بسم الله الرحمن الرحیم

دانشکده مهندسی مکانیک و مکاترونیک

گروه مکاترونیک

طراحی یک استراتژی کنترل بهینه برای کنترل ردیابی مسیر در فضای کار توسط ربات دلتا

نگارنده: زهرا حسن‌شاهی

اساتید راهنما

دکتر حبیب احمدی

دکتر سید مجتبی واردی کولایی

شهریور ۱۳۹۸

تقدیم اثر

ما حصل آموخته‌هایم را تقدیم می کنم به آنان که مهر آسمانی‌شان، آرام بخش آلام زمینی‌ام است:

به استوارترین تکیه گاهم ، پدرم

که همواره چتر محبتش بر سرم است

و به سرسبزترین نگاه زندگیم ، مادرم

که دامان پرمهرش یگانه پناهم است

هر آنچه آموختم در مکتب شما آموختم و هر چه بکوشم ، قطره‌ای از دریای بی کران مهربانیتان را سپاس نتوانم بگویم .

تشکر و قدردانی

حمد و سپاس یکتای بی‌همتا را که هستی‌مان بخشید و به طریق علم و دانش رهنمونمان شد، ادای شکرش را هیچ زبان و دریای فضلش را هیچ کران نیست و اگر در این وادی هستیم، همه محبت اوست.

با تشکر از خانواده عزیزم که لذت دانستن، جسارت خواستن، عظمت رسیدن و تمام تجربه‌های ناب و یکتای زندگیم، مدیون حضور سبز آن‌هاست.

همچنین با تقدیر و تشکر شایسته از استادان فرهیخته جناب دکتر احمدی و جناب دکتر واردی کولایی که در کمال شکیبایی، از هیچ کمکی دریغ ننمودند و راهنمایی‌هایشان در تمام مسیر، روشنگر راهم بود.

تعهدنامه

اینجانب **زهرا حسن‌شاهی** دانشجوی دوره کارشناسی ارشد رشته **مکاترونیک** دانشکده مکانیک دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **طراحی یک استراتژی کنترل بهینه برای کنترل ردیابی مسیر در فضای کار توسط ربات دلتا** تحت راهنمائی **دکتر حبیب احمدی** و **دکتر سید مجتبی واردی کولایی** متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایح اصلی پایان‌نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان‌نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان‌نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می‌باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .

استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

چکیده

مزایای موجود در ربات‌های موازی موجب افزایش استفاده از آن‌ها در صنعت و همچنین افزایش توجه به آن‌ها در مطالعات علمی شده است. در این تحقیق، ابتدا به مدل‌سازی سینماتیکی و دینامیکی یک ربات موازی سه درجه آزادی به نام ربات دلتا و کنترل حرکت آن پرداخته شده است. به دلیل وجود زنجیره سینماتیکی موازی و با توجه به روابط هندسی زنجیره ربات دلتا، معادلات سینماتیک معکوس بصورت معادلات حلقه بسته استخراج شده است. در این پژوهش برای به دست آوردن معادلات دینامیکی ربات دلتا از روش لاگرانژ، که یکی از کاربردی‌ترین راه‌های به دست آوردن معادلات دینامیک ربات می‌باشد، استفاده شده است. با توجه به وجود نامعینی‌ها در مدل دینامیکی و به منظور کنترل موقعیت ربات جهت تعقیب مسیر مطلوب، دو استراتژی کنترلی، کنترل‌کننده تطبیقی و مود لغزشی، برای کنترل این ربات در نظر گرفته شده است. کنترل‌کننده تطبیقی جهت مقابله با نامعینی‌ها بر اساس تخمین آن‌ها و کنترل مود لغزشی به دلیل مقاوم بودن در مقابل عدم قطعیت‌های پارامتری و اغتشاش بار طراحی شده‌اند. نتایج نشان می‌دهند که کنترل‌کننده مود لغزشی پیشنهادی عملکرد بهتری نسبت به کنترل‌کننده تطبیقی دارد. همچنین برای بهبود عملکرد کنترل‌کننده مود لغزشی، از یک روش بهینه‌سازی به نام الگوریتم انبوه ذرات، برای پیدا کردن مقادیر بهینه ضرایب کنترل‌کننده استفاده شده است. نتایج شبیه‌سازی، کاهش خطا و همچنین کاهش گشتاورهای کنترلی را توسط کنترل‌کننده مود لغزشی بهینه نسبت به کنترل مود لغزشی بدون بهینه‌سازی نتیجه می‌دهد.

کلمات کلیدی: ربات دلتا، کنترل تطبیقی، کنترل مود لغزشی، بهینه‌سازی.

لیست مقالات مستخرج از پایان‌نامه

۱- مدل‌سازی سینماتیکی و دینامیکی و کنترل تطبیقی ربات موازی دلتا، بیست و هفتمین همایش سالانه بین المللی مهندسی مکانیک ایران، اردیبهشت ۱۳۹۸.

فهرست مطالب

فهرست جداول

فهرست اشکال

فصل ۱: مقدمه

۱-۱ پیشگفتار

مکاترونیک یک رشته علمی از نوع بین‌رشته‌ای بوده که شامل ترکیبی از مهندسی مکانیک، الکترونیک، مهندسی کامپیوتر، مهندسی مخابرات، مهندسی سیستم‌ها و مهندسی کنترل می‌باشد. هدف مکاترونیک، کنار هم قرار دادن یکپارچه این علوم برای داشتن سیستمی ساده‌تر، اقتصادی‌تر و با قابلیت اطمینان بیشتر است. ربات‌ها نمونه‌ای از سیستم‌های مکاترونیکی به شمار می‌روند که دارای جنبه‌های مختلفی از الکترونیک، مکانیک و علوم کامپیوتر برای انجام وظایف گوناگون می‌باشند. تاریخچه ربات به زمانی بازمی‌گردد که ریاضیدان یونانی با نام ارخوطس یک پرنده مکانیکی ساخت که با استفاده از بخار حرکت می‌کرد. این اولین تلاش انسان برای ساخت ابزار اتوماتیک بود. بخش عمده کارها در رباتیک، در قرن بیستم صورت پذیرفت. نخستین بار در اوایل دهه ۱۹۲۰ کلمه ربات با مفهوم امروزی توسط "کارل چاپک" در نمایشنامه خود به عنوان آدم مصنوعی بکار گرفت. ریشه این کلمه به زبان چک برمی‌گردد و از کلمه "ربوتا" به معنی "برده و کارگر" گرفته شده است. کلمه رباتیک اولین بار توسط ایزاک آسیموف در یک داستان کوتاه ارائه شد[۱].

یکی از انگیزه‌های اصلی برای استفاده از ربات‌ها افزایش بهره‌وری با استفاده از افزایش سرعت ربات است. در واقع یک رویکرد عملی برای به حداقل رساندن انرژی مصرفی موتور و حداقل‌سازی سطح ولتاژ مورد نیاز حرکت موتور جهت انجام وظایف محوله با توجه به محدودیت‌های فیزیکی است. به طور کلی می‌توان تحقیقات انجام گرفته توسط محققان را در زمینه بازوان رباتیک به سه دسته طراحی مسیر، تخمین موقعیت و کنترل تقسیم‌بندی نمود. در این پایان نامه در فصل دوم معادلات سینماتیک و دینامیکی ربات تشریح می‌شود. در فصل سوم طراحی کنترل تطبیقی انجام می‌شود و در فصل چهارم طراحی کنترل مود لغزشی و بهینه‌سازی آن صورت می‌پذیرد. در فصل پنجم نیز نتایج مورد بررسی قرار می‌گیرند.

۱-۲ ادبیات موضوع

عصر رباتها با اولین ربات آدم‌نما توسط جرج دوول شروع شد. امروزه ۹۰ درصد ربات‌ها، ربات‌های صنعتی هستند، یعنی ربات‌هایی که در کارخانه‌ها، آزمایشگاه‌ها، انبارها، نیروگاه‌ها، بیمارستان‌ها و بخش‌های مشابه به کار گرفته می‌شوند. در سال‌های قبل، اکثر ربات‌های صنعتی در کارخانه‌های خودروسازی به کار گرفته می‌شدند ولی امروزه ربات‌های صنعتی زیادی ساخته شده‌اند و تنها حدود نیمی از ربات‌های موجود در دنیا در کارخانه‌های خودروسازی به کار گرفته می‌شوند. در سال ۱۹۶۰ انجمن صنایع رباتیک این تعریف را برای ربات صنعتی ارائه کرده است: **" ربات صنعتی یک وسیله چندکاره و با قابلیت برنامه ریزی چند باره است که برای جابجایی قطعات، مواد، ابزارها یا وسایل خاص به وسیله حرکات برنامه ریزی شده، برای انجام کارهای متنوع استفاده می‌شود[۱]."**

به طور کلی ربات‌ها بر اساس زنجیره سینماتیکی به دو دسته سری و موازی تقسیم می‌شوند. به ربات‌هایی که زنجیره سینماتیکی باز داشته باشند ربات سری گویند و ربات‌های با زنجیره سینماتیکی بسته را ربات موازی می‌نامند. از مزیت‌هایی که ربات موازی نسبت به ربات سری دارد، می توان به دقت بالا، سفتی زیاد و نسبت بار به وزن بالا اشاره کرد و از معایب آن فضای کاری محدود، روابط سینماتیک مستقیم پیچیده و تحلیل های دینامیکی دشوار می باشد. مزایای موجود در مکانیزم موازی منجر به توزیع نیروی بهتر و بالا رفتن سرعت و شتاب ربات می‌شود [۲]. از این رو استفاده از آنها به جای مکانیزم‌های سری در صنعت و همچنین در مطالعات علمی افزایش یافته است.

ربات دلتا یکی از معروف‌ترین و کاربردی‌ترین ربات‌های موازی است که دارای سه درجه آزادی می‌باشد. این ربات شامل یک صفحه ثابت است که سه موتور به آن متصل است. موتورها باعث حرکت سه بازو می‌شوند که به یک صفحه متحرک متصل هستند. حرکت این سه بازو که دارای هندسه متوازی‌الاضلاع هستند، باعث می‌شود که

صفحه متحرک همیشه درموازات صفحه ثابت حرکت کند. ایده اولیه طراحی ربات‌های موازی دلتا استفاده از متوازی‌الاضلاع است. یک متوازی‌الاضلاع اجازه می‌دهد که حرکت لینک خروجی با توجه به حرکت لینک ورودی در یک جهت ثابت باقی بماند. استفاده از سه متوازی‌الاضلاع در مکانیزم این ربات به طور کامل جهت گیری صفحه متحرک را مهار می‌کند که تنها سه درجه آزادی انتقالی دارد. لینک‌های ورودی سه متوازی‌الاضلاع بر روی بازوهای چرخشی با مفاصل درونی متصل می‌شوند. مفاصل دورانی بازوهای دوار به دو روش موتورهای چرخشی و یا با عملگرهای خطی عمل می‌کنند. در برخی از ربات‌های دلتا با استفاده از لینک چهارم برای انتقال حرکت دورانی از پایه به مجری نهایی متصل به صفحه متحرک استفاده می‌شود.

برای مطالعه هر ربات، باید معادلات سینماتیک مستقیم و معکوس و همچنین معادلات دینامیک ربات را یافت. برای یافتن معادلات سینماتیک معکوس ربات‌های موازی، به دلیل داشتن زنجیره سینماتیکی بسته، معمولا از روش معادلات حلقه بسته استفاده می‌شود. با توجه به معادلات سینماتیک مستقیم و معکوس، ماتریس ژاکوبین ربات نیز حاصل می‌شود. ماتریس ژاکوبین مستقیم و معکوس ربات حاوی اطلاعاتی است که کمک می‌کند تا بتوان برخی از پیکربندی‌های ویژه را شناسایی کرد. منظور از پیکربندی ویژه، حالتی است که ربات در تکینگی قرار دارد. روش‌های متفاوتی برای به دست آوردن معادلات دینامیکی ربات‌ها وجود دارد که از جمله آنها روش لاگرانژ، قانون دوم نیوتن و کار مجازی می‌باشد. روش لاگرانژ یکی از بهترین روش‌های موجود برای حل معادلات دینامیکی ربات است. همچنین به دلیل وجود قیود اضافی در ربات‌های موازی، روش لاگرانژ تعمیم یافته یک روش مناسب برای یافتن معادلات دینامیکی این چنین ربات‌ها است.

بسیاری از سیستم‌های دینامیکی غیرخطی که بایستی کنترل شوند، پارامترهای نامعلوم دارند که یا ثابت هستند و یا به آهستگی تغییر می‌کنند. برای مثال، ربات‌های حمل مواد ممکن است اجسام بزرگی را با پارامترهای اینرسی نامعلوم حمل کنند. هدف اصلی از طراحی کنترل‌کننده در این موارد، این است که با وجود عدم قطعیت

در برخی از پارامترهای ربات نظیر جرم و ممان اینرسی جسم میانی، بتوان موقعیت ربات را کنترل نمود[۳]. کنترل‌کننده‌های مقاوم یکی از ابزارهای با ارزش کنترلی می‌باشند که قادر به مقابله با عدم قطعیت‌هایی با اندازه کران‌دار مانند اغتشاش خارجی و دینامیک‌های مدل نشده، می‌باشد. کنترل تطبیقی روش دیگر برای کنترل این چنین سیستم‌ها است که قادر به مقابله با عدم قطعیت‌های پارامتری می‌باشد. همچنین می‌توان با به‌کارگیری هردو ابزار کنترل تطبیقی و کنترل مقاوم به طراحی کنترل‌کننده برای کنترل بهتر ربات متحرک پرداخت.

موضوع کنترل بهینه، یافتن یک قانون کنترلی برای سیستم به نحوی است که معیار بهینگی معینی حاصل شود. ورودی اعمال شده به سیستم در بهترین شرایط و معیار بهترین راه یا عملکرد را به ترتیب کنترل بهینه و شاخص عملکرد یا تابع هزینه می‌گویند. یک مسئله کنترل بهینه دارای یک تابع هزینه است که این تابع هزینه، تابعی از متغیرهای حالت و کنترلی می‌باشد. مهم‌ترین هدف کنترل بهینه، تعیین سیگنال کنترلی است که باعث شود یک فرآیند در برخی محدودیت‌ها یا قیود فیزیکی صدق کرده و در عین حال یک تابع هزینه بهینه شود. برای داشتن بهترین شرایط کنترلی جهت کنترل ربات دلتا، از استراژی کنترل بهینه استفاده می‌شود.

وجود مسائل پیچیده طراحی ربات، مسیر و کنترل ربات‌ها باعث می‌شود که روش‌های نوین بهینه‌سازی کارایی زیادی در این زمینه داشته باشند. اخیرا به دلیل زمان‌بر بودن و پیچیدگی روش‌های دقیق، استفاده از روش‌های بهینه‌سازی هوشمند رواج یافته است. تاکنون روش‌های بهینه‌سازی متعددی معرفی شده‌اند که از مهمترین آن‌ها می‌توان به الگوریتم‌های تکاملی، الگوریتم تپه نوردی، الگوریتم شبیه‌ساز سرد کردن فلزات، الگوریتم بهینه‌سازی انبوه ذرات، الگوریتم جستجوی ممنوع، الگوریتم بهینه‌سازی مورچه‌ها، خودکارهای یادگیر و غیره اشاره نمود.

۳-۱ مروری بر کارهای پیشین

استفاده از ربات‌های موازی برای اولین بار، توسط استوارت و با معرفی یک مکانیزم موازی با شش درجه آزادی آغاز شد. این مکانیزم، دارای شش پایه است که توسط شش موتور کنترل می‌شوند و هر کدام دارای یک پایه زمین هستند[۴]. پس از آن، انواع ربات‌های موازی با ساختارهای مختلف مورد بررسی و استفاده قرار گرفتند. ایده ربات دلتا برای اولین بار در اوایل سال ۱۹۸۰ توسط یک تیم تحقیقاتی به رهبری پروفسور ریموند کلاول در موسسه فدرال تکنولوژی در لوزان سوئیس مطرح شد و در سال ۱۹۸۷ این ربات بـا نام ویپو[1] ثبت اختــراع گردید[۵]. هدف از ســاخت این نوع ربات، جابه‌جایی اجسام سبک و کوچک با حداکثــر سرعت بود. این ربات به دلیل دارا بودن سرعت بالا، برای برداشتن و گذاشتن اجسام سبک در صنعت مورد استفاده قرار می‌گیرد. پس از آن در سال ۱۹۸۷ یک شرکت سوئیسی به نام دمارکس[2] مجوز ساخت ربات دلتا را خریداری کرد و شروع به تولید ربات‌های دلتا برای صنعت بسته‌بندی نمود.

در سال ۱۹۸۹ یک ربات موازی دلتا توسط تیم EPFL[3] طراحی شد. طراحی این ربات به گونه‌ای بود که از مزایای ربات های موازی و همچنین از سهولت مدل‌سازی ربات سری برخوردار بود. آن‌ها تلاش کردند تا با در نظر گرفتن پارامترهای سینماتیکی مطلوب و طرح چند فرضیه‌ی ساده‌سازی برای مدل استاتیکی و دینامیکی آن، سینماتیک و دینامیک کامل ربات را ارائه دهند [۶]. در شکل (۱-۱) یک ربات متخصص مورد استفاده در زمینه جراحی مشاهده می‌شود که توسط یک شرکت سوئدی تولید شده است[۵].

[1] wipo
[2] Demaurex
[3] Ecole Polytechnique Federale de Lausanne

شکل ۱-۱. ربات متخصص جراح [۶]

در تحقیق [۷] آقای هاشم زاده و همکارانش سعی کردند تا به ملزومات مورد استفاده در جهت ساخت پرینتر سه بعدی با ساختار دلتا از لحاظ سخت افزاری و نرم افزاری بپردازند. همچنین پس از بررسی تحقیقات پیشین بر روی ربات دلتا، در مورد مراحل طراحی و ساخت در چاپگرهای سه بعدی بحث شده است. سپس به منظور رسیدن به فضای کاری دلخواه و بهینه کردن طراحی و مکانیزم، به طراحی دستگاه و ساخت و خرید قطعات مکانیکی و الکتریکی لازم و نهایتا به بخش انتخاب ماده برای ساخت پرینتر پرداخته شده است. در بحث طراحی و ساخت، مباحثی همچون فضای کاری، جلوگیری از ارتعاشات ناخواسته، بهینه کردن مکانیزم‌های قبلی و همچنین سرعت انجام عملیات توسط دستگاه را الگو قرار داده‌اند.

یک ربات آموزشی دلتا، در سال ۲۰۱۷ توسط هومام ابوالکباش و همکاران که از دانشجویان کارشناسی مهندسی مکانیک دانشگاه پلی تکنیک فلسطین بودند، طراحی و اجرا شد که در شکل (۲-۱) مشاهده می‌شود. این ربات

آموزشی از جنبه‌های مختلف از جمله طراحی مکانیکی، حسگرها، میکروکنترلرها و تکنیک‌های کنترل، مورد توجه قرار گرفته است [۸].

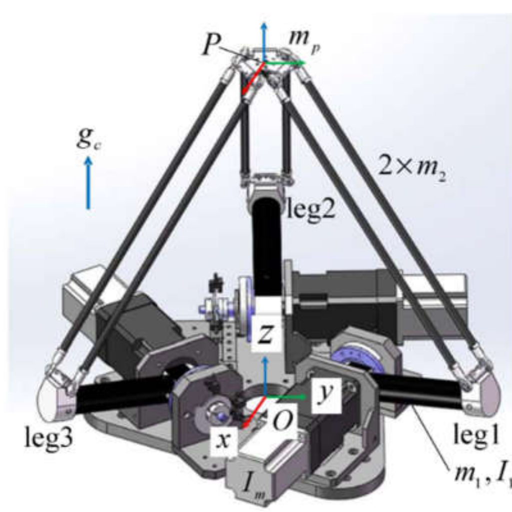
شکل ۲-۱. ربات دلتا آموزشی [۸]

در تحقیق [۹] از تحلیل سینماتیک ربات دلتا استفاده شده تا سرعت مجری نهایی از ترم سرعت زاویه‌ای مفاصل به دست آید و بنابراین ماتریس‌های ژاکوبین حاصل می‌شود. از برابر قرار دادن دترمینان ماتریس ژاکوبین با صفر چندین موقعیت تکینگی از ساختار ربات استخراج می‌شود. تجزیه و تحلیل ماتریس ژاکوبین معکوس نشان می-دهد که تکینگی‌ها زمانی رخ می‌دهند که عضوهای متعلق به زنجیره سینماتیکی روی یک صفحه قرار گیرند. دو ساختار احتمالی که با این شرایط مطابقت دارند، زمانی است که ربات به طور کامل باز یا بسته شده که در این صورت مرزهای فضای کاری را نشان می‌دهد.

در تحقیق [۱۰] یک روش ساده بر اساس اصل کار مجازی برای مدل‌سازی دینامیک ربات‌های موازی پیشنهاد شده است. این روش به ربات دلتا اعمال شده است که منجر به یک مدل بسیار کارآمد شده که برای کنترل ربات

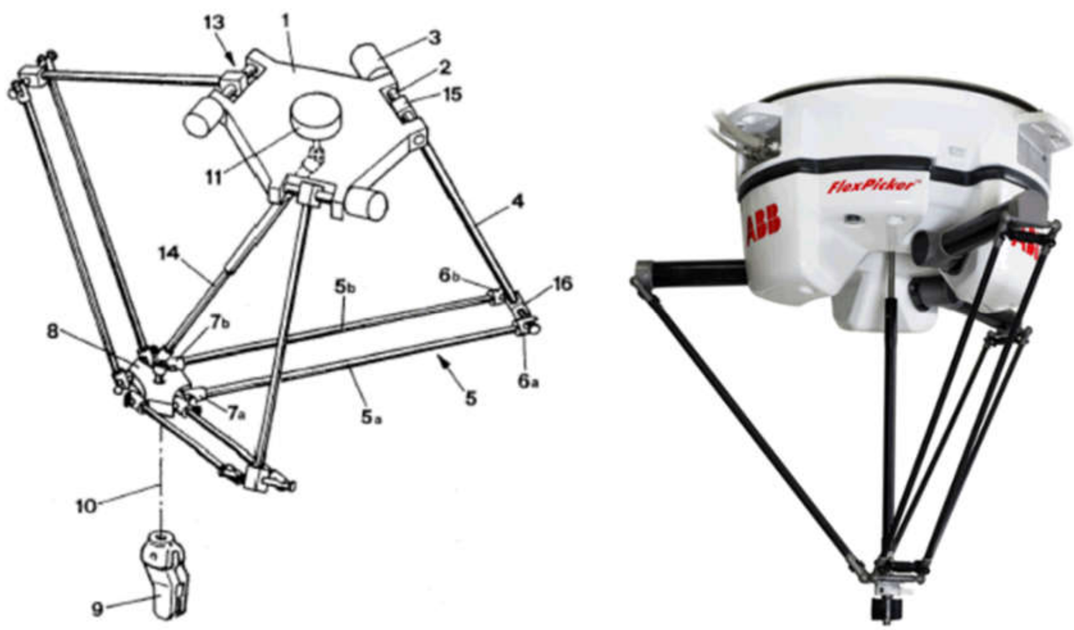
مورد استفاده قرار گرفته است. مدل پیشنهادی به اندازه کافی ساده است که بتواند در زمان واقعی محاسبه شود و از آن مدل، برای کنترل گشتاور محاسبه شده استفاده کرد. همچنین در سال ۲۰۱۵ یو و همکارانش نیز معادلات ربات دلتا را بر اساس کار مجازی به دست آوردند و از کنترل‌کننده PD برای کنترل این ربات استفاده کردند[۱۱].

در رابطه با مدل‌سازی دینامیکی و سینماتیکی ربات دلتا، پیروت و همکاران تلاش کردند تا با در نظر گرفتن پارامترهای سینماتیکی مطلوب و طرح چند فرضیه‌ی ساده سازی برای مدل استاتیکی و دینامیکی آن، سینماتیک و دینامیک کامل ربات را ارائه دهند[۶]. در سال ۲۰۱۳ پارک و همکارانش یک نوع ربات دلتا را مدل‌سازی کردند و سینماتیک معکوس آن را با استفاده از معادلات حلقه بسته و دینامیک آن را با روش لاگرانژ تعمیم یافته بدست آوردند. به طور معمول، صفحه ثابت ربات دلتا و همچنین موتورها در بالا و صفحه متحرک آن در پایین قرار دارند اما در برخی از انواع ربات دلتا، صفحه متحرک در بالا قرار میگیرد. شکل (۳-۱) یک نوع ربات دلتا را نشان می‌دهد که پارک و همکارانش بر روی آن آزمایش کردند و موتورهای آن در پایین قرار دارند [۱۲].



شکل ۳-۱. ربات دلتا مورد مطالعه پارک و همکاران [۱۲]

همچنین آقای ویلیامز در سال ۲۰۱۶ روابط کامل سینماتیکی و دینامیکی یک نوع ربات دلتا با ۴ درجه آزادی را بدست آورده است. این ۴ درجه آزادی شامل ۳ درجه آزادی برای حرکت انتقالی در سه جهت محور مختصات و همچنین یک ساق درونی چهارم برای کنترل یک درجه آزادی چرخشی صفحه مجری نهایی می‌باشد. شکل (۴-۱) این ربات را نشان می‌دهد که ساق چهارم در جهت محور عمود بر صفحه ثابت و متحرک می‌باشد [۱۳].



شکل ۴-۱. ربات دلتا با یک درجه آزادی اضافی [۱۳]

سیستم‌های فیزیکی ذاتاً غیرخطی هستند و ربات دلتا نیز دارای معادلات دینامیکی غیرخطی می‌باشد. بدین منظور برای کنترل این ربات باید از روش‌های کنترل غیرخطی استفاده کرد. یکی از ساده‌ترین راه‌های کنترل این ربات، کنترل PD می‌باشد. در این زمینه خانگ و همکاران سه کنترل‌کننده مختلف، شامل کنترل‌کننده PD بر پایه‌ی گشتاور محاسبه‌شده، کنترل‌کننده مد لغزشی مرتبه اول و کنترل‌کننده مد لغزشی با کمک شبکه‌ی

عصبی را بر روی ربات دلتا تست کردند و نتایج حاصل را با یکدیگر مقایسه نمودند. نتایج نشان می‌دهد که روش کنترل PD فقط برای زمانی دارای پاسخ مناسب است که پارامترهای مدل دقیقا شناسایی شده باشد. در مورد پارامترهای نامشخص، از دو روش دیگر استفاده می‌شود. هر سه کنترل‌کننده شبیه‌سازی شده است و نتایج شبیه‌سازی نشان می‌دهد که استفاده از کنترل‌کننده مد لغزشی با کمک شبکه عصبی در مقایسه با دو روش دیگر بهتر است[۱۴].

در زمینه‌ی کنترل ربات دلتا، اوزونوویچ و همکاران سینماتیک معکوس ربات را به کمک شبکه‌ی عصبی استخراج کردند و سپس از این شبکه، به عنوان بخشی از سیستم کنترل استفاده نمودند. آنها همچنین برای اندازه‌گیری سرعت و اغتشاش از مشاهده‌گرها استفاده کردند[۱۵]. همچنین، ژانگ و لیهانگ شی بر مبنای نظریه کنترل فازی یک روش برای کنترل ربات سه درجه آزادی دلتا ارائه دادند. تجزیه و تحلیل نتایج نشان می‌دهد که این روش کنترلی، کنترل مختصات را برای این ربات حل نموده است. بهبود دقت کنترل و ردیابی مسیر در زمان واقعی از مزایای این طرح بوده است[۱۶].

در سال ۲۰۱۱ طراحی کنترل‌کننده منطق فازی نوع ۲ (IT2FLC[1]) با وجود عدم قطعیت‌های مقاوم توسط اندرج لیندا و میلوس منیک انجام شد و بر روی مساله کنترل موقعیت ربات موازی دلتا مورد بررسی قرار گرفت. روش ارائه شده با طراحی کنترل فازی نوع ۱ (IT1FLC[2]) آغاز شده و سپس طراحی کنترل فازی نوع ۲ صورت گرفته است. منبع اصلی عدم قطعیت‌ها شامل نویز و پارامترهای نامشخص سیستم می‌باشد. نتایج کار ارائه شده را می‌توان به شرح زیر خلاصه کرد: زمانی که طراحی مناسب مجموعه های فازی نوع ۲ انجام شده، IT2FLC عملکرد بهتر و قابلیت اطمینان بیشتری نسبت به IT1FLC دارد. افزایش مقدار فازی نوع ۲ به طور یکنواختی

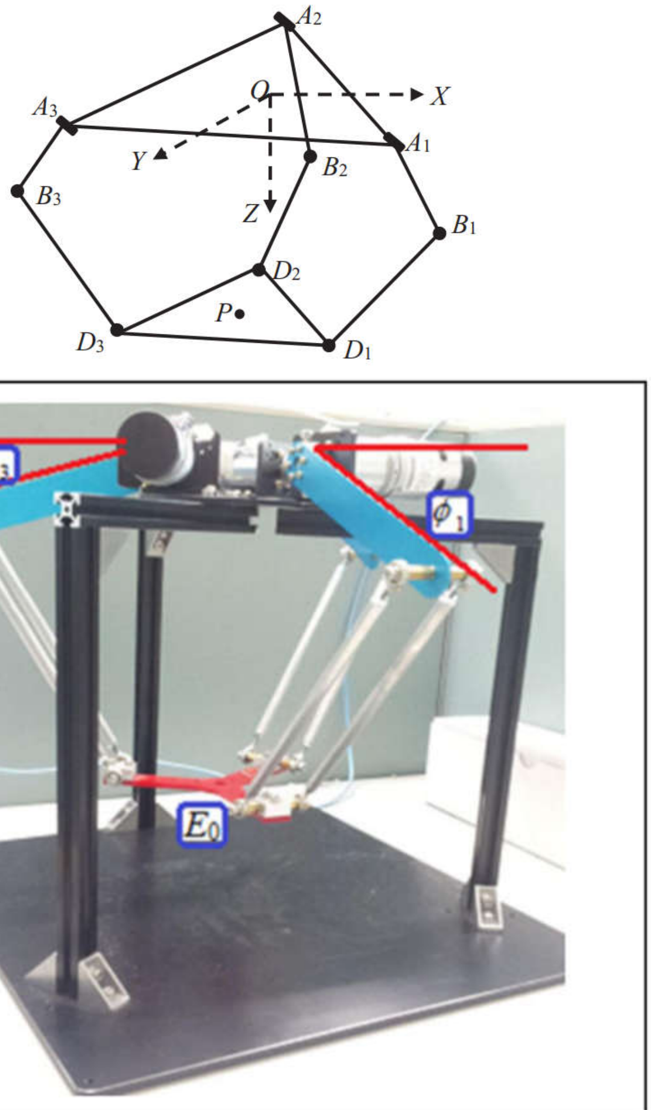
[1] Interval Type-2 Fuzzy Logic Controller
[2] Interval Type-1 Fuzzy Logic Controller

موجبات عدم اطمینان IT2FLC را تا نقطه اوج آن افزایش می‌دهد. همچنین مقدار بیش از حد فازی نوع ۲ و خیلی مبهم بودن توابع عضویت فازی، منجر به تخریب عملکرد سریع سیستم می‌شود[۱۷].

در سال ۲۰۱۵ راچدی و همکاران، دو طرح مبتنی بر مدل، شامل کنترل مقاوم H∞ و کنترل گشتاور محاسبه شده CTC[1] را بر روی ربات دلتا اجرا کردند و عملکرد کنترل‌کننده‌ها را برای یک مسیر مطلوب سهموی مقایسه کردند. نتایج شبیه‌سازی‌شده بر روی ربات دلتا نشان داد که کنترل‌کننده H∞ عملکرد بهتر و استحکام بیشتری نسبت به کنترل‌کننده CTC داشته است[۱۸]. ایشان همچنین کنترل مقاوم H∞ را با کنترل PID اعمال شده بر روی ربات دلتا را با هم مقایسه کردند. آزمایشات با اعمال بارهای متفاوتی به ربات انجام شد که در وزن های بالا، خطای کنترل H∞ حدود ۸۰ درصد خطای کنترل PID بوده است. آزمایشات تنوع بار ثابت کردند که کنترل H∞ قوی‌تر از PID است و میانگین خطای متوسط مربع H∞ در حالت پایدار کمتر از ۶۰٪ خطای بدست آمده از کنترل‌کننده PID است[۱۹].

یانگ لی کو و پنگ یو هانگ کنترل حرکت ربات دلتا را با روش مبتنی بر مدل دینامیکی انجام دادند. طرح پیشنهادی آن‌ها بدین صورت است که ابتدا گشتاور اعمال شده بر موتورها را با کمک سیگنال بازگشتی از انکودر، مبتنی بر سینماتیک و دینامیک ربات محاسبه می‌کند. سپس کنترل محاسبه شده برای کنترل بازخورد به کنترل‌کننده‌ها فرستاده می‌شود[۲۰]. شکل (۵-۱) یک ربات دلتا آزمایشگاهی که طراحی شده است را نشان می‌دهد.

[1] Copmuted Torqe Control



شکل ۵-۱. ربات دلتا آزمایشگاهی[۲۰]

لوئیس انگل و همکاران یک استراتژی کنترل تطبیقی برای حل مشکل ردیابی مسیر یک ربات دلتا با مدل نامتعارف دینامیکی که در شکل (۶-۱) مشاهده می‌شود را طراحی کرده‌اند. الگوریتم تطبیقی بدون در نظر گرفتن اندازه‌گیری سرعت مفاصل ربات طراحی شده است. ردیابی مسیر مرجع در این تحقیق، موفقیت آمیز بوده و در تمام آزمایشات، طرح تطبیقی عملکرد بهتر را نسبت به کنترل‌کننده PID معمولی نشان داده است[۲] .

شکل ۶-۱. ربات دلتا با مدل نامتعارف دینامیکی [۲]

ایده اصلی در کنترل تطبیقی این است که پارامترهای نامعلوم سیستم و یا پارامترهای کنترل‌کننده آن بر اساس سیگنال‌های اندازه‌گیری شده به صورت بهنگام تخمین زده شوند. در انجام محاسبات ورودی کنترل، از پارامترهای تخمین زده شده استفاده شود. در سال ۲۰۱۵ ماریو رامیرز و همکاران خود، مساله ردیابی مسیر برای این ربات را با کنترل‌کننده ردگیری اغتشاش فعال مبتنی بر مشاهدات را حل نموده‌اند. طرح کنترل پیشنهادی به شناخت دقیق سیستم نیازی ندارد. پس جایگزین خوبی برای طرح‌های کنترل کلاسیک مانند روش گشتاور محاسبه شده می‌باشد[۲۱].

در سال ۱۹۹۸ باردت و همکار خود برای یافتن پارامترهای معادله دینامیک و بهبود کنترل ساختار ربات، کنترل‌کننده تطبیقی غیرخطی را بر روی دو ساختار دو و سه درجه آزادی که همان ربات دلتا بود، آزمایش

کردند. نتایج تجربی نشان داد که کنترل‌کننده پیشرو تطبیقی (AFFC[1]) برای مشخص شدن پارامترهای مجهول دینامیک ربات حتی در صورت وجود اصطکاک و نویز مناسب است. عملکرد کنترل در طول این اعمال این روش بهتر از زمانی است که پارامترهای اندازه گیری شده استفاده می‌شود. این کار همچنین یک کنترل قوی و پایدار را فراهم می‌کند و باعث عملکرد بهتر می‌شود[۲۲].

لو و لیو در سال ۲۰۱۷ برای ردیابی مسیر مطلوب ربات دلتا یک طرح کنترل خودکار شبکه عصبی فازی نوع ۲ پیشنهاد دادند. این کنترل‌کننده دارای ساختار موازی است که ترکیبی از یک کنترل‌کننده شبکه عصبی فازی نوع ۲ (IT2FNN[2]) و یک کنترل‌کننده PD سنتی است. از کنترل‌کننده PD برای جبران عملکرد گذرای سیستم و از IT2FNN برای یافتن پارامترهای دینامیکی سیستم استفاده می‌شود. نتایج شبیه‌سازی عددی نشان داده است که کنترل‌کننده شبکه عصبی فازی عملکرد کنترلی بهتری نسبت به کنترل‌کننده سنتی PD برای کنترل ردیابی مسیر از نظر بهبود دقت ردیابی و استحکام داشته است[۲۳].

دقت ردیابی و قابلیت اطمینان در کنترل ربات‌های موازی، بسیار اهمیت دارد. کنترل مود لغزشی مساله ردیابی دقیق را حل می‌کند اما به دلیل پدیده‌های لرزش به اندازه کافی قوی نمی‌باشد. پیاده سازی منطق فازی در مود لغزشی باعث افزایش قابلیت اطمینان می‌شود. به همین خاطر در مقاله [۲۴] با استفاده از مدل دینامیک ربات موازی دلتا و با توجه به عدم قطعیت در پارامترهای دینامیکی ربات، کنترل مود لغزشی فازی به این ربات اعمال شده است. نتایج شبیه‌سازی و مقایسه آنها با نتایج کنترل مود لغزشی مزایای استفاده از منطق فازی در روش‌های کنترل را نشان می‌دهد.

مساله کنترل بهینه برای سیستم تحت محدودیت‌ها، توسط پونتریاگین و دیگر محققین مورد اشاره قرار گرفته است و نتایج آن در یک اصل مشارکتی با نام اصل حداقل پونتراگین ارائه شده است. تحقیقات بسیاری نیز در این

[1] Adaptive FeedForward Controller
[2] Interval Type-2 Fuzzy Neural Network

زمینه انجام شده است. اصل پونتریاگین شرایط لازم برای بهینگی یک سیستم دینامیکی را فراهم می‌کند. در تحقیق [۲۵] شرایط کافی برای بهینگی کنترل یک سیستم غیرخطی تحت محدودیت‌های حالت و کنترل و با معلوم بودن زمان‌های اولیه و نهایی ارائه شده است.

