



دانشکده برق و رباتیک  
گروه کنترل  
پایان نامه کارشناسی ارشد

کنترل بهینه مود لغزشی یک ربات در حضور عدم قطعیت با استفاده از  
الگوریتم پرندگان

سید محمد هاشم زاده

استاد راهنما:

دکتر محمد حداد ظریف

استاد مشاور:

دکتر محمد مهدی فاتح

شهریور ۱۳۹۲





دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه کنترل

کنترل بهینه مود لغزشی یک ربات در حضور عدم قطعیت با استفاده از الگوریتم پرنندگان

دانشجو :

سید محمد هاشم زاده

استاد راهنما :

دکتر محمد حداد ظریف

استاد مشاور:

دکتر محمد مهدی فاتح

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۲



دانشکده برق و رباتیک

گروه کنترل

پایان نامه کارشناسی ارشد آقای سید محمد هاشم زاده

تحت عنوان:

کنترل بهینه مود لغزشی یک ربات در حضور عدم قطعیت با استفاده از الگوریتم پرنندگان

در تاریخ ..... توسط کمیته تخصص زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی :		نام و نام خانوادگی :
	نام و نام خانوادگی :		نام و نام خانوادگی :

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی :		نام و نام خانوادگی :
			نام و نام خانوادگی :
			نام و نام خانوادگی :
			نام و نام خانوادگی :

## تقدیم به

مادرم که مشعل امید را در تاریک‌نمای زندگی برایم به ارمغان آورد

پدرم که نیک اندیشیدن و آزاداندیشی را به من آموخت

برادرانم که همواره پشتیبانم بوده اند

## تشکر و قدردانی:

در ابتدا خداوند یکتا را شاکرم که توان اتمام این پایان‌نامه را به من داد. سپس جا دارد از پرفسور فاتح، استاد مشاور اینجانب کمال تشکر را داشته که بدون شک نه حق استادی که حق پدری را درباره اینجانب به اتمام رساندند. امید است که رهنمودهای اخلاقی ایشان همانند نکات علمی که تجربه و دانش غنی ایشان را نمودار می‌سازد، همواره سرلوحه کار نگارنده باشد. زحمات دکتر حدادظریف نیز که به عنوان استاد راهنما زحمت ویرایش این پژوهش را متحمل شدند باید ستود.

در اتمام این پایان‌نامه افراد بسیاری به نویسنده کمک کرده‌اند. در ابتدا از پدر و مادرم که همواره به من دلگرمی داده‌اند تشکر می‌کنم. از عمه‌جانم که حق بسیار بر گردن من و همه برادرانم دارد نیز بسیار سپاسگزارم. سپس برادرانم سید محسن و سید کاظم هاشم‌زاده که همواره مشوقم بوده‌اند کمال امتنان را دارم. سپس یادی از دوستانم امیر نگهداری و محمد رضا شفیع چافی باید کرد. در پایان نیز از مهندس پویان بشلی و مهندس بهنام سبزی کمال تشکر را دارم.

# تعهد نامه

اینجانب سید محمد هاشم زاده دانشجوی دوره کارشناسی ارشد رشته مهندسی برق-کنترل دانشکده برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه کنترل بهینه مود لغزشی یک ربات در حضور عدم قطعیت با استفاده از الگوریتم پرندگان تحت راهنمایی دکتر محمد حداد ظریف و پرفسور محمد مهدی فاتح متعهد می شوم .

- تحقیقات در این پایان نامه 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 پایان نامه 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 می باشد 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 نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است.

۱۳۹۲/۶/۱۵

سید محمد هاشم زاده

## مالکیت نتایج و حق نشر

- کلیه حقوق مادی و معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه ای، نرم‌افزارها و تجهیزات ساخته شده) 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 پایان نامه بدون ذکر مرجع مجاز نمی باشد.

