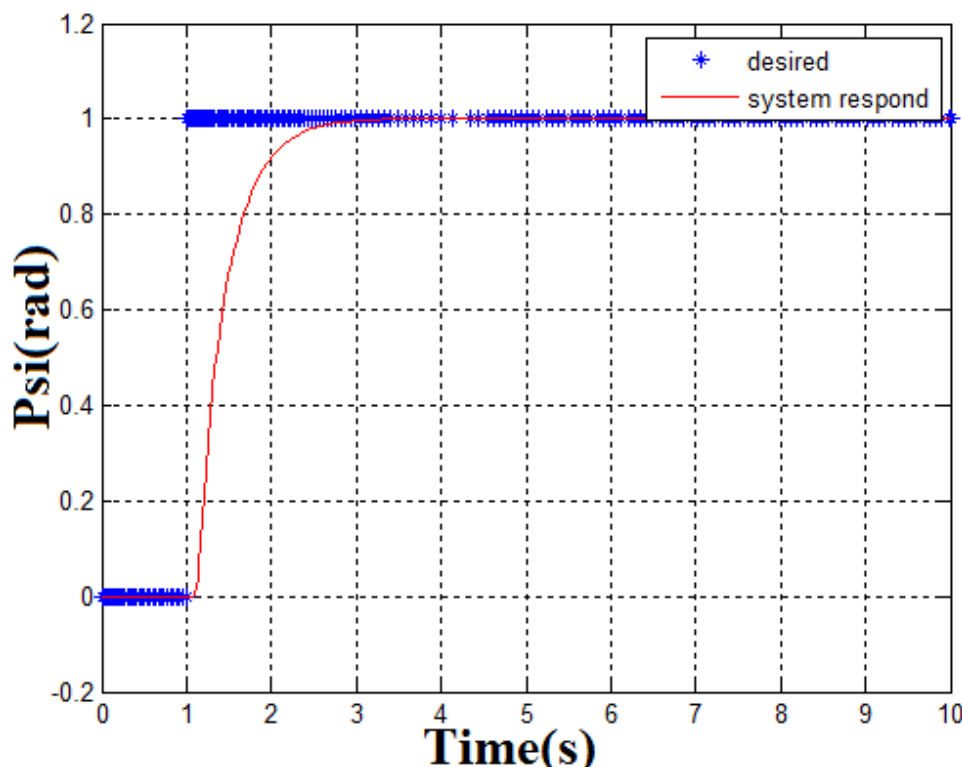
مقادیر ثابت می کند که کارای کنترل کننده تلفیقی بسیار بیشتر از کنترل شبکه عصبی به صورت واحد می باشد. با بررسی نتایج به دست آمده کیفیت و صحت کنترل کننده تلفیقی به طور کامل قابل اثبات است که نشان می دهد که این نوع کنترل کننده قابل استفاده مطلوب برای عمودپرواز است.