سیستم‌های کنترل بهینه، دسته مهمی از روش‌ها را شامل می‌شوند که در کنار کاربردهای فنی و صنعتی، کاربردهای فراوانی در حوزه‌های مختلف مدیریت، اقتصاد و علوم مالی دارد. چن در سال ۱۹۹۱ مساله طرحی مسیر بهینه برای بازوان رباتیک را به عنوان یک مساله کنترل بهینه با محدودیت‌های حالت و کنترل مطرح کرد و سپس نتیجه کنترل بهینه حل شده را به صورت عددی حل نمود[۲۶] . همچنین در تحقیقات [۲۷] و [۲۸] از روش کنترل بهینه غیرخطی با معادله ریکاتی وابسته به حال برای کنترل مسیر و هدایت انواع سیستم‌ها رباتیکی استفاده شد. در تحقیق [۲۹] یک روش سنتز کنترل بهینه غیرخطی جدید که روش دو مرحله‌ای نامیده می‌شود، برای بهبود بهینه‌سازی یک ربات ناوبری معادن ذغال سنگ (CMRR[1]) طراحی شده است. در این روش اولین قدم این است که سیستم خطی شود و حل مساله کنترل خطی مرتبط با آن انجام شود. گام دوم راه حل تنظیم عددی از گام اول به منظور اجرای محدودیت‌های بهینه و پایداری است. نتایج نشان داده است که روش دو مرحله‌ای می‌تواند مسیر مطلوب برای CMRR را در محیط ناشناخته به دست آورد. در سال ۲۰۱۴ نیز تحقیقی مبتنی بر کنترل بهینه و بررسی کمترین زمان محدود برای سیتم‌های متغیر با زمان با حالت و کنترل غیر خطی صورت پذیرفت که در آن، از معادله ریکاتی وابسته به حالت استفاده شد[۳۰].

در خصوص ربات دلتا، در سال ۲۰۰۷ وانگ و همکاران ایده جدیدی را برای تعیین مجموعه‌ای از پارامترهای طراحی بهینه یک ربات دلتای خطی ارائه داده‌اند. روش طراحی بهینه را بر اساس سه الگوریتم معرفی کردند که

[1] Coal mine rescue robot

دو الگوریتم برای تعیین فضای کاری قابل دسترس طراحی شده‌اند و الگوریتم دیگر برای حل مشکل بهینه استفاده می‌شود و نتیجه روی ربات دلتای خطی اعمال شده است[۳۱].

فابین و همکاران خود در سال ۲۰۱۶ ردیابی مسیر رباب دلتا را انجام دادند. برای این منظور یک نمونه از این ربات با استفاده از طراحی ساختاری مکانیکی که قبلا توسط ریموند کلاول پیشنهاد شده بود، ساخته شد. برای ردیابی مسیر توسط این ربات دو روش مختلف طراحی کنترل‌کننده بر اساس قرارگیری قطب‌ها و بازخورد حالت استفاده شده است. اولین روش طراحی کنترل کننده PD بود و روش دوم LQR[۱] بود که در آن پارامترهای PD با استفاده از موقعیت قطب‌ها بدست آمد. کنترل‌کننده‌های طراحی شده بر اساس کنترل موقعیت زاویه‌ای موتورها طراحی شدند و کنترل مجری نهایی از سینماتیک معکوس حاصل می‌شود. از لحاظ خطای ردیابی کنترل LQR عملکرد بهتری نسبت به PD داشته است اما کنترل LQR نیاز به اندازه‌گیری سیگنال‌های بیشتری دارد و ممکن است هزینه بیشتری داشته باشد[۳۲].

اخیرا محققان به دلیل سادگی و کارآیی بالا از روش‌های نوین بهینه‌سازی برای هدف بهینه‌سازی کنترل و همچنین طراحی بهینه ربات و مسیر استفاده می‌کنند. در این خصوص، لاریبی و رومدهان در سال ۲۰۰۷ با استفاده از الگوریتم ژنتیک به پیدا کردن کوچکترین فضای کاری ربات دلتا می‌پردازند که در آن ربات دلتا بتواند به تمام نقاط مورد نظر که از قبل تعیین شده‌اند، دسترسی پیدا کند[۳۳]. در سال ۲۰۱۱ ارگین و همکاران خود یک مدل ربات دلتا را با ابعاد بهینه ساختند. در واقع آن‌ها یک مساله بهینه‌سازی طراحی چندمنظوره برای به حداکثر رساندن شفافیت دستگاه طراحی کردند و شاخص های عملکرد سینماتیک و دینامیک را به طور همزمان بهینه نمودند. دستگاه با استفاده از کنترل امپدانس حلقه باز کنترل شده است. دستگاه کنترل شده با امپدانس

[1] Linear Quadratic Regulator

حاصل می‌تواند یک مسیر مرجع سینوسی را با خطای متوسط ۲.۳٪ ردیابی کند، همچنین می‌تواند داده‌های استاتیکی را با ضریب کمتر از ۰.۱٪ ارائه دهد[۳۴].

کلایایا و همکاران خود در سال ۲۰۱۲ یک روش برای بهینه‌سازی چند منظوره یک ربات دلتای خطی را پیشنهاد دادند. روش پیشنهادی، بدین صورت است که ابتدا مدل‌سازی هندسی و سینماتیکی ساختار انجام می-شود و معیارهای عملکرد، همانند فضای کاری و سختی مورد ارزیابی قرار می‌گیرد و سپس محدودیت‌های مربوط به ساختار ربات تعریف می‌شود. پس از آن فرمول‌بندی ریاضی مساله بهینه‌سازی چند منظوره صورت می‌پذیرد و در نهایت الگوریتم ژنتیک برای حل مساله به کار گرفته می‌شود[۳۵].

با توجه به مشکل تنظیم پارامترهای کنترل منطق فازی (FLC[1]) برای کنترل مسیر ربات دلتا در سال ۲۰۱۵ لو و لیو یک رویکرد مبتنی بر بهینه سازی ذرات (PSO[2]) برای تنظیم خودکار پارامترهای توابع عضویت تابع فازی پیشنهاد کردند. آنها علاوه بر استفاده از کنترل فازی و بهینه کردن پارامترهای مربوط به توابع عضویت FLC، از کنترل کننده PID نیز استفاده کردند و ضرایب کنترل کننده را با استفاده از همین روش پیشنهادی بهینه کردند و این دو روش را با هم مقایسه کردند. همچنین مقاوم بودن کنترل کننده ها در سه مورد از عدم قطعیت های مختلفی که در کاربردهای مهندسی با آن مواجه می‌شوند مورد ارزیابی قرار گرفت. نتایج شبیه سازی نشان می-دهد که کنترل کننده FLC تنظیم شده توسط PSO میتواند عملکرد بهتری را در مقایسه با کنترل‌کننده PID تنظیم شده با PSO در برخورد با سیستم‌های غیرخطی مانند ربات‌های موازی از جمله ربات دلتا بدست آورد همچنین به این نتیجه رسیدند که تنظیم خودکار پارامترهای FLC با استفاده از روش PSO بسیار ساده‌تر از روش سنتی آزمون و خطا بر اساس تجربیات است و کارایی بیشتری دارد[۳۶].

[1] fuzzy logic controller
[2] particle swarm optimization

در سال ۲۰۱۵ روش جدیدی برای طراحی و تنظیم بهینه از کنترل‌کننده منطقی فازی نوع ۲ برای نوع PID پیشنهاد شد که برای کنترل ردیابی مسیر ربات موازی دلتا استفاده شده است. با توجه به پیچیدگی ساختار کنترل‌کننده، این روش به دو مرحله تقسیم می‌شود. روش ارائه شده با یک مساله طراحی بهینه از IT2FLC آغاز می شود و سپس، یک مساله بهینه‌سازی چند منظوره برای تنظیم پارامترهای IT2FLC طراحی شده است. نتایج شبیه‌سازی، عملکرد کنترل بهینه شده در مورد کاربرد در ربات موازی دلتا را تایید می‌کند[۳۷].

لو و لیو همچنین در سال ۲۰۱۶ یک روش طراحی و بهینه‌سازی کنترل منطق فازی نوع ۲ پیشنهاد کردند و عملکرد آن را بر روی مساله کنترل ردیابی مسیر ربات دلتا مورد ارزیابی قرار دادند. شبیه سازی کنترل‌کننده بهینه فازی نوع دوم در حضور عدم قطعیت داخلی و خارجی اعتبار این کنترل‌کننده را نشان می‌دهد. همچنین، تجزیه و تحلیل کنترل منطق فازی نوع ۱ بهینه شده برای کنترل مسیر نهایی نیز ربات دلتا انجام شد. عملکرد کنترل‌کننده در حضور سطوح مختلف عدم قطعیت داخلی و خارجی مورد آزمایش قرار گرفت و در نهایت این دو روش کنترل فازی بهینه را با هم مقایسه کردند. نتایج شبیه‌سازی و مقایسه این دو روش نشان می‌دهد که کنترل‌کننده فازی نوع دوم که بهینه شده است، می تواند عملکرد مسیریابی بهتری را فراهم کند[۳۸].

چن و همکاران خود در سال ۲۰۱۷ یک روش برنامه‌ریزی مسیریابی مطلوب را برای ربات دلتا بر اساس منحنی لاما ارائه دادند. بر اساس منحنی لامای سه پارامتر، بهینه‌سازی برای به دست آوردن پارامترهای مطلوب از جمله دستور منحنی و شعاع محوری کوچک و بزرگ تحت محدودیت‌های سینتیک و دینامیک انجام می‌شود. نتایج شبیه‌سازی نشان می‌دهد که این روش برنامه‌ریزی مسیر بهینه می‌تواند مسیر صاف را در فضای دکارتی به دست آورد و اجازه می‌دهد تا فرآیند را برای سرعت بالا برای ربات دلتا با خطای ردیابی کم انجام شود[۳۹].

در سال ۲۰۱۹ آقای زیدان و همکاران خود یک روش جدید تنظیم اتوماتیک برای یک کنترل‌کننده حرکت P/PI یک ربات دلتا پیشنهاد دادند. تنظیم خودکار با استفاده از الگوریم بهینه‌سازی چندمنظوره طراحی شده که با

کمک الگوریتم جهانی A226 به دست آمده است. ساختار کنترل اجرایی شامل دو کنترل‌کننده است: یک کنترل کننده پیش فرض بر اساس یک مدل دینامیکی معکوس ربات و یک کنترل‌کننده برای جبران اثرات غیرمنتظره. تنظیم خودکار در بهینه‌سازی پارامترهای کنترل در سه مرحله حاصل می‌شود. در مرحله اول پارامترهای کنترل بازخورد پس از نادیده گرفتن ترم کنترل پیشخورد محاسبه می‌شوند. هدف، به حداقل رساندن خطای موقعیت در ردیابی است که برای شناخت پارامترهای مدل دینامیکی در مرحله دوم استفاده می‌شود. پس از آن ترم جبران پیشخورد، بر اساس مسیر مطلوب محاسبه می‌شود. پس از اضافه کردن کنترل‌کننده پیشخورد در مرحله نهایی پارامترهای P/PI دوباره بهینه می‌شوند. نتایج تجربی ارائه شده ثابت کردند که تنظیم خودکار پارامترهای کنترل همراه با شناسایی دینامیک ربات، عملکرد ردیابی مسیر مطلوب را به میزان قابل توجهی بهبود بخشیده است[۴۰].

فصل ۲: سینماتیک و دینامیک ربات

۲-۱ ربات موازی دلتا

ربات دلتا متعلق به گروه ربات‌های موازی می‌باشد. ربات‌های موازی دارای اتصالات چندگانه هستند که با هدف انجام حرکت، به مجری نهایی متصل می‌باشند. در اغلب موارد، موتورهایی که جهت هدایت ربات‌ها به کار می‌روند در قسمت ثابتی طراحی شده‌اند. در هنگامی که جهت جابجایی و بلند کردن، سرعت عمل لازم باشد، این دسته از ربات‌ها می‌توانند عملکرد عالی داشته باشند. ربات‌هایی که موازی نیستند به عنوان ربات سری شناخته می‌شوند. ربات‌های سری فقط یک اتصال به مجری نهایی دارند و غالبا موتورها به قسمت متحرک ربات متصل هستند. هر کدام از انواع ربات‌های سری یا موازی دارای نقاط ضعف و قدرتی می‌باشند.

ربات‌های سری دارای فضای کاری ساده و بزرگی می‌باشند که این مزیت باعث می‌شود که ربات‌های سری در کارخانه‌ها زمانی که وظیفه حمل اتاقک‌های نسبتا بزرگ را داشته باشند، موثرتر عمل کنند. ربات‌های سری ضعف‌های زیادی نیز دارند. در هر لینک ربات سری به دلیل ساختار سری این ربات‌ها، هر موتور باید بر وزن و اینرسی موتورهای پایین غلبه کند که این امر منجر به کندی سرعت و کارآیی کمتر آن می‌شود. علاوه بر این خطایی که برای هر لینک وجود دارد، در اتصالاتی که ترکیبی از لینک‌ها هستند نیز خطای بزرگتری در موقعیت مجری نهایی وجود خواهد داشت که این امر باعث دقت پایین ربات سری می‌شود.

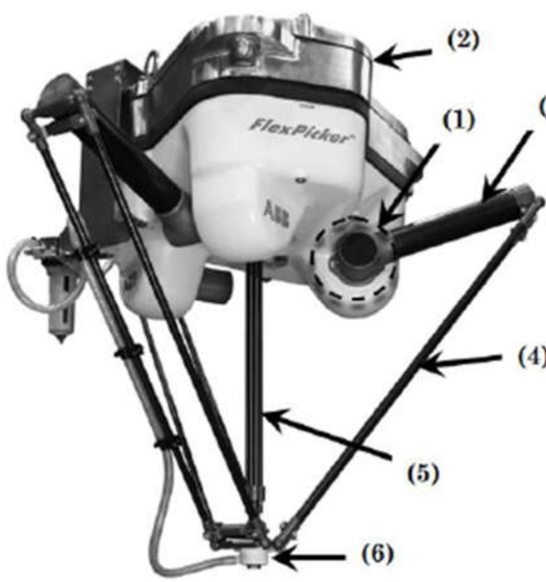
ربات‌های موازی نیز نقاط قوت و ضعفی دارند که بزرگترین نقطه ضعف این ربات‌ها، فضای کاری پیچیده و کوچک آن‌ها است، در نتیجه در مواقعی که مراحل کار در یک محیط کوچک انجام می‌شوند و نیاز به فضای کاری بزرگی ندارند، ربات‌های موازی مورد استفاده قرار می‌گیرند. از آن‌جایی که موتور این ربات‌ها بر روی صفحه ثابتی قرار می‌گیرد، وزن و اینرسی قسمت‌های متحرک کمتر است و این امر موجب سرعت عمل بیشتر و شتاب بالاتر می‌شود. به دلیل استفاده از لینک‌های متعدد در ساختار ربات موازی، وزن بار در مجری نهایی تقسیم می‌شود و این امر باعث عملکرد بهتر ربات در برداشتن و گذاشتن بار می‌شود.

جدول ۱-۲. مقایسه نقاط قوت و ضعف ربات سری و موازی[۴۱]

ویژگی	ربات سری	ربات موازی
فضای کاری	بزرگ	کوچک و پیچیده
حل سینماتیک مستقیم	آسان	سخت
حل سینماتیک معکوس	سخت	آسان
خطای موقعیت	متراکم و انباشته	عادی
خطای نیرو	عادی	متراکم
ماکزیمم نیرو	محدود شده توسط نیروی عملگر	مجموع همه نیروهای عملگر
سختی	کم	زیاد
مشخصه‌های دینامیکی	فقیر	خیلی زیاد
مدل‌سازی و حل دینامیک	دارای روابط ساده	خیلی پیچیده
اینرسی	بزرگ	کوچک
نسبت ظرفیت بار به وزن ربات	پایین	بالا
سرعت و شتاب	پایین	بالا
دقت	پایین	بالا
کالیبراسون	روابط ساده	پیچیده
نسبت فضای کاری به اندازه ربات	بالا	پایین

از دیگر مزیت‌های ربات موازی می‌توان به دقت بالای آن‌ها اشاره کرد. علت این امر این است که بر خلاف ربات-های سری، خطای مرتبط با مجری نهایی ترکیبی نبوده و به صورت میانگین می‌باشد. همچنین چندین لینک بطور همزمان مجری نهایی را هدایت می‌کنند که این امر منجر به کشش کمتر می‌شود. در جدول فوق نقاط ضعف و قوت ربات‌های سری و موازی ارائه شده است.

شکل (۱-۲) مولفه‌های اصلی این ربات را نشان می‌دهد که از سه یا چهار زنجیره سینماتیک حلقه بسته تشکیل شده است. این ربات دارای سه درجه آزادی است. متوازی‌الاضلاع‌ها جهت گیری دائمی بین صفحه ثابت و متحرک را تضمین می‌کنند که فقط حرکات انتقالی صفحه متحرک را اجازه می‌دهد. مجری نهایی بازوی ربات در صفحه متحرک قرار دارد [۴۲].



شکل ۱-۲. اجزای تشکیل دهنده ربات دلتا

ترکیبی از حرکات محدود شده بازوهای اتصال صفحه متحرک به صفحه پایه منجر به سه درجه آزادی می‌شود. به عنوان یک گزینه، با یک محور چرخشی اضافی در نقطه مرکز مجری نهایی، ۴ درجه آزادی امکان‌پذیر است.

با توجه به شکل ۱-۲ ربات تشکیل شده از:

۱. سه محرک یا عملگر
۲. صفحه ثابت یا پایه
۳. بازوی ربات بالا
۴. بازوی ربات پایین
۵. بازوی چرخشی (اختیاری، ۴ درجه آزادی)
۶. صفحه متحرک.

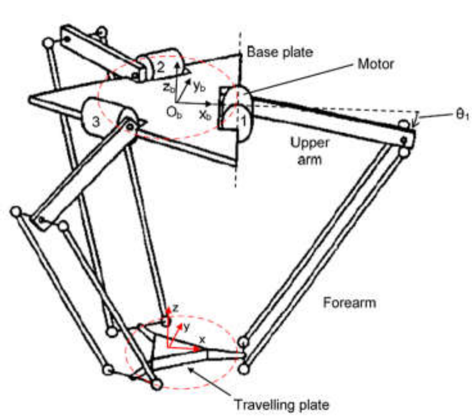
بازوهای بالایی ربات به طور مستقیم به محرک‌ها (عملگرها) متصل می‌شوند تا پایداری بالا را تضمین کنند و سه محرک بر روی صفحه پایه با فاصله ۱۲۰ درجه قرار دارند (به طورثابت و صلب). هر یک از سه بازوی پایینی ربات شامل دو میله موازی تشکیل شده است که بازوی بالایی را با صفحه متحرک از طریق مفاصل کروی متصل می‌کند. این امر باعث می‌شود نیروهای اصطکاک کمتری ایجاد شود. یک میله چهارم، محور چرخشی، برای مکانیسم ربات به عنوان یک گزینه انتخابی در دسترس است. محرک این میله در سمت بالای صفحه پایه ربات متصل می‌شود. میله دورانی به طور مستقیم به ابزار متصل می‌شود و برای حرکت چرخشی اضافی تضمین می‌شود[۴۳].

۲-۲ سینماتیک معکوس

هدف از تحلیل سینماتیک معکوس برای ربات موازی، محاسبه دقیق موقعیت عملگرها در هر موقعیت خاص از مجری نهایی است. بنابراین در تحلیل سینماتیک معکوس با داشتن موقعیت و جهت‌گیری مجری نهایی، باید موقعیت عملگرها را تعیین کرد. در این بخش به استخراج معادلات سینماتیک معکوس پرداخته شده است. این امر با استفاده از هندسه ربات و روش معادلات حلقه بسته انجام شده است.

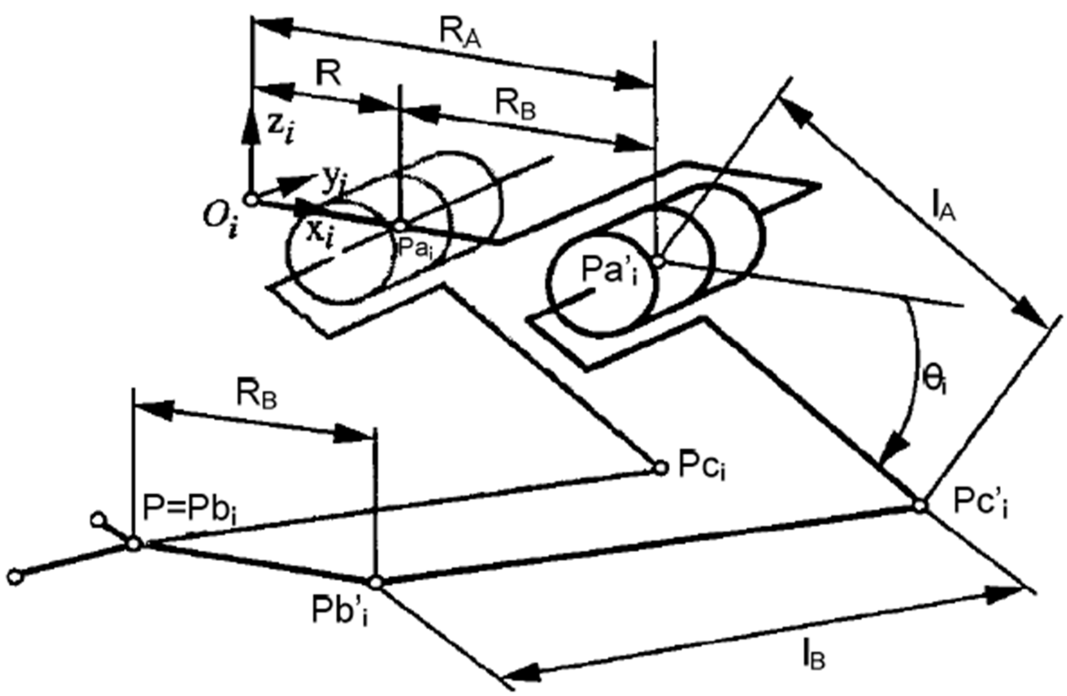
همانطور که در شکل (۲-۲) مشاهده می‌شود، از آنجایی که ربات دلتای ۳ درجه آزادی یک ساختار حلقه بسته است، محاسبه سینماتیک بسیار دشوار است. جهت ساده سازی مدل و کاهش تعداد پارامترها، فرض‌های زیر در نظر گرفته می‌شوند:

۱. صفحه متحرک همیشه با صفحه پایه موازی باقی می ماند و جهت‌گیری آن پیرامون محور عمود بر صفحه پایه به طور مداوم صفر است. بنابراین، مفاصل نوع متوازی‌الاضلاع (ساعد) را می توان با میله‌بهای ساده بدون تغییر رفتار سینماتیک ربات جایگزین کرد.



شکل۲-۲. تشریح یک ربات دلتا که در آن صفحه متحرک همیشه موازی با صفحه پایه است.

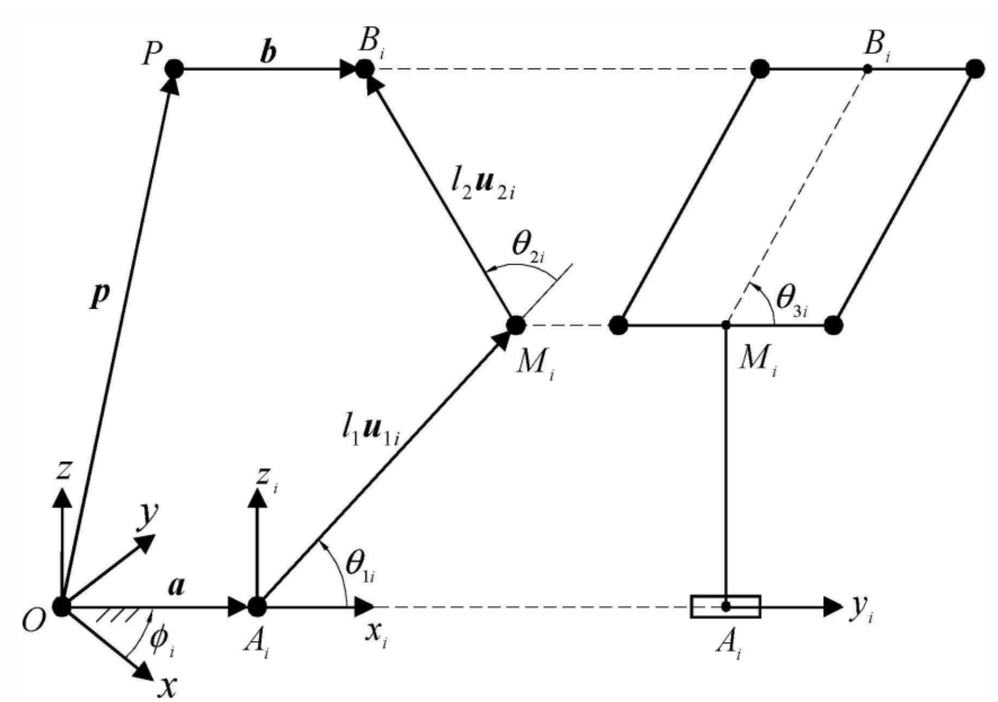
۲. مفاصل دورانی (بین صفحه پایه و بازوهای بالایی و بین ساعدها و صفحه متحرک) مطابق با شکل (۲-۲) عینا در یک دایره قرار می‌گیرند. بنابراین مطابق با شکل (۳-۲) صفحه متحرک را می‌توان با یک نقطه P که سه ساعد متصل به آن قرار دارد جایگزین کرد.



شکل۳-۲. نمایش مدل ساده شده‌ای از ربات دلتای سه درجه آزادی

انتخاب مختصات مرجع همان طور که در شکل ۲-۲ نشان داده شده در مرکز دایره با z به سمت بالا ومحور x عمود بر محور موتور۱ است. به دلیل تقارن بازوهای ربات دلتا، هر بازو می‌تواند به طور جداگانه رفتارکند[۴۴].

برای تجزیه و تحلیل سینماتیک معکوس، هر زنجیره سینماتیک به طور جداگانه در نظر گرفته می‌شود. هندسه هر سه ساق ربات دلتا دقیقا مشابه هم هستند که می‌توان معادلات را برای سه ساق تعمیم داد. تنها تفاوت میان معادلات مربوط به زنجیره سینماتیک مربوط به هر ساق، زاویه آنها با دستگاه مختصات مرجع می‌باشد. φ_i زاویه هر ساق با دستگاه مختصات مرجع را نشان می‌دهد و مقدار ثابتی دارد. برای سادگی می‌توان دستگاه مختصات مرجع را روی یک ساق قرار داد و معادلات را برای ساق اول با زاویه صفر به دست آورد. سپس با در نظر گرفتن اینکه هر زنجیره سینماتیک ۱۲۰ درجه نسبت به هم اختلاف دارند، معادلات را برای دو ساق دیگر نیز به دست آورد. شکل ۴-۲ می‌تواند زنجیره سینماتیک مربوط به هر کدام از ساق‌های ربات دلتا را نشان دهد. متغیر i در معادلات نشان دهنده شماره ساق است.



شکل ۴-۲. حلقه سینماتیکی بسته [۱۲]

با توجه به هندسه ربات، معادلات حلقه بسته مطابق شکل ۴-۲ به صورت زیر است:

$$l_{1i}u_{1i} + l_{2i}u_{2i} = (p + b - a)^i \quad (۲-۱)$$

که در آن

$$u_{1i} = \begin{bmatrix} \cos\theta_{1i} \\ 0 \\ \sin\theta_{1i} \end{bmatrix} \quad (۲-۲)$$

$$u_{1i} = \begin{bmatrix} \sin\theta_{3i}\cos(\theta_{1i} + \theta_{2i}) \\ \cos\theta_{3i} \\ \sin\theta_{3i}\sin(\theta_{1i} + \theta_{2i}) \end{bmatrix}$$

$$b^i = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix} ،a^i = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} ، p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

$$p^i = R^T p \,, R = \begin{bmatrix} \cos\varphi_i & -\sin\varphi_i & 0 \\ \sin\varphi_i & \cos\varphi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

همانطور که گفته شد در مساله سینماتیک معکوس، مجهولات مساله موقعیت عملگرها می‌باشد که متغیرهای θ_i موقعیت عملگرها را بیان می‌کنند. برای به دست آوردن این متغیرها باید دستگاه معادلات (۱-۲) حل شود. جهت حل برای یک ساق، ابتدا با جای‌گذاری پارامترهای (۲-۲) در معادلات (۱-۲)، معادلات (۳-۲) حاصل می‌شود.

$$a - b - p_1\cos\varphi - p_2\sin\varphi + l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2)\cos\theta_3 = 0 \quad (۲-۳-۱)$$

$$p_1\sin\varphi - p_2\cos\varphi + l_2\cos\theta_3 = 0 \quad (۲-۳-۲)$$

$$l_1 \sin\theta_1 - p_3 + l_2 \sin(\theta_1+\theta_2)\sin\theta_3 = 0 \quad (۲-۳-۳)$$

برای ساده سازی، پارامترهای (۲-۴) به طور کلی به صورت زیر تعریف می‌شود.

$$
\begin{aligned}
b_{1i} &= \cos\varphi_i\, p_1 + \cos\varphi_i\, p_2 + b - a \quad (۲-۴)\\
b_{2i} &= -\sin\varphi_i\, p_1 + \cos\varphi_i\, p_2\\
b_{3i} &= p_3\\
g_{1i} &= l_1 + l_2 \cos\theta_{2i} \sin\theta_{3i}\\
g_{2i} &= l_2 \sin\theta_{2i} \sin\theta_{3i}\\
k_i &= (b_{1i}^2 + b_{2i}^2 + b_{3i}^2 - l_1^2 - l_2^2)/(2 l_1 l_2 \sin\theta_{3i})
\end{aligned}
$$

از معادله (۲-۳-۲) داریم:

$$l_2 \cos\theta_3 = p_1 \sin\varphi - p_2 \cos\varphi \quad (۵-۲)$$

و از آنجایی که $b_{2i} = -\sin\varphi_i\, p_1 + \cos\varphi_i\, p_2$، پس معادله (۲-۶) حاصل می‌شود و در نتیجه آن معادله، یکی از مجهولات مساله در (۲-۷) به دست می‌آید.

$$l_2 \cos\theta_3 = b_2 \quad (۲-۶)$$

$$\theta_3 = \cos^{-1}\left(\frac{b_2}{l_2}\right) \quad (۲-۷)$$

اگر معادلات (۱-۳-۲) و (۳-۳-۲) با توجه به پارامترهای تعریف شده (۴-۲) به صورت زیر بازنویسی شود:

$$l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2)\cos\theta_3 = b_1 \quad (۲-۸-۱)$$

$$l_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2)\sin\theta_3 = b_3 \quad (۲-۸-۲)$$

و همچنین اگر مربع دو طرف معادله‌های (۱-۸-۲) و (۲-۸-۲) با هم جمع شود، معادله زیر حاصل می‌شود.

$$(l_1 \cos\theta_1 + l_2 \cos(\theta_1+\theta_2)\cos\theta_3)^2 + (l_1 \sin\theta_1 + l_2 \sin(\theta_1+\theta_2)\sin\theta_3)^2 = {b_1}^2 + {b_3}^2 \qquad (۲-۹)$$

پس از ساده سازی معادله (۹-۲)، معادله (۱۰-۲) به صورت زیر حاصل می‌شود.

$${b_1}^2 + {b_2}^2 + {b_3}^2 - {l_1}^2 - {l_2}^2 = 2l_1l_2 \sin\theta_3 \cos\theta_2 \qquad (۲-۱۰)$$

از معادله (۲-۱۰)، یکی دیگر از مجهولات مساله سینماتیک معکوس در (۲-۱۱) به دست می‌آید.

$$\theta_2 = \cos^{-1}\frac{{b_1}^2 + {b_2}^2 + {b_3}^2 - {l_1}^2 - {l_2}^2}{2l_1l_2\sin\theta_3} \qquad (۱۱-۲)$$

با کمک روابط مثلثاتی، معادله (۱-۳-۲) به صورت (۱۲-۲) بازنویسی می‌شود و از آن معادله (۱۳-۲) و در نتیجه معادله (۱۴-۲) حاصل می‌شود:

$$l_1\cos\theta_1 + l_2\sin\theta_3\cos\theta_1\cos\theta_2 - l_2\sin\theta_3\, sin\,\theta_1\sin\theta_2 = b_1 \qquad (۱۲-۲)$$

$$\cos\theta_1\,(l_1 + l_2\sin\theta_3\cos\theta_2) = b_1 + l_2\sin\theta_3\, sin\,\theta_1\sin\theta_2 \qquad (۱۳-۲)$$

$$cos\,\theta_1 = \frac{b_1 + l_2\, sin\,\theta_3\, sin\,\theta_1\, sin\,\theta_2}{l_1 + l_2\, sin\,\theta_3\, cos\,\theta_2} \qquad (۱۴-۲)$$

همچنین با کمک روابط مثلثاتی معادله (۳-۳-۲) نیز به صورت (۱۵-۲) بازنویسی و از آن معادله (۱۶-۲) و در نتیجه معادله (۱۷-۲) حاصل می‌شود.

$$l_1\,sin\,\theta_1 + l_2 \sin\theta_3 \sin\theta_1 \cos\theta_2 + l_2 \sin\theta_3\, cos\,\theta_1 \sin\theta_2 = b_3 \qquad (۲-۱۵)$$

$$\sin\theta_1\,(l_1 + l_2 \sin\theta_3 \cos\theta_2) = b_3 - l_2 \sin\theta_3\, cos\,\theta_1 \sin\theta_2 \qquad (۲-۱۶)$$

$$sin\,\theta_1 = \frac{b_3 - l_2 \sin\theta_3\, cos\,\theta_1 \sin\theta_2}{l_1 + l_2\, sin\,\theta_3\, cos\,\theta_2} \qquad (۲-۱۷)$$

با توجه به پارامترهای تعریف شده (۴-۲)، معادله (۱۳-۲) به صورت (۱۸-۲) بازنویسی می‌شود.

$$g_1 \cos\theta_1 = b_1 + g_2\, sin\,\theta_1 \qquad (۲-۱۸)$$

پس از جای‌گذاری معادله (۱۷-۲) در (۱۸-۲) و ساده‌سازی آن، معادله (۱۸-۲) به فرم (۱۹-۲) درمی‌آید.

$$({g_1}^2 + g_2) \cos\theta_1 = b_1 g_1 + b_3 g_2 \qquad (۲-۱۹)$$

همچنین با توجه به پارامترهای تعریف شده (۴-۲)، معادله (۱۶-۲) به صورت (۲۰-۲) بازنویسی می‌شود.

$$g_1\, sin\,\theta_1 = b_3 - g_2\, cos\,\theta_1 \qquad (۲-۲۰)$$

پس از جای‌گذاری معادله (۱۴-۲) در (۲۰-۲) و ساده‌سازی آن، معادله (۲۰-۲) به فرم (۲۱-۲) درمی‌آید.

$$({g_1}^2 + g_2) \sin\theta_1 = b_3 g_1 - b_1 g_2 \qquad (۲-۲۱)$$

حال، با تقسیم معادله (۲۱-۲) بر معادله (۱۹-۲)، مجهول دیگر مساله سینماتیک معکوس ربات دلتا در (۲۲-۲) به دست می‌آید.

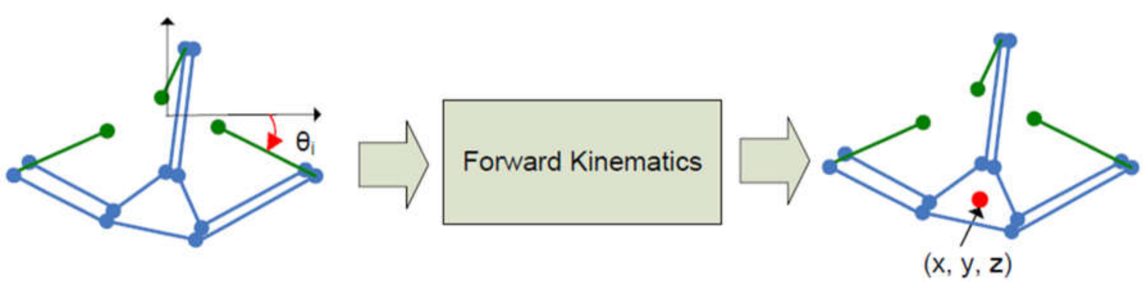
$$\theta_1 = \tan^{-1}\left(\frac{b_3 g_1 - b_1 g_2}{b_1 g_1 + b_3 g_2}\right) \qquad (۲-۲۲)$$

با توجه به پارامترهای تعریف شده (۴-۲)، نتایج سینماتیک معکوس که برای هر سه ساق ربات تعمیم داده شده است، در (۲۳-۲) قابل مشاهده می‌باشد.

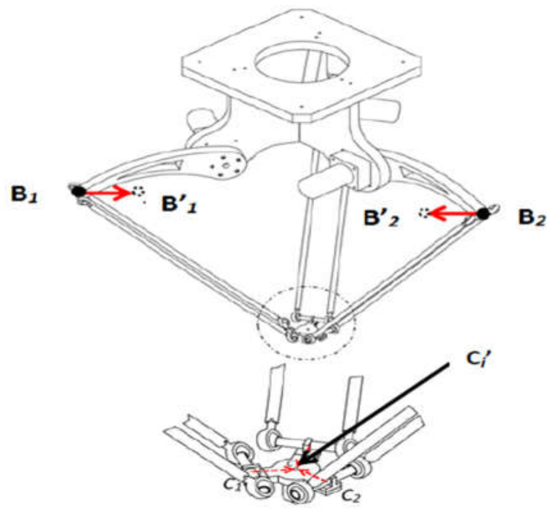
$$\theta_{3i} = \cos^{-1}\left(\frac{b_{2i}}{l_2}\right)$$
$$\theta_{2i} = \cos^{-1}(k_i)$$
$$\theta_{1i} = atan2(-g_{2i}b_{1i} + g_{1i}b_{3i}, g_{1i}b_{1i} + g_{2i}b_{3i}) \qquad (۲-۲۳)$$

۳-۲ سینماتیک مستقیم [۴۷]

سینماتیک مستقیم همچنین به نام سینماتیک مستقیم، تعیین موقعیت (x، y، z) مجری نهایی (صفحه متحرک) با توجه به پیکربندی هر زاویه θ_i از مفاصل دورانی فعال است.