## چکیده

در این پایان‌نامه مسئله‌ی کنترل مود لغزشی بهینه‌ی بازوی رباتیک تحت عدم قطعیت بررسی شده است. حداقل کردن سیگنال کنترل و خطای ردگیری، از اهداف بسیار مهم در سیستم‌های کنترل است که باید مورد توجه قرار گیرد. نوآوری این تحقیق استفاده از الگوریتم پرنده‌گان در حالت خارج از خط و ترکیب آن با تخمین حدود عدم قطعیت در حالت بر خط جهت بهینه کردن پارامترهای سیستم کنترل است. مزیت عمده‌ی این روش بر دیگر روش‌های معمول در بهینه‌سازی که عمدتاً در حالت خارج از خط سیستم را بهینه می‌کنند، برخط بودن آن است. در حالت برخط با استفاده از تخمین حدود عدم قطعیت می‌توان مدل نامی را به مدل واقعی نزدیک کرد. بنابراین، بهینه‌سازی الگوریتم پرنده‌گان برای سیستم‌های دینامیکی واقعی قابل اجرا خواهد بود. دیگر مزیت روش پیشنهادی، محاسبات کمتر در حالت برخط است. زیرا تمامی محاسبات زمانبر الگوریتم پرنده‌گان در حالت خارج از خط انجام شده و نتایج به صورت جدول در دسترس کنترل‌کننده است. در این تحقیق مطالعه‌ی موردی برای ربات پیوما ۵۶۰ آورده شده است. این ربات، سیستمی چند ورودی-چند خروجی، غیرخطی دارای عدم قطعیت می‌باشد که برای کنترل تحت عدم قطعیت آن باید به روش‌های کنترل مقاوم همچون کنترل مد لغزشی رو آورد. آنچه که بهینه‌سازی را برای این ربات و بسیاری از سیستم‌ها مشکل می‌سازد، وجود عدم قطعیت است که باعث تفاوت مدل نامی با مدل واقعی می‌شود. در این پژوهش با تخمین برخط عدم قطعیت و استفاده از جدول بهره‌ی بهینه در حالت خارج از خط، به اهداف ردگیری و کنترل بهینه‌ی ربات پرداخته شده است. نتایج شبیه‌سازی برای ربات پیوما ۵۶۰، موثر بودن روش پیشنهادی برای کنترل بهینه را تایید می‌کند.

**کلمات کلیدی:** کنترل مد لغزشی، الگوریتم پرنده‌گان، عدم قطعیت، کنترل بهینه، بازوی رباتیک



## فهرست مقالات مستخرج از پایان نامه

## فهرست مطالب

فصل اول: مقدمه.....	۱
۱-۱ مروری بر کارهای گذشته.....	۳
۲-۱ مروری بر ساختار پایان نامه.....	۸
فصل دوم: الگوریتم پرندگان.....	۱۰
۲-۱ ایده های اولیه.....	۱۲
۲-۲ بیان قدم به قدم الگوریتم پرندگان.....	۱۴
۱-۲-۲ قدم اول: تعریف دسته ذرات.....	۱۴
۲-۲-۲ قدم دوم: مقداردهی اولیه ذرات.....	۱۵
۳-۲-۲ قدم سوم: تعیین گراف ارتباط ذرات.....	۱۵
۴-۲-۲ قدم چهارم: مشخص کردن جا به جایی بعدی ذره.....	۱۶
۳-۲-۲ نخستین فرمول بندی مسئله.....	۱۷
۱-۳-۲ یافتن فرمول برای نحوه حرکت ذرات.....	۱۸
۲-۳-۲ دو خطای رایج در معادله حرکت.....	۲۲
۴-۲ درباره محدودیت ها؛ خروج ذرات از فضا و مقداردهی اولیه.....	۲۳
۱-۴-۲ مقداردهی اولیه.....	۲۳
۲-۴-۲ جلوگیری از خروج ذرات از فضای جستجو.....	۲۴
۱-۲-۴-۲ محدودیت شماره صفر (توقف در مرز).....	۲۴
۲-۲-۴-۲ محدودیت شماره یک (برگشت قطعی).....	۲۵
۳-۲-۴-۲ محدودیت شماره دو (بازگشت تصادفی).....	۲۵
۴-۲-۴-۲ محدودیت شماره سه (هاپربولیک).....	۲۵