شکل ۱۰- چگونگی کنترل زاویه ϕ



شکل ۱۱- چگونگی کنترل زاویه θ



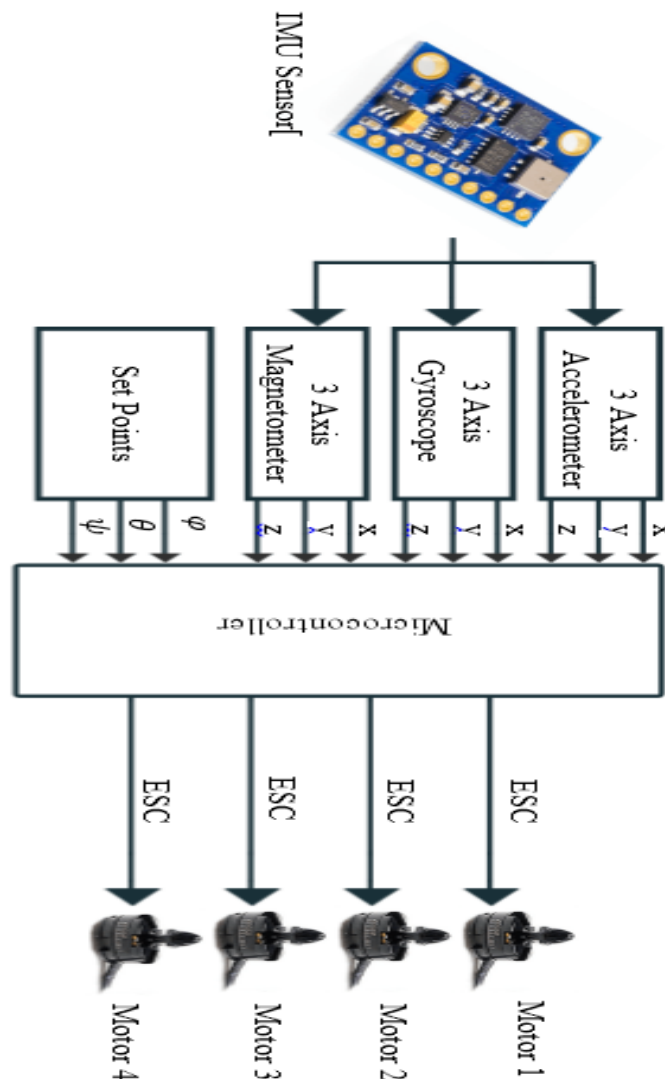
شکل ۱۲- چگونگی زاویه ψ

۷- گرفتن داده‌های کنترل پرواز واقعی برای تعلیم شبکه عصبی

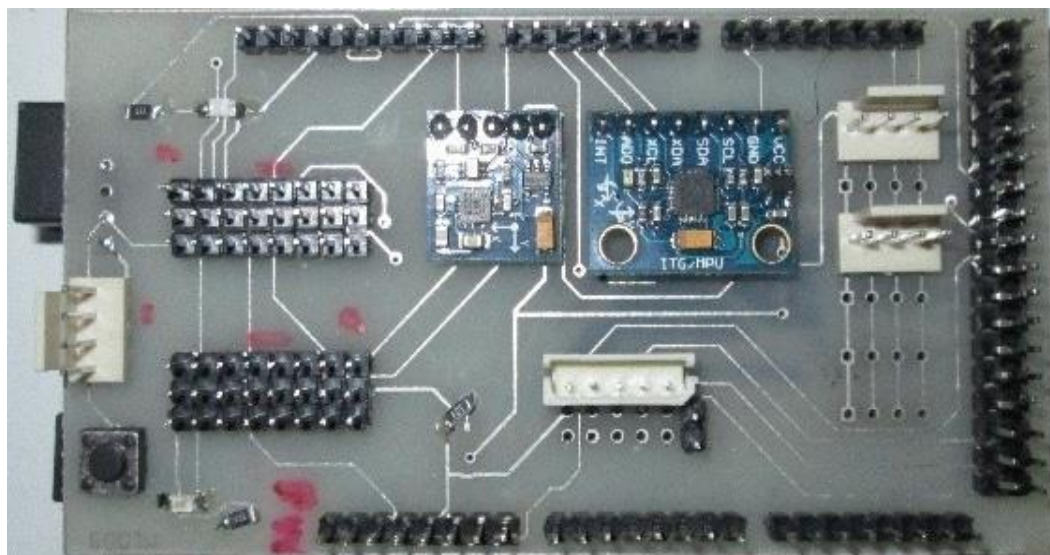
پس از به نتیجه رسیدن کامل طراحی و آزمایش کردن شبکه عصبی در محیط شبیه‌سازی کامپیوتری این عملیات در محیط واقعی و با ربات عمودپرواز به صورت عملی مورد تحلیل قرار می‌گیرد. برای این کار نیازمند دستگاهی هستیم که بتواند داده‌های ورودی و خروجی به کنترل پرواز را به طور کامل ذخیره‌سازی کند. بدین منظور از یک برد کامل طراحی شده با آردوینو استفاده می‌شود.