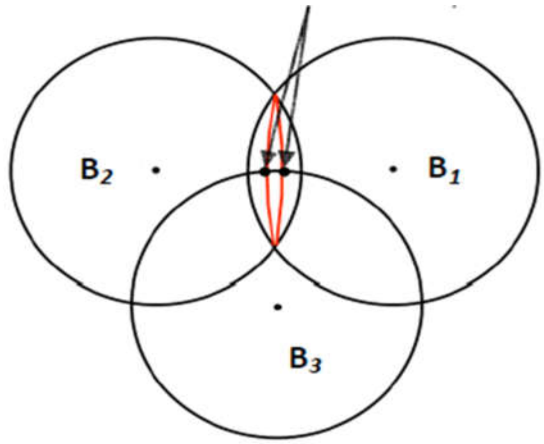
شکل ۵-۲. سینماتیک مستقیم زوایای θ_3 و θ_2 و θ_1 مفاصل ربات به موقعیت در مختصات z و y و x



شکل ۶-۲. پیکربندی انتخاب شده برای تجزیه و تحلیل سینماتیک مستقیم

برای حل سینماتیک مستقیم ابتدا سه کره هر کدام را با مرکزیت در آرنج B_i ازهر زنجیره بازوی ربات و با طول ساعد l_b به عنوان شعاع در نظر گرفته شده است. مدل سینماتیک مستقیم برای یک ربات موازی دلتا می‌تواند با کمک تقاطع بین این سه کره محاسبه شود. هنگام تجسم این سه کره، آنها در دو مکان دروسط متقاطع می شوند.

یک نقطه تقاطع جایی که z مثبت است و یک نقطه تقاطع که مختصات z منفی است. بر اساس دستگاه مختصات پایه که در آن محور z- مثبت است (به طرف بالا)، وقتی که مختصات z منفی باشد، مرکز صفحه متحرک، نقطه‌ی تقاطع است. شکل (۷-۲) تقاطع بین سه کره، جایی که دو کره در یک دایره متقاطع است و سپس کره سوم این دایره را در دو مکان قطع می‌کند را نشان می‌دهد.



شکل ۷-۲. دو کره متقاطع شده در یک دایره و یک کره سوم که دایره را در دو مکان قطع می‌کند.

بر اساس مدل پیشنهادی ساخته شده، بردار B_i که مختصات آرنج را توصیف می‌کند برای هر یک از سه بازو به صورت زیر می‌باشد:

$$B_i = [f + l_a \cos(\theta_i) \quad 0 \quad l_a \sin(\theta_i)]^T \tag{۲-۲۴}$$

برای محاسبه سینماتیک مستقیم، مرکز کره‌ها به داخل حرکت داده می‌شود (از نقاط B_iبه نقاط $\acute{B}_i$ به ترتیب برای ۳,۲,۱ = i). پس از این انتقال، سه کره در نقطه مرکز صفحه متحرک متقاطع می‌شوند.

$$B_i' = [(f - e) + l_a \cos(\theta_i) \quad 0 \quad l_a \sin(\theta_i)]^T \tag{۲-۲۵}$$

درحالتی که $e = B_1\acute{B}_1 = B_2\acute{B}_2 = B_3\acute{B}_3$ طول فاصله جابجایی است که به وضوح در شکل (۶-۲) نشان داده شده است. برای رسیدن به یک ماتریس که تمام سه نقطه در مختصات پایه را توضیح می‌دهد باید B_i در ماتریس دوران R_z^0 ضرب شود.

$$R_z^o = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{۲-۲۶}$$

درنتیجه ماتریس B_i' :

$$B' = R_z^o\, B_i' = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} [(f-e) + l_a \cos(\theta_i) \quad 0 \quad l_a \sin(\theta_i)]^T \tag{۲-۲۷}$$

و سپس:

$$B' = \begin{bmatrix} \cos(\alpha)\, B_{i,x}' \\ \sin(\alpha)\, B_{i,x}' \\ B_{i,z}' \end{bmatrix} = \begin{bmatrix} \cos(\alpha)[(f-e) + l_a \cos(\theta_i)] \\ \sin(\alpha)[(f-e) + l_a \cos(\theta_i)] \\ l_a \sin(\theta_i) \end{bmatrix} = \begin{bmatrix} S_{i.x} \\ S_{i.y} \\ S_{i.z} \end{bmatrix} \tag{۲-۲۸}$$

سپس می‌توان سه کره را با طول ساعد l_b به عنوان شعاع و مراکز آنها را در B_i ایجاد کرد. معادله کلی برای یک کره به صورت (۲-۲۹) است:

$$(x - S_{i.x})^2 + (y - S_{i,y})^2 + (s - S_{i,z})^2 = r^2 \tag{۲-۲۹}$$

این سه معادله برای سه لینک به ترتیب $i = 1,2,3$، می‌باشد. برای لینک (۱) بازو به سمت بالا به موازات محور x و عمود بر محور y است، بنابراین زاویه چرخش $\alpha_1 = 0$، اما دو لینک دیگر دارای زاویه چرخش $\alpha_2 = 120$ برای لینک (۲) و $\alpha_3 = -120$ برای لینک (۳) هستند.

$$\begin{aligned} &(x - \cos(\alpha_1)\,[(f-e) + l_a \cos(\theta_1)])^2 + (y - \sin(\alpha_1)\,[(f-e) + l_a \cos(\theta_1)])^2 \\ &\qquad + (z - l_a \sin(\theta_1))^2 = l_b^2 \\ &(x - \cos(\alpha_2)\,[(f-e) + l_a \cos(\theta_2)])^2 + (y - \sin(\alpha_2)[(f-e) + l_a \cos(\theta_2)])^2 \\ &\qquad + (z - l_a \sin(\theta_2))^2 = l_b^2 \end{aligned} \tag{۲-۳۰}$$

$$(x-\cos(\alpha_3)\,[(f-e)+l_a\cos(\theta_3)])^2+(y-\sin\,(\alpha_3)[(f-e)+l_a\cos(\theta_3)])^2 \\ +(z-l_a\sin(\theta_3))^2=l_b^2$$

پس از جای‌گذاری مقادیر $\alpha_1 = 0$ و $\alpha_2 = 120$ و $\alpha_3 = -120$ در معادله فوق، معادلات سه کره حاصل می-شود:

$$(x-[(f-e)+l_a\cos(\theta_1)])^2+(y)^2+(z-l_a\sin(\theta_1))^2=l_b^2 \quad (۲-۳۱)$$

$$\left(x+{}^1\!/_2\,[(f-e)+l_a\cos(\theta_2)]\right)^2+\left(y-({}^{\sqrt{3}}\!/_2)[(f-e)+l_a\cos(\theta_2)]\right)^2 \\ +(z-l_a\sin(\theta_2))^2=l_b^2$$

$$\left(x+{}^1\!/_2\,[(f-e)+l_a\cos(\theta_3)]\right)^2+\left(y+({}^{\sqrt{3}}\!/_2)[(f-e)+l_a\cos(\theta_3)]\right)^2 \\ +(z-l_a\sin(\theta_3))^2=l_b^2$$

با مرتب کردن این معادله، معادلات زیر حاصل می‌شود:

$$(x+k_{11})^2+(y+k_{12})^2+(z+k_{13})^2=l_b^2 \quad (۲-۳۲)$$
$$(x+k_{21})^2+(y+k_{22})^2+(z+k_{23})^2=l_b^2$$
$$(x+k_{31})^2+(y+k_{32})^2+(z+k_{33})^2=l_b^2$$

که در آن:

$$k_{11}=(f-e)+l_a\cos(\theta_1) \quad (۲-۳۳)$$
$$k_{12}=o$$
$$k_{13}=-l_a\sin(\theta_1)$$
$$k_{21}={}^1\!/_2\,[(f-e)+l_a\cos(\theta_2)]$$
$$k_{22}=-({}^{\sqrt{3}}\!/_2)[(f-e)+l_a\cos(\theta_2)]$$
$$k_{23}=-l_a\sin(\theta_2)$$

$$k_{31} = {}^{1}\!/_{2}\,[(f - e) + l_a \cos(\theta_3)]$$
$$k_{32} = {}^{\sqrt{3}}\!/_{2}\,[(f - e) + l_a \cos(\theta_3)]$$
$$k_{33} = -l_a \sin(\theta_3)$$

پس از بازکردن معادله (۲-۳۲)، معادله (۲-۳۴) حاصل می‌شود:

$$x^2 + y^2 + z^2 + 2k_{i1}x + 2k_{i2}y + 2k_{i3}z = l_b^2 - \left({k_{i1}}^2 + {k_{i2}}^2 + {k_{i3}}^2\right) \quad (۲\text{-}۳۴)$$
$$i = 1,2,3$$

با کم کردن معادله (۲-۳۴) با ۲= i از معادله (۲-۳۴) با ۱= i نتیجه زیر حاصل می‌شود:

$$2(k_{11} - k_{21})x + 2(k_{12} - k_{22})y + 2(k_{13} - k_{21})z = (k_{21}^2 + k_{22}^2 + k_{23}^2) - (k_{11}^2 + k_{12}^2 + k_{13}^2) \quad (۲\text{-}۳۵)$$

همچنین با کم کردن معادله (۲-۳۴) با ۳= i از معادله (۲-۳۴) با ۱=i نتیجه زیر حاصل می‌شود:

$$2(k_{11} - k_{31})x + 2(k_{12} - k_{32})y + 2(k_{13} - k_{33})z = (k_{31}^2 + k_{32}^2 + k_{33}^2) - (k_{11}^2 + k_{12}^2 + k_{13}^2) \quad (۲\text{-}۳۶)$$

با ساده سازی، معادله (۲-۳۵) و (۲-۳۶) به صورت (۲-۳۷) درمی‌آیند:

$$a_1x + b_1y + c_1z = d_1 \quad (۲\text{-}۳۷)$$
$$a_2x + b_2y + c_2z = d_2$$

که در آن:

$$a_1 = 2(k_{11} - k_{21}) \quad (۲\text{-}۳۸)$$
$$b_1 = 2(k_{12} - k_{22})$$
$$c_1 = 2(k_{13} - k_{23})$$
$$a_2 = 2(k_{11} - k_{31})$$

$$b_2 = 2(k_{12} - k_{32})$$
$$c_2 = 2(k_{13} - k_{33})$$
$$d_1 = (k_{21}^2 + k_{22}^2 + k_{23}^2) - (k_{11}^2 + k_{12}^2 + k_{13}^2)$$
$$d_2 = (k_{31}^2 + k_{32}^2 + k_{33}^2) - (k_{11}^2 + k_{12}^2 + k_{13}^2)$$

با تنظیم معادلات (۳۷-۲) در یک فرم ماتریس به شکل زیر درمی‌آید:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1 - c_1 z \\ d_2 - c_2 z \end{bmatrix} \tag{۲-۳۹}$$

تعریف می‌شود $\Delta = a_1b_2 - a_2b_1$ ، سپس برای $\Delta \notin 0$ ،

$$\Delta x = (d_1 - c_1 z)\, b_2 - (d_2 - c_2 z)\, b_1 = (b_2 d_1 - b_1 d_2) + (b_1 c_2 - b_2 c_1) z \tag{۲-۴۰}$$
$$\Delta x = (d_2 - c_2 z)\, a_2 - (d_1 - c_1 z)\, a_1 = (a_1 d_2 - a_2 d_1) + (a_2 c_1 - a_1 c_2) z$$

و در نتیجه:

$$x = {}^{\Delta x}/_{\Delta} = \frac{b_2 d_1 - b_1 d_2}{\Delta} + \frac{b_1 c_2 - b_2 c_1}{\Delta} \tag{۲-۴۱}$$
$$y = {}^{\Delta y}/_{\Delta} = \frac{a_1 d_2 - a_2 d_1}{\Delta} + \frac{a_2 c_1 - a_1 c_2}{\Delta}$$

با فرض:

$$f_1 = \frac{b_2 d_1 - b_1 d_2}{\Delta} \quad , \quad f_2 = \frac{b_1 c_2 - b_2 c_1}{\Delta} \tag{۲-۴۲}$$
$$f_x = \frac{b_1 c_2 - b_2 c_1}{\Delta} \quad , \quad f_y = \frac{a_2 c_1 - a_1 c_2}{\Delta}$$

سپس معادلات زیر به دست می‌آید:

$$x = f_1 + f_x z \qquad (۲-۴۳)$$
$$y = f_2 + f_y z$$

با جای‌گزینی معادله (۴۳-۲) در معادله (۳۲-۲) برای i = ۳؛ نتیجه زیر حاصل می‌شود:

$$(1 + f_x^2 + f_y^2)z^2 + 2\left([f_x f_1 + f_x k_{31}] + \left[f_y f_2 + f_y k_{32}\right] + k_{33}\right)z + f_{11}^2 + f_{22}^2 + k_{33}^2 - l_b^2 \qquad (۲-۴۴)$$

عبارات زیر تعریف می‌شود:

$$A = (1 + f_x^2 + f_y^2) \qquad (۲-۴۵)$$
$$B = 2\left([f_x f_1 + f_x k_{31}] + \left[f_y f_2 + f_y k_{32}\right] + k_{33}\right)$$
$$C = f_{11}^2 + f_{22}^2 + k_{33}^2 - l_b^2$$

که در آن:

$$f_{11} = f_1 + k_{31} \qquad (۲-۴۶)$$
$$f_{22} = f_2 + k_{32}$$

برای حل معادله $Az^2+Bz+C=0$ از معادله زیر استفاده می‌شود:

$$z = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \qquad (۲-۴۷)$$

ازمعادله (۴۷-۲)، می توان معادله (۴۳-۲) را ارزیابی کرد.

با توجه به پارامترهای تعریف شده، نتایج کلی به دست آمده برای سینماتیک مستقیم در (۲-۴۸) قابل مشاهده است:

$$x = f_1 + f_x z \qquad (۲-۴۸)$$
$$y = f_2 + f_y z$$

$$z = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

سینماتیک معکوس یک راه حل واحد را ارائه می‌دهد اما سینماتیک مستقیم معمولا دو راه حل دارد. علت آن هم این می‌باشد که زوایای مفصل غیرفعال تشکیل شده بین بازوی بالا و بازو پایین توسط معادلات سینماتیک تعیین نمی‌شود. پس راه حل این است که باید در فضای کاری ربات انتخاب شود.

راه حل خروجی به چهار حالت ممکن است:

۱) راه حل عمومی، دو پاسخ در تقاطع یک دایره و یک کره به دست می‌آید.

۲) راه حل منحصر به فرد، هنگامی که کره مماس بر دایره حاصل از تقاطع دو کره دیگر است. از این رو تنها یک پاسخ ممکن وجود دارد.

۳) راه حل منحصر به فرد، مرکز هر دو کره برهم منطبق شده باشد، در نتیجه تعداد نامحدودی از پاسخ‌ها وجود دارد. این یک پیکربندی بعید برای ساختارها است. به جز وضعیتی که:

$$\theta_1 = \frac{\pi}{2} \quad \theta_2 = \frac{\pi}{2} \quad \theta_3 = \frac{\pi}{2} \qquad (۲-۴۹)$$

۴) بدون راه حل، سه کره در یک نقطه مشترک همدیگر را قطع نمی‌کنند.

۴-۲ دینامیک

در این بخش، مدل دینامیکی ربات بر مبنای مدل سینماتیکی ارائه شده در بخش قبل، استخراج می‌شود. معادلات دینامیک ربات معمولا به فرم مرسوم زیر نوشته می‌شوند:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \qquad (۲-۵۰)$$

که $M(q)$ ماتریس $n \times n$ جرم بازوی ربات، $C(q,\dot{q})$ یک بردار $n \times 1$ نیروی گریز از مرکز و کریولیس و $G(q)$ بردار $n \times 1$ نیروی گرانش است. n تعداد عملگرها را نشان می‌دهد که در این تحقیق، هر دو ترم $C(q,\dot{q})\dot{q}$ و $G(q)$ در یک ماتریس $n \times 1$ با نام $C(q,\dot{q})$ آورده شده است.

برای استخراج مدل دینامیکی جهت اعمال کنترل‌کننده‌های پیشنهادی به آن با هدف ردیابی مسیر مطلوب، از روش لاگرانژ استفاده می‌شود. با توجه به اینکه معادلات قید مکانیزم موازی، در مختصات تعمیم یافته هستند، فرمول‌بندی معادلات لاگرانژ برای سیستم مقید مطابق رابطه زیر بکار گرفته می‌شود.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^{k} \lambda_i \frac{\partial \Gamma_i}{\partial q_j} \qquad (۲-۵۱)$$

که در آن λ_i ضرایب لاگرانژ و Γ_i قیود معادله هستند. معادلات قید ربات موازی از طریق هندسه ربات مطابق با شکل (۴-۲) به صورت زیر استفاده شده است.

$$\Gamma_i = \overline{M_\iota B_\iota}^2 - l_2^{\,2} \qquad (۲-۵۲)$$

که $\overline{M_\iota B_\iota}$ یک بردار شامل مولفه های x و y و z است که در شکل (۴-۲) قابل مشاهده است.

$$\Gamma_i = (M_i B_i)_x^{\,2} + (M_i B_i)_y^{\,2} + (M_i B_i)_z^{\,2} - l_2^{\,2} \qquad (۲-۵۳)$$

مطابق شکل (۴-۲)، بردار به صورت زیر باز می‌شود.

$$\Gamma_i = (P_1 + b\cos\varphi_i - a\,cos\,\varphi_i - l_1\cos\varphi_i\,cos\,\theta_i)^2 + (P_2 + b\sin\varphi_i - a\,sin\,\varphi_i - l_1\sin\varphi_i\,cos\,\theta_i)^2 + (P_3 - l_1\sin\theta_i)^2 - {l_2}^2 \quad (۲-۵۴)$$

تجزیه و تحلیل دینامیک این ربات را می توان با استفاده از سه مختصات تعمیم یافته انجام داد، زیرا این یک مکانیزم سه درجه آزادی است. با این حال، با توجه به سینماتیک پیچیده، سه مختصات فرعی $(p_1،p_2،p_3)$ به مختصات اصلی که موقعیت عملگرها می‌باشد، اضافه شده است. بنابراین مختصات تعمیم یافته برای توصیف سیستم به صورت (۵۵-۲) می باشند.

$$q = [p \quad \theta]^T \quad (۲-۵۵)$$

که p حاوی مولفه‌های موقعیت مجری نهایی یعنی x و y و z ، و θ مولفه‌های موقعیت عملگرها می‌باشند.

برای به دست آوردن تابع لاگرانژین مطابق با معادله (۵۶-۲) نیاز به محاسبه انرژی جنبشی و انرژی پتانسیل ربات است.

$$L = K - U \quad (۲-۵۶)$$

که K انرژی جنبشی و U انرژی پتانسیل کل ربات است. جهت محاسبه انرژی جنبشی، باید انرژی جنبشی مجری نهایی و انرژی جنشی‌های هر ساق که مشابه هم هستند را محاسبه و با هم جمع کرد. انرژی جنبشی کل به صورت (۵۷-۲) خواهد بود:

$$K = K_P + \sum_{i=1}^{3}(K_{1i} + K_{2i}) \quad (۲-۵۷)$$

با جای‌گذاری انرژی جنبشی هر جزء، معادله زیر حاصل می‌شود:

$$K=\frac{1}{2}m_P\sum_{i=1}^{3}\dot{P_i}^2+\sum_{i=1}^{3}\left(\frac{1}{2}(\gamma^2 I_m+I_1)\dot{\theta_i}^2+\frac{1}{2}m_2\sum_{i=1}^{3}\dot{P_i}^2+\frac{1}{2}m_2{I_1}^2\dot{\theta_i}^2\right) \tag{۲-۵۸}$$

همچنین برای محاسبه انرژی پتانسیل نیز باید انرژی پتانسیل‌های همه اجزا را محاسبه و با جمع کرد که به صورت (۵۹-۲) می‌باشد.

$$U=U_P+\sum_{i=1}^{3}(U_{1i}+U_{2i}) \tag{۲-۵۹}$$

با جای‌گذاری انرژی پتانسیل هر جزء، معادله زیر حاصل می‌شود:

$$U=-m_P g_c P_3-\sum_{i=1}^{3}\big(m_1 g l_{c1}\sin\theta_i+m_2 g(P_3+l_1\sin\theta_i)\big) \tag{۲-۶۰}$$

پس با توجه به معادله (۵۶-۲) در نهایت با کم کردن انرژی پتانسیل از انرژی جنبشی، تابع لاگرانژین به صورت زیر حاصل می‌شود.

$$\begin{aligned}L=\frac{1}{2}m_P\sum_{i=1}^{3}\dot{P_i}^2+m_P g_c P_3+\sum_{i=1}^{3}\left(\frac{1}{2}(\gamma^2 I_m+I_1)\dot{\theta_i}^2+\frac{1}{2}m_2\sum_{i=1}^{3}\dot{P_i}^2+\frac{1}{2}m_2{I_1}^2\dot{\theta_i}^2\right)\\+\sum_{i=1}^{3}\big(m_1 g l_{c1}\sin\theta_i+m_2 g(P_3+l_1\sin\theta_i)\big)\end{aligned} \tag{۲-۶۱}$$

برای بدست آوردن معادله دینامیکی ربات، ابتدا نیاز به محاسبه ضرایب لاگرانژ می‌باشد. برای به دست آوردن ضرایب لاگرانژ از رابطه لاگرانژ به صورت (۶۲-۲) استفاده می‌شود.

$$\sum_{i=1}^{k} \lambda_i \frac{\partial \Gamma_i}{\partial q_j} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} - \hat{Q}_j \tag{۲-۶۲}$$

که در آن $\hat{Q}_j$ نیروهای خارجی را نشان می‌دهد. برای استفاده از این معادله، متغیر j از ۱ تا ۳ خواهد بود و باید از سه مختصات تعمیم یافته استفاده کرد.

ابتدا برای مختصات اول روابط زیر برقرار است:

$$\begin{gathered} j = 1 \\ q_1 = P_1 \end{gathered} \tag{۲-۶۳}$$

با جای‌گذاری روابط (۶۳-۲) در معادله (۶۲-۲)، معادله زیر حاصل می‌شود.

$$\begin{aligned} 2\sum_{i=1}^{k} \lambda_i (P_1 + b\cos\varphi_i - a\cos\varphi_i - l_1\cos\varphi_i\cos\theta_i) \\ = \frac{d}{dt}\big((m_P + 3m_2)\dot{P}_1\big) - 0 - f_1 \end{aligned} \tag{۲-۶۴}$$

معادله فوق پس از ساده‌سازی به معادله (۶۵-۲) تبدیل می‌شود.

$$2\sum_{i=1}^{k} \lambda_i (P_1 + b\cos\varphi_i - a\cos\varphi_i - l_1\cos\varphi_i\cos\theta_i) = (m_P + 3m_2)\ddot{P}_1 - f_1 \tag{۲-۶۵}$$

سپس برای مختصات دوم روابط زیر برقرار است:

$$j = 2 \qquad (۲-۶۶)$$
$$q_2 = P_2$$

با جای‌گذاری روابط (۶۶-۲) در معادله (۶۲-۲)، معادله زیر حاصل می‌شود.

$$2\sum_{i=1}^{k}\lambda_i(P_2 + b\,sin\,\varphi_i - a\sin\varphi_i - l_1\sin\varphi_i\cos\theta_i) = \frac{d}{dt}\big((m_P + 3m_2)\dot{P}_2\big) - 0 - f_2 \qquad (۲-۶۷)$$

معادله فوق پس از ساده‌سازی به معادله (۶۸-۲) تبدیل می‌شود.

$$2\sum_{i=1}^{k}\lambda_i(P_2 + b\sin\varphi_i - a\sin\varphi_i - l_1\sin\varphi_i\cos\theta_i) = (m_P + 3m_2)\ddot{P}_2 - f_2 \qquad (۲-۶۸)$$

و سپس برای مختصات سوم نیز روابط زیر برقرار است:

$$j = 3 \qquad (۲-۶۹)$$
$$q_3 = P_3$$

با جای‌گذاری روابط (۶۹-۲) در معادله (۶۲-۲)، معادله زیر حاصل می‌شود.

$$2\sum_{i=1}^{k}\lambda_i(P_3 - l_1\,sin\,\theta_i) = \frac{d}{dt}\big((m_P + 3m_2)\dot{P}_3\big) - (m_P + 3m_2)g - f_3 \qquad (۲-۷۰)$$

معادله فوق پس از ساده‌سازی به معادله (۷۱-۲) تبدیل می‌شود.

$$2\sum_{i=1}^{k}\lambda_i(P_3 - l_1\sin\theta_i) = (m_P + 3m_2)(\ddot{P}_3 - g) - f_3 \qquad (۲-۷۱)$$

حال، با استفاده از سه معادله (۶۵-۲) و (۶۸-۲) و (۷۱-۲)، سه مجهول ضرایب لاگرانژ به دست می‌آیند.

پس از یافتن ضرایب لاگرانژ، مجموعه دوم معادلات مربوط به نیروهای فعال را بدین صورت می‌توان نوشت:

$$Q_j = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} - \sum_{i=1}^{k}\lambda_i\frac{\partial \Gamma_i}{\partial q_j} \qquad (۲-۷۲)$$

که در این معادله Q_j گشتاور محرک‌ها می‌باشد و متغیر j از ۴ تا ۶ می‌باشد و نیز از مختصات اصلی استفاده می‌شود. با استفاده از ضرایب لاگرانژ حاصل و قرار دادن آنها در معادله (۷۲-۲) در نهایت، معادلات دینامیکی ربات به صورت سه معادله دیفرانسیل غیرخطی به صورت (۷۳-۲) به دست می‌آید.

$$\begin{aligned}\tau_i = {} & \frac{d}{dt}\left((\gamma^2 I_m + I_1 + m_1{l_1}^2)\dot{\theta}_i\right) - \left(g\cos\theta_i\,(m_1 l_{c1} + m_2 l_1)\right) \\ & - 2\lambda_i\big((P_1 + b\cos\varphi_i - a\cos\varphi_i \\ & - l_1\cos\varphi_i\cos\theta_i)(l_1\cos\varphi_i\sin\theta_i) \\ & + (P_2 + b\,sin\,\varphi_i - a\sin\varphi_i - l_1\sin\varphi_i\cos\theta_i)(l_1\sin\varphi_i\sin\theta_i) \\ & + (P_3 - l_1\,sin\,\theta_i)(-l_1\,cos\,\theta_i)\big)\end{aligned} \qquad (۲-۷۳)$$

که در آن i از ۱ تا ۳ می‌باشد. پس از مشتق‌گیری و ساده‌سازی معادله فوق به صورت زیر حاصل می‌شود.

$$\begin{aligned}\tau_i = {} & \left(\gamma^2 I_m + I_1 + m_1{l_1}^2\right)\ddot{\theta}_i - \left((m_1 l_{c1} + m_2 l_1)g\cos\theta_i\right) \\ & - 2\lambda_i l_1\left((P_1\cos\varphi_i + P_2\sin\varphi_i + b - a)\sin\theta_i + P_3\,cos\,\theta_i\right)\end{aligned} \qquad (۲-۷۴)$$

به صورت کلی، مرسوم است که معادلات دینامیکی ربات را به شکل ماتریسی متعارفی تبدیل کرد و همچنین برای اهداف کنترلی نیز از این فرم متعارف استفاده می‌شود. معادله دینامیکی ربات دلتا به فرم متعارف ماتریسی در (۷۵-۲) نشان داده شده است.

$$\tau = M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right) \qquad (۲\text{-}۷۵)$$

که در این معادله، $M(\theta)$ یک ماتریسی مربعی 3×3 که به ماتریس اینرسی معروف است و $C\left(\theta, \dot{\theta}\right)$ یک بردار 1×3 می‌باشد.

فصل ۳: کنترل تطبیقی

۳-۱ طراحی کنترل‌کننده تطبیقی

در زبان روزمره کلمه تطبیق به معنای تغییر رفتار برای وفق یافتن با وضع جدید است. کنترل تطبیقی از جمله کنترل‌کننده‌هایی است که بتواند رفتارش را در پاسخ به تغییر دینامیک فرایند و اغتشاش‌ها تغییر دهد. بسیاری از سیستم‌های دینامیکی غیرخطی که بایستی کنترل شوند، پارامترهای نامعلوم دارند که یا ثابت هستند و یا به آهستگی تغییر می‌کنند. هدف اصلی از طراحی کنترل‌کننده در این موارد، این است که با وجود عدم قطعیت در برخی از پارامترهای ربات نظیر جرم و ممان اینرسی جسم میانی، بتوان موقعیت ربات را کنترل نمود. کنترل تطبیقی یک روش برای کنترل این چنین سیستم‌ها است که قادر به مقابله با عدم قطعیت‌های پارامتری می-باشد.

ایده اصلی در کنترل تطبیقی این است که پارامترهای نامعلوم سیستم و یا پارامترهای کنترل‌کننده آن بر اساس سیگنال‌های اندازه‌گیری شده به صورت بهنگام تخمین زده شوند و در انجام محاسبات ورودی کنترل، از پارامترهای تخمین زده شده استفاده شود.

به صورت کلی، معادله دینامیکی ربات مورد نظر به صورت زیر می‌باشد.

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) \quad (۳-۱)$$

که در فصل قبل شرح داده شد. جهت هدف کنترل تطبیقی، باید بتوان مدل دینامیکی فوق را می‌توان به صورت رابطه (۳-۲) نوشت[۳].

$$\tau = YP \quad (۳-۲)$$

که P ماتریس پارامترهای نامعلومی است که باید تخمین زده شوند و Y ماتریس رگرسیون می‌باشد. این ماتریس، شامل همه پارامترهای مدل دینامیکی به جز گشتاورها و پارامترهای نامعلوم است که گشتاورها در سمت چپ معادله قرار دارند و پارامترهای نامعلوم نیز در ماتریس P قرار گرفته‌اند که در ادامه بیشتر توضیح داده خواهد شد.

در این تحقیق فرض بر این است که جرم‌ها و اینرسی‌های اجزای ربات دارای مقدار ثابت و نامعلومی هست. در معادله دینامیکی ربات، همه‌ی جرم‌ها و اینرسی‌ها در سه پارامتر خلاصه می‌شوند که ماتریس P در معادله (۲-۳) را تشکیل می‌دهند. با توجه به نیاز این پارامترها در کنترل‌کننده، این پارامترها باید تخمین زده شوند.

برای بدست آوردن گشتاورهای عملگرها جهت ردگیری مسیر مطلوب، قانون کنترلی به صورت (۳-۳) پیشنهاد می‌شود.

$$\tau = \widehat{M}\left(\ddot{\theta_d} + K_d\dot{e} + K_p e\right) + \hat{C} \qquad (۳-۳)$$

که ماتریس‌های $\widehat{M}$ و $\hat{C}$ دارای پارامترهای نامعلوم مذکور هستند و $\ddot{\theta}_d$ شتاب مسیر مطلوب را نشان می‌دهد. پارامترهای K_p و K_d نیز ماتریس ضرایب کنترل‌کننده هستند. با توجه به معادلات (۱-۳) و (۲-۳)، معادله (۴-۳) را می‌توان نوشت.

$$M\ddot{\theta} + C = YP \qquad (۳-۴)$$

رابطه (۴-۳) با در نظر گرفتن تخمین پارامترهای نامعلوم به صورت (۵-۳) حاصل خواهد شد.

$$\widehat{M}\ddot{\theta} + \hat{C} = Y\widehat{P} \qquad (۳-۵)$$

با کم کردن معادله (۵-۳) از معادله (۴-۳)، عبارت (۶-۳) به دست می‌آید.

$$\left(M - \widehat{M}\right)\ddot{\theta} + C - \hat{C} = Y(P - \hat{P}) \tag{۳-۶}$$

با جای‌گذاری قانون کنترلی (۳-۳) در معادله دینامیکی (۱-۳) و اضافه کردن عبارت $\widehat{M}\ddot{\theta}$ به دو طرف تساوی، عبارت زیر حاصل می‌شود.

$$\left(M - \widehat{M}\right)\ddot{\theta} + C - \hat{C} = \widehat{M}\left(\ddot{e} + K_d\dot{e} + K_p e\right) \tag{۳-۷}$$

با مقایسه معادلات (۶-۳) و (۷-۳) می‌توان دینامیک خطا را با معادله (۸-۳) نمایش داد.

$$\widehat{M}\left(\ddot{e} + K_d\dot{e} + K_p e\right) = Y(P - \hat{P}) \tag{۳-۸}$$

معادله (۸-۳) را میتوان به صورت (۹-۳) بازنویسی نمود.

$$\ddot{e} = -K_d\dot{e} - K_p e + \widehat{M}^{-1}Y(P - \hat{P}) \tag{۳-۹}$$

فرم فضای حالت برای سیستم را با توجه به معادله (۹-۳) می‌توان به صورت (۱۰-۳) نوشت.

$$\dot{X} = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} X + Bw \tag{۳-۱۰}$$

که در آن $X = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$ و $w = \widehat{M}^{-1}Y\left(P - \hat{P}\right)$. از آن‌جایی که در حالت کلی رابطه زیر برقرار است،

$$\dot{X} = AX + Bw \tag{۳-۱۱}$$

پس با مقایسه معادله فوق با معادله (۱۰-۳) نتایج زیر حاصل می‌شود.

$$B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{۳-۱۲}$$

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}$$

همچنین متغیرهای حالت X به صورت زیر هستند:

$$\begin{cases} x_1 = e \\ x_2 = \dot{e} \end{cases} \tag{۳-۱۳}$$

و معادلات حالت به شکل زیر است.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -k_p x_1 - k_d x_2 + w \end{cases} \tag{۳-۱۴}$$

برای طراحی کنترل‌کننده تطبیقی از روش لیاپانوف استفاده می‌شود. بدین منظور تابع لیاپانوف به صورت زیر تعریف می‌شود[۳].

$$V = \frac{1}{2} X^T S X + \frac{1}{2\gamma} (P - \hat{P})^T (P - \hat{P}) \tag{۳-۱۵}$$

جهت تضمین پایداری سیستم بایستی مشتق تابع لیاپانوف منفی باشد. لذا مشتق تابع لیاپانوف به صورت (۱۶-۳) خواهد بود.

$$\dot{V} = \frac{1}{2} \dot{X}^T S X + \frac{1}{2} X S \dot{X}^T - \frac{1}{\gamma} \dot{\hat{P}}^T (P - \hat{P}) \tag{۳-۱۶}$$

با جای‌گذاری معادله فضای حالت (۱۰-۳) در معادله (۱۶-۳)، در نهایت مشتق تابع لیاپانوف به صورت (۱۷-۳) به دست می‌آید.

$$\dot{V} = \frac{1}{2}(AX + Bw)^T SX + \frac{1}{2}XS(AX + Bw) - \frac{1}{\gamma}\dot{\hat{P}}^T(P - \hat{P}) \tag{۳-۱۷}$$

و سپس

$$\dot{V} = \frac{1}{2}(X^T A^T SX + wB^T SX + X^T SAX + X^T SBw) - \frac{1}{\gamma}\dot{\hat{P}}^T(P - \hat{P}) \tag{۳-۱۸}$$

پس از ساده‌سازی معادله فوق، معادله زیر حاصل می‌شود.

$$\dot{V} = \frac{1}{2}X^T(A^T S + SA)X + X^T SBw - (\frac{1}{\gamma}\dot{\hat{P}}^T)(P - \hat{P}) \tag{۳-۱۹}$$

و بعد از جای‌گذاری مقدار w، معادله زیر به دست می‌آید.

$$\dot{V} = \frac{1}{2}X^T(A^T S + SA)X + (X^T SB\widehat{M}^{-1}Y - \frac{1}{\gamma}\dot{\hat{P}}^T)(P - \hat{P}) \tag{۳-۲۰}$$

با انتخاب عبارت $A^T S + SA$ به صورت $-Q$ که یک ماتریس منفی معین است و باتوجه به اینکه برای اثبات پایداری سیستم، باید مشتق تابع لیاپانوف، منفی باشد، و اینکه جمله اول به دلیل مثبت معین بودن ماتریس Q، عبارتی منفی می‌باشد، لذا عبارت دوم باید برابر با صفر انتخاب شود:

$$\left(X^T SB\widehat{M}^{-1}Y - \frac{1}{\gamma}\dot{\hat{P}}^T\right)\left(P - \hat{P}\right) = 0 \tag{۳-۲۱}$$

که در نتیجه:

$$\left(X^T SB\widehat{M}^{-1}Y - \frac{1}{\gamma}\dot{\hat{P}}^T\right) = 0 \tag{۳-۲۲}$$

از صفر بودن این عبارت، معادله (۳-۲۳) حاصل می شود که برای تخمین پارامترهای نامعلوم سیستم، از این معادله استفاده می‌شود.

$$\dot{\hat{P}}^T = 2\Upsilon X^T SB\widehat{M}^{-1}Y \qquad (۳\text{-}۲۳)$$

در این قسمت به بررسی ماتریس رگرسیون و تبدیل فرم متعارف مدل دینامیکی به فرم رگرسیون پرداخته شده است. همان‌طور که قبلا گفته شد، مدل دینامیکی متعارف به فرم زیر است:

$$\tau = M(\theta)\ddot{\theta} + C\left(\theta,\dot{\theta}\right) \qquad (۳\text{-}۲۴)$$

و فرم رگرسیون که جهت هدف کنترل تطبیقی مورد استفاده قرار گرفته، به شکل زیر است.

$$\tau = YP \qquad (۳\text{-}۲۵)$$

برای تبدیل معادله (۳-۲۴) به (۳-۲۵)، ابتدا باید معادله (۳-۲۴) به صورت زیر باز شود:

$$\tau = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \qquad (۳\text{-}۲۶)$$

پس از تفکیک معادله فوق، سه معادله مجزای زیر حاصل می‌شود.

$$\tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + M_{13}\ddot{\theta}_3 + C_1 \qquad (۳\text{-}۲۷)$$

$$\tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + M_{23}\ddot{\theta}_3 + C_2 \qquad (۳\text{-}۲۸)$$

$$\tau_3 = M_{31}\ddot{\theta}_1 + M_{32}\ddot{\theta}_2 + M_{33}\ddot{\theta}_3 + C_3 \quad (۳-۲۹)$$

ماتریس P شامل سه مولفه زیر می‌باشد،

$$P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (۳-۳۰)$$

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} N \\ M_{mp} \\ M_m \end{bmatrix}$$

که این مولفه‌ها شامل مقادیر مرتبط با جرم‌ها و اینرسی ربات هستند که مقادیر ثابت ولی نامعلومی دارند و به صورت زیر تعریف شده اند.

$$N = {I_1}^2 m_2 + I_m \gamma^2 + I_1 \quad (۳-۳۱)$$

$$M_{mp} = m_p + 3m_2$$

$$M_m = I_1 m_2 + I_{1c} m_1$$

که با توجه هندسه ربات، m_1 جرم میله بالایی هر ساق، m_2 جرم میله پایینی، m_p جرم صفحه متحرک و I_1 اینرسی میله بالایی می‌باشد.