۲۷	فصل سوم: روش کنترل مد لغزشی ؛ پیشینه ریاضی و معرفی.....
۲۹	۱-۳ سیستم های دینامیک ناپیوسته.....
۳۳	۲-۳ مختصری درباره کنترل مد لغزشی.....
۳۶	۳-۳ کنترل مد لغزشی، روش اسلوتین.....
۳۶	۱-۳-۳ یک مثال ساده برای حالت تک ورودی تک خروجی.....
۴۲	۲-۳-۳ قانون کنترل مد لغزشی اسلوتین برای حالت چند ورودی چند خروجی.....
۴۹	فصل چهارم: دینامیک و سینماتیک پیوما ۵۶۰.....
۵۱	۱-۴ بررسی سینماتیک مستقیم و وارون.....
۵۳	۲-۴ دینامیک ربات پیوما ۵۶۰.....
۵۵	۳-۴ کنترل گشتاور پیوما ۵۶۰.....
۶۳	فصل پنجم: معرفی یک روش جدید برای کنترل بهینه تحت عدم قطعیت.....
۶۴	۱-۵ درباره عدم قطعیت و الگوریتم پرندگان.....
۶۷	۲-۵ بهینه سازی یک سیستم ساده با الگوریتم پرندگان.....
۶۹	۳-۵ یک روش جدید برای کنترل بهینه تحت عدم قطعیت.....
۷۲	۱-۳-۵ بررسی نتایج برای سیستم ساده.....
۷۶	۴-۵ شبیه سازی و نتایج برای پیوما ۵۶۰.....
۷۶	۱-۴-۵ پیاده سازی روش جدید برای ربات.....
۷۷	۲-۴-۵ نتایج شبیه سازی.....
۸۵	فصل ششم: نتیجه گیری.....
۸۶	۱-۶ نتیجه گیری و جمع بندی.....
۸۷	۲-۶ پیشنهادات.....

٨٨.....منابع

٩٢.....پيوست

## فهرست شکل‌ها

- شکل (۱-۱) مروری بر روش‌های بهینه‌سازی پویا..... ۳
- شکل (۱-۲) ارتباط تعداد اطلاع دهنده‌ها با انتشار اطلاعات و گستردگی..... ۱۶
- شکل (۲-۲) تعیین مکان بعدی ذره..... ۱۷
- شکل (۳-۲) متوازی السطوح با حاشیهٔ  $\epsilon$  از هر ضلع..... ۲۴
- شکل (۳-۲) محدودیت شماره صفر (توقف در مرز)..... ۲۵
- شکل (۴-۲) محدودیت شماره یک (برگشت قطعی)..... ۲۵
- شکل (۵-۲) محدودیت شماره دو (بازگشت تصادفی)..... ۲۵
- شکل (۶-۲) محدودیت شماره سه (هایپربولیک)..... ۲۶
- شکل (۱-۳) جهت بردارها در اطراف سطح ناپیوستگی..... ۳۲
- شکل (۱-۴) نمادگذاری دناویت-هارتنبرگ برای پیوما ۵۶۰..... ۵۱
- شکل (۲-۴) انواع پیکربندی های پیوما ۵۶۰..... ۵۳
- شکل (۳-۴) مسیر مطلوب حرکت بازوی ربات..... ۵۶
- شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین (الف) خطای  $\dot{q}$ ..... ۵۷
- شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین (ب) خطای  $\dot{q}$ ..... ۵۸
- شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین (ج) خطای  $q$ ..... ۵۹
- شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین (د) سیگنال کنترل..... ۶۰
- شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین (ه) خطای کنترل موقعیت..... ۶۱
- شکل (۱-۵) نمودار های سیستم ساده غیر بهینه..... ۶۹
- شکل (۲-۵) نمودار های سیستم ساده بهینه شده با الگوریتم پرنده‌گان در حالت خارج از خط..... ۷۳

- شکل (۳-۵) نمودار های سیستم ساده بهینه شده با روش جدید..... ۷۳
- شکل (۳-۵) نمودار های سیستم ساده بهینه شده با روش جدید(ادامه)..... ۷۴
- شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط (الف) سیگنال کنترل..... ۷۸
- شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط (ب) خطای  $\dot{q}$ ..... ۷۹
- شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط (ج) خطای  $\ddot{q}$ ..... ۸۰
- شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط (د) خطای  $q$ ..... ۸۱
- شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط(ه) خطای کنترل موقعیت..... ۸۲