در این برد خودکار مجموعه‌ای از حسگرها و مدارات مجتمع الکترونیکی قرار داده شده است که وظیفه کنترل درایو موتورها و مجموعه‌ای از وظایف کنترلی دیگر از جمله پایدارسازی پهپاد و تعقیب نقاط تنظیم را بر عهده دارد. مهم‌ترین حسگر این لایه از ترکیبی از شتاب سنچ، جایرو و قطب‌نما ساخته شده است که IMU نامیده می‌شود. حسگر IMU زوایای مختلف و سرعت‌های دورانی پرنده در جهات گوناگون را اندازه‌گیری می‌کند. این سیستم از یک پردازنده مستقل و تجهیزات لازم برای اجرای الگوریتم کنترلی بهره می‌برد و توسط کابل USB به رایانه متصل و با کمک زبان سی برنامه‌نویسی و با زبان آردوینو وارد برد می‌شود. این برد قابلیت تجهیز به حسگر مکان‌یابی جهت بازگرداندن مستقل پرنده به محل شروع پرواز را داراست. همان‌گونه که در شکل (۱۳) نشان داده شده است میکرو کنترلر سیگنال آنالوگ را از شتاب سنچ، ژيروسکوپ و قطب‌نما می‌خواند. این سیگنال به مقادیر دیجیتال تبدیل می‌شود تا بتواند در میکروکنترلر مورد استفاده قرار بگیرد. میکروکنترلر با توجه به مقادیر خوانده شده و ورودی مطلوب منظور شده برای کوادروتور برای چهار موتور سیگنال PWM تولید می‌کند و به این ترتیب سرعت چرخش هر پيشران را معین می‌کند. همان‌گونه که در شکل (۱۳) هم مشخص است سیگنال PWM تولید شده توسط میکروکنترلر وارد کنترل کننده‌های سرعت (ESC) می‌شود و در

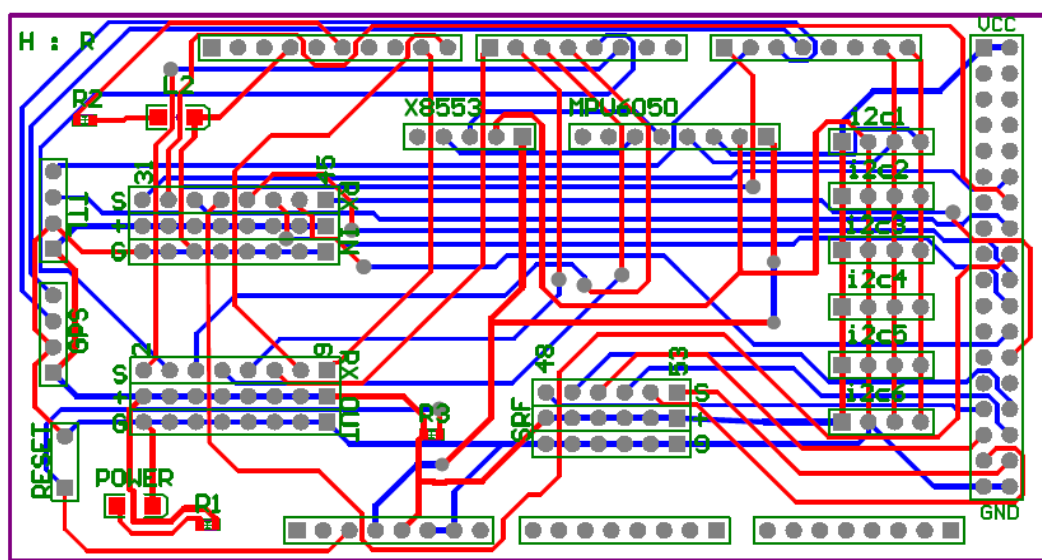
خروجی سرعت موتورهای را تعیین می‌کند. مقادیر شتاب سنج و ژيروسکوپ از یک حسگر IMU مبتنی بر حسگرهای بسیار کوچک به دست می‌آیند. با توجه به اینکه حسگر شتاب‌سنج موقعیت زاویه‌ای را با منظور کردن کسینوس زاویه شتاب خطی و شتاب زمین محاسبه می‌کند قادر به محاسبهٔ موقعیت زاویه در راستای یابو نیست، بدین منظور از قطب‌نما برای محاسبه زاویه یابو استفاده شده است. ماژول MPU6050 در شکل (۱۳) برای محاسبه موقعیت زاویه‌ای با حسگرهای شتاب سنج، ژيروسکوپ و قطب‌نما مورد استفاده قرار گرفته است. به صورت کلی سه استراتژی برای ترکیب داده‌ها موجود است. اولین روش با عنوان روش اصلاح شناخته می‌شود، در این روش اطلاعات خروجی از یک حسگر برای اصلاح دیگری استفاده می‌شود. استراتژی دوم اختلاط نام دارد، که با ترکیب و اختلاط خروجی‌های چندین حسگر اقدام به تولید داده می‌کند. روش سوم و بهترین روش برای ترکیب اطلاعات حسگرها، ادغام است. در این روش مقادیر هر حسگر بامنظور کردن ضرایب وزنی و آماری مختلف با همدیگر ترکیب می‌شوند تا بهینه‌ترین پاسخ ممکن را در خروجی تولید کند. به صورت کلی الگوریتم‌های ادغام از فیلتر کالمن برای تشخیص سیگنال نویزی در طول زمان استفاده می‌کنند تا سیگنالی نزدیک به مقدار واقعی تولید شود. برد مدار کنترل در شکل‌های (۱۴) و (۱۵) به تصویر درآمده است.