با باز کردن معادله (۲۵-۳)، فرم زیر حاصل می‌شود.

$$\tau = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (۳-۳۲)$$

و سپس سه معادله مجزای زیر به دست می‌آید.

$$\tau_1 = Y_{11}P_1 + Y_{12}P_2 + Y_{13}P_3 \qquad (۳-۳۳)$$

$$\tau_2 = Y_{21}P_1 + Y_{22}P_2 + Y_{23}P_3 \qquad (۳-۳۴)$$

$$\tau_3 = Y_{31}P_1 + Y_{32}P_2 + Y_{33}P_3 \qquad (۳-۳۵)$$

ماتریس M شامل عباراتی است که مولفه‌های اول و دوم ماتریس P را دربردارد و همچنین ماتریس C شامل عباراتی است که مولفه دوم و سوم ماتریس P را دربردارد. پس می‌توان سه معادله (۲۷-۳) و (۲۸-۳) و (۲۹-۳) را به صورت زیر نوشت.

$$\tau_1 = (A_{11}P_1 + A_{12}P_2)\ddot{\theta}_1 + (B_{11}P_1 + B_{12}P_2)\ddot{\theta}_2 + (C_{11}P_1 + C_{12}P_2)\ddot{\theta}_3 + (D_{12}P_2 + D_{13}P_3) \qquad (۳-۳۶)$$

$$\tau_2 = (A_{21}P_1 + A_{22}P_2)\ddot{\theta}_1 + (B_{21}P_1 + B_{22}P_2)\ddot{\theta}_2 + (C_{21}P_1 + C_{22}P_2)\ddot{\theta}_3 + (D_{22}P_2 + D_{23}P_3) \qquad (۳-۳۷)$$

$$\tau_3 = (A_{31}P_1 + A_{32}P_2)\ddot{\theta}_1 + (B_{31}P_1 + B_{32}P_2)\ddot{\theta}_2 + (C_{31}P_1 + C_{32}P_2)\ddot{\theta}_3 + (D_{32}P_2 + D_{33}P_3) \qquad (۳-۳۸)$$

که A_{ij} و B_{ij} و C_{ij} و D_{ij} شامل عبارات معلوم در معادله دینامیکی ربات هستند. پس از ساده‌سازی، معادلات فوق به شکل زیر درمی‌آیند.

$$\tau_1 = A_{11}P_1\ddot{\theta}_1 + A_{12}P_2\ddot{\theta}_1 + B_{11}P_1\ddot{\theta}_2 + B_{12}P_2\ddot{\theta}_2 + C_{11}P_1\ddot{\theta}_3 + C_{12}P_2\ddot{\theta}_3 + D_{12}P_2 + D_{13}P_3 \qquad (۳-۳۹)$$

$$\tau_2 = A_{21}P_1\ddot{\theta}_1 + A_{22}P_2\ddot{\theta}_1 + B_{21}P_1\ddot{\theta}_2 + B_{22}P_2\ddot{\theta}_2 + C_{21}P_1\ddot{\theta}_3 + C_{22}P_2\ddot{\theta}_3 + D_{22}P_2 + D_{23}P_3 \qquad (۳-۴۰)$$

$$\tau_3 = A_{31}P_1\ddot{\theta}_1 + A_{32}P_2\ddot{\theta}_1 + B_{31}P_1\ddot{\theta}_2 + B_{32}P_2\ddot{\theta}_2 + C_{31}P_1\ddot{\theta}_3 + C_{32}P_2\ddot{\theta}_3 + D_{32}P_2 + D_{33}P_3 \qquad (۳-۴۱)$$

برای به دست آوردن ماتریس رگرسیون، اگر در معادلات فوق، از مولفه‌های ماتریس P فاکتور بگیریم معادلات به شکل زیر حاصل می‌شوند.

$$\tau_1 = (A_{11}\ddot{\theta}_1 + B_{11}\ddot{\theta}_2 + C_{11}\ddot{\theta}_3)P_1 + (A_{12}\ddot{\theta}_1 + B_{12}\ddot{\theta}_2 + C_{12}\ddot{\theta}_3 + D_{12})P_2 + D_{13}P_3 \qquad (۳\text{-}۴۲)$$

$$\tau_2 = (A_{21}\ddot{\theta}_1 + B_{21}\ddot{\theta}_2 + C_{21}\ddot{\theta}_3)P_1 + (A_{22}\ddot{\theta}_1 + B_{22}\ddot{\theta}_2 + C_{22}\ddot{\theta}_3 + D_{22})P_2 + D_{23}P_3 \qquad (۳\text{-}۴۳)$$

$$\tau_3 = (A_{31}\ddot{\theta}_1 + B_{31}\ddot{\theta}_2 + C_{31}\ddot{\theta}_3)P_1 + (A_{32}\ddot{\theta}_1 + B_{32}\ddot{\theta}_2 + C_{32}\ddot{\theta}_3 + D_{32})P_2 + D_{33}P_3 \qquad (۳\text{-}۴۴)$$

حال، با مقایسه معادلات فوق با معادلات (۳-۳۳) و (۳-۳۴) و (۳-۳۵)، ضرایب مولفه‌های ماتریس P مشخص شده و در نتیجه مولفه‌های ماتریس رگرسیون به شکل زیر حاصل شده است.

$$Y_{11} = A_{11}\ddot{\theta}_1 + B_{11}\ddot{\theta}_2 + C_{11}\ddot{\theta}_3 \qquad (۳\text{-}۳۶)$$
$$Y_{12} = A_{12}\ddot{\theta}_1 + B_{12}\ddot{\theta}_2 + C_{12}\ddot{\theta}_3 + D_{12}$$
$$Y_{13} = D_{13}$$

$$Y_{21} = A_{21}\ddot{\theta}_1 + B_{21}\ddot{\theta}_2 + C_{21}\ddot{\theta}_3$$
$$Y_{22} = A_{22}\ddot{\theta}_1 + B_{22}\ddot{\theta}_2 + C_{22}\ddot{\theta}_3 + D_{22}$$
$$Y_{23} = D_{23}$$

$$Y_{31} = A_{31}\ddot{\theta}_1 + B_{31}\ddot{\theta}_2 + C_{31}\ddot{\theta}_3$$
$$Y_{32} = A_{32}\ddot{\theta}_1 + B_{32}\ddot{\theta}_2 + C_{32}\ddot{\theta}_3 + D_{32}$$
$$Y_{33} = D_{33}$$

پس ماتریس رگرسیون به طور کامل به شکل زیر می‌باشد.

$$Y = \begin{bmatrix} A_{11}\ddot{\theta}_1 + B_{11}\ddot{\theta}_2 + C_{11}\ddot{\theta}_3 & A_{12}\ddot{\theta}_1 + B_{12}\ddot{\theta}_2 + C_{12}\ddot{\theta}_3 + D_{12} & D_{13} \\ A_{21}\ddot{\theta}_1 + B_{21}\ddot{\theta}_2 + C_{21}\ddot{\theta}_3 & A_{22}\ddot{\theta}_1 + B_{22}\ddot{\theta}_2 + C_{22}\ddot{\theta}_3 + D_{22} & D_{23} \\ A_{31}\ddot{\theta}_1 + B_{31}\ddot{\theta}_2 + C_{31}\ddot{\theta}_3 & A_{32}\ddot{\theta}_1 + B_{32}\ddot{\theta}_2 + C_{32}\ddot{\theta}_3 + D_{32} & D_{33} \end{bmatrix} \qquad (۳-۷)$$

۲-۳ شبیه‌سازی و نتایج

در این بخش به شبیه‌سازی ربات و اعمال کنترل‌کننده طراحی‌شده بر روی آن پرداخته می‌شود و نتایج شبیه‌سازی سیستم حلقه بسته برای کنترل ربات دلتا ارائه میگردد. پارامترهای هندسی ربات دلتای مورد نظر که جهت شبیه‌سازی سیستم، مورد نیاز است، مطابق با جدول (۱-۳) در نظر گرفته می‌شوند.

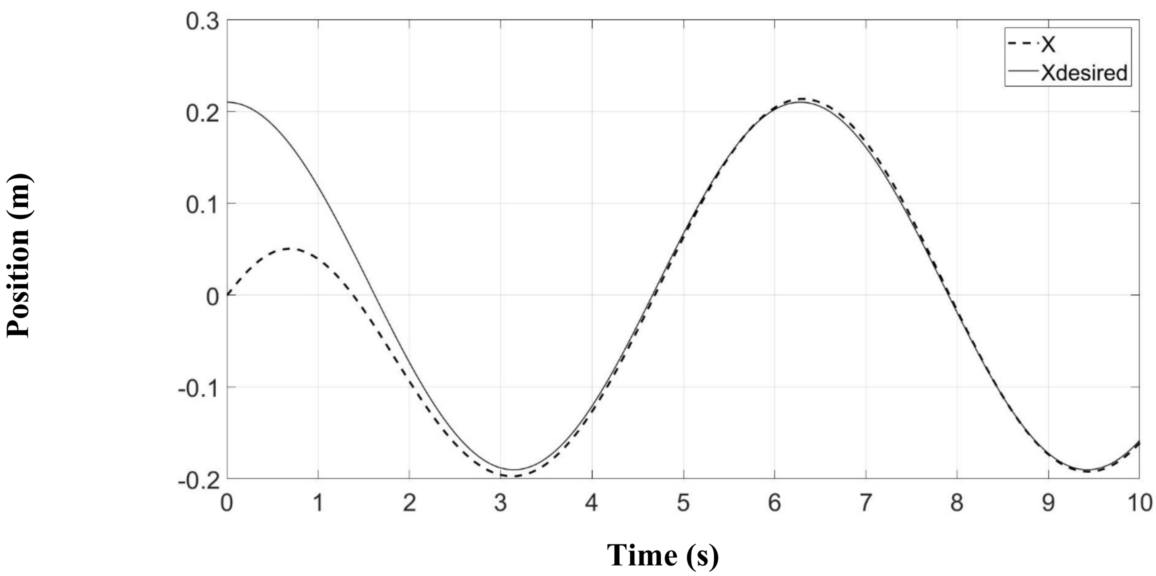
همچنین، برای شبیه‌سازی سیستم، یک مسیر دلخواه (۳۸-۳) به ربات داده می‌شود که با اعمال کنترل‌کننده، این مسیر را دنبال کند.

$$x_d = 0.01 + 0.2\cos t \qquad (۳-۳۸)$$
$$y_d = 0.01 + 0.2\sin t$$
$$z_d = 0.4$$

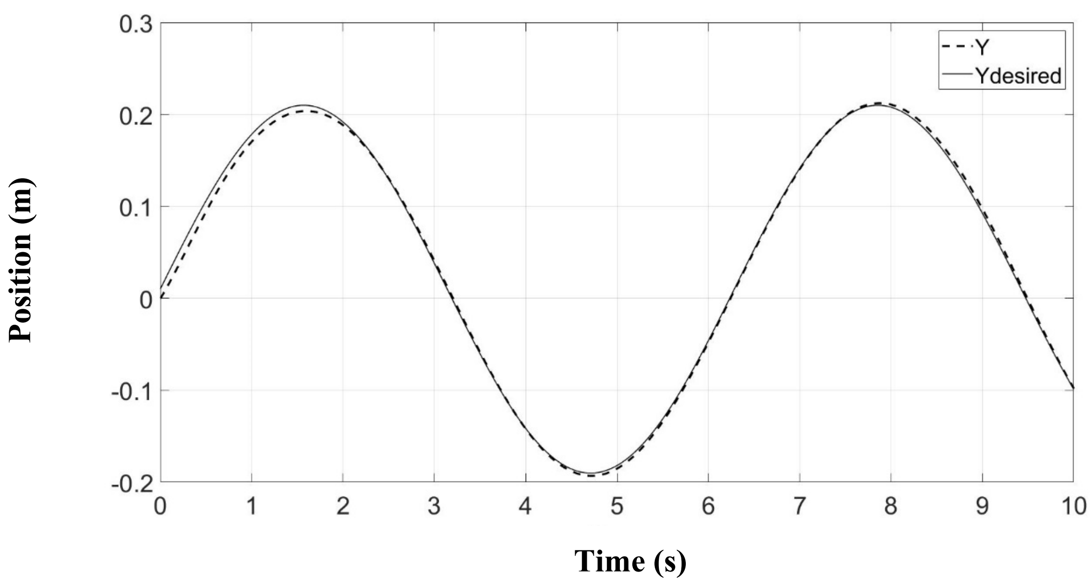
جدول ۱-۳. جدول پارامترهای هندسی ربات دلتا [۱۲]

ردیف	پارامتر	تعریف	مقدار
۱	l_1	طول میله اول در هر ساق	*0.171 m*
۲	l_2	طول میله دوم در هر ساق	*0.396 m*
۳	a	فاصله OA_i در صفحه ثابت	*0.1 mm*
۴	b	فاصله $\mathrm{PB_i}$ در صفحه متحرک	*0.0225 m*
۵	φ_1	زاویه $\mathrm{OA_1}$ با محور x	*π /3 rad*
۶	φ_2	زاویه $\mathrm{OA_2}$با محور x	*π rad*
۷	φ_3	زاویه $\mathrm{OA_3}$ با محور x	*- π /3 rad*
۸	m_1	جرم میله اول در هر ساق	*0.426 g*
۹	m_2	جرم میله دوم در هر ساق	*0.069 g*
۱۰	m_p	جرم صفحه متحرک	*0.096 g*

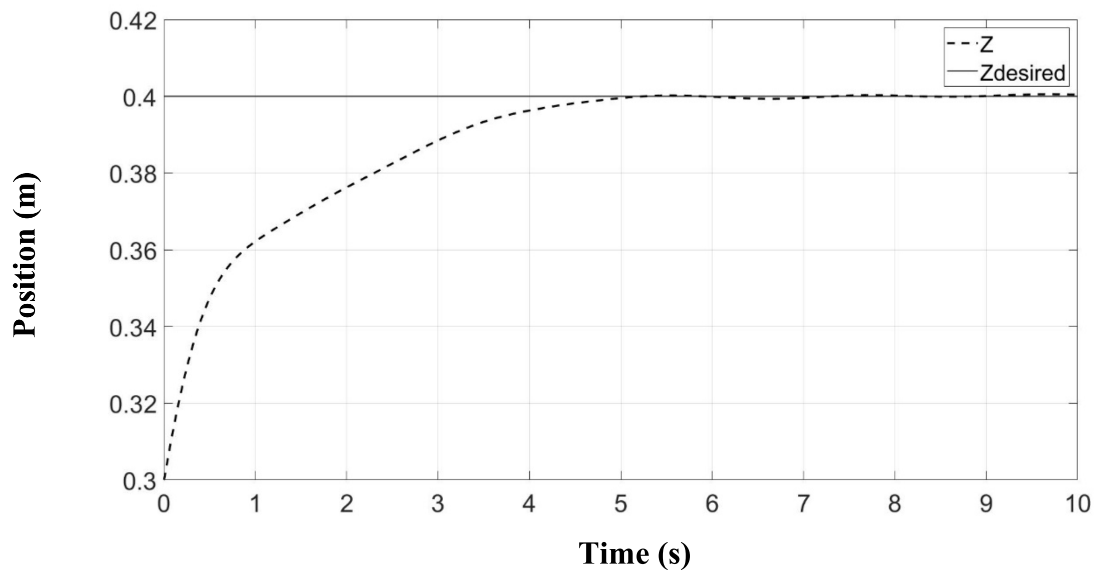
نتایج شبیه‌سازی برای بررسی عملکرد ردیابی، در شکل‌های زیر ارائه شده است. منحنی شکل (۱-۳) تعقیب مسیر مطلوب را برای مختصات x، منحنی (۲-۳) برای مختصات y ومنحنی (۳-۳) برای مختصات z نشان می‌دهد.



شکل ۱-۳. نمودار ردیابی مجری نهایی در مسیر x

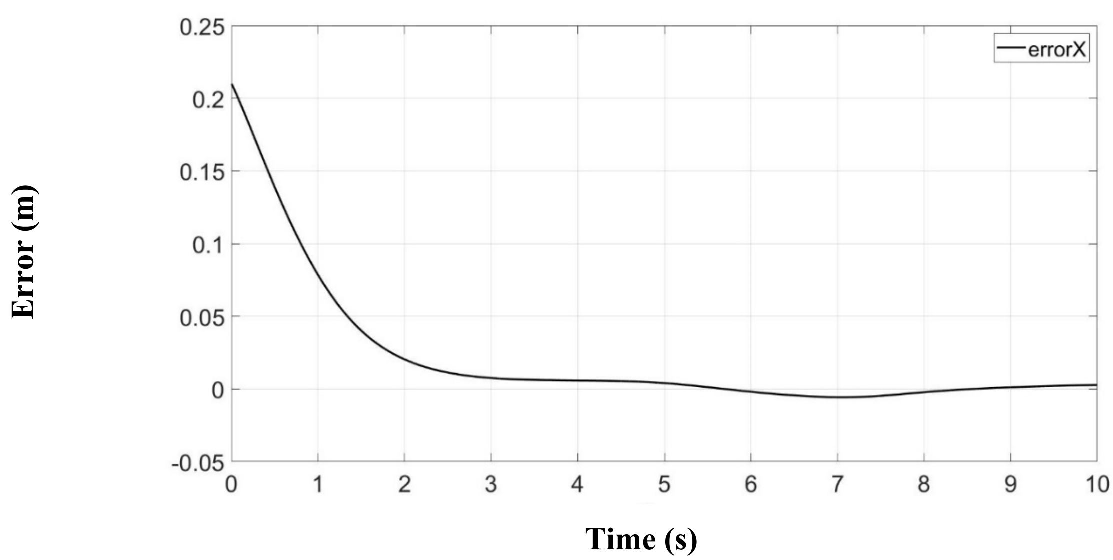


شکل ۲-۳. نمودار ردیابی مجری نهایی در مسیر y

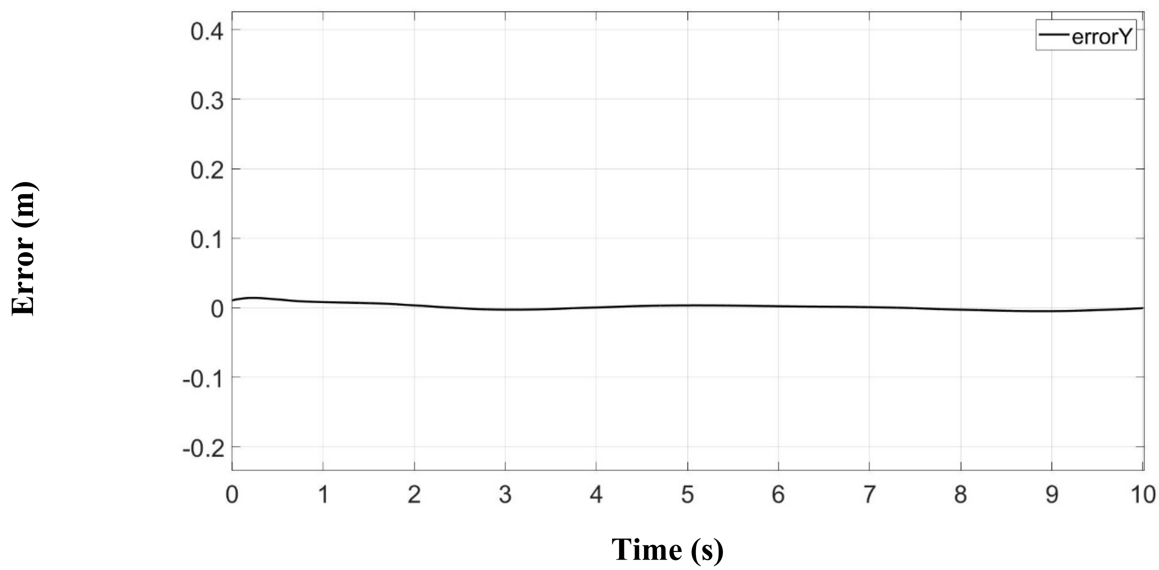


شکل ۳-۳. نمودار ردیابی مجری نهایی در مسیر z

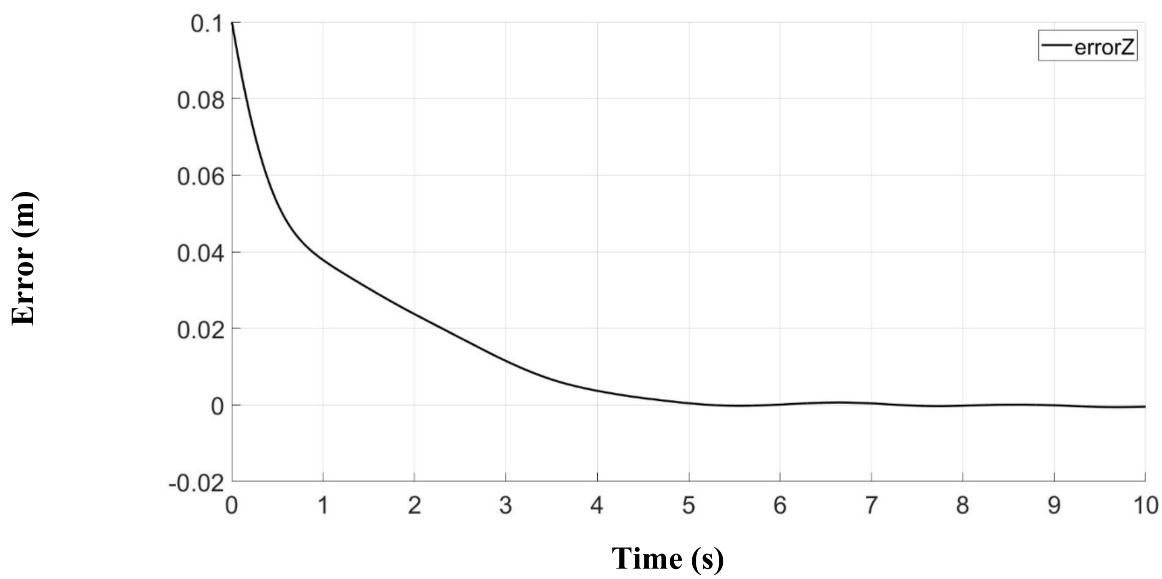
منحنی‌های شکل (۴-۳)، (۵-۳) و (۶-۳) اختلاف بین مسیرهای مطلوب و تعقیب آن مسیرها توسط ربات را نشان می‌دهد.



شکل ۴-۳. نمودار خطای ردیابی مسیر مجری نهایی در مختصات x

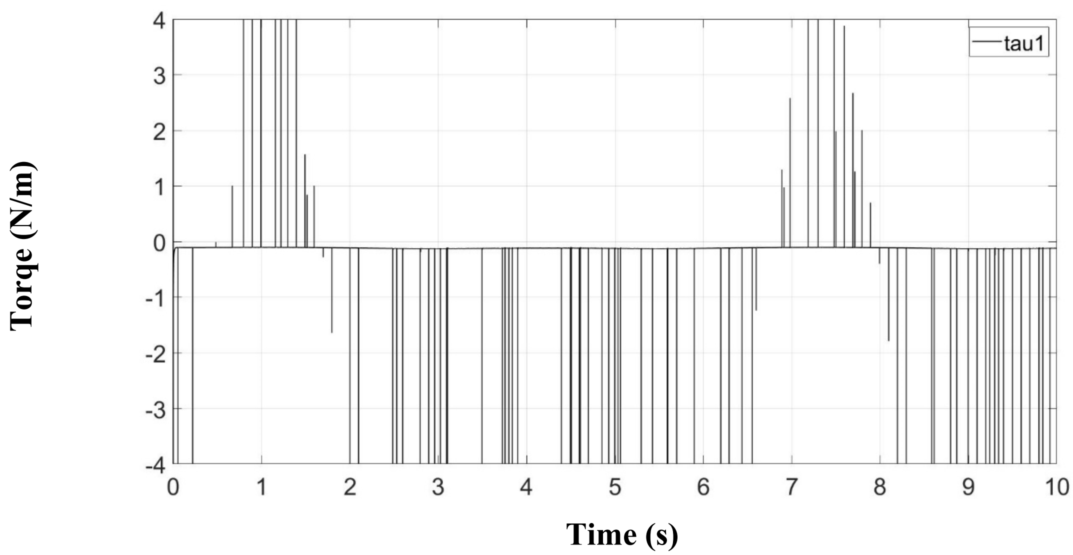


شکل ۵-۳. نمودار خطای ردیابی مسیر مجری نهایی در مختصات y

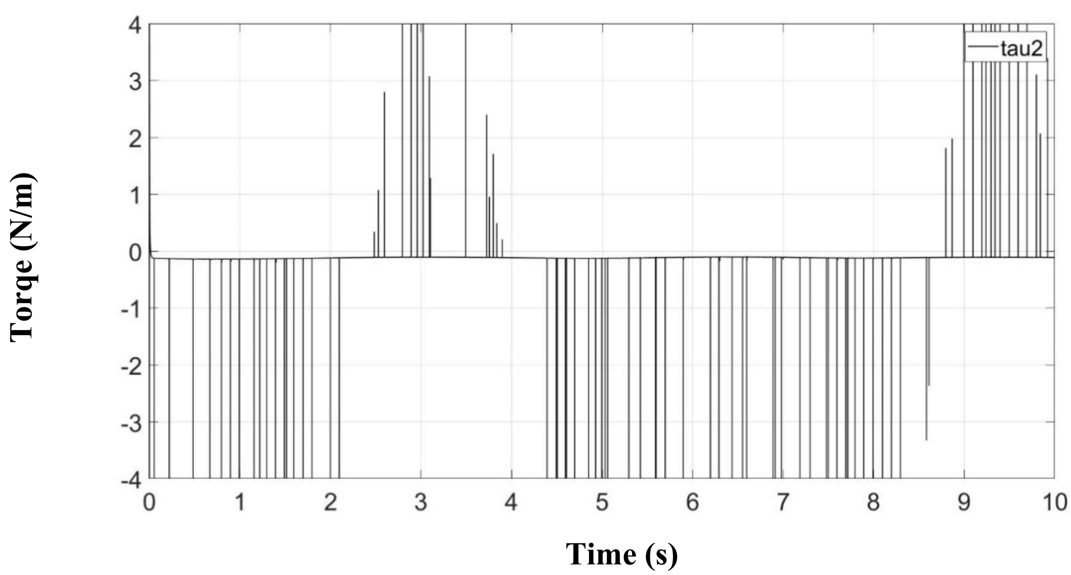


شکل ۶-۳. نمودار خطای ردیابی مسیر مجری نهایی در مختصات z

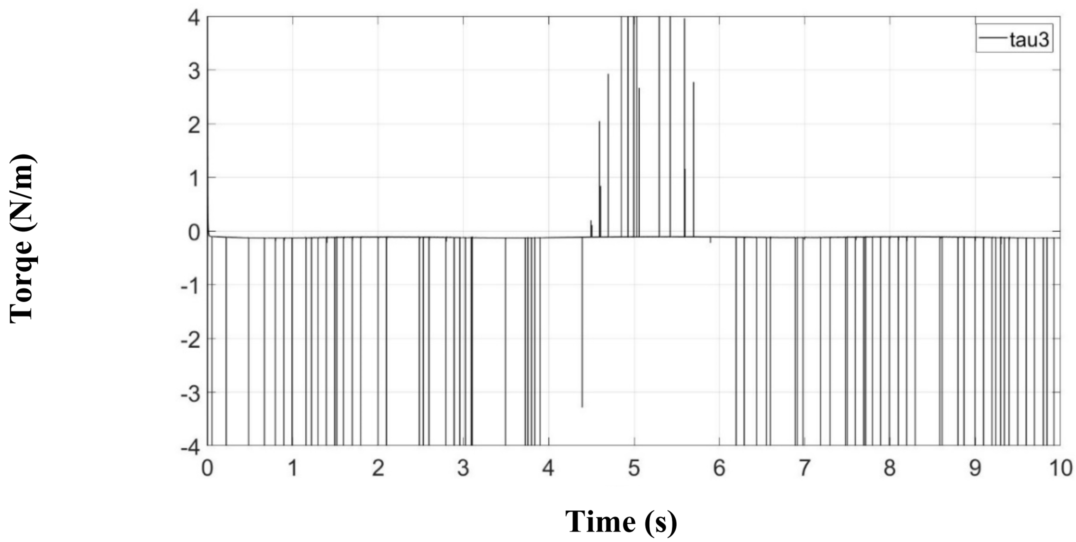
گشتاور کنترلی به دست آمده، ماتریسی شامل سه گشتاور بوده که منحنی‌های شکل (۳-۷)، (۳-۸) و (۳-۹) نشان‌دهنده این گشتاورهای کنترلی می‌باشد.



شکل ۳-۷. نمودار گشتاور کنترلی اول

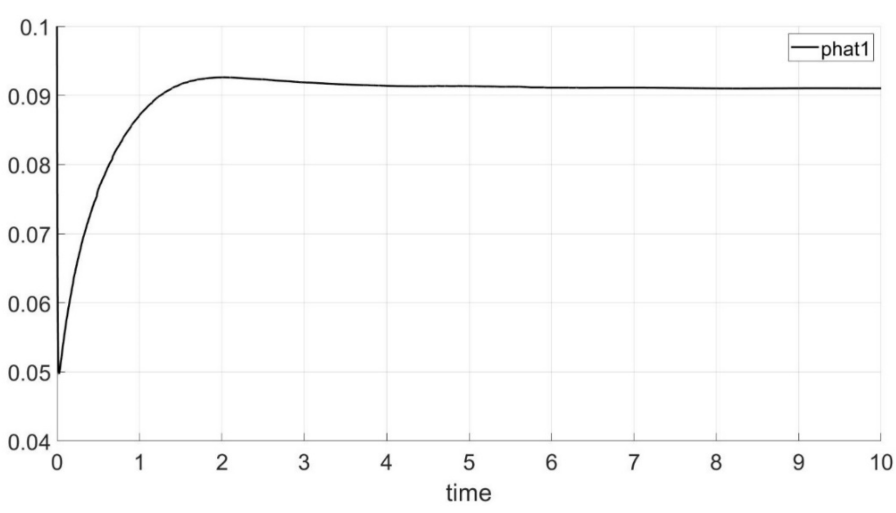


شکل ۳-۷. نمودار گشتاور کنترلی دوم

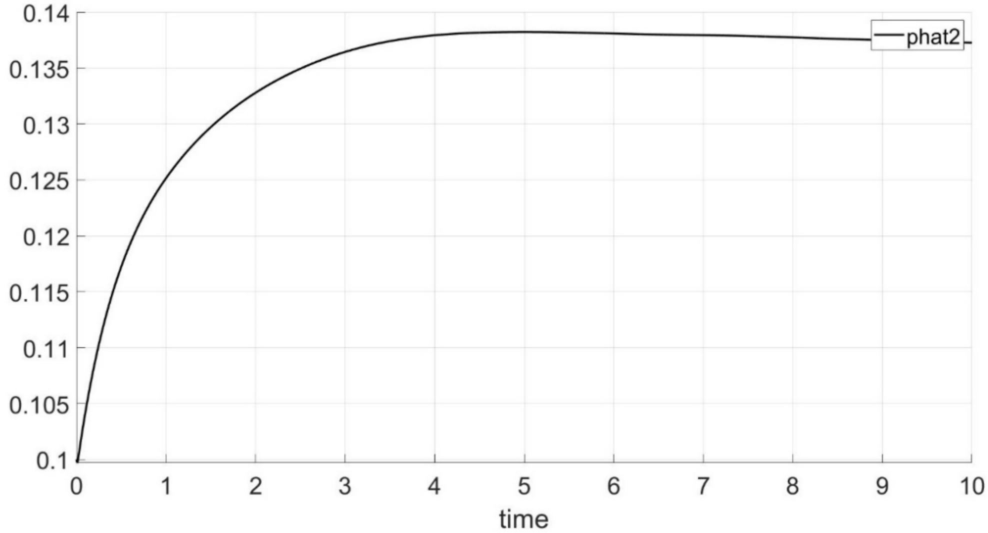


شکل ۷-۳. نمودار گشتاور کنترلی سوم

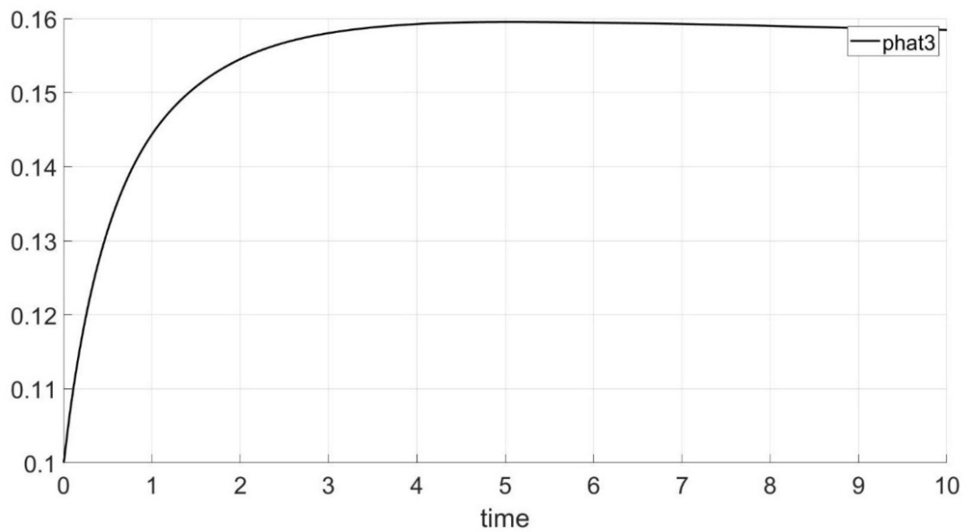
سه پارامتر موردنظر که در این تحقیق با استفاده از کنترل‌کننده تطبیقی تخمین زده شدند، در شکل (۱۰-۳)، (۱۱-۳) و (۱۲-۳) مشاهده می‌شود.



شکل ۱۰-۳. نمودار تخمین زده شده اول



شکل ۱۱-۳. نمودار پارامتر تخمین زده شده دوم



شکل ۱۲-۳. نمودار پارامتر تخمین شده سوم

فصل ۴: بهینه‌سازی کنترل مود لغزشی

۴-۱ کنترل مود لغزشی

از میان روش‌های کنترلی و همچنین کنترل مقاوم، کنترل مود لغزشی به دلیل مقاوم بودن در مقابل عدم قطعیت‌های پارامتری و اغتشاش بار، سادگی محاسبات و پاسخ دینامیکی سریع قابل توجه است. در این بخش با توجه به عدم قطعیت‌ها جهت ردیابی مسیر دلخواه طراحی کنترل‌کننده مود لغزشی برای ربات دلتا صورت می‌پذیرد. نقطعه ضعف این روش، لرزشی است که ایجاد می‌کند و در فرکانس‌های بالا باعث عملکرد نامطلوب سیستم می‌شود. در حین طراحی کنترل‌کننده، نیاز به ضرایب کنترل‌کننده می‌باشد که تاثیر مستقیمی بر نتیجه ردیابی و همچنین انرژی سیستم دارد که بایستی درست انتخاب شوند. در این بخش، ضرایب به صورت دلخواه و به روش صحیح و خطا انتخاب شده‌اند اما روش صحیح و خطا روشی مناسبی جهت انتخاب ضرایب نمی‌باشد. برای بهبود عملکرد سیستم می‌توان از روش‌های بهینه‌سازی برای انتخاب ضرایب کنترل‌کننده مطلوب استفاده کرد که در بخش بعد انجام شده است و بیشتر توضیح داده خواهد شد.

ایده اصلی در طراحی کنترل‌کننده مود لغزشی، تعریف یک سطح خطی یا غیرخطی به نام s است که در حالت کلی تابعی از متغیرهای حالت سیستم و زمان در صفحه فاز سیستم می‌باشد که به سطح لغزش معروف است. این سطح باید دارای دینامیک پایدار باشد یعنی پاسخ معادله $s = 0$ محدود باشد زیرا ورودی کنترلی باید به صورتی حاصل شود که مسیرهای حالت را به سمت سطح برده و بدون تغییر بر روی آن نگه دارد. در این حالت به دلیل پایدار بودن دینامیک سطح لغزش، حالت‌های سیستم به سمت نقطه تعادل سیستم حرکت خواهد کرد و سیستم پایدار خواهد شد. این موضوع در مورد مساله ردیابی نیز صدق می‌کند که در آن s تابعی از متغیرهای خطای سیستم بوده و مسیرهای حالت با ردیابی متغیرهای مطلوب یه سمت خطای صفر حرکت خواهند کرد.

هدف یک کنترل‌کننده مود لغزشی علاوه بر ردیابی مسیر مطلوب، این است که عدم قطعیت پارامتری، مثلا عدم دقت در خواص جرم واینرسی یا بارها، عدم دقت‌ها در ثابت‌های گشتاور، اصطکاک و غیره را جبران کند. همچنین حضور دینامیک‌های مدل‌نشده را نیز به حساب آورد.

ضمن یادآوری معادله دینامیکی ربات دلتا که به شکل زیر است.