## فهرست جداول

- جدول (۱-۲) خلاصه پارامترهای استفاده شده در الگوریتم پرندگان استاندارد..... ۲۳
- جدول (۱-۳) خلاصه روش های حل معادلات دیفرانسیل ناپیوسته..... ۳۰
- جدول (۱-۴) پارامترهای دناویت-هارتنبرگ برای پیوما ۵۶۰..... ۵۲
- جدول (۲-۴) ثابت های جاذبه (N.m) ..... ۵۴
- جدول (۳-۴) ثابت های اینرسی ( $\text{kg.m}^2$ ) ربات پیوما ۵۶۰..... ۵۵
- جدول (۴-۴) پارامترهای روش اسلوتین..... ۵۶
- جدول (۵-۴) مقایسه شاخص های کارایی برای ربات پیوما ۵۶۰ برای مدل نامی..... ۶۲
- جدول (۱-۵) جدول بهره بهینه شده با الگوریتم پرندگان برای سیستم ساده..... ۷۱
- جدول (۲-۵) شاخص کارایی برای سیستم ساده..... ۷۵
- جدول (۳-۵) مقایسه زمان نشست و مجموع خطای ردگیری برای ربات پیوما ۵۶۰..... ۸۲
- جدول (۴-۵) مقایسه شاخص های کارایی برای ربات پیوما ۵۶۰..... ۸۳

# فصل اول

## مقدمه



رباتیک یکی از میدان های فناوری جدید است که از مرزهای مهندسی سنتی عبور کرده است. این میدان که اکنون به رشته مستقلی تبدیل شده است، نیاز به دانش مهندسی برق، مکانیک، صنایع، علوم کامپیوتر، اقتصاد و ریاضیات دارد. همچنین رشته های جدیدی چون مهندسی ساخت و تولید و مفاهیمی نظیر بینایی ماشین و هوش مصنوعی نیز در آن وارد شده است.

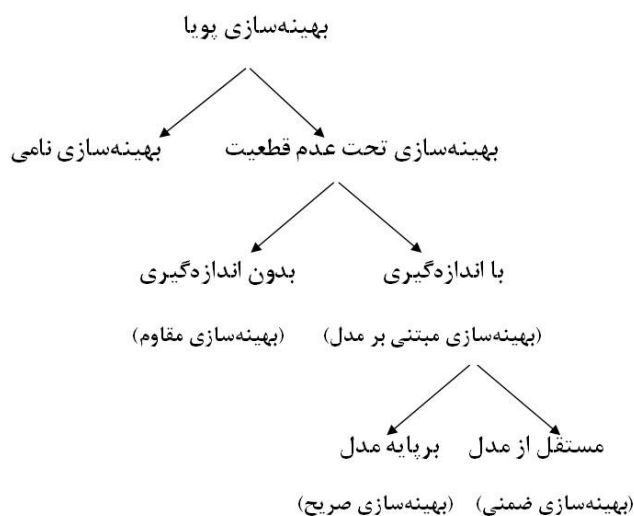
ربات، سیستمی غیرخطی، چند ورودی- چند خروجی با دامنه ی کاری وسیع و دارای عدم قطعیت است. ربات ها امروزه رفته رفته در همه ی بخش های خط تولید جایگزین کارگران شده اند تا کارهای تکراری و مشخص را با دقت بالاتر، هزینه ی کمتر و سرعت بیشتر انجام دهند. پیشرفت چشمگیر برخی صنایع بزرگ در کشورهای پیشرفته ای چون ژاپن و آلمان، بدون شک مرهون وجود ربات ها بوده است. کنترل و نیز کنترل بهینه ی چنین سیستمی در چند دهه ی اخیر یکی از زمینه های فعال تحقیقاتی برای محققین بوده است. از روش های بسیار محبوب کنترل مقاوم می توان به کنترل مد لغزشی اشاره کرد. به کمک این روش که یک روش کنترل ساختار متغیر است با مقادیری ناپیوسته برای سیگنال کنترل، یک سیستم دلخواه که دارای عدم قطعیت ساختاری و پارامتری است، کنترل می شود.