شکل ۱۳- معماری داخلی خلبان خودکار ساخته شده



شکل ۱۴- خلبان خودکار طراحی شده با آردوینو



شکل ۱۵- برد پوسته خلبان خودکار طراحی شده در نرم افزار پروتل

این پوسته به صورت دوطرفه و با طراحی خروجی هایی که در آینده کاری این ربات مورد استفاده خواهد گرفت انجام شده است. این موارد شامل GPS و ماژول مافوق صوت و IMU می باشند.

برنامه نویسی بر مبنای نرم افزار آردوینو انجام شد که البته بسیار شبیه به نرم افزارهای متداول برنامه نویسی میکرو کنترلرها می باشد طراحی و نوشته شد تا بتواند تمام سیگنال های PWM که در رادیو کنترل کننده و در درون کنترل پرواز برای موتورها به وجود می آید را ذخیره کند و بدون تداخلی دوباره به هدفشان بفرستد تا از پرواز پرنده جلوگیری نشود، این داده ها که به دست می آید بعداً در طراحی شبکه عصبی نقش اساسی دارند.

برد ساخته شده که بر روی آردوینو سوار می شود می تواند جریان زیادی را القا کند. همین طور پدیده تشدید و خطای میکرو نکته قابل ذکر دیگر تأخیر دستورات در درون برنامه است که طبیعتاً خطای این کار را زیاد

می‌کند نوع شبکه عصبی مورد استفاده پیش‌خور و نوع نارس بود که هر دو نوع شبکه عصبی، امتحان شده است.

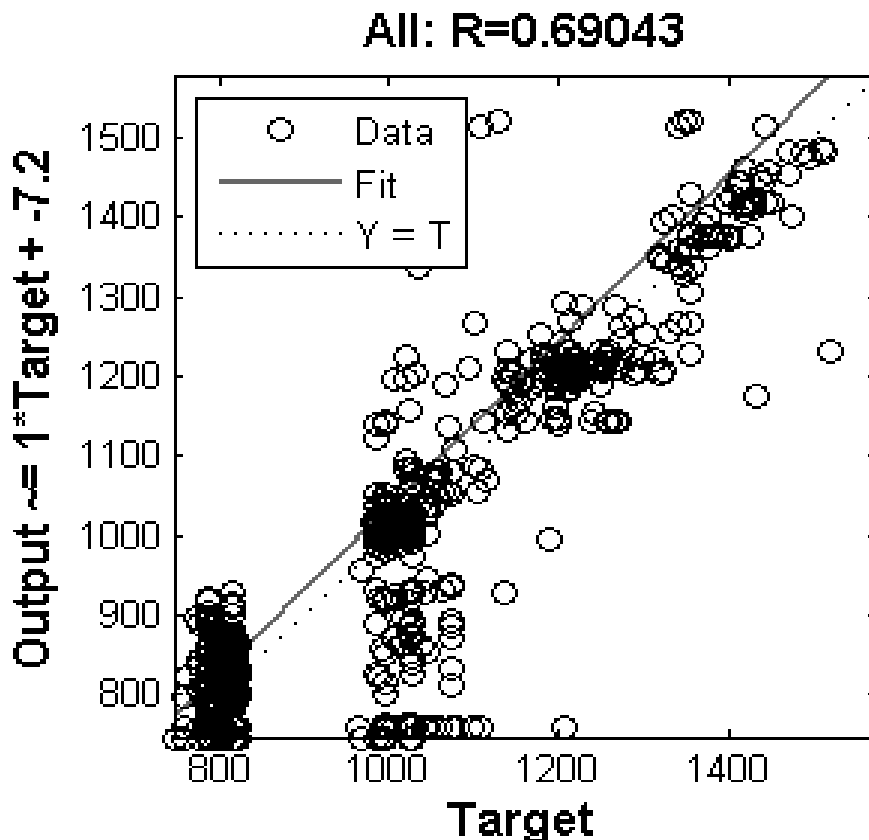
پس‌ازاینکه این اطلاعات از کنترل‌کننده استخراج شد و توانستیم این اطلاعات را به موتورها و کنترل پرواز برگردانیم برای این ذخیره‌سازی از حافظه *EEPROM* میکروکنترلر استفاده می‌شود. دربردهای آردوینو مگا میکروکنترلر مورد استفاده ATMEGA 2560 است که حافظه آن ۴ کیلوبایت می‌باشد.