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) \tag{۴-۱}$$

لازم به ذکر است که به دلیل بزرگ بودن درایه‌های ماتریس‌های سازنده معادله دینامیکی ربات مورد بررسی فقط فرم کلی آن‌ها بیان شده است. در این تحقیق تمامی پارامترهای سیستم، دارای نامعینی با کران مشخص می-باشند که به صورت کلی ماتریس‌های دینامیک سیستم به شکل زیر خواهند بود.

$$M = \widehat{M} + \Delta M \tag{۴-۲}$$
$$C = \hat{C} + \Delta C$$

که ماتریس های $\widehat{M}$ و $\hat{C}$ به ترتیب قسمت معلوم ماتریس‌های M و C و همچنین ماتریس‌های ΔM و ΔC شامل عدم قطعیت‌های سیستم می‌باشند. جهت طراحی کنترل‌کننده مود لغزشی به یک سطح لغزش نیاز است که برای سیستم های مرتبه دو به صورت زیر تعریف می‌گردد.

$$s = \dot{e} + \lambda e \tag{۴-۳}$$

که در آن s یک جمع وزن‌دار از خطاهای سرعت و خطاهای موقعیت است و λ یک عدد ثابت مثبت است. به دلیل ماتریسی بودن مقدار خطا و در نتیجه سطح لغزش، λ نیز یک ماتریس قطری بدین شکل خواهد بود.

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{۴-۴}$$

خطای سیستم شامل خطای موقعیت، سرعت و شتاب نیز به شکل زیر تعریف می‌شود.

$$\begin{aligned} e &= \theta_d - \theta \\ \dot{e} &= \dot{\theta}_d - \dot{\theta} \\ \ddot{e} &= \ddot{\theta}_d - \ddot{\theta} \end{aligned} \tag{۴-۵}$$

به صورت کلی این سطح لغزش به شکل زیر تعریف می‌گردد

$$s = (\frac{d}{dt} + \lambda)^{n-1} e \tag{۴-۶}$$

که در آن n مرتبه سیستم را نشان می‌دهد. همان‌طور که مشاهده می‌شود برای سیستم مرتبه دوم معادله فوق به فرم (۳-۴) درمی‌آید. همچنین برای هدف کنترلی، یک شرط لغزش به صورت زیر تعریف می‌گردد

$$\frac{d}{dt}\left(\frac{1}{2} s^T s\right) \leq -\eta |s| \tag{۴-۷}$$

که η یک بردار ستونی شامل مولفه های مثبت می‌باشد

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \tag{۴-۸}$$

اگر مشتق مربوط به عبارت فوق گرفته شود، رابطه زیر برقرار خواهد بود.

$$s\dot{s} \leq -\eta|s| \tag{۴-۹}$$

با توجه به تعریف سطح لغزش برای سیستم مرتبه دو، پس مشتق آن به شکل زیر خواهد بود.

$$\dot{s} = \ddot{e} + \lambda\dot{e} \tag{۴-۱۰}$$

و از آن‌جایی که

$$\mathrm{sgn}(s) = \frac{s}{|s|} \tag{۴-۱۱}$$

پس عبارت (۱۱-۴) به شکل زیر درمی‌آید

$$\dot{s}\,\mathrm{sgn}(s) \leq -\eta \tag{۴-۱۲}$$

در حالت کلی اگر سیستم به فرم زیر باشد

$$\ddot{\theta} + f = u \tag{۴-۱۳}$$

و تمام عدم قطعیت‌ها به شکل زیر در ترم Δf باشد

$$f=\hat{f}+\Delta f \tag{۴-۱۴}$$

با توجه به معادله مشتق سطح لغزش (۱۰-۴) و همچنین تعریف خطای شتاب، می‌توان شرط سطح لغزش (۱۲-۴) را به شکل زیر نوشت

$$(\ddot{\theta}_d - \ddot{\theta} + \lambda e)\,\mathrm{sgn}(s) \le -\eta \tag{۴-۱۵}$$

حال با جای‌گذاری معادله سیستم در معادله فوق و همچنین با توجه به عبارت (۱۳-۴)، معادله زیر حاصل خواهد شد.

$$(\ddot{\theta}_d - u + f + \lambda \dot{e})\,\mathrm{sgn}(s) \le -\eta \tag{۴-۱۶}$$

قانون کنترلی زیر پیشنهاد می‌شود

$$u = \ddot{\theta}_d - \hat{f} + \lambda \dot{e} + k\,\mathrm{sgn}(s) \tag{۴-۱۷}$$

که ترم $k\,\mathrm{sgn}(s)$ برای حذف عدم قطعیت‌های سیستم لحاظ شده است و باقی ترم‌ها جهت کنترل قسمت‌های معلوم سیستم می‌باشد.

با جای‌گذاری قانون کنترلی پیشنهادی و ساده‌سازی، معادله (۱۶-۴) به فرم زیر حاصل می‌شود

$$(\Delta f - k\,\mathrm{sgn}(s))\,\mathrm{sgn}(s) \le -\eta \tag{۴-۱۸}$$

برای به دست آوردن مقدار k، معادله فوق به شکل زیر نوشته می‌شود

$$\Delta f\,\mathrm{sgn}(s) - \eta \le k \tag{۴-۱۹}$$

اگر $|\Delta f| \le F$ ، پس می‌توان مقدار k را به صورت زیر نوشت.

$$k = F + \eta \tag{۴-۲۰}$$

مطابق با قانون کنترلی پیشنهادی برای سیستم کلی، قانون کنترلی برای ربات مورد نظر به صورت زیر پیشنهاد می‌شود.

$$\tau = u_{eq} + k\,\mathrm{sgn}(s) \tag{۴-۲۱}$$

که u_{eq} برای قسمت معلوم دینامیک می‌باشد و ترم دوم آن برای حذف عدم قطعیت‌های سیستم است.

با توجه به مقدار k که برای سیستم کلی در معادله (۴-۲۰) حاصل شد و با توجه به فرم دینامیک این ربات، می‌توان مقدار k را برای ربات مورد نظر به فرم زیر نوشت.

$$k = \Delta M\ddot{\theta}_d + \Delta C + \eta \tag{۴-۲۲}$$

هدف به صفر رساندن سطح لغزش و مشتق آن است که برای به دست آوردن u_{eq}، از همین ویژگی سطح لغزش استفاده می‌شود.

$$\dot{s} = 0 \tag{۴-۲۳}$$

با جای‌گذاری خطای شتاب در مشتق سطح لغزش، معادله زیر حاصل می‌شود.

$$\dot{s} = \left(\ddot{\theta}_d - \ddot{\theta}\right) + \lambda\dot{e} \tag{۴-۲۴}$$

از معادله دینامیکی ربات، معادله زیر حاصل می‌شود.

$$\ddot{\theta} = M^{-1}(\tau - C) \tag{۴-۲۵}$$

از آن‌جایی که هدف به دست آوردن ورودی مربوط به قسمت معلوم مساله می‌باشد پس از معادله دینامیکی به نحو زیر استفاده می‌شود.

$$\ddot{\theta} = \widehat{M}^{-1}\left(u_{eq} - \hat{C}\right) \tag{۴-۲۶}$$

و سپس با جای‌گذاری معادله دینامیکی سیستم در معادله(۴-۲۴)، عبارت زیر حاصل می‌شود.

$$\dot{s} = \ddot{\theta}_d - \widehat{M}^{-1}\left(u_{eq} - \hat{C}\right) + \lambda\dot{e} \tag{۴-۲۷}$$

پس از ساده‌سازی معادله فوق به شکل زیر نوشته می‌شود.

$$\dot{s} = \ddot{\theta}_d - \widehat{M}^{-1}u_{eq} + \widehat{M}^{-1}\hat{C} + \lambda\dot{e} \tag{۴-۲۸}$$

با جای‌گذاری مقدار معادله $\dot{s}$ طبق معادله فوق در شرط $\dot{s} = 0$ ، می‌توان معادله زیر را به دست آورد.

$$\ddot{\theta}_d - \widehat{M}^{-1}u_{eq} + \widehat{M}^{-1}\hat{C} + \lambda\dot{e} = 0 \tag{۴-۲۹}$$

و سپس مقدار u_{eq} به شکل حاصل می‌شود.

$$u_{eq} = \widehat{M}\left(\ddot{\theta}_d + \lambda\dot{e}\right) + \hat{C} \tag{۴-۳۰}$$

پس با توجه به قانون کنترلی پیشنهادی برای ربات دلتا مورد نظر و همچنین با توجه مقدار u_{eq} که در معادله فوق قابل مشاهده می‌باشد و نیز همچنین با توجه به مقدار k که در بالا حاصل گردید، معادله زیر برای کنترل مورد لغزشی ربات دلتا قابل استفاده می‌باشد.

$$\tau = \widehat{M}\left(\ddot{\theta}_d + \lambda\dot{e}\right) + \hat{C} + (\Delta M\ddot{\theta}_d + \Delta C + \eta)\,\mathrm{sgn}(s) \tag{۴-۳۱}$$

برای اثبات پایداری سیستم در استفاده از این قانون کنترلی، از تابع لیاپانوف استفاده می‌شود. تابع لیاپانوف به شکل زیر تعریف می‌گردد.

$$V = \frac{1}{2}s^T s \tag{۴-۳۲}$$

همانطور که مشاهده می‌شود، تابع لیاپانوف مثبت است. برای اثبات پایداری باید مشتق این تابع منفی معین باشد. بدین منظور از تابع لیاپانوف مشتق گرفته می‌شود که در زیر مشاهده است.

$$\dot{V} = s\dot{s} \tag{۴-۳۳}$$

با جای‌گذاری معادله (۴-۲۴) در معادله فوق، عبارت زیر حاصل می‌شود.

$$\dot{V} = s\left(\ddot{\theta}_d - \ddot{\theta} + \lambda\dot{e}\right) \tag{۴-۳۴}$$

و سپس با جای‌گذاری معادله (۴-۲۵) در معادله فوق، عبارت زیر حاصل می‌شود

$$\dot{V} = s\left(\ddot{\theta}_d - M^{-1}(\tau - C) + \lambda\dot{e}\right) \tag{۴-۳۵}$$

همچنین پس از جای‌گذاری قانون کنترلی پیشنهادی (۴-۲۱) در عبارت فوق و ساده‌سازی، معادله زیر حاصل می‌شود.

$$\dot{V} = s\big(\ddot{\theta}_d - M^{-1}\big(u_{eq} + k\,\mathrm{sgn}(s)\big) + M^{-1}C + \lambda\dot{e}\big) \tag{۴-۳۶}$$

و اگر به جای u_{eq} مقدار $M(\ddot{\theta}_d + \lambda\dot{e}) + C$ قرار داده شوند.

$$\dot{V} = s\big(\ddot{\theta}_d - M^{-1}\big(M\ddot{\theta}_d + M\lambda\dot{e} + C + k\,\mathrm{sgn}(s)\big) + M^{-1}C + \lambda\dot{e}\big) \tag{۴-۳۷}$$

و پس از ساده‌سازی نتیجه به شکل زیر حاصل می‌شود.

$$\dot{V} = sM^{-1}k\,\mathrm{sgn}(s) \tag{۴-۳۸}$$

حال می‌توان یک متغیر جدید تعریف کرد:

$$\acute{k} = M^{-1}k \tag{۴-۳۹}$$

از طرفی رابطه (۴-۱۱) زیر برقرار است، در نهایت مشتق تابع لیاپانوف به شکل زیر خواهد بود

$$\dot{V} = -\acute{k}s \tag{۴-۴۱}$$

که $\acute{k}$ مقدار مثبتی دارد، پس مشتق تابع لیاپانوف منفی معین خواهد بود.

$$\dot{V} \leq 0 \tag{۴-۴۲}$$

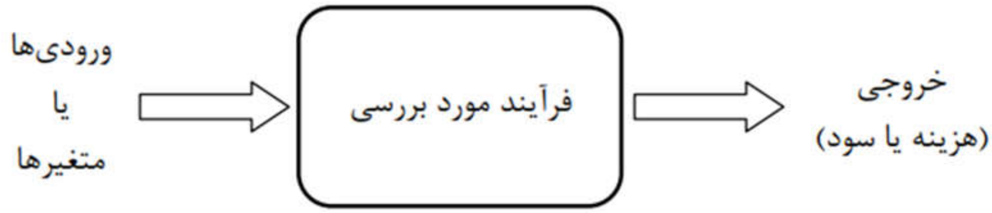
۴-۲ بهینه‌سازی کنترل‌کننده طراحی شده با الگوریتم pso

۴-۲-۱ مقدمه

بهینه‌سازی فرآیندی است که برای بهتر کردن چیزی دنبال می‌شود. فکر، ایده و یا طرحی که به وسیله یک دانشمند یا یک مهندس مطرح می‌شود، طی روال بهینه‌سازی بهتر می‌شود. در هنگام بهینه‌سازی، شرایط اولیه با روش‌های مختلف مورد بررسی قرار می‌گیرد و اطلاعات به دست آمده، برای بهبود بخشیدن به یک فکر یا روش مورد استفاده قرار می‌گیرند. بهینه‌سازی ابزاری ریاضی است که برای یافتن بسیاری از پرسش ها در خصوص چگونگی راه حل مسائل مختلف به کار می‌رود.

در بهینه‌سازی از یافتن بهترین جواب برای یک مساله صحبت به میان می‌آید. لفظ بهترین به طور ضمنی بیان می‌کند که بیش از یک جواب برای مساله مورد نظر وجود دارد که البته دارای ارزش یکسانی نیستند. تعریف بهترین جواب، به مساله مورد بررسی، روش حل و همچنین میزای خطای مجاز وابسته است. بنابراین نحوه فرمول بندی مساله نیز بر چگونگی تعریف بهترین جواب تاثیر مستقیم دارد.

بهینه‌سازی، تغییر دادن ورودی‌ها و خصوصیات یک دستگاه، فرآیند ریاضی و یا آزمایش تجربی است به نحوی که بهترین خروجی یا نتیجه به دست بیاید (شکل (۱-۴)). ورودی‌ها متغیرهای فرآیند یا تابع مورد بررسی هستند که به نام‌های تابع هدف، تابع هزینه یا تابع برازندگی نامیده می‌شود. خروجی نیز به صورت هزینه، سود یا برازندگی تعریف می‌شود. معمولا مسائل بهینه‌سازی به صورت کمینه‌سازی مقدار یک تابع هزینه در نظر گرفته می‌شود. به راحتی می‌توان نشان داد که هر نوع مساله بهینه‌سازی را می‌توان در قالب یک مساله کمینه‌سازی تعریف نمود.



شکل ۱-۴. فرآیند یا تابعی که بهینه‌سازی می‌شود.

جیمز کندی روان‌شناس اجتماعی و راسل سی‌ابرهارت مهندس برق، صاحبان اصلی ایده الگوریتم بهینه‌سازی انبوه ذرات (pso) می‌باشند. آن‌ها قصد داشتند که با بهره‌گیری از مدل‌های اجتماعی و روابط موجود اجتماعی، نوعی از هوش محاسباتی را به وجود بیاورند که به توانایی‌های فردی ویژه نیازی نداشته باشد. اولین شبیه‌سازی آن‌ها که در سال ۱۹۹۵ انجام گردید، آن‌ها را به سمت شبیه‌سازی رفتار پرندگان برای یافتن دانه رهنمون کرد. این کار تحت تاثیر کار هپنر و گرنادر بود که در سال ۱۹۹۰ برای شبیه‌سازی رفتار پرندگان به صورت یک سیستم غیرخطی انجام گرفته بود.

کار کندی و ابرهارت منجر به ایجاد الگوریتمی قوی برای بهینه‌سازی شد که با نام الگوریتم بهینه‌سازی ذرات یا pso معروف است. در الگوریتم pso تعدادی از موجودات وجود دارند که به آن‌ها ذره گفته می‌شود و در فضای جستجوی تابعی که قصد کمینه کردن (یا بهینه کردن) مقدار آن را داریم، پخش شده‌اند. هر ذره در موقعیتی از فضا که در آن قرار گرفته است، مقدار تابع هدف را محاسبه می‌کند. سپس با استفاده از ترکیب اطلاعات محل فعلی‌اش و بهترین محلی که در گذشته در آن بوده است و همچنین اطلاعات یک یا چند ذره از بهترین ذرات موجود در جمع، جهتی را برای حرکت انتخاب می‌کند. همه ذرات جهتی برای حرکت انتخاب می‌کنند و پس از انجام حرکت، یک مرحله از الگوریتم به پایان می‌رسد. این مراحل چندین بار تکرار می‌شوند تا آن که جواب مورد

نظر به دست بیاید. در واقع انبوه ذرات که مقدار کمینه یک تابع را جستجو می‌کنند، همانند دسته‌ای از پرندگان عمل می‌کنند که به دنبال غذا می‌گردند.

هر ذره در الگوریتم pso از سه بردار d بعدی تشکیلی شده است که d بعد فضای جستجو می‌باشد. برای ذره iام این سه بردار عبارتند از: x^{i} موقعیت فعلی ذره، v^{i} سرعت حرکت ذره و $x^{i,best}$ بهترین موقعیتی که ذره تا به حال تجربه کرده است. x^{i} مجموعه ای از مختصات است که موقعیت فعلی ذره را نمایش می‌دهد. در هر مرحله‌ای که الگوریتم تکرار می‌شود، x^{i} به عنوان یک جواب برای مساله محاسبه می‌شود. اگر این موقعیت بهتر از جواب‌های پیشین باشد در x^{i} ذخیره می‌شود. f^{i} مقدار تابع هدف در x^{i} و $f^{i,best}$ مقدار تابع هدف در $x^{i,best}$ است که هر دو از عناصر تشکیل دهنده هر ذره به حساب می‌آیند. ذخیره کردن مقدار $f^{i,best}$ برای انجام مقایسه‌های بعدی ضروری است اما ذخیره کردن مقدار f^{i} ضروری نمی‌باشد. در هر تکرار x^{i} و v^{i} جدیدی به دست می‌آیند و منظور از اجرای الگوریتم، بهتر کردن $x^{i,best}$ و به احتمال x^{i} است.

الگوریتم pso چیزی فراتر از یک مجموعه ذرات است. هیچ‌کدام از ذرات قدرت حل هیچ مساله‌ای را ندارند بلکه هنگامی می‌توان به حل مساله امیدوار شد که آن‌ها با همدیگر ارتباط و تعامل داشته باشند. در واقع برای انبوه ذرات، حل مساله، یک مفهوم اجتماعی است که از رفتار تک تک ذرات و تعامل میان آن‌ها به وجود می‌آید. بهترین موقعیتی که به وسیله همه ذرات پیدا شده است، به صورت x^{gbest} نشان داده می‌شود که با مقایسه مقادیر $f^{i,best}$ به ازای همه ذرات و از میان $x^{i,best}$ها انتخاب می‌شود. مقدار تابع هدف در x^{gbest} به صورت f^{gbest} نشان داده می‌شود. اگر تعداد ذرات موجود در جمعیت n باشد، آن‌گاه می‌توان روابط زیر را نوشت:

$$x^{i,best}[t] = \underset{\tau\le t}{\arg\min} f\left(x^{i}[\tau]\right) = \arg\min\left\{f\left(x^{i}[t]\right), f\left(x^{i,best}[t-1]\right)\right\} \quad (۴\text{-}۴۳)$$

$$f^{i,best}[t] = f\left(x^{i,best}[t]\right) = \min_{\tau\le t} f^{i}[\tau] = \min\left\{f^{i}[t], f^{i,best}[t-1]\right\} \quad (۴\text{-}۴۴)$$

$$x^{gbest}[t] = \underset{i=1,\dots,n}{\arg\min} f\left(x^{i,best}[t]\right) \tag{۴-۴۵}$$

$$f^{gbest}[t] = f(x^{gbest}[t]) = \min_{i=1,\dots,n} f^{i,best}[t] \tag{۴-۴۶}$$

در مرحله ابتدایی الگوریتم، ذرات با موقعیت‌ها و سرعت‌های تصادفی ایجاد می‌شوند. در طی اجرای الگوریتم، موقعیت و سرعت هر ذره در مرحله $t+1$ام از الگوریتم، از روی اطلاعات مرحله قبلی ساخته می‌شود. اگر z_j مولفه jام از بردار z باشد، آن‌گاه روابطی که سرعت و موقعیت ذرات را تغییر می‌دهند، عبارتند از:

$$v_j^i[t+1] = wv_j^i[t] + c_1r_1\left(x_j^{i,best}[t] - x_j^i[t]\right) + c_2r_2\left(x_j^{gbest}[t] - x_j^i[t]\right) \tag{۴-۴۷}$$

$$x_j^i[t+1] = x_j^i[t] + v_j^i[t+1] \tag{۴-۴۸}$$

در این روابط، w ضریب اینرسی، r_1 و r_2 اعدادی تصادفی در بازه $[0,1]$ با توزیع یکنواخت و همچنین c_1 و c_2 ضرایب یادگیری هستند. r_1 و r_2 باعث می‌شوند که نوعی گوناگونی در جواب‌ها به وجود بیاید و به این نحو جستجوی کاملی روی فضا انجام پذیرد. c_1 ضریب یادگیری مربوط به تجارب شخصی هر ذره است و در مقابل c_2 ضریب یادگیری مربوز به تجارب کل جمع می‌باشد. از معادله (۴-۴۷) می‌توان به این نتیجه رسید که هر ذره به هنگام حرکت، (الف) جهت حرکت قبلی خود، (ب) بهترین موقعیتی را که در آن قرار داشته است و (پ) بهترین موقعیتی را که به وسیله کل جمع تجربه شده است، در نظر می‌گیرد. در برخی موارد روابط (۴-۴۷) و (۴-۴۸) به صورت زیر برای همه ابعاد جمع‌بندی می‌شود:

$$v^i[t+1] = wv^i[t] + c_1R_1 \otimes \left(x^{i,best}[t] - x^i[t]\right) + c_2R_2 \otimes \left(x^{gbest}[t] - x^i[t]\right) \tag{۴۹-۴}$$

$$x^i[t+1] = x^i[t] + v^i[t+1] \tag{۴-۵۰}$$

که در آن، R_1 و R_2 دو بردار هم‌اندازه با بعد فضای جستجو هستند که مولفه‌هایشان اعداد تصادفی مستقل با توزیع یکنواخت و در بازه $[0,1]$ هستند. همچنین علامت $\otimes$ نشان دهنده عمل ضرب عضو به عضو برای ماتریس‌ها است. به منظور محدود کردن میزان حرکت هر ذره، مقدار مولفه‌های سرعت ذرات در بازه $[-v_{max}, v_{max}]$ درنظر گرفته می‌شود و مقادیر بزرگتر یا کوچکتر نیز به این بازه تصویر می‌شوند. البته فرض بر این است که عرض فضای جستجو در تمام ابعاد ثابت و برابر با s باشد. در این صورت به طور معمول، $v_{max} = ps$ در نظر گرفته می‌شود که $p \in [0.1,1]$ است. در شکل زیر، مراحل الگوریتم pso که در بالا توضیح داده شد، مشاهده می‌شود.

الگوریتم بهینه‌سازی انبوه ذرات

n ذره بساز.
برای تمام ذرات، سرعت و موقعیتی تصادفی ایجاد کن.
تا زمانی که شرایط خاتمه محقق نشده‌اند:
- یک واحد به t اضافه کن.
- مقدار تابع هدف را به ازای هر ذره محاسبه کن.
- **به ازای** i از یک تا n:
  - $x^{i,best}[t]$ را محاسبه کن.
- **مقدار بعدی** i.
- $x^{gbest}[t]$ را محاسبه کن.
- **به ازای** i از یک تا n:
  - **به ازای** j از یک تا d:

$$v_j^i[t+1] = wv_j^i[t] + c_1r_1\left(x_j^{i,best}[t] - x_j^i[t]\right) + c_2r_2\left(x_j^{gbest}[t] - x_j^i[t]\right)$$

$$x_j^i[t+1] = x_j^i[t] + v_j^i[t+1]$$

  - **مقدار بعدی** j.
- **مقدار بعدی** i.

شکل ۲-۴. مراحل الگوریتم pso [۴۶]

۴-۲-۲ بهینه‌سازی ضرایب کنترلی

همانطور که در بخش قبل گفته شد، انتخاب ضرایب کنترلی به روش صحیح و خطا مناسب نیست. لذا الگوریتم‌های بهینه‌سازی می‌توانند روش مناسبی برای این امر باشند. برای این امر باید بتوان یک تابع هدف مناسب یافت که با هدف کمینه کردن این تابع، ضرایب کنترلی مناسب انتخاب شوند. از آن‌جایی که خطای ردیابی باید به کمترین حد خود برسد و همچنین برای صرف انرژی کمتر، باید کمترین میزان گشتاور استفاده شود، تابع هزینه مورد نیاز در بهینه‌سازی به گونه‌ای انتخاب می‌شود که هر دو شرط حداقل‌سازی خطا و گشتاور را ارضا کند.

طبق توضیحاتی که در بالا داده شد، الگوریتم بهینه‌سازی به این صورت عمل می‌کند که در ابتدا مقادیر اولیه‌ای برای ضرایب کنترل‌کننده در نظر گرفته می‌شود و با این مقادیر، کنترل‌کننده طراحی شده، بر روی ربات اعمال می‌شود. در نتیجه خطای ردیابی مفاصل و همچنین گشتاور حاصل می‌شود که نهایتا تابع هزینه به دست می‌آید. حال برای بهینه شدن کنترل‌کننده و حداقل‌سازی تابع هزینه، ضرایب کنترلی اتفاقی با توجه به محدودیت تعیین شده برای آن‌ها، انتخاب می‌شود. ضرایب کنترلی جدید به کنترل‌کننده و در نتیجه به ربات اعمال می‌شود و باز هم تابع هزینه مطابق با خطای ردیابی مفاصل و گشتاورها حاصل می‌شود.

پس از بررسی نتایج حاصله و مقایسه با نتایج مرحله قبل، یکی از ضرایب کنترلی که نتیجه بهتری داده باشد به عبارتی میزان تابع هزینه را کمتر کرده باشد، به عنوان ضریب کنترلی اصلی در نظر گرفته می‌شود. سپس برای نتیجه‌گیری بهتر، ضرایب کنترلی اتفاقی دیگری انتخاب می‌شود و مراحل قبل تکرار می‌شود. در هر مرحله با مقایسه با بهترین نتایج مراحل قبل، ضریبی انتخاب می‌شود که بهترین نتیجه را داده باشد. بهترین ضریب، ضریبی است که با استفاده از آن در کنترل‌کننده تابع هزینه را حداقل کند.

تابع هزینه پیشنهادی برای این بهینه سازی به شکل زیر تعریف می‌شود:

$$y = e^2 + u^2 \qquad (۴-۵۱)$$

که e اندازه خطای ردیابی مفاصل و u اندازه گشتاور اعمالی به ربات می‌باشد. اندازه خطا به فرم زیر می‌باشد.

$$e = \sqrt{{e_1}^2 + {e_2}^2 + {e_3}^2} \tag{۴-۵۲}$$

که e_1, e_2, e_3 شامل خطای ردیابی مفاصل θ_1 مربوط به سه ساق ربات هستند یعنی:

$$\begin{aligned} e_1 &= \theta_{11d} - \theta_{11} \\ e_2 &= \theta_{12d} - \theta_{12} \\ e_3 &= \theta_{13d} - \theta_{13} \end{aligned} \tag{۴-۵۳}$$

گشتاور اعمالی به سه محرک ربات شامل τ_1, τ_2, τ_3 می‌باشد که اندازه آن به فرم زیر می‌باشد.

$$u = \sqrt{{\tau_1}^2 + {\tau_2}^2 + {\tau_3}^2} \tag{۴-۵۴}$$

لازم به ذکر است که الگوریتم pso به گونه ای نوشته می‌شود که تابع هزینه را ماکزیمم می‌کند. پس تابع هزینه باید طوری انتخاب شود که با به حداکثر رسیدن خود، خطا و گشتاور را به حداقل مقدار خود برساند. برای این امر، تابع هزینه قرینه تابع مورد نظر در نظر گرفته می‌شود.

$$y = -(e^2 + u^2) \tag{۴-۵۵}$$

بنابراین تابع هزینه برای بهینه‌سازی ضرایب کنترل‌کننده مود لغزشی به شکل فوق انتخاب شده است. پس از بررسی نتایج، مشاهده می‌شود که اندازه گشتاور و اندازه خطا به یک میزان و در یک رنج نیستند و مقدار اندازه گشتاور بسیار زیاد بوده است. با جمع کردن اندازه گشتاور با اندازه خطا که مقدار بسیار کمتری دارد، نتایج

مطلوبی حاصل نخواهد شد و در این صورت خطا نادیده گرفته می‌شود. به عبارتی حداقل‌سازی فقط برای اندازه گشتاور صورت می‌پذیرد. بنابراین برای یکسان کردن رنج مقادیر خطا و گشتاور، تابع هزینه به شکل زیر تغییر می‌یابد.

$$y = -\left(e^2 + \frac{u^2}{1000}\right) \tag{۴-۵۶}$$

۴-۳ شبیه‌سازی و نتایج

۴-۳-۱ نتایج شبیه‌سازی اعمال کنترل‌کننده مود لغزشی

در این بخش نتایج شبیه‌سازی سیستم حلقه بسته برای کنترل مود لغزشی ربات دلتا روی همان مسیر دلخواه (۳-۳۸) قابل مشاهده می‌باشد.

جهت یادآوری:

$$\begin{aligned} x_d &= 0.01 + 0.2\cos t \\ y_d &= 0.01 + 0.2\sin t \\ z_d &= 0.4 \end{aligned} \tag{۳-۳۸}$$

همچنین مقادیر ضرایب کنترلی λ و η که به روش صحیح و خطا انتخاب شده‌اند، در زیر قابل مشاهده می‌باشد.

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{۴-۵۷}$$

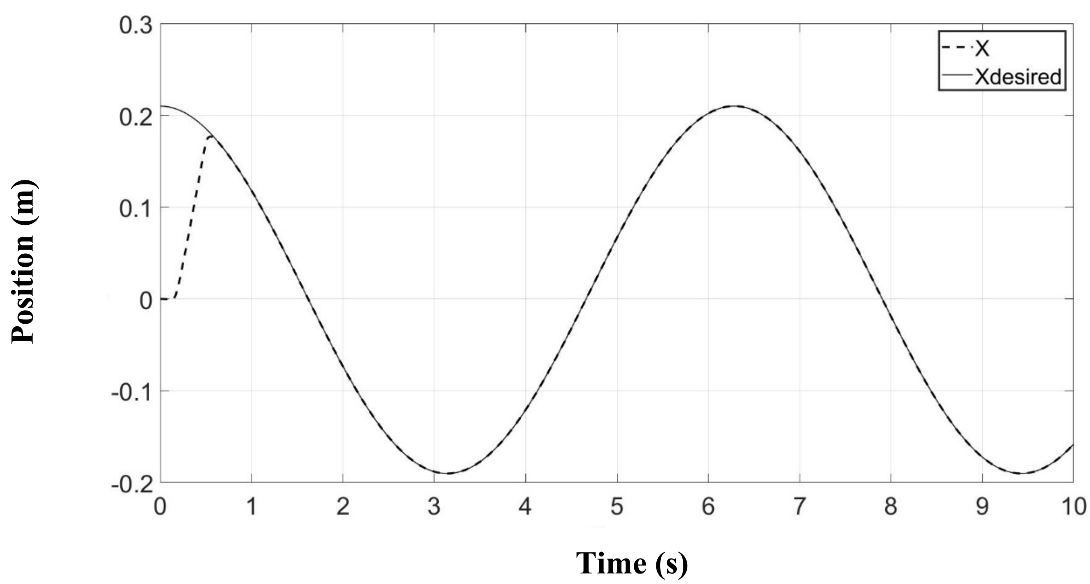
$$\lambda_1 = 30\,, \lambda_2 = 30\,, \lambda_3 = 30$$

و

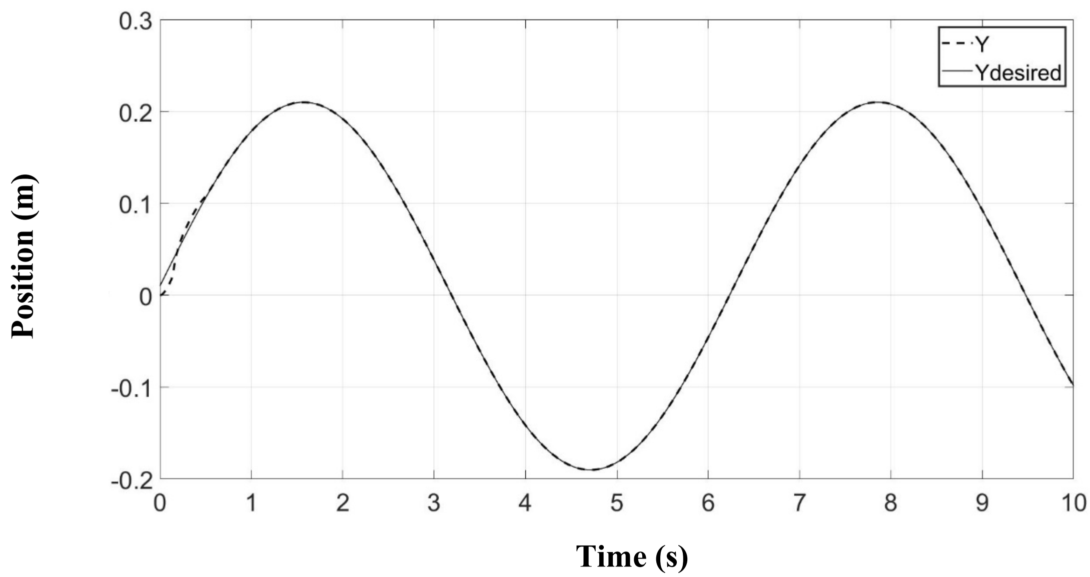
$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \tag{۴-۵۸}$$

$$\eta_1 = 1 \,, \eta_2 = 1 \,, \eta_3 = 1$$

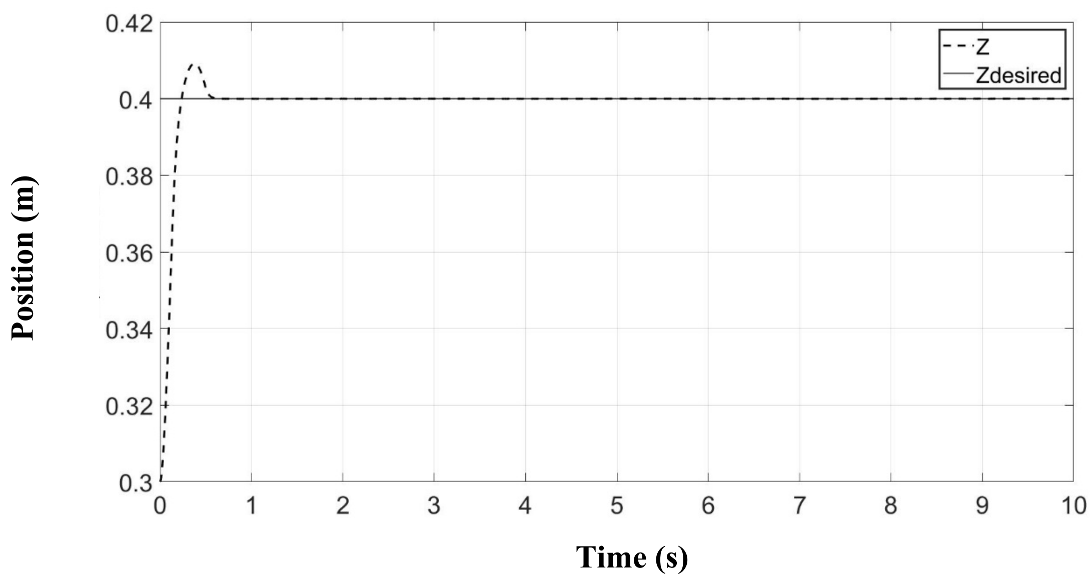
نتایج شبیه‌سازی برای بررسی عملکرد ردیابی، در شکل‌های زیر ارائه شده است. منحنی شکل (۳-۴) تعقیب مسیر مطلوب را برای مختصات x، منحنی (۴-۴) برای مختصات y ومنحنی (۵-۴) برای مختصات z نشان می-دهد.