در ربات ها و نیز بسیاری از سیستم های غیرخطی، نمی توان از روش های سنتی بهینه سازی بهره جست. در این سیستم ها معمولاً تعداد متغیرها بسیار زیاد و تاثیرات آن ها روی هم در کل نامشخص است. به این موارد، بازه های مجاز، توابع بدتعریف، شاخص های مختلف کارایی و نیز بحث مقاوم بودن (که تا حدی با بهینگی در تعارض است) نیز اضافه می شود. برای بهینه سازی در چنین شرایطی دیگر نمی توان از روش های سنتی مثل روش نیوتن یا روش گرادیان استفاده کرد. یک دسته از الگوریتم ها که به کار می آیند، روش های تکاملی یا هوشمند هستند. الگوریتم پرندگان از جمله ی این الگوریتم ها است که می تواند در شرایطی که روش های سنتی از حل مسئله عاجزند، بهینه سازی را انجام دهد.

پارامترهای ربات دارای عدم قطعیت هستند و اختلال خارجی نیز مانند دیگر سیستم‌های فیزیکی، در اینجا به طور نامشخصی اضافه خواهد شد. در چنین شرایطی، برای بهینه‌سازی، با مسئله‌ی بهینه‌سازی تحت عدم قطعیت روبه‌رو می‌شویم. این مسئله به صورت بهینه‌سازی در شرایطی که همه‌ی تاثیرات را نمی‌دانیم بیان می‌شود. در ادامه‌ی این فصل ابتدا مروری بر کارهای گذشته خواهد شد و سپس ساختار کلی پایان‌نامه ارائه خواهد شد.

## ۱-۱ مروری بر کارهای گذشته

در [۱] به بهینه‌سازی پویا<sup>۱</sup> برای فرایندهای دسته‌ای<sup>۲</sup> پرداخته شده است. در ابتدا مروری بر روش‌های بهینه‌سازی شده است که در شکل (۱-۱) آمده است.



شکل (۱-۱) مروری بر روش‌های بهینه‌سازی پویا [۱]

<sup>۱</sup> Dynamic Optimization

<sup>۲</sup> Batch Processes

همانطور که در شکل (۱-۱) آمده است، اگر عدم قطعیت در نظر گرفته نشود، روش‌های معمول، چه روش‌های سنتی چه تکاملی، بهینه‌سازی را انجام می‌دهند. در حالت دارای عدم قطعیت، نیاز یا عدم نیاز به اندازه‌های بدست آمده از سنسورها، دو دسته بهینه‌سازی را تعریف می‌کند.

در صورت عدم نیاز به اندازه‌های بدست آمده از سنسورها، بهینه‌سازی را مقاوم می‌نامند. در این روش، عدم قطعیت پارامتری را می‌توان با روش‌های شناسایی پارامتر بدست آورد. برای اختلال نیز به صورت تابع توزیع احتمالی در نظر گرفته می‌شود و بر آن اساس به بهینه‌سازی پرداخته می‌شود.

دسته‌ی دیگر از روش‌های بهینه‌سازی تحت عدم قطعیت، بهینه‌سازی بر اساس اندازه‌های بدست آمده از سنسورهاست. همانطور که در شکل (۱-۱) دیده می‌شود، با توجه به اینکه وابسته به مدل هستیم یا خیر، دو نوع بهینه‌سازی خواهیم داشت: روش‌های بر پایه مدل<sup>۱</sup> و روش‌های مستقل از مدل<sup>۲</sup>. اگر بهینه‌سازی بر پایه مدل باشد آن را بهینه‌سازی صریح<sup>۳</sup> و اگر مستقل از مدل باشد آن را بهینه‌سازی ضمنی<sup>۴</sup> گویند.