به‌طور کلی نباید انتظار داشت که نتایج به‌دست‌آمده در مدل عملی دقت و صحت ساختار شبیه‌سازی کامپیوتر را داشته باشد به‌خصوص که به خاطر حافظه محدود *EEPROM* مجبور به داده‌برداری در دفعات متعدد شدیم.

۸- نتایج شبکه عصبی مدل عملی

راه‌اندازی شبکه عصبی پیش‌خور به‌طور مفصل در بخش‌های گذشته بحث شد. دلیل اهمیت این نوع شبکه این است که شبکه عصبی پیش‌خور در مدار سخت‌افزاری (به کمک میکروکنترلر) به‌سادگی طراحی می‌شود و با انجام چند رابطه می‌توانیم جواب شبکه عصبی را وارد سیستم کنیم و سیگنال خروجی که همان سیگنال PWM برای موتورها است را دریافت نماییم.

میزان پراکندگی داده‌ها و نمودارهایی که برازش شده در شکل (۱۶) مشاهده می‌شود. لازم به توضیح است که میزان دقت نمودار برازش شده نسبت به مقدار تابع هدف و با توجه به کم بودن داده‌های آموزش قابل قبول است.



شکل ۱۶- پراکندگی داده‌ها در مدل عملی

به همان شیوه‌ای که در مرحله قبل شبکه عصبی در سیستم اجرا شد این بار همان تابع فعال‌سازی و همان ساختار جمع‌آوری وزن و بایاس شبکه برای شبیه‌سازی شبکه عصبی در درون میکروکنترلر امتحان می‌شود. در این حالت نیازی به کنترل‌کننده پرواز پرنده نداریم و خود میکروکنترلر و برد طراحی شده کنترل پرواز ما هستند و به صورت مستقل کنترل پرنده را انجام می‌دهند. با توجه به نتایج به دست آمده و ماتریس‌های آموزش‌دهنده شبکه عصبی می‌توان ماتریس وزن و بایاس شبکه عصبی را مورد تحلیل قرارداد و این ماتریس‌های خروجی در میکروکنترلر آردوینو به صورت کامل همراه با فرمول‌های فعال‌سازی تابع برای هر لایه محاسبه و پیاده شده است و در آخر پرنده همراه با این کدها آماده پرواز شد. نمایی از ربات پرنده ساخته شده در دو حالت نشسته و در حال پرواز در شکل (۱۷) نمایش داده شده است. فیلم کوتاهی از پرواز این ربات پرنده از پیوند زیر قابل دسترس است.

<https://drive.google.com/file/d/0B6igBODAFzaEVIRLN0VJVGtXSWc/view?usp=sharing>



شکل ۱۷- ربات پرنده ساخته شده

در جدول (۱) معماری‌های مختلف شبکه عصبی برای ساخت ربات پرنده را در یک نگرش کلی با هم مقایسه کرده‌ایم. در شبیه‌سازی کامپیوتری به دلیل وجود داده‌های آموزشی زیاد صحت عملکرد هر دو نوع شبکه عصبی پیش‌خور و نارکس عالی است. درعین حال شبکه نارکس منطق عملکرد بازگشتی و پیچیده‌ای داراست. در مدل عملی به دلیل محدودیت مقدار داده‌های آموزشی عملکرد شبکه نارکس بسیار مناسب‌تر از عملکرد شبکه پیش‌خور است. در حقیقت پیچیدگی الگوریتم بازگشتی نارکس بدان کمک می‌کند تا با وجود داده‌های کم رفتار پیچیده دینامیکی سیستم را به خوبی تفسیر کند. عیب این نوع شبکه پیاده‌سازی سخت‌افزاری بسیار پیچیده آن است که نیازمند پردازنده و حافظه بسیار قوی‌تری برای اجرا روی کنترل کننده‌هاست.