شکل ۳-۴. نمودار ردیابی مجری نهایی در مسیر x

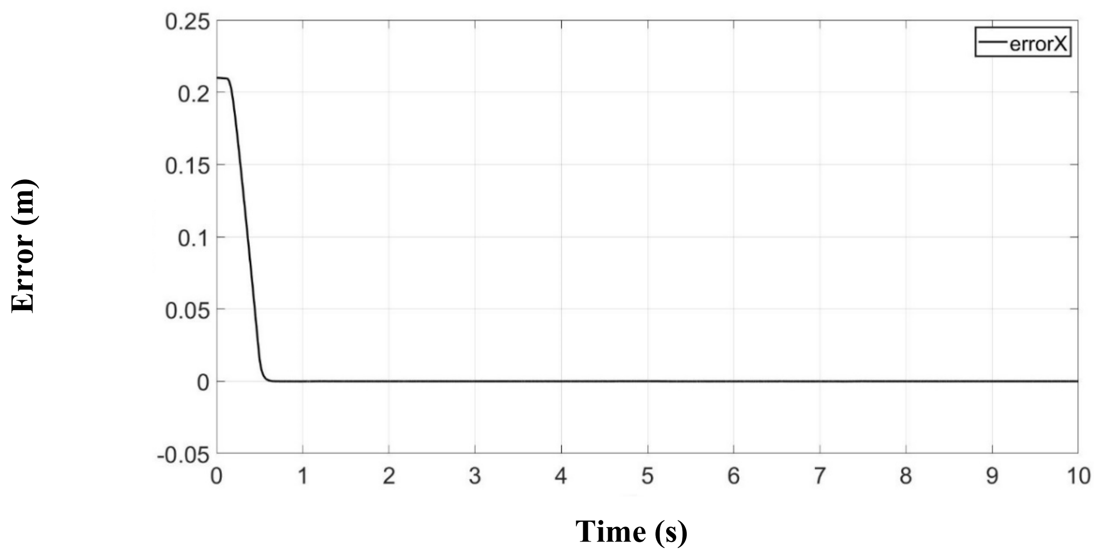


شکل ۴-۴. نمودار ردیابی مجری نهایی در مسیر y

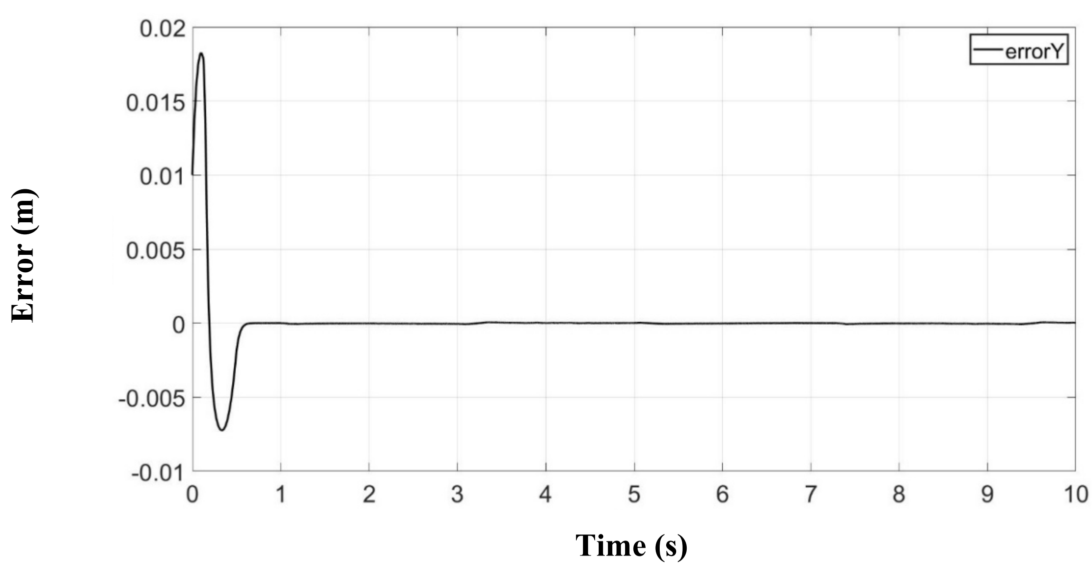


شکل ۴-۵. نمودار ردیابی مجری نهایی در مسیر z

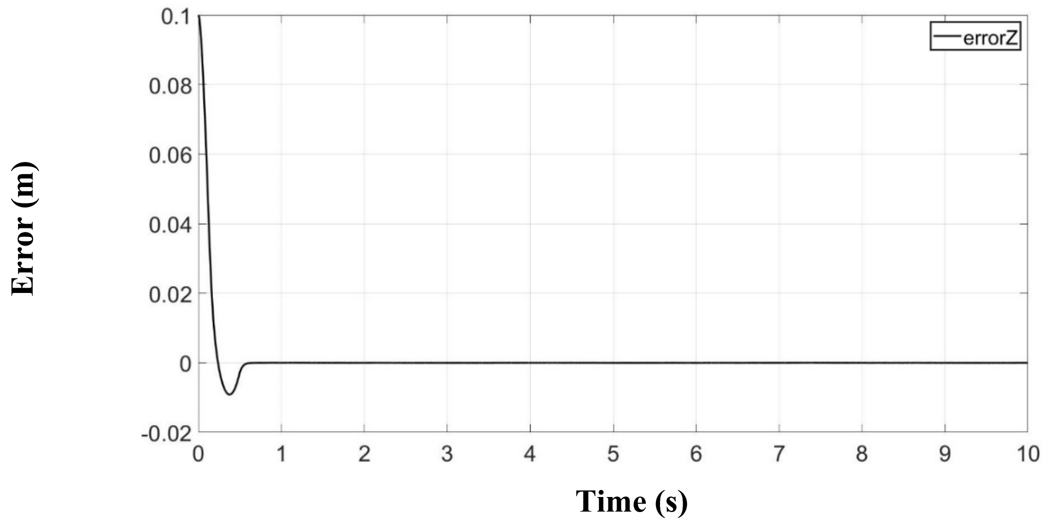
منحنی‌های شکل (۶-۴)، (۷-۴) و (۸-۴) اختلاف بین مسیرهای مطلوب و تعقیب آن مسیرها توسط ربات را نشان می‌دهد.



شکل ۶-۴. نمودار خطای ردیابی مسیر مجری نهایی در مختصات x

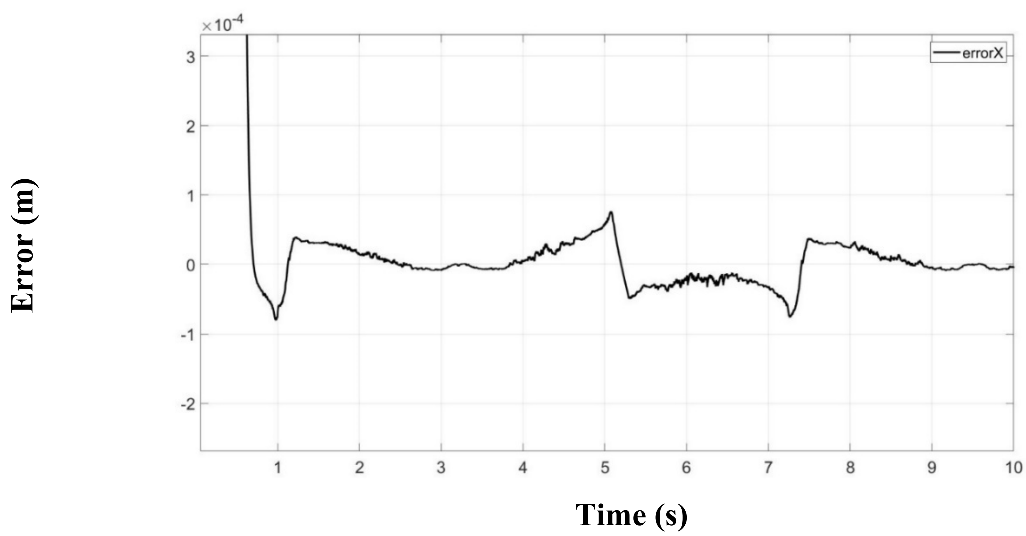


شکل ۷-۴. نمودار خطای ردیابی مسیر مجری نهایی در مختصات y

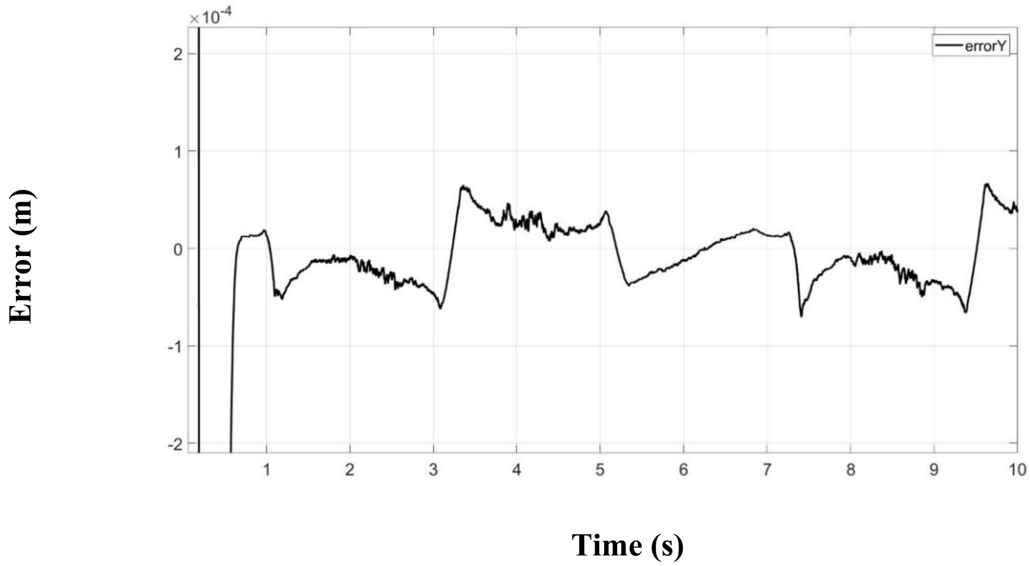


شکل ۸-۴. نمودار خطای ردیابی مسیر مجری نهایی در مختصات z

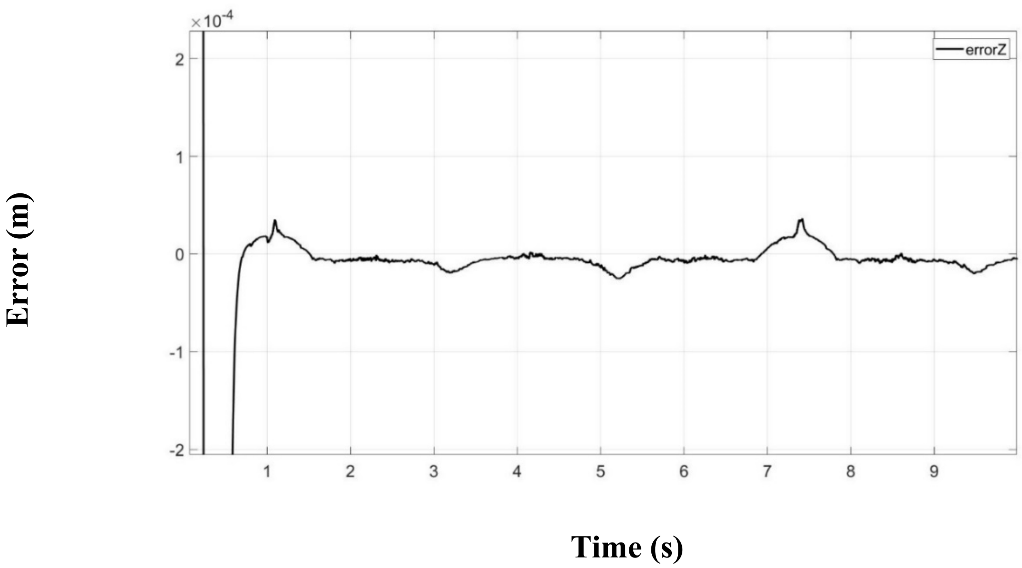
همچنین منحنی‌های شکل (۹-۴)، (۱۰-۴) و (۱۱-۴) نیز اختلاف بین مسیرهای مطلوب و تعقیب آن مسیرها توسط ربات که برای بهتر دیده شدن، بزرگ‌نمایی شده است را نشان می‌دهد.



شکل ۹-۴. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات x

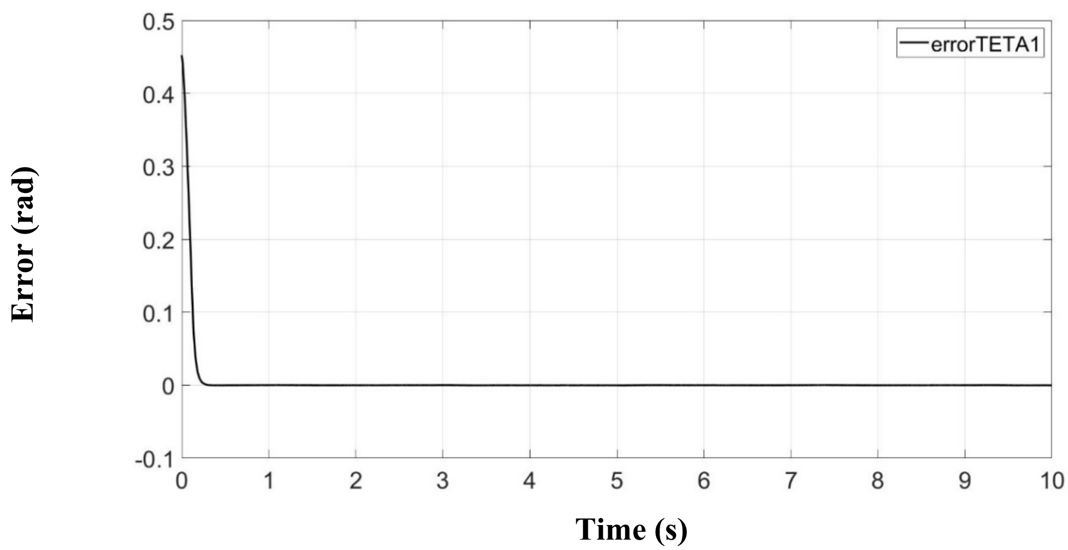


شکل ۱۰-۴. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات y

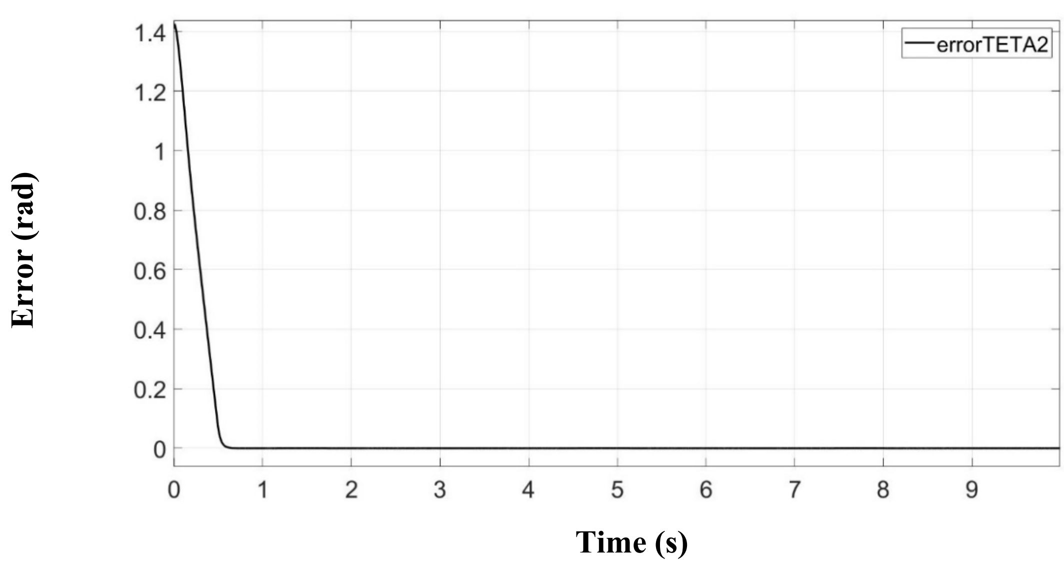


شکل ۱۱-۴. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات z

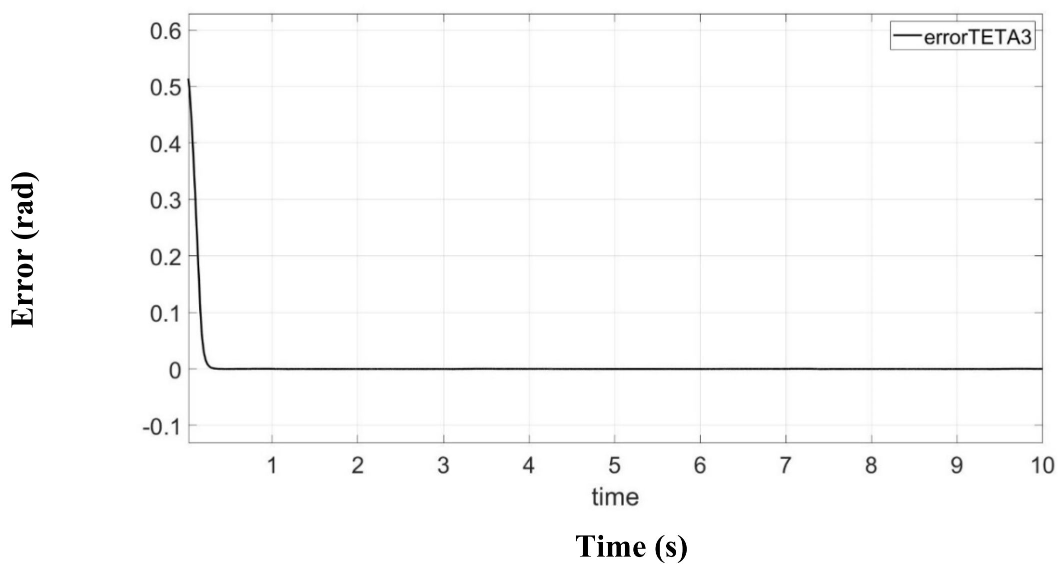
منحنی‌های شکل (۴-۱۲)، (۴-۱۳) و (۴-۱۴) اختلاف بین زوایای مفصلی مطلوب و تعقیب آن توسط ربات را نشان می‌دهد.



شکل ۴-۱۲. نمودار خطای زاویه مفصلی ساق اول

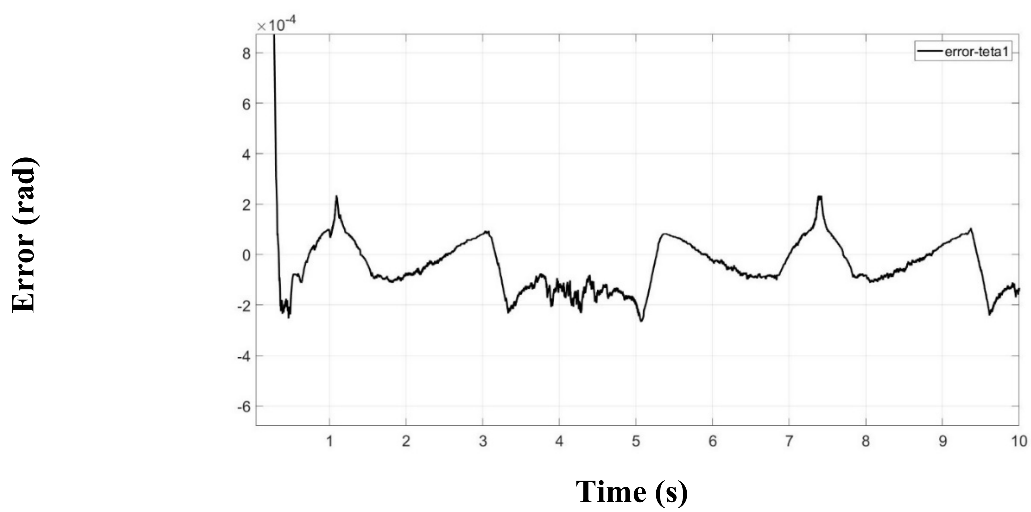


شکل ۴-۱۳. نمودار خطای زاویه مفصلی ساق دوم

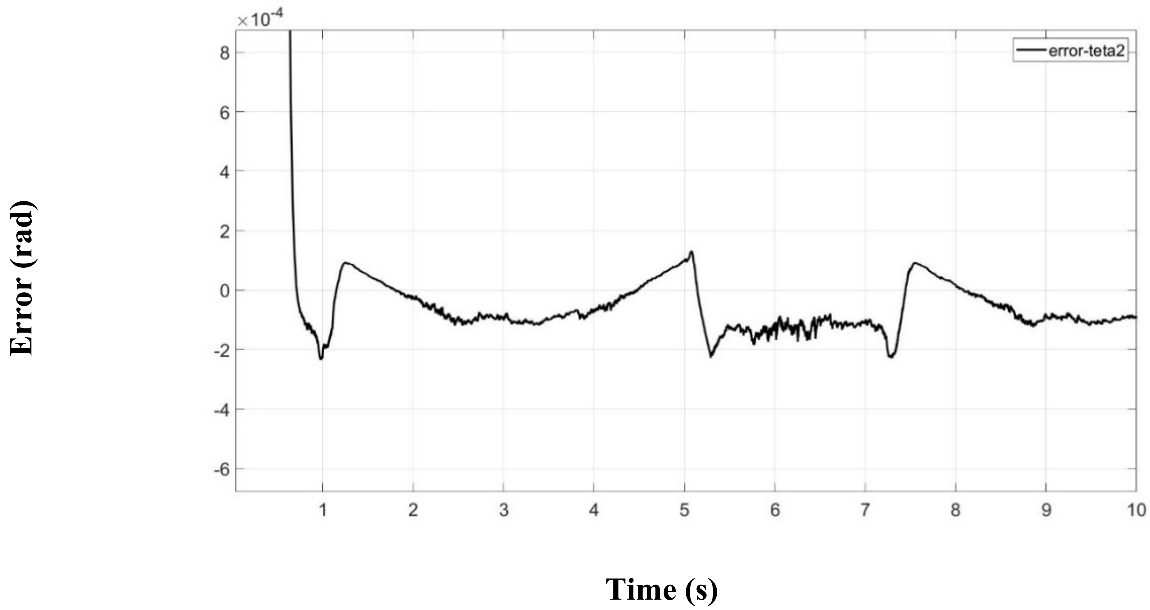


شکل ۱۴-۴. نمودار خطای زاویه مفصلی ساق سوم

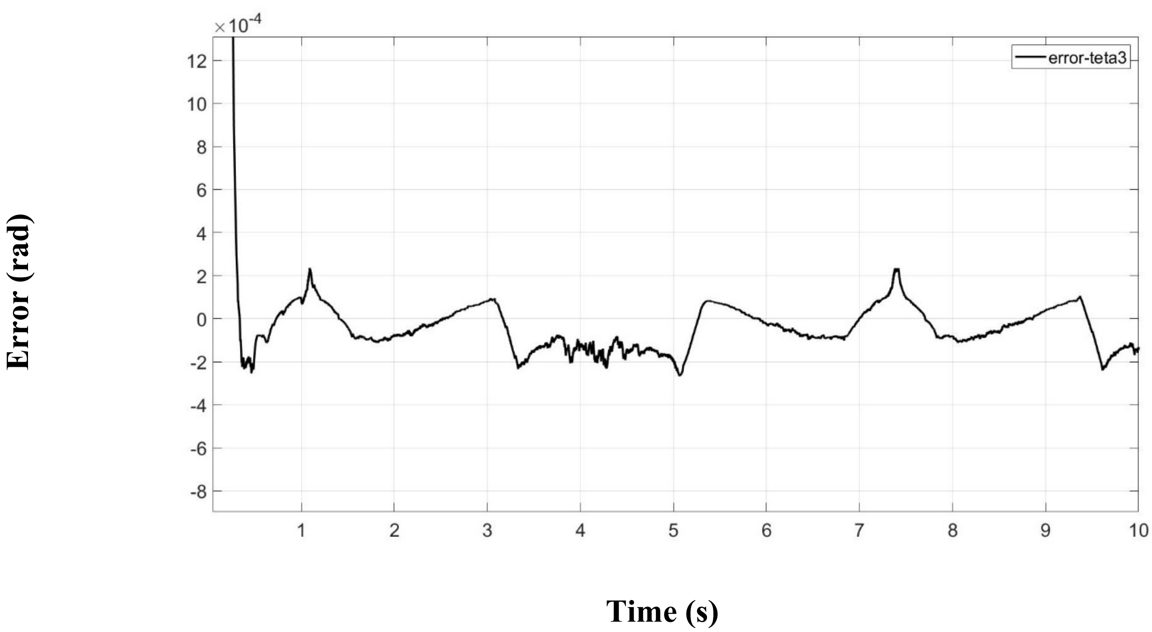
همچنین منحنی‌های شکل (۱۵-۴)، (۱۶-۴) و (۱۷-۴) نیز اختلاف بین زوایای مفصلی مطلوب و تعقیب آن توسط ربات که برای بهتر دیده شدن، بزرگ‌نمایی شده است را نشان می‌دهد.



شکل ۱۵-۴. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق اول

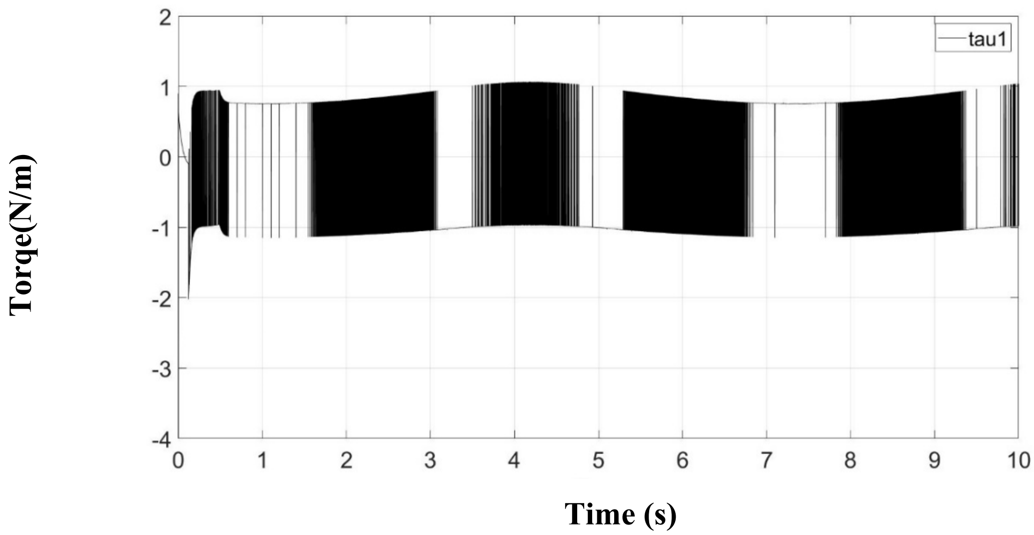


شکل ۱۶-۴. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق دوم

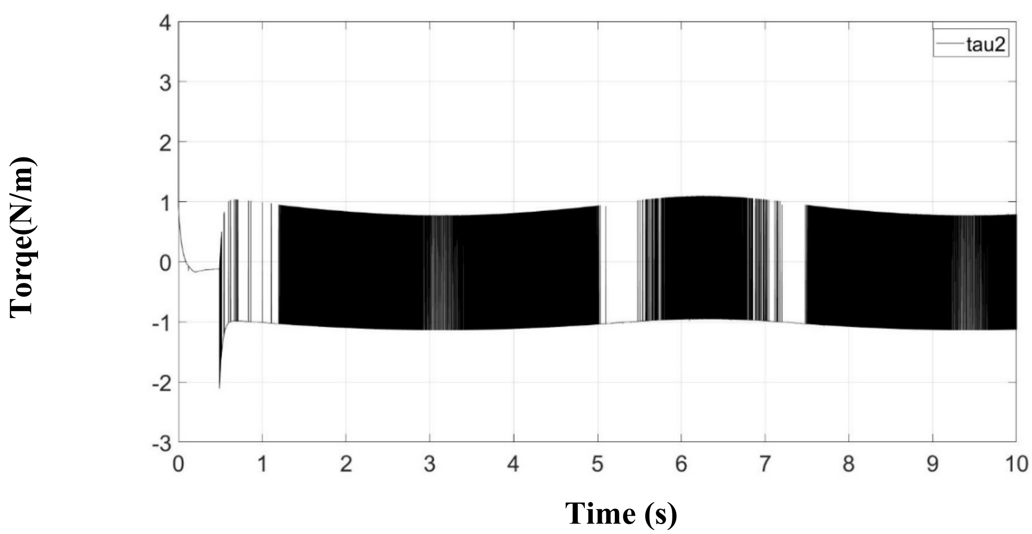


شکل ۱۷-۴. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق سوم

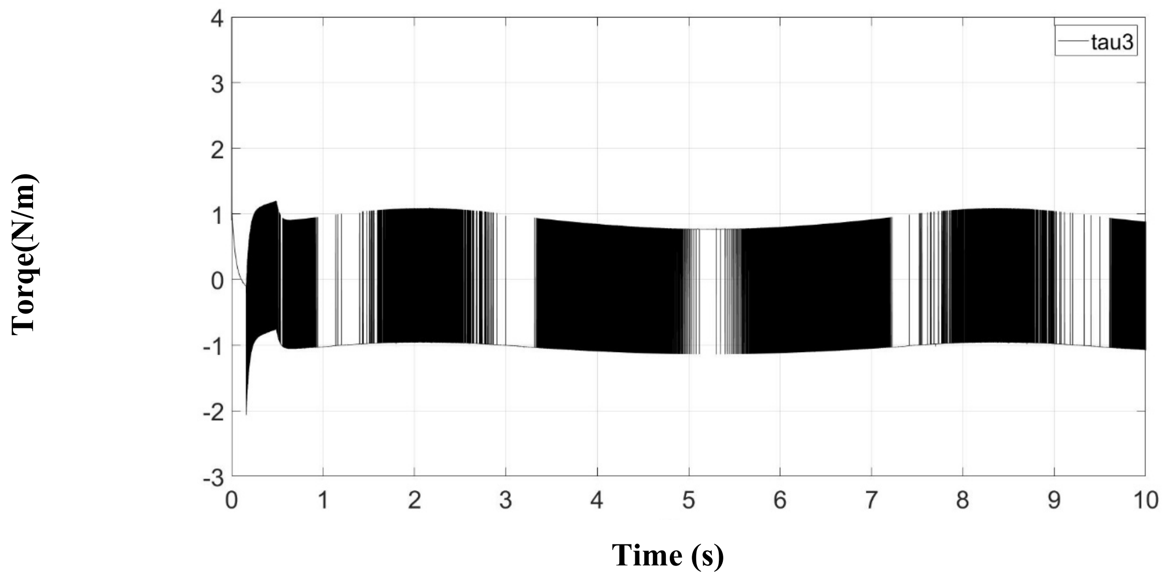
گشتاور کنترلی به دست آمده، برداری شامل سه گشتاور بوده که منحنی‌های شکل (۴-۱۸)، (۴-۱۹) و (۴-۲۰) نشان‌دهنده این گشتاورهای کنترلی می‌باشد.



شکل ۴-۱۸. نمودار گشتاور کنترلی اول



شکل ۴-۱۹. نمودار گشتاور کنترلی دوم



شکل ۲۰-۴. نمودار گشتاور کنترلی سوم

۴-۳-۲ نتایج شبیه‌سازی اعمال کنترل‌کننده مود لغزشی بهینه‌سازی‌شده با الگوریتم pso

در این بخش نتایج شبیه‌سازی سیستم حلقه بسته برای کنترل مود لغزشی ربات دلتا که ضرایب آن با استفاده از الگوریتم بهینه‌سازی انبوه ذرات بهینه شده‌اند، روی همان مسیر دلخواه (۳۸-۳) قابل مشاهده می‌باشد.

همچنین مقادیر ضرایب کنترلی بهینه λ و η که با استفاده از روش بهینه‌سازی pso حاصل شده‌اند، در زیر قابل مشاهده می‌باشد.

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{۴-۵۹}$$

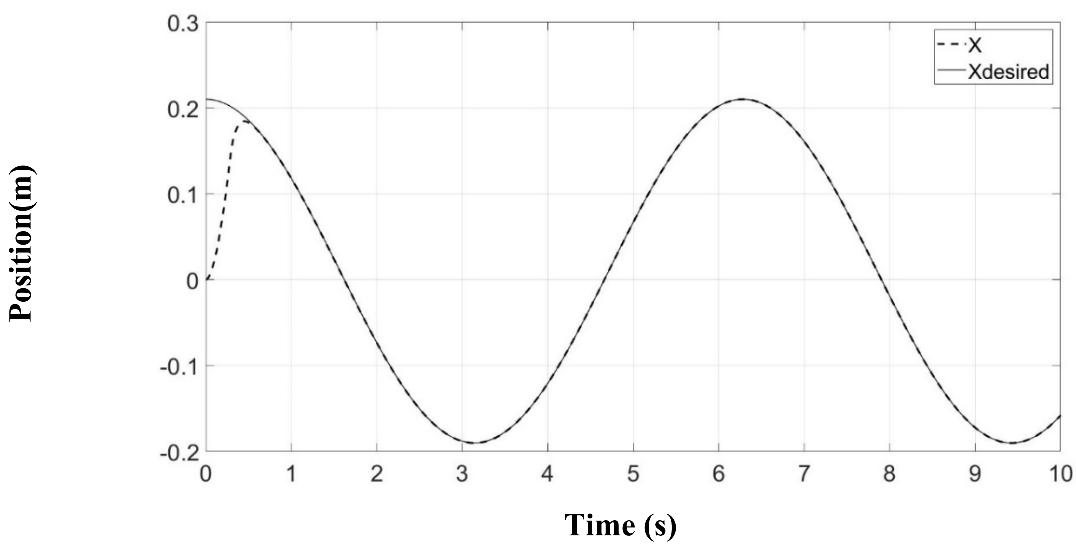
$$\lambda_1 = 27.3370\,, \lambda_2 = 14.7761\,, \lambda_3 = 10.5024$$

و

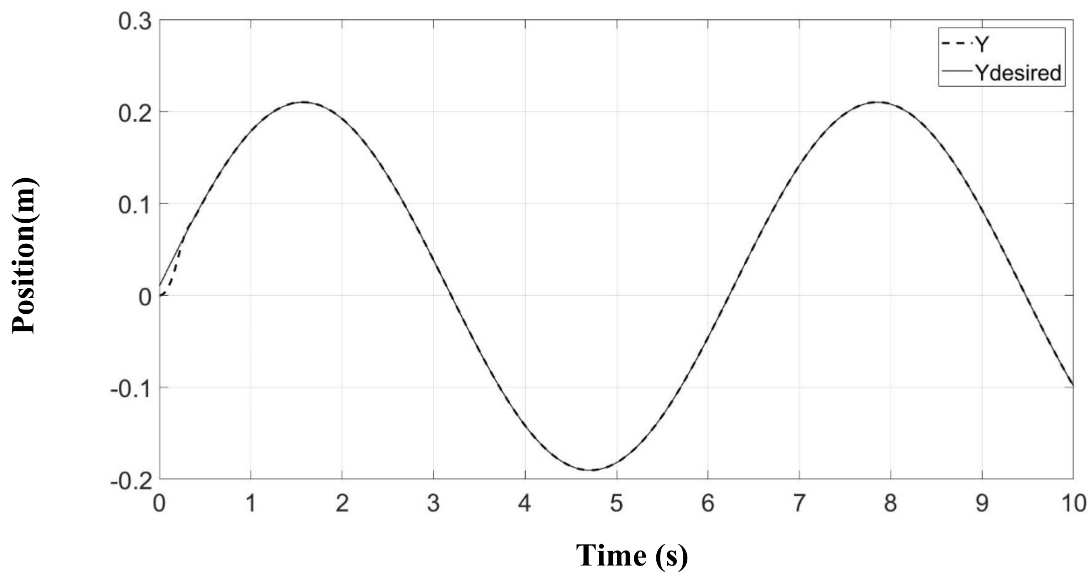
$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \tag{۴-۶۰}$$

$$\eta_1 = 0.3374\,, \eta_2 = 0.8018\,, \eta_3 = 0.2382$$

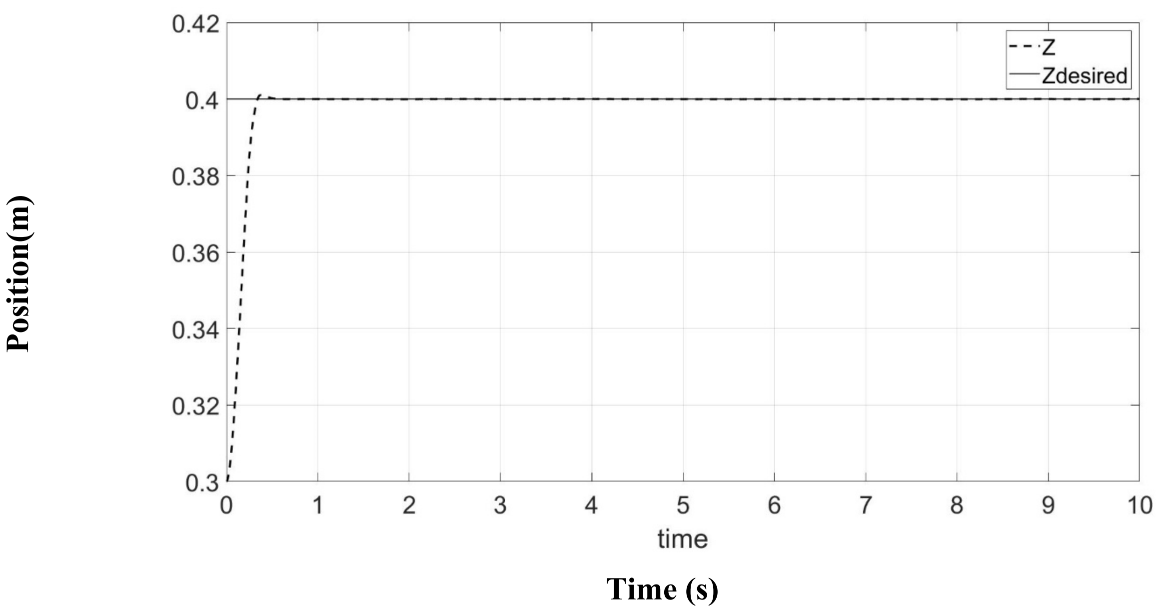
نتایج شبیه‌سازی برای بررسی عملکرد ردیابی، در شکل‌های زیر ارائه شده است. منحنی شکل (۲۱-۴) تعقیب مسیر مطلوب را برای مختصات x، منحنی (۲۲-۴) برای مختصات y ومنحنی (۲۳-۴) برای مختصات z نشان می‌دهد.