در روش بهینه‌سازی بر پایه مدل (صریح)، بر اساس داده‌هایی که در حالت برخط وارد سیستم می‌شود، بار دیگر بهینه‌سازی انجام می‌شود. از این رو این روش را بهینه‌سازی مکرر<sup>۵</sup> نیز می‌نامند، زیرا هر بار با توجه به داده‌های بدست آمده، عمل بهینه‌سازی مرتباً تکرار می‌شود. در یکی از این روش‌ها، که بهینه‌سازی اجرا تا اجرا<sup>۶</sup> نامیده می‌شود، فقط داده‌ها برای حالت خارج از خط موجود است. نکته‌ای که به کمک طراح در این روش می‌آید این است که در بسیاری از فرایندهای صنعتی، فرایند تکراری است و به

<sup>۱</sup> Model-based Methods

<sup>۲</sup> Model-free Methods

<sup>۳</sup> Explicit Optimization

<sup>۴</sup> Implicit Optimization

<sup>۵</sup> Repeated Optimization

<sup>۶</sup> Run-to-run Optimization

این روش می‌توان تا حدی، برخی فرایندهای تکراری را بهینه‌سازی کرد. عیب این روش، در نظر نگرفتن اختلالاتی است که در حالت برخط وارد سیستم می‌شود.

در روش مستقل از مدل (ضمنی)، بر اساس گسسته‌سازی فضای پارامترها [۱]، جواب‌های حالت بهینه بدست آمده و ذخیره می‌شوند. این ذخیره‌سازی می‌تواند به صورت تعلیم یک شبکه عصبی باشد. آنگاه سیستمی که بهترین انطباق را دارد برگزیده شده و بر آن اساس، بهینه‌سازی انجام می‌شود. روشی که در پیشنهاد شده است، در دسته‌ی بهینه‌سازی مستقل از مدل (ضمنی) جای می‌گیرد که ردگیری مرجع<sup>۱</sup> نام دارد. در این روش البته مدل نامی باید در دسترس باشد و سپس به کمک روش‌های عددی می‌توان مقدار متغیرهای حالت و سیگنال کنترل بهینه را یافت. در این روش با توجه به اینکه محدودیت‌ها و حساسیت، چه در حضور و چه در نبود عدم قطعیت یکسان است، شرایطی پیشنهاد شده که چگونه مسیر مرجع سیگنال انتخاب شود. قانون کنترلی که بتواند حساسیت نسبت به مسیر و حالت نهایی را صفر نگه دارد، می‌تواند بهینگی تحت عدم قطعیت را تضمین کند. حسن این روش محاسبات کم و تنظیم پارامتر آسان است. عیب این روش نیز دشوار بودن انتخاب سیگنال کنترل است.

در [۲] روشی برای کار با عدم قطعیت در الگوریتم‌های تکاملی ارائه شده است. در یک طبقه‌بندی دیگر، راهکار برخورد با عدم قطعیت، افزایش نرخ سیگنال به نویز<sup>۲</sup> است که می‌تواند به صورت افزایش مقدار سیگنال یا کاهش مقدار عدم قطعیت باشد. در این تحقیق هر دو راهکار با هم آورده شده است. در بخش اول یک الگوریتم تکاملی ترتیبی و در بخش دوم روش کار با عدم قطعیت آورده شده است. در بخش اول از روش تطبیق ماتریس کوواریانس استفاده شده که در حالت برخط و نیز برای الگوریتم‌های تکاملی

<sup>۱</sup> Reference Tracking

<sup>۲</sup> Signal to noise ratio (SNR)

با تعداد ذرات کم، بهتر جواب می‌دهد. در بخش دوم ابتدا به کمی‌سازی<sup>۱</sup> عدم قطعیت پرداخته شده و سپس روشی جهت کار با عدم قطعیت ارائه شده است. این روش بررسی می‌کند که عدم قطعیت تا چه حد کیفیت یک راه‌حل را تغییر می‌دهد و نرخ جمعیت را بر اساس آن تغییر می‌دهد تا نسبت سیگنال به نویز زیادتر شده و نهایتاً سیستم رفتار بهینه‌تری از خود در حضور عدم قطعیت نشان دهد.