۹- نتیجه‌گیری

در این مقاله در ابتدا بعد از تحلیل دینامیکی پرنده و شناختن فیزیک ربات عمودپرواز، به شبیه‌سازی کامپیوتری کنترل کننده مشتق‌گیر-تناسبی بر روی سیستم پرداختیم. این کنترل کننده به‌تنهایی نمی‌تواند داده‌های آموزشی و هدف شبکه عصبی را به ما بدهد. برای حل این مشکل کنترل کننده بازگشت به عقب که کارایی و دقت عملکرد مناسب‌تری را دارد را طراحی و شبیه‌سازی کردیم. کنترل گر بازگشت به عقب به‌عنوان یک هدف و کنترل کننده PD را به‌عنوان کمک‌کننده شبکه عصبی انتخاب شد. تابع هدف این شبکه عصبی اختلاف کنترل کننده بازگشت به عقب با PD بود. پیاده‌سازی این کنترل کننده و بررسی نتایج آن، نشان از عملکرد مناسب و خوب کنترل طراحی شده داشت. پس از آن به‌منظور کسب درک بهتری نسبت به نتایج به‌دست آمده یک مدل عملی عمودپرواز منطبق بر تمام ساختار شبیه‌سازی و محاسبات تئوری، ساخته شد و با امتحان کردن این نتایج و مدل‌های طراحی الگوریتمی کنترل کننده و بالا بردن عملکرد که در مراحل قبلی به طور کامل توضیح دادیم، ارائه شد، این بار بر روی ربات پرنده واقعی نتایج و ایده‌ها امتحان شد و ثابت کرد که این نتایج حتی در مدل عملی کاملاً قابل‌اتکا هستند.

جدول ۱- نگرش کلی ایجادشده در معماری‌های مختلف شبکه عصبی

نوع سیستم	نوع معماری	مقدار داده‌ها	پیاده‌سازی در مدل	صحت عملکرد
شبیه‌سازی کامپیوتری	پیش‌خور	مطلوب	منطق ساده	عالی
شبیه‌سازی کامپیوتری	نارکس	مطلوب	منطق پیچیده	عالی
مدل عملی	پیش‌خور	محدود	منطق ساده	نه‌چندان مناسب
مدل عملی	نارکس	محدود	منطق پیچیده	خوب

مراجع

[1] Bolandi, H., Rezaei, M., Mohsenipour, R., Nemati, H., and Smailzadeh, S., "Attitude Control of a Quadrotor with Optimized PID Controller", Intelligent Control and Automation, Vol. 4, No. 3, pp. 123-129, (2013).

- [2] Bouabdallah, S., Murrieri, P., and Siegwart, R., "Design and Control of an Indoor Micro Quadrotor", Proceedings of the IEEE International Conference of Robotic and Automation, ICRA '04., IEEE, 26 April-1 May, New Orleans, LA, USA, (2004).
- [3] Tayebi, A., and McGilvary, S., "Attitude Stabilization of a VTOL Quadrotor Aircraft", IEEE Transactions on Control Systems Technology, IEEE, Vo1. 14(3), pp. 562-571, (2006).
- [4] Erginer, B., and Altug, E., "Modeling and PD Control of a Quadrotor VTOL Vehicle", IEEE Intelligent Vehicles Symposium, IEEE, June 13-15, Istanbul, Turkey, (2007).
- [5] Sanca, A.S., Alisina, P.J., and Cerqueira, J.F., "Dynamic Modelling of a Quad Rotor Aerial Vehicle with Nonlinear Input", IEEE Latin American Robotic Symposium, IEEE, October 29-30, Salvador, Brazil, (2008).
- [6] Yali, Y., Changhong, J., and Haiwei, W., "Backstopping Control of Each Channel for a Quadrotor Aerial Robot", Computer Mechatronics Control and Electronic Engineering(CMCE), Vol. 3, No. 2, pp. 403-407, (2012).
- [7] De Vries, E., and Sbarao, K., "Backstepping Based Nested Multi-loop Control Laws for a Quadrotor", 11th International Conference on Control Automation Robotics and Vision, IEEE, December 07-10, Singapore, (2010).
- [8] Morata, F., "Intelligent Fuzzy Controller of Quad Rotor", IEEE International Conference on Intelligent Systems and Knowledge Engineering, IEEE, November 15-16, Hangzhou, China, (2011).
- [9] Abeywardena, D.M., Amaratunga, L.A., and Shakoor, S.A., "A Velocity Feedback Fuzzy Logic Controller for Stable Hovering of a Quad Rotor UAV", International Conference on Industrial and Information Systems (ICIIS), IEEE, December 28-31, Peradeniya, Sri Lanka, (2009).
- [10] Voos, H., "Nonlinear Control of Quadrotor Micro-UAV using Feedback-linearization", IEEE International Conference on Mechatronics, IEEE, April 14-17, Malaga, Spain, (2009).
- [11] Shakev, N.G., Topalov, A.V., Kaynak, O., and Shiev, K.B., "Comparative Results on Stabilization of the Quad-rotor Rotorcraft using Bounded Feedback Controllers", Journal of Intelligent and Robotic Systems, Vol. 65, pp. 389-408, (2012).
- [12] Besnard, L., Shtessel, Y.B., and Landrum, B., "Control of a Quadrotor Vehicle using Sliding Mode Disturbance Observer", IEEE American Control Conference, ACC '07, IEEE, July 09-13, New York, NY, USA, (2007).
- [13] Besnard, L., Shtessel, Y.B., and Landrum, B., "Quadrotor Vehicle Control via Sliding Mode Controller Driven by Sliding Mode Disturbance Observer", Journal of the Franklin Institute, Vol. 349(2), pp. 658-684, (2012).
- [14] Abeywardena, D.M., and Munasinghe, S.R., "Performance Analysis of Kalman Filter Based Attitude Estimator for a Quad Rotor UAV Ultra-modern", International Congress on Ultra Modern Telecommunications and Control Systems, IEEE, October 18-20, Moscow, Russia, (2010).