شکل ۲۱-۴. نمودار ردیابی مجری نهایی در مسیر x

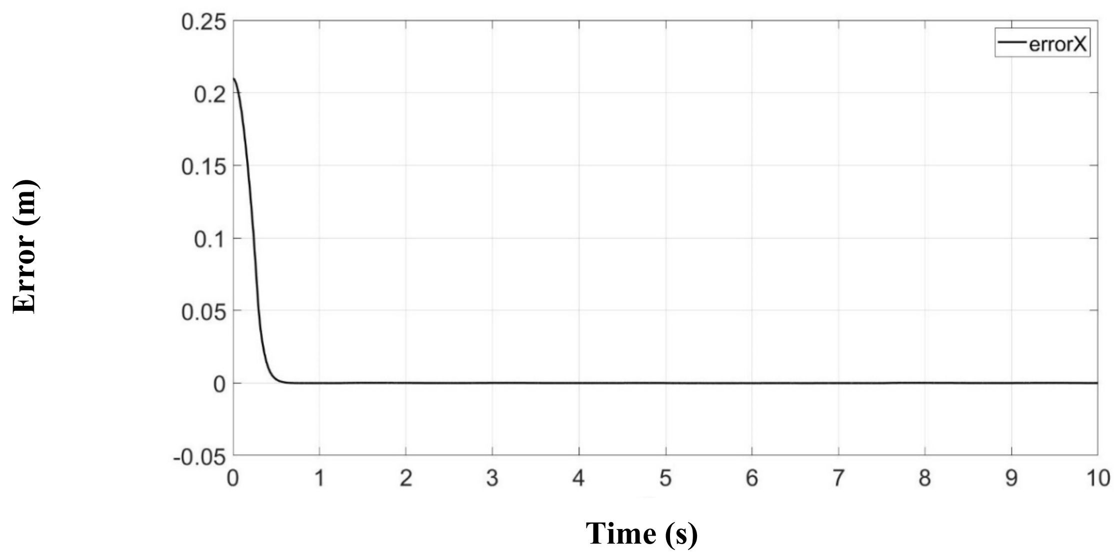


شکل ۴-۲۲. نمودار ردیابی مجری نهایی در مسیر y

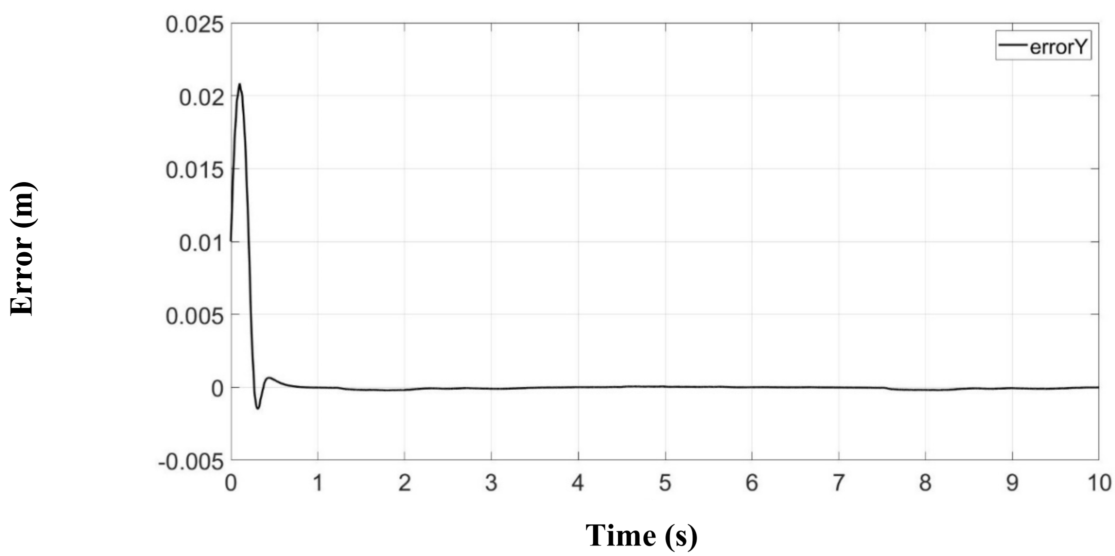


شکل ۴-۲۳. نمودار ردیابی مجری نهایی در مسیر z

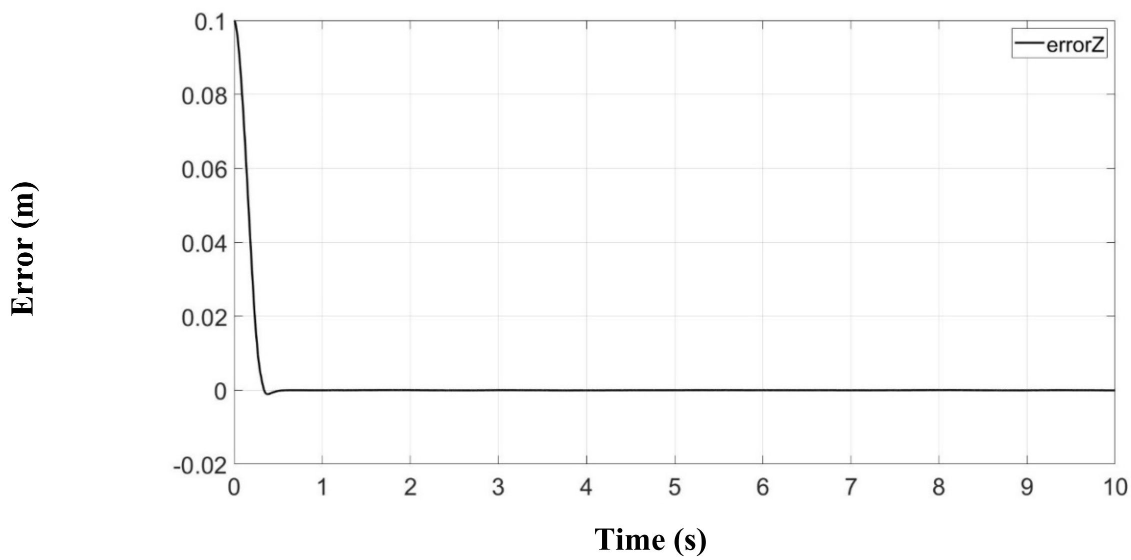
منحنی‌های شکل (۴-۲۴)، (۴-۲۵) و (۴-۲۶) اختلاف بین مسیرهای مطلوب و تعقیب آن مسیرها توسط ربات را نشان می‌دهد.



شکل ۴-۲۴. نمودار خطای ردیابی مسیر مجری نهایی در مختصات x

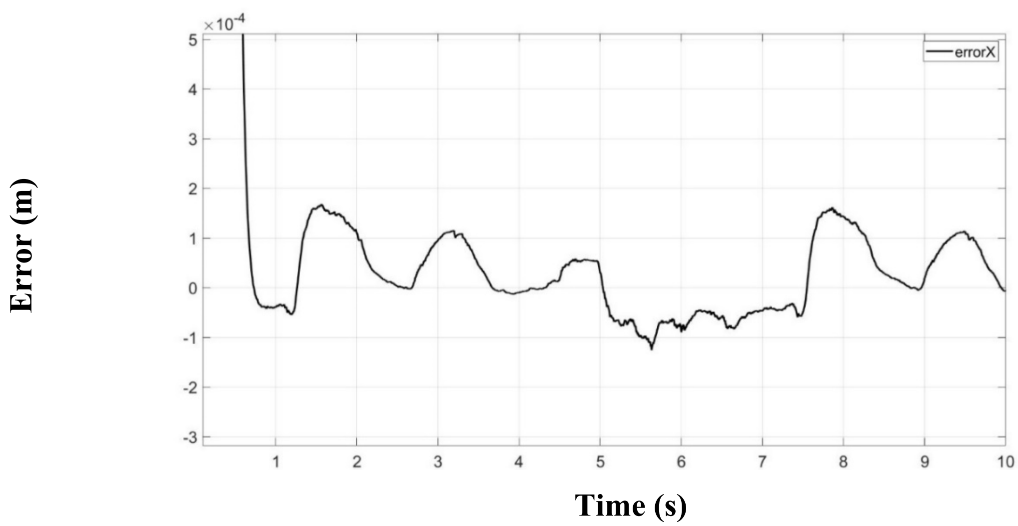


شکل ۴-۲۵. نمودار خطای ردیابی مسیر مجری نهایی در مختصات y

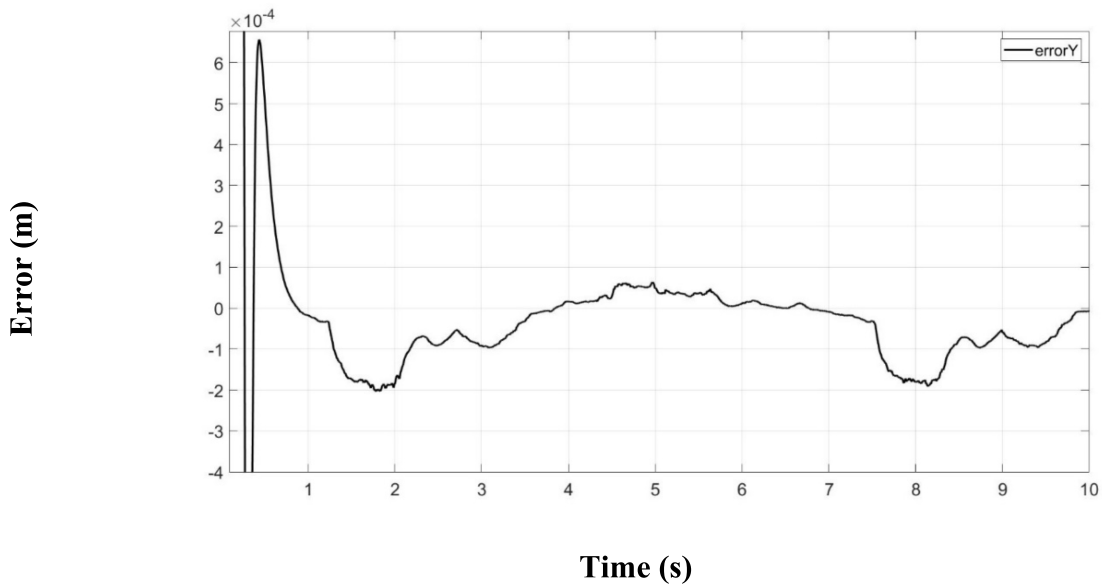


شکل ۴-۲۶. نمودار خطای ردیابی مسیر مجری نهایی در مختصات z

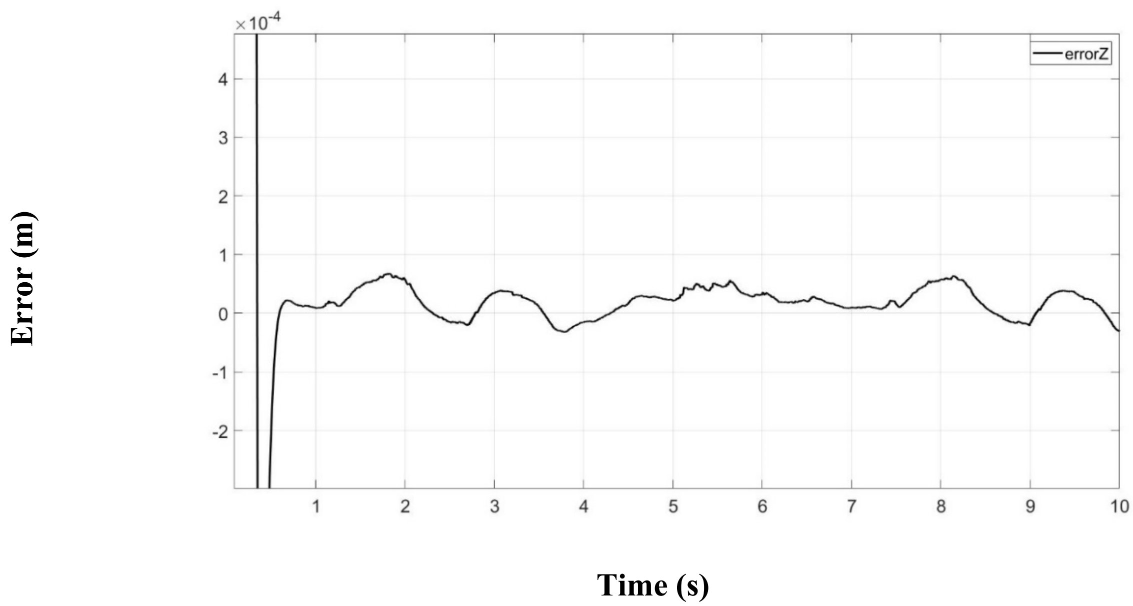
همچنین منحنی‌های شکل (۴-۲۷)، (۴-۲۸) و (۴-۲۹) نیز اختلاف بین مسیرهای مطلوب و تعقیب آن مسیرها توسط ربات که برای بهتر دیده شدن، بزرگ‌نمایی شده است را نشان می‌دهد.



شکل ۴-۲۷. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات x

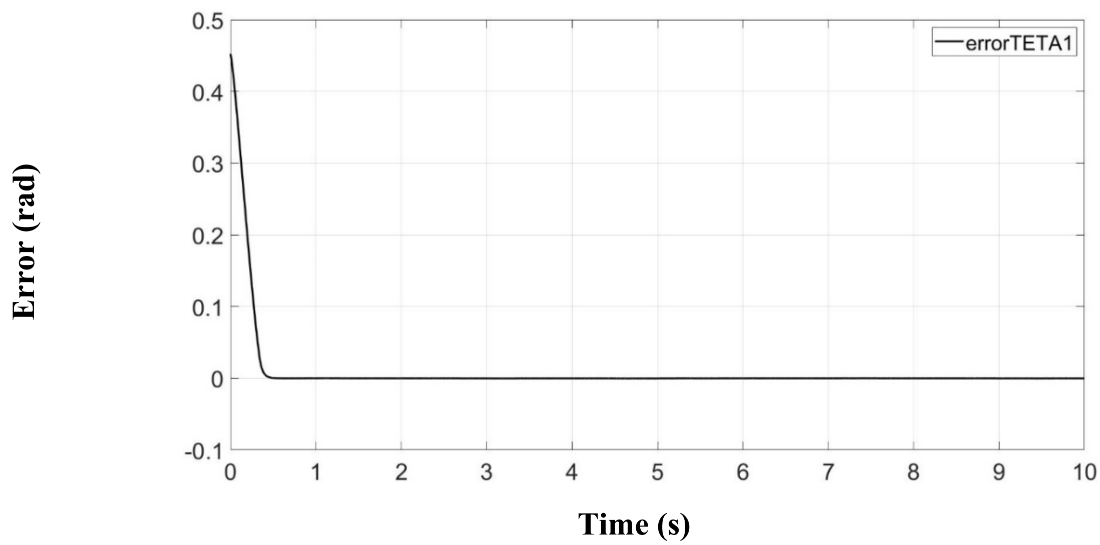


شکل ۴-۲۸. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات y

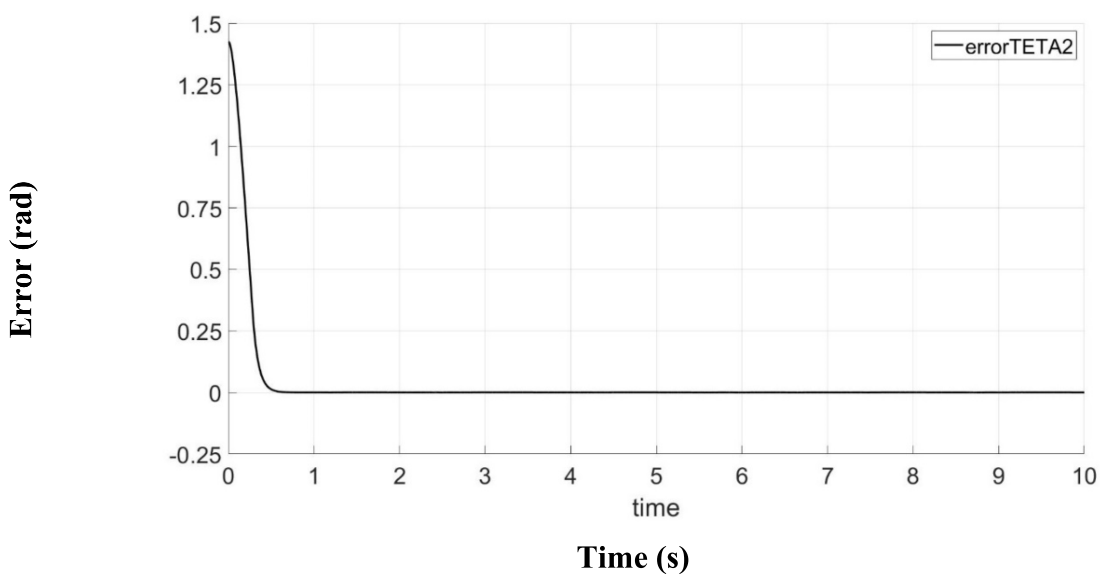


شکل ۴-۲۹. نمودار بزرگ‌نمایی شده خطای ردیابی مسیر مجری نهایی در مختصات z

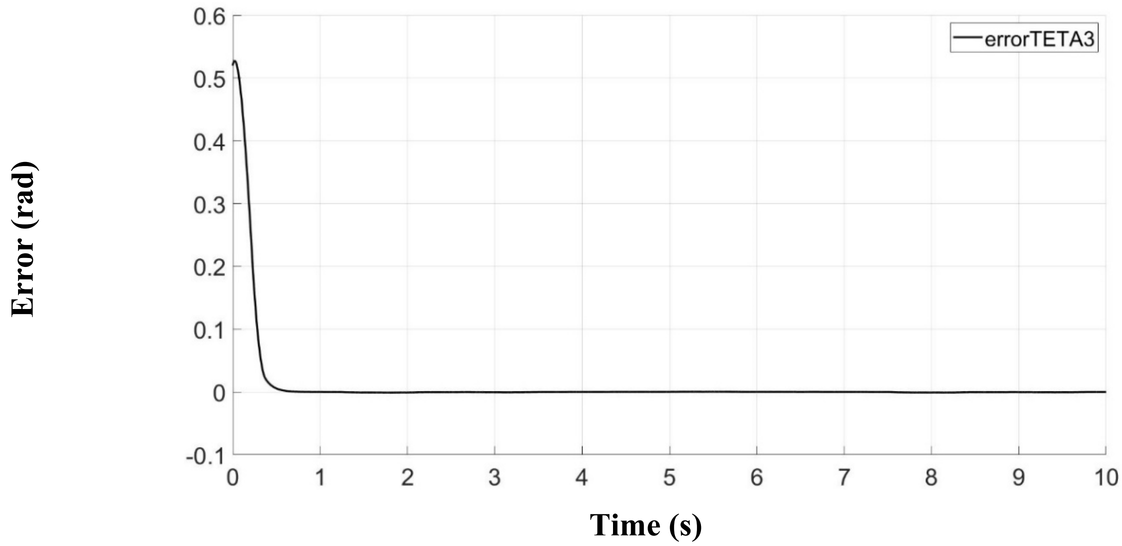
منحنی‌های شکل (۴-۳۰)، (۴-۳۱) و (۴-۳۲) اختلاف بین زوایای مفصلی مطلوب و تعقیب آن توسط ربات را نشان می‌دهد.



شکل ۴-۳۰. نمودار خطای زاویه مفصلی ساق اول

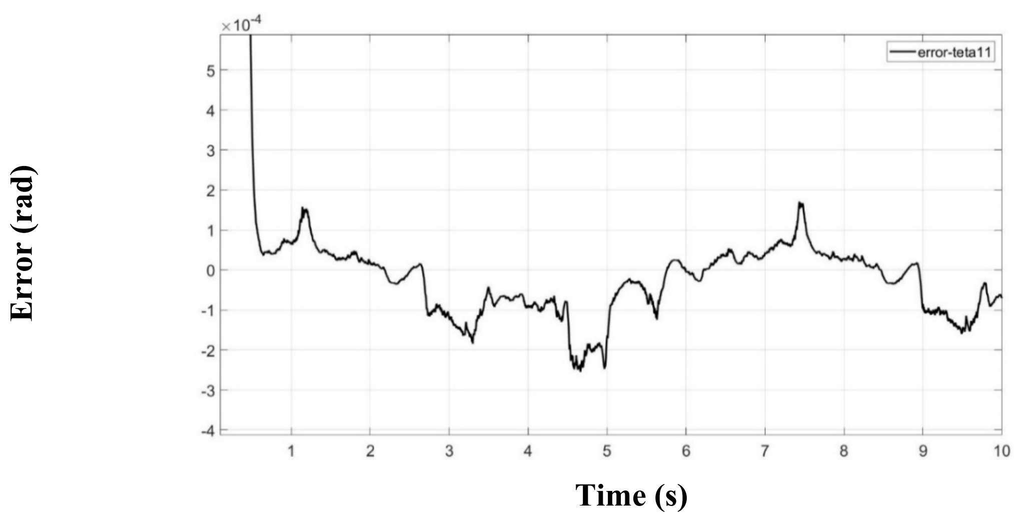


شکل ۴-۳۱. نمودار خطای زاویه مفصلی ساق دوم

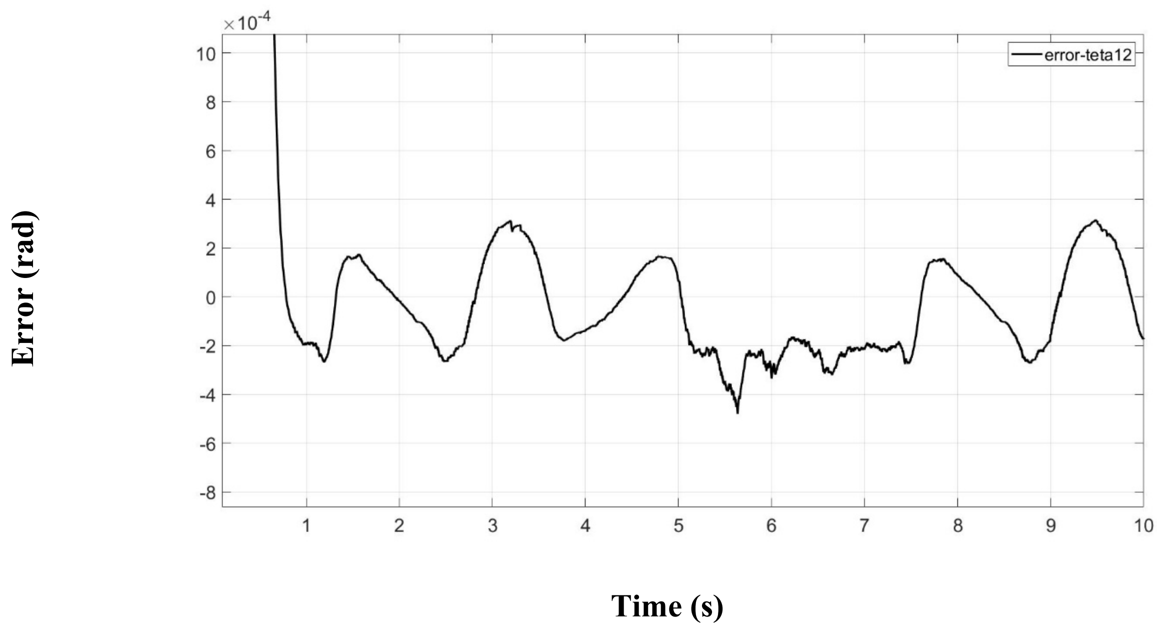


شکل ۴-۳۲. نمودار خطای زاویه مفصلی ساق سوم

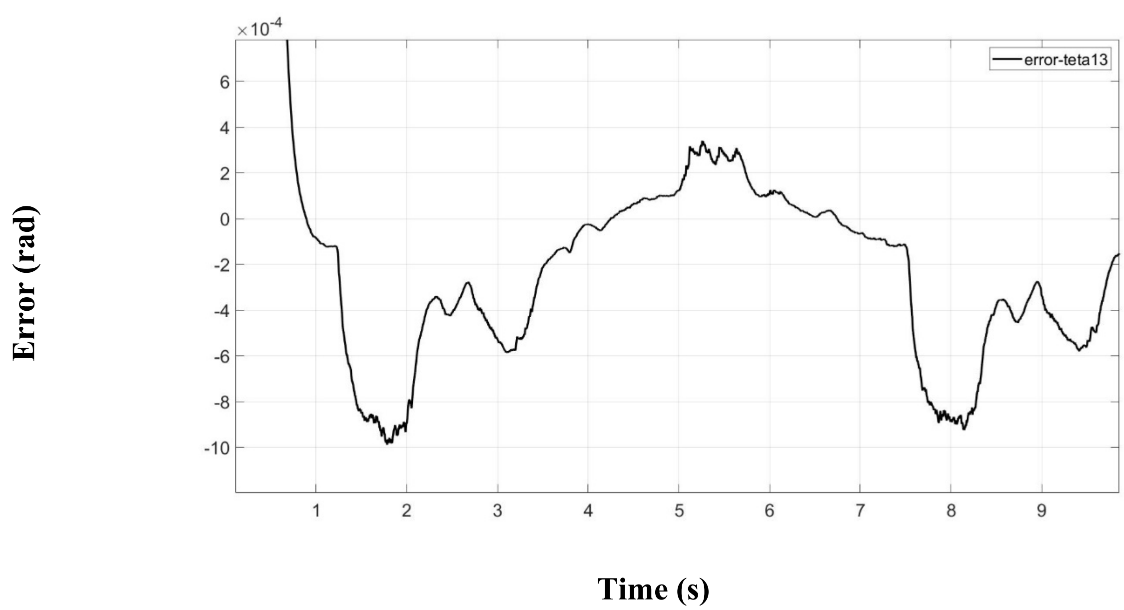
همچنین منحنی‌های شکل (۴-۳۳)، (۴-۳۴) و (۴-۳۵) نیز اختلاف بین زوایای مفصلی مطلوب و تعقیب آن توسط ربات که برای بهتر دیده شدن، بزرگ‌نمایی شده است را نشان می‌دهد.



شکل ۴-۳۳. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق اول

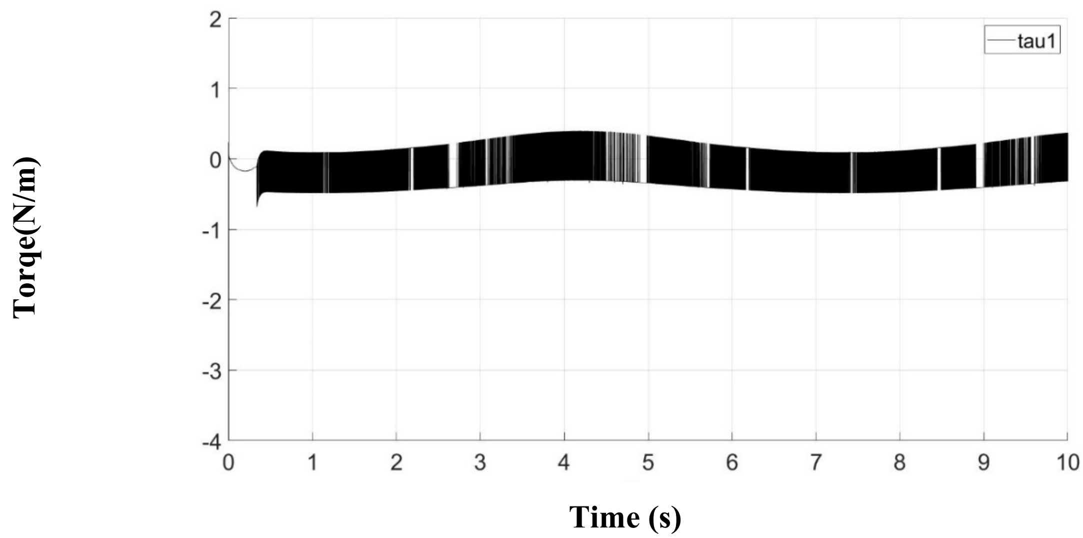


شکل ۴-۳۴. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق دوم

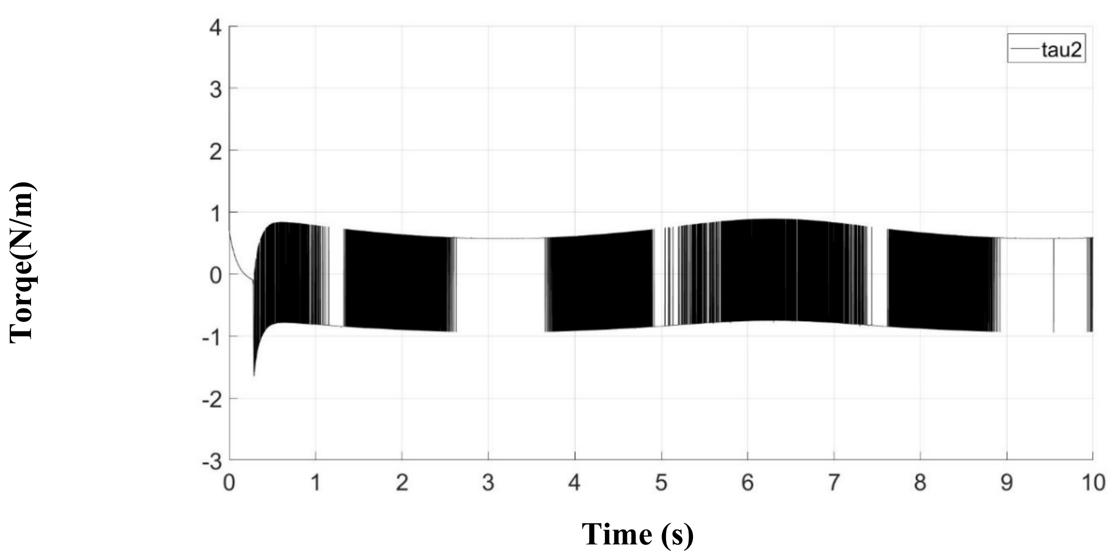


شکل ۴-۳۵. نمودار بزرگ‌نمایی شده خطای زاویه مفصلی ساق سوم

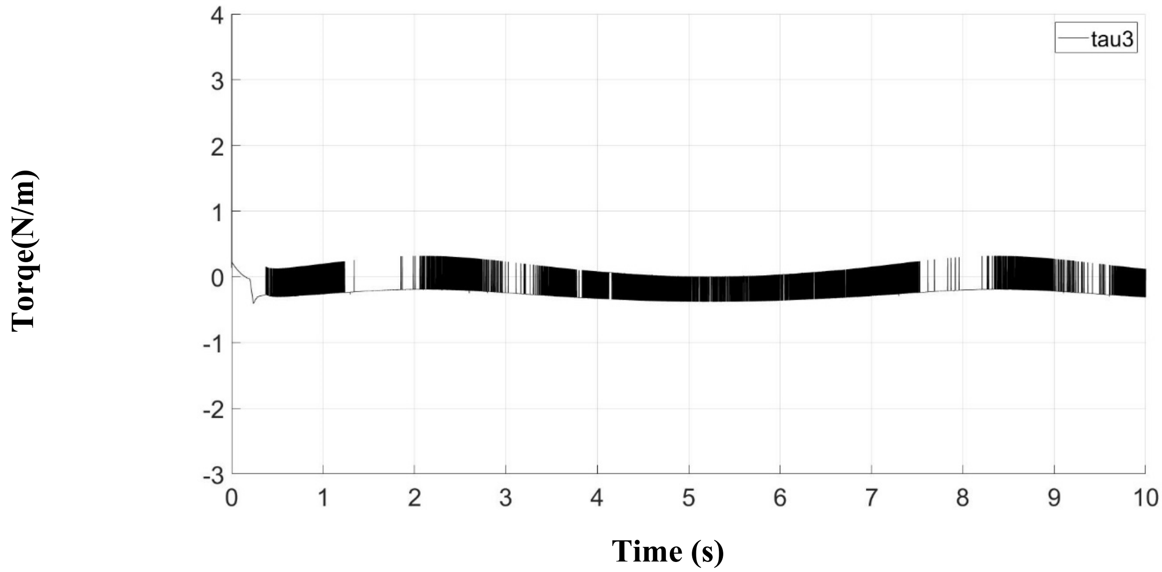
گشتاور کنترلی به دست آمده، ماتریسی شامل سه گشتاور بوده که منحنی‌های شکل (۳۶-۴)، (۳۷-۴) و (۳۸-۴) نشان‌دهنده این گشتاورهای کنترلی می‌باشد.



شکل ۳۶-۴. نمودار گشتاور کنترلی اول



شکل ۳۷-۴. نمودار گشتاور کنترلی دوم



شکل ۳۸-۴. نمودار گشتاور کنترلی سوم

فصل ۵: بررسی نتایج

۵-۱ مقایسه نتایج کنترل‌کننده‌ها و نتیجه‌گیری

در این تحقیق، علاوه بر طراحی و شبیه‌سازی کنترل‌کننده مود لغزشی بهینه، طراحی و شبیه‌سازی یک کنترل‌کننده تطبیقی نیز انجام پذیرفت که به وسیله آن بتوان یک دیدگاه مقایسه ای برای تحلیل و بررسی عملکرد کنترل‌کننده مود لغزشی بهینه ایجاد کرد. همان‌طور که مشاهده شد، نتایج شبیه‌سازی‌ها برای هر کنترل‌کننده در هر بخش ارائه شده است که شامل نتایج ردیابی مسیرهای مطلوب توسط مجری نهایی ربات، خطای ردیابی مجری نهایی، خطای ردیابی مفاصل و توابع کنترلی مورد نیاز برای هر یک از آن‌ها می‌باشد.

برای شبیه‌سازی ربات و اعمال کنترل‌کننده بر روی آن یک مسیر دلخواه که در بخش قبل‌های قبل ذکر شد، جهت ردیابی مسیر توسط ربات دلتا انتخاب شد. این مسیر برای تمام کنترل‌کننده‌های طراحی شده یکسان بوده تا بتوان عملکرد کنترل‌کننده‌ها را در ردیابی مسیر مطلوب مشاهده نمود. همچنین پارامترهای فیزیکی ربات دلتا که در جدول (۳-۱) آمده است نیز برای تمام شبیه‌سازی‌ها یکسان می‌باشد.

هر یک از روش‌های کنترلی انجام شده دارای مزایا و معایبی هستند. از معایب کنترل‌کننده تطبیقی می‌توان به این نکته اشاره کرد که برای طراحی این کنترل‌کننده، نیاز به تغییر فرم دینامیک ربات از فرم متعارف (۵-۱) به شکل (۵-۲) می‌باشد.

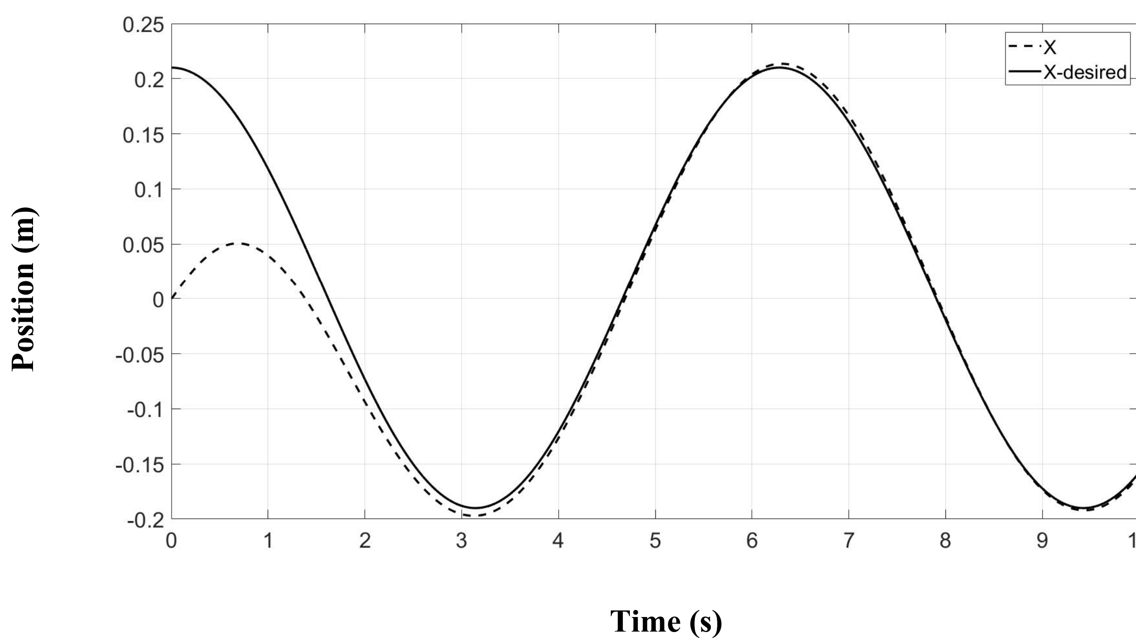
$$\tau = M(\theta)\ddot{\theta} + C\big(\theta, \dot{\theta}\big) \tag{۵-۱}$$

$$\tau = YP \tag{۵-۲}$$

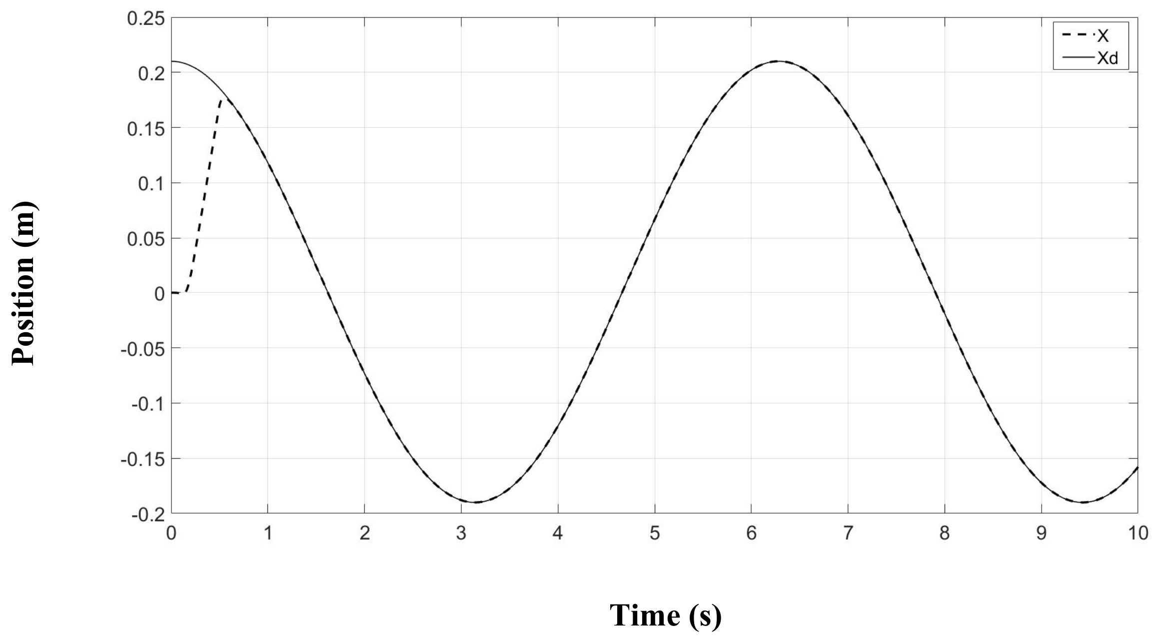
با توجه به اینکه ربات دلتا از دسته ربات‌های موازی است و ربات‌های موازی دارای دینامیک پیچیده‌تری نسبت به ربات‌های سری هستند، تغییر فرم دینامیک آن امری بسیار دشوار می‌باشد و نیاز به محاسبات پیچیده‌ای دارد.