در [۳] به کمک کنترل مد لغزشی ترکیبی<sup>۲</sup>، که ترکیبی از کنترل مد لغزشی با سیستم‌های فازی-عصبی بازگشتی<sup>۳</sup> است، به بهینه‌سازی سیستم کنترل و نیز حذف پدیده‌ی لرزش<sup>۴</sup> در سیستم‌های متصل به هم در ابعاد بزرگ<sup>۵</sup> پرداخته است. الگوریتم تکاملی در این روش، الگوریتم ژنتیک است. برای حذف کامل پدیده‌ی لرزش و نیز جبران موارد غیرخطی کنترل‌کننده‌ی تک ورودی - تک خروجی ممدانی پیشنهاد شده است. عیب این روش نیز خارج از خط بودن و نداشتن تخمین درست از عدم قطعیت در حالت برخط است، چراکه الگوریتم ژنتیک الگوریتمی خارج از خط است.

در [۴] با استفاده از یک روش جدید به کنترل بهینه تحت عدم قطعیت پرداخته است. ابتدا مسئله‌ی کنترل مقاوم تحت عدم قطعیت به یک مسئله‌ی کنترل بهینه تبدیل می‌شود و سپس با توجه به ارتباط این دو روش، [۵]، مسئله‌ی کنترل بهینه‌ی معادل با کنترل مقاوم حل می‌شود. البته در این روش برآوردی از عدم قطعیت در حالت برخط در دسترس نیست. یک پارامتر در کنترل‌کننده نیز باید به نحوی انتخاب شود که روش انتخاب آن به آینده واگذار شده است.

---

<sup>۱</sup> Quantification

<sup>۲</sup> Hybrid

<sup>۳</sup> Recurrent fuzzy neural network (RFNN)

<sup>۴</sup> Chattering Phenomenon

<sup>۵</sup> Large-scale Interconnected systems

در [۹-۶]، که نمونه‌های بسیار بیشتری از آن‌ها نیز در نشریات موجود است، از روش‌های مقاوم مانند کنترل مد لغزشی استفاده شده است و سپس در حالت خارج از خط توسط الگوریتم‌های تکاملی نظیر الگوریتم ژنتیک یا الگوریتم پرندگان، مدل نامی بهینه شده است. به دلیل استفاده از روش‌های کنترل مقاوم، عدم قطعیت‌ها فقط باعث ناپایداری نمی‌شوند و اختلالی در کار سیستم رخ نمی‌دهد ولی اینگونه نیست که سیستم بهینه باشد. البته به دلیل کوچک بودن نسبی عدم قطعیت‌ها، معمولاً این روش بسیار به حالت بهینه نزدیک است ولی عدم قطعیت فقط در حالت برخاسته بدست می‌آید که این مورد عیب این روش است.

در [۱۰] با استفاده از ترکیب روش کنترل مد لغزشی سیمپلکس<sup>۱</sup> و بهینه‌سازی آشوب، روشی برای بهینه‌سازی سیستم‌های غیرخطی دارای عدم قطعیت ارائه شده است. روش کنترل مد لغزشی سیمپلکس تا حدی پدیده‌ی لرزش را بهبود می‌دهد. در کنترل مد لغزشی استاندارد، به تعداد متغیرهای کنترل، متغیر لغزش در نظر گرفته می‌شود. به طور مثال اگر  $m$  ورودی کنترل (برابر تعداد متغیرهای لغزش) در اختیار باشد، فضا به  $2^m$  زیر فضا افراز می‌شود که هنگام عبور از مرز هر کدام از این زیرفضاها به دیگری، عمل سوئیچینگ انجام می‌شود. در روش کنترل مد لغزشی سیمپلکس، تنها  $m+1$  زیرفضا خواهیم داشت که تعداد دفعات سوئیچینگ کاهش و در نتیجه پدیده‌ی لرزش بهبود خواهد داشت. در روش بهینه‌سازی آشوب بر خلاف روش‌های تکاملی و دیگر روش‌های برمبنای جستجوی تصادفی، جستجو به صورت قطعی می‌باشد. نقطه شروع متفاوت در روش آشوب، جواب‌هایی به غایت متفاوت می‌دهد. در این مقاله ابتدا متغیرها آشوبناک شده و سپس طراحی کنترل مد لغزشی ساده انجام می‌شود. در مرحله