- [15] Bouabdallah, S., and Siegwart, R., "Towards Intelligent Miniature Flying Robots", Part of the Springer Tracts in Advanced Robotics Book Series (STAR), In Proc. of "*Field and Service Robotics*", Springer, Berlin, Heidelberg, Vol. 25, pp. 429-440, (2005).
- [16] Murray, R.M., Li, Z., and Sastry, S.S., "*A Mathematical Introduction to Robotic Manipulation*", 1st Edition, CRC Press, Boca Raton, USA, (1994).
- [17] Leishman, J.G., "*Principles of Helicopter Aerodynamics (Cambridge Aerospace Series, Series Number 12)*", 2nd Edition, Cambridge University Press, (2016).
- [18] Fay, G., "Derivation of the Aerodynamic Forces for the Mesicopter Simulation", Engineering, Physics, pp. 1-8, (2001).
- [19] Shepherd, J.f., and Tumer, K., "Robust Neuro-control for a Micro Quadrotor", GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, July, <https://doi.org/10.1145/1830483.1830693>, pp. 1131–1138, (2010).
- [20] Efe, M.O., "Neural Network Assisted Computationally Simple $PI\lambda D\mu$ Control of a Quadrotor UAV", IEEE Transactions on Industrial Informatics, IEEE, Vol. 7(2), May, pp. 354-361, (2011).
- [21] Dunfied, J., Tarbouchi, M., and Labonte, G., "Neural Network Based Control of a Four Rotor Helicopter", IEEE International Conference on Industrial Technology, IEEE ICIT '04, December 8-10, Hammamet, Tunisia, (2004).
- [22] Jingya, S., Penghui, F., and Kai-Yuan, C., "Attitude Control of Quadrotor Aircraft via Nonlinear PID", Journal of Beijing University of Aeronautics and Astronautics, Vol. 03, No. 9, pp. 234-242, (2011).
- [23] Bouabdallah, S., Noth, A., and Siegwart, R., "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 28 September-02 October, Sendai, Japan, (2004).
- [24] Zulu, A., and John, S., "A Review of Control Algorithms for Autonomous Quadrotors", Open Journal of Applied Sciences, Vol. 4, pp. 547-556, (2014).

Abstract

In this paper, an intelligent neural network controller is designed that can balance a quadrotor. After comparing the two types of independent reverse controllers and PID in the simulation environment, their differences are stored as data in the software. By specifying the input data of the controller and the target data and using the feedforward neural network architecture and a Narx architecture, intelligent controllers are designed and the obtained results are shown in several stability detection diagrams. The obtained results show that the equilibrium and vertical flight control are quite acceptable. Finally, the results are tested in a practical model on a real vertical plane. In the practical model, due to the limited amount of training data, the performance of Narx network is much more appropriate than the performance of the feed network. In fact, the complexity of the Narx recursion algorithm helps to interpret the complex dynamic behavior of the system well despite the low data. The disadvantage of this type of network is its very complex hardware implementation, which requires a much more powerful processor and memory to run on the controllers.