نتایج ردیابی نشان می‌دهند که کنترل‌کننده مودلغزشی نسبت به کنترل‌کننده تطبیقی، ردیابی بهتر و دقیق‌تری داشته است. همچنین این نتایج و نتایج خطای ردیابی نشان می‌دهند که سریع‌تر به مسیر مطلوب رسیده است، به عبارتی سریع‌تر خطا به سمت صفر میل کرده است.

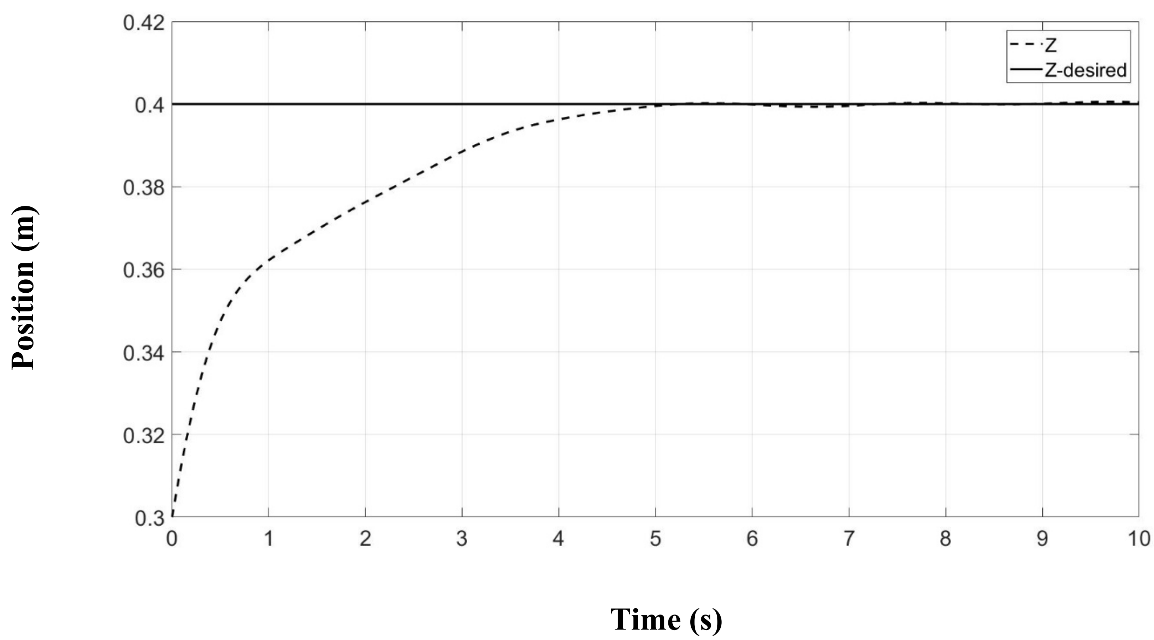
برای نمونه، نمودارهای ردیابی مجری نهایی در مسیرهای x و z که ربات با استفاده از کنترل مودلغزشی و کنترل تطبیقی حرکت می‌کند، در ادامه آورده شده است. نمودار شکل (۱-۵) و (۲-۵) ، ردیابی مسیر توسط ربات با کنترل تطبیقی را نشان‌می‌دهد و نمودارهای (۳-۵) و (۴-۵) ردیابی مسیر توسط ربات با کنترل مود لغزشی را نشان می‌دهند.



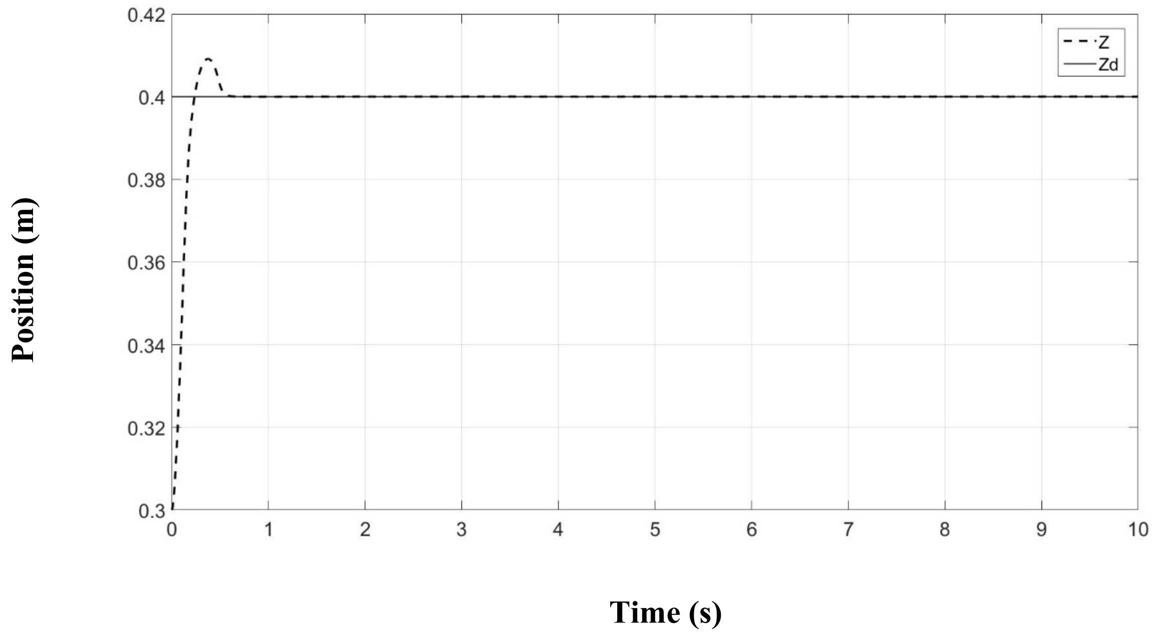
شکل ۱-۵. نمودار ردیابی مجری نهایی در مسیر x با اعمال کنترل تطبیقی



شکل ۲-۵. نمودار ردیابی مجری نهایی در مسیر x با اعمال کنترل مود لغزشی

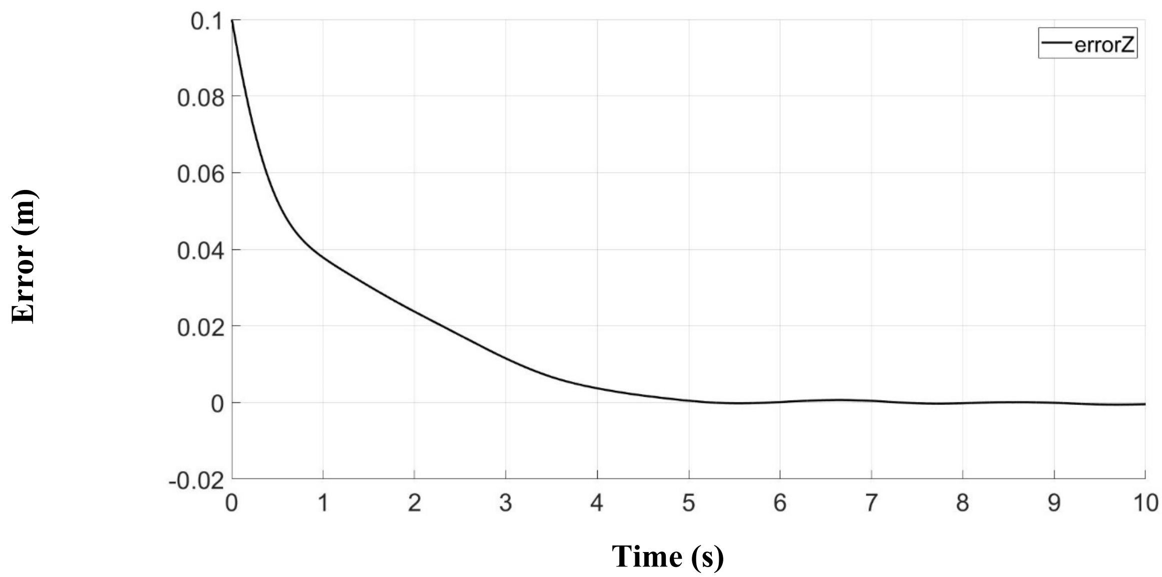


شکل ۳-۵. نمودار ردیابی مجری نهایی در مسیر z با اعمال کنترل تطبیقی

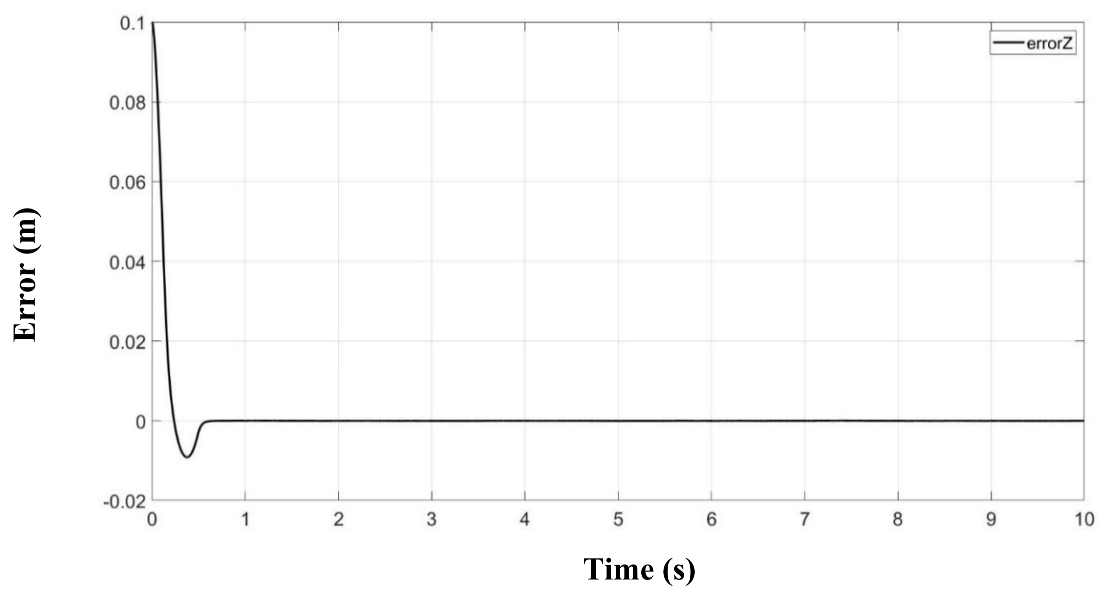


شکل ۴-۵. نمودار ردیابی مجری نهایی در مسیر z با اعمال کنترل مود لغزشی

همانطور که مشاهده می‌شود، کنترل مود لغزشی عملکردی بهتری برای ردیابی مسیر مطلوب توسط ربات دلتا را داشته است. همچینن نمودار خطا نیز این برتری عملکرد کنترل مود لغزشی را نشان می‌دهد. نمودار شکل (۵-۵) خطای ردیابی مسیر z تحت کنترل تطبیقی و نمودار شکل (۶-۵) خطای ردیابی مسیر z تحت کنترل مود لغزشی را نشان می‌دهد.

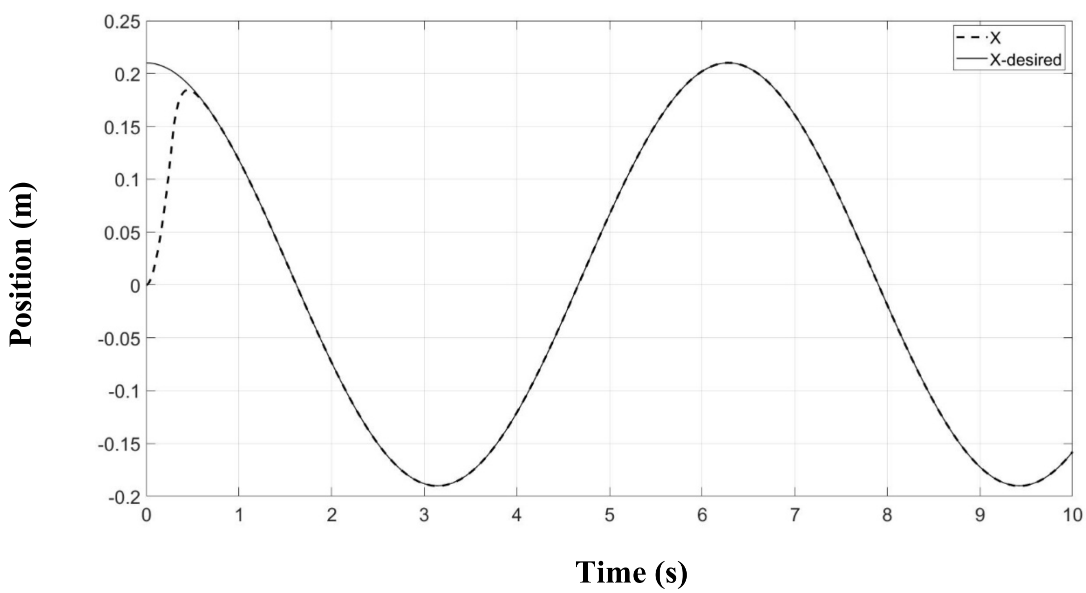


شکل ۵-۵. نمودار خطای ردیابی مسیر مجری نهایی در مختصات z با اعمال کنترل تطبیقی

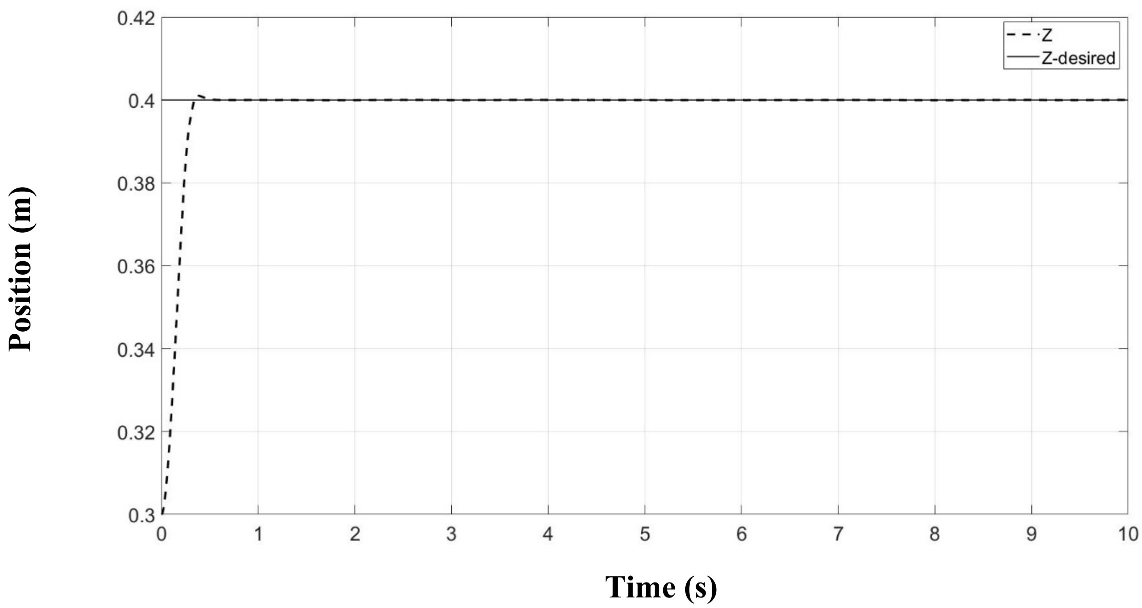


شکل ۶-۵. نمودار خطای ردیابی مسیر مجری نهایی در مختصات z با اعمال کنترل مود لغزشی

همچنین ردیابی مسیر توسط ربات دلتا با کنترل مود لغزشی بهینه نتایج بهتری نسبت به کنترل مود لغزشی داشته است. نمودارهای زیر، ردیابی مجری نهایی در مسیر x و z با اعمال کنترل‌کننده مود لغزشی بهینه را نشان می‌دهد که با مقایسه آن‌ها با عملکرد ردیابی ربات با کنترل مود لغزشی، برتری عملکرد کنترل بهینه مشاهده می‌شود.



شکل ۷-۵. نمودار ردیابی مجری نهایی در مسیر x با اعمال کنترل مود لغزشی بهینه



شکل ۷-۵. نمودار ردیابی مجری نهایی در مسیر z با اعمال کنترل مود لغزشی بهینه

همان‌طور که در فصل چهارم گفته شد، برای بهینه‌سازی، نیاز به اکسترمم کردن یک تابع هزینه است که با توجه به نیاز تعریف می‌شود. تابع هزینه مورد استفاده در این الگوریتم به شکل زیر تعریف شد:

$$y = -\left(e^2 + \frac{u^2}{1000}\right) \quad (۵-۳)$$

که در الگوریتم بهینه‌سازی انبوه ذرات، این تابع ماکزیمم می‌شود. هدف، مینیمم کردن اندازه خطای ردیابی و همچنین مینیمم کردن اندازه گشتاور کنترلی می‌باشد. با اعمال ضرایب کنترلی که برای اعمال کنترل مود لغزشی به صورت صحیح و خطا به دست آمده بود، نتایج اندازه خطا و اندازه گشتاور کنترلی به صورت زیر حاصل شد:

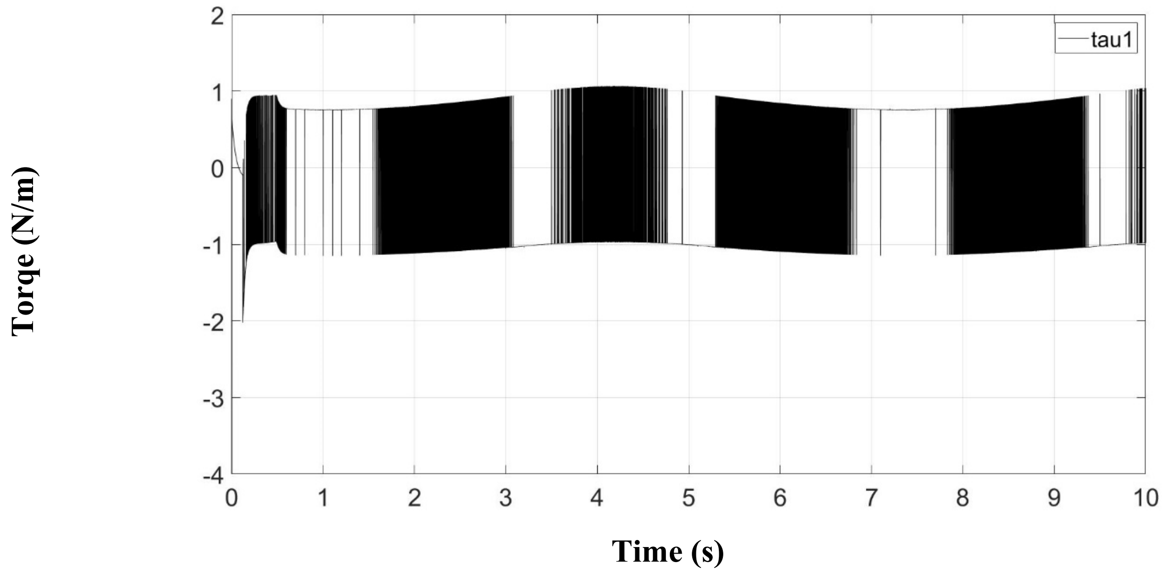
$$e^2 = 462.3988 \tag{۵-۴}$$
$$\frac{u^2}{1000} = 86.7447$$
$$y = -549.1434$$

اما با اعمال ضرایب کنترلی که با روش بهینه‌سازی انبوه ذرات حاصل شد، نتایج اندازه خطا و اندازه گشتاورکنترلی به صورت زیر حاصل شد:

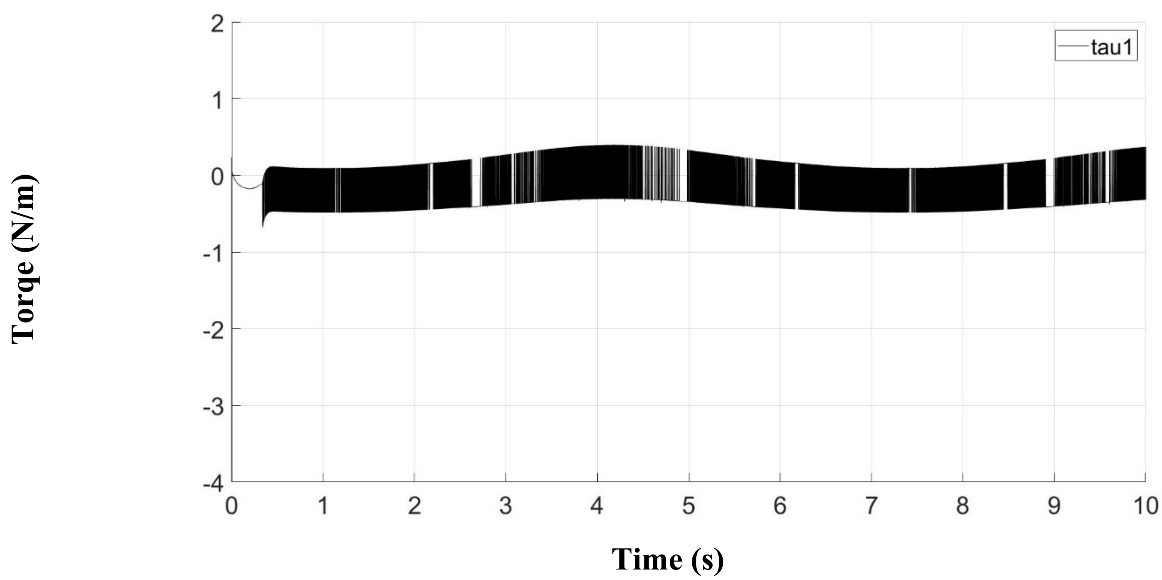
$$e^2 = 58.1669 \tag{۵-۵}$$
$$\frac{u^2}{1000} = 21.7674$$
$$y = -79.9342$$

مشاهده می‌شود که با بهینه‌سازی کنترل‌کننده مود لغزشی، هدف مینیمم کردن اندازه خطای ردیابی و همچنین مینیمم کردن اندازه گشتاور کنترلی برای صرف انرژی کمتر ارضا شده است.

با مشاهده نمودارهای گشتاور کنترلی نیز می‌توان به این نتیجه رسید که گشتاور کنترلی که توسط کنترل‌کننده مود لغزشی بهینه به ربات اعمال می‌شود اندازه بسیار کمتری نسبت به گشتاور اعمالی توسط کنترل‌کننده مود لغزشی دارد. برای نمونه، نمودار گشتاور کنترلی ساق اول به ازای اعمال کنترل‌کننده مود لغزشی و کنترل‌کننده مود لغزشی بهینه در زیر مشاهده می‌شود:

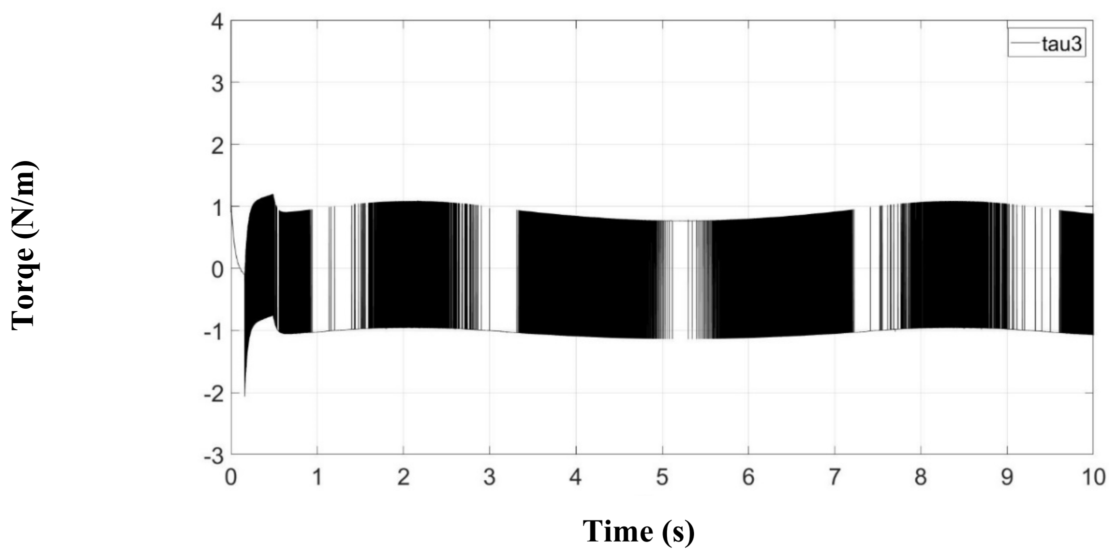


شکل ۸-۵. نمودار گشتاور کنترلی اول با اعمال کنترل مود لغزشی

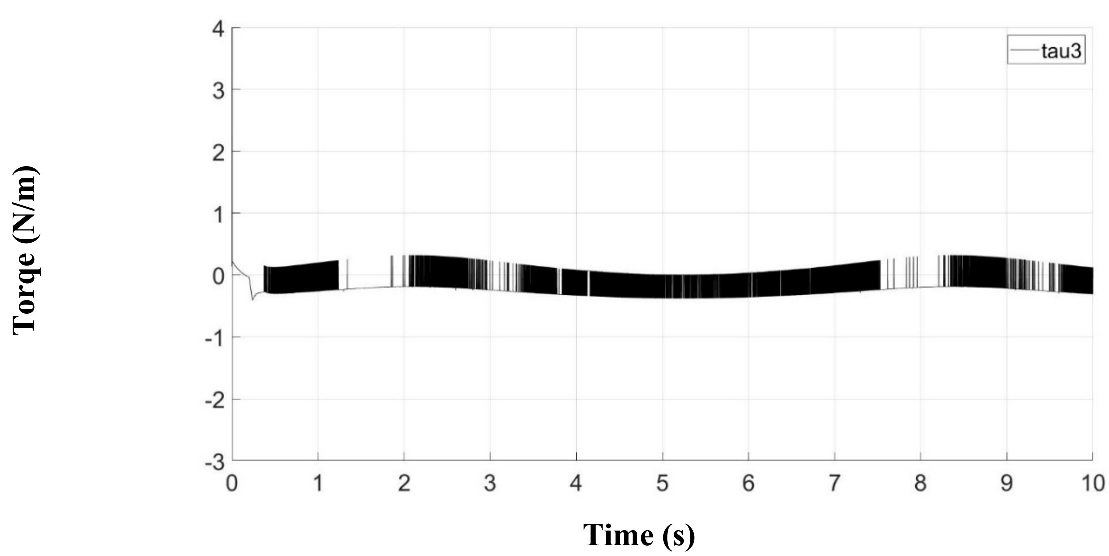


شکل ۹-۵. نمودار گشتاور کنترلی اول با اعمال کنترل مود لغزشی بهینه

همچنین نمودار گشتاور کنترلی ساق سوم به ازای اعمال کنترل‌کننده مود لغزشی و کنترل‌کننده مود لغزشی بهینه نیز در زیر مشاهده می‌شود:



شکل ۱۰-۵. نمودار گشتاور کنترلی سوم با اعمال کنترل مود لغزشی



شکل ۱۱-۵. نمودار گشتاور کنترلی سوم با اعمال کنترل مود لغزشی بهینه

که به طور واضح مشخص است که گشتاور کنترلی با بهینه‌سازی کنترل‌کننده مود لغزشی کاهش یافته است.

۵-۲ پیشنهادات

با توجه به تحقیقات پیشین، برای مطالعه و تحقیق راجع به ربات دلتا و ادامه این پژوهش، موارد زیر پیشنهاد می‌شود.

۱. بهینه‌سازی با الگوریتم‌های بهینه‌سازی دیگر

۲. طراحی کنترل بهینه با استفاده از معادله هامیلتون ژاکوبی بلمن

مراجع

[1] بوستانی، ص. (۱۳۹۵، تیر). *رباتیک از ابتدا تا انتها*. سورس باران.

[7] هاشم‌زاده بشری، س. خراسانی، م. کریم‌زاده عبدالجبار ع. (۱۳۹۵، خرداد). طراحی و ساخت دلتا ربات خطی با کاربری پرینتر سه بعدی. *اولین کنفرانس بین المللی دستاورهای نوین پژوهشی در مکانیک مکاترونیک و بیومکانیک*.

[46] مطیع قادر، ح. لطفی، ش. سید اسفهان، م.م. (۱۳۸۹). مروری بر برخی از روش های بهینه سازی هوشمند. *دانشگاه ازاد اسلامی واحد شبستر*.

[2] Castañeda, L. A., Luviano-Juárez, A., & Chairez, I. (2014). Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Transactions on control systems technology*, *23*(4), 1387-1398.

[3] Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice hall.

[4] Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, *180*(1), 371-386.

[5] Bonev, I. (2001). Delta parallel robot—the story of success. *Newsletter,* no. 4, pp. 1–8.

[6] Pierrot, F., Reynaud, C., & Fournier, A. (1990, March). DELTA: a simple and efficient parallel robot. *Robotica*, *8*(2), 105-109.

[8] Abu-Alkebash, H &, Bader, A. & IyadHashlamon. (2017). Three Degree of FreedomDelta Robot , Implementation for Educational Purposes Design , Control , and Implementation for Educational Purpose. vol. 11, no. April, pp. 126–132.

[9] López, M., Castillo, E., García, G., & Bashir, A. (2006). Delta robot: inverse, direct, and intermediate Jacobians. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *220*(1), 103-109.

[10] Codourey, A. (1998). Dynamic modeling of parallel robots for computed-torque control implementation. *The International Journal of Robotics Research*, *17*(12), 1325-1336.

[11] Hao, J., Xie, X., Bian, G., Feng, Z., Gao, Z., Hou, Z., ... & Vladareanu, L. (2015, August). Dynamic modeling and control simulation of a modified delta manipulator. In *2015 IEEE International Conference on Information and Automation* (pp. 1573-1578). IEEE.

[12] Park, S. B., Kim, H. S., Song, C., & Kim, K. (2013, October). Dynamics modeling of a delta-type parallel robot (2013). In *IEEE ISR 2013* (pp. 1-5). IEEE.

[13] “The Delta Parallel Robot : Kinematics Solutions Robert L . Williams II , Ph . D ., williar4wil@ohio.edu Mechanical Engineering , Ohio University , October 2016 Table of Contents,” no. 4, pp. 1–46, 2016.

[14] Van Khang, N., Hoang, N. Q., Sang, N. D., & Dung, N. D. (2015). A comparison study of some control methods for delta spatial parallel robot. *Journal of Computer Science and Cybernetics*, *31*(1), 71-81.

[15] Uzunovic, T., Golubovic, E., Baran, E. A., & Sabanovic, A. (2013, November). Configuration space control of a parallel Delta robot with a neural network based

inverse kinematics. In *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 497-501). IEEE.
[16] Zhang, J., Lian, C., Gao, R., & Shi, L. (2010, August). 3-Degree-of-freedom parallel robot control based fuzzy theory. In *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics* (Vol. 1, pp. 221-224). IEEE.
[17] Linda, O., & Manic, M. (2011). Uncertainty-robust design of interval type-2 fuzzy logic controller for delta parallel robot. *IEEE transactions on industrial informatics*, *7*(4), 661-670.
[18] Rachedi, M., Bouri, M., & Hemici, B. (2015, July). Robust control of a parallel robot. In *2015 International Conference on Advanced Robotics (ICAR)* (pp. 428-433). IEEE.
[19] Rachedi, M., Hemici, B., & Bouri, M. (2015). Design of an H∞ controller for the Delta robot: experimental results. *Advanced Robotics*, *29*(18), 1165-1181.
[20] Kuo, Y. L., & Huang, P. Y. (2017). Experimental and simulation studies of motion control of a Delta robot using a model-based approach. *International Journal of Advanced Robotic Systems*, *14*(6), 1729881417738738.
[21] Ramírez-Neria, M., Sira-Ramírez, H., Luviano-Juárez, A., & Rodrguez-Ángeles, A. (2015). Active disturbance rejection control applied to a delta parallel robot in trajectory tracking tasks. *Asian Journal of Control*, *17*(2), 636-647.
[22] Burdet, E., Codourey, A., & Rey, L. (1998). Experimental evaluation of nonlinear adaptive controllers. *IEEE control systems magazine*, *18*(2), 39-47.
[23] Lu, X., Zhao, Y., & Liu, M. (2018). Self-learning interval type-2 fuzzy neural network controllers for trajectory control of a Delta parallel robot. *Neurocomputing*, *283*, 107-119.
[24] Dehghani, M., & Shabaninia, F. (2007, August). Chattering elimination with fuzzy sliding mode control in parallel robots. In *First Joint Congress on Fuzzy and Intelligent Systems Ferdowsi University of Mashhad*.
[25] Mangasarian, O. L. (1966). Sufficient conditions for the optimal control of nonlinear systems. *SIAM Journal on control*, *4*(1), 139-152.
[26] Chen, Y. C. (1991). Solving robot trajectory planning problems with uniform cubic B-splines. *Optimal Control Applications and Methods*, *12*(4), 247-262.
[27] Nekoo, S. R. (2013). Nonlinear closed loop optimal control: a modified state-dependent Riccati equation. *ISA transactions*, *52*(2), 285-290.
[28] Geranmehr, B., & Nekoo, S. R. (2015). Nonlinear suboptimal control of fully coupled non-affine six-DOF autonomous underwater vehicle using the state-dependent Riccati equation. *Ocean Engineering*, *96*, 248-257.
[29] Liu, W., Wang, Z., & Guo, M. (2011, August). Two-step nonlinear optimal control approach for coal mine rescue robot navigation. In *2011 Second International Conference on Digital Manufacturing & Automation* (pp. 1380-1383). IEEE.
[30] Korayem, M. H., & Nekoo, S. R. (2015). Finite-time state-dependent Riccati equation for time-varying nonaffine systems: Rigid and flexible joint manipulator control. *ISA transactions*, *54*, 125-144.
[31] Wang, Z., Wang, G., Ji, S., Wan, Y., & Yuan, Q. (2007, December). Optimal design of a linear delta robot for the prescribed cuboid dexterous workspace. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 2183-2188). IEEE.
[32] Fabian, J., Monterrey, C., & Canahuire, R. (2016, August). Trajectory tracking control of a 3 DOF delta robot: a PD and LQR comparison. In *2016 IEEE XXIII International*

Congress on Electronics, Electrical Engineering and Computing (INTERCON) (pp. 1-5). IEEE.

[33] Laribi, M. A., Romdhane, L., & Zeghloul, S. (2007). Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace. *Mechanism and machine theory*, *42*(7), 859-870.

[34] Ergin, M. A., Satici, A. C., & Patoglu, V. (2011, April). Design optimization, impedance control and characterization of a modified delta robot. In *2011 IEEE International Conference on Mechatronics* (pp. 737-742). IEEE.

[35] Kelaiaia, R., Company, O., & Zaatri, A. (2012). Multiobjective optimization of a linear Delta parallel robot. *Mechanism and Machine Theory*, *50*, 159-178.

[36] Lu, X., & Liu, M. (2015, November). A fuzzy logic controller tuned with PSO for delta robot trajectory control. In *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society* (pp. 004345-004351). IEEE.

[37] Lu, X., & Liu, M. (2016). Optimal design and tuning of PID-type interval type-2 fuzzy logic controllers for delta parallel robots. *International Journal of Advanced Robotic Systems*, *13*(3), 96.

[38] Lu, X. G., Liu, M., & Liu, J. X. (2017). Design and optimization of interval type-2 fuzzy logic controller for delta parallel robot trajectory control. *International Journal of Fuzzy Systems*, *19*(1), 190-206.

[39] Chen, W., Fang, H., Yang, Y., & He, W. (2017, July). Optimal Trajectory Planning for Delta Robot Based on Three-Parameter Lamé Curve. In *2017 2nd International Conference on Cybernetics, Robotics and Control (CRC)* (pp. 39-44). IEEE.

[40] Zidan, A., Kaczor, D., Tappe, S., & Ortmaier, T. (2019). Optimization of a P/PI Cascade Motion Controller for a 3-DOF Delta Robot. In *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter* (pp. 217-226). Springer Vieweg, Berlin, Heidelberg.

[41] Pandilov, Z., & Dukovski, V. (2014). copparison of the characteritics between serial and parallel robots. *Acta Technica Corvininesis-Bulletin of Engineering*, *7*(1).

[42] Li, Y., & Xu, Q. (2009). Dynamic modeling and robust control of a 3-PRC translational parallel kinematic machine. *Robotics and Computer-Integrated Manufacturing*, *25*(3), 630-640.

[43] Olsson, A. (2009, February). Modeling and control of a Delta-3 robot. *Department of Automatic Control, Control Lund*. ISSN 0280-5316.

[44] Gkountas, K., Ntekoumes, G., & Tzes, A. (2017, July). Dynamics and control of an Unmanned Aerial Vehicle employing a delta-manipulator. In *2017 25th Mediterranean Conference on Control and Automation (MED)* (pp. 1207-1212). IEEE.

[45] Alashgar, E. H. A. (2013). Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator. *Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator*.

Abstract

The advantages of parallel robots have increased their usage in the industry as well as increasing their attention in scientific studies. In this study, first, the dynamic and kinematic modeling of a parallel robot called the delta robot and its motion control is investigated. Due to the existence of parallel kinematics, the inverse kinematics equation is extracted as closed loop equations. In this study, to obtain the dynamic equations of the delta robot lagrange method is used which is one of the most practical ways to obtain robot dynamics equations. In order to control the position of the robot to track the desired path, two control strategies, adaptive and sliding mode controller are considered to control the robot. Adaptive controller is designed to compensate the uncertainties based on their estimation and sliding mode control due to compensate the parametric uncertainties and load disturbance. The results show that the proposed sliding mode controller has better performance than the adaptive controller. Also, to improve the sliding mode controller performance, a swarm optimization method is used to find optimal values of controlling coefficients. The simulation results show that the error is reduced as well as reduction of the control inputs by the optimal sliding mode controller rather than the sliding mode control without optimization.

Keywords: Delta Robot, Adaptive Control, Sliding Mode Control, Optimization.

Shahrood University of Technology

Faculty of mechanical and mechatronics engineering
M.Sc. Thesis in applied mechatronics engineering

Design of an optimal control strategy for trajectory tracking in work space by delta robot

By: Zahra hassanshahi

Supervisor:
Dr. habib ahmadi
Dr. seyed mojtaba varedi koulaei

Sepember 2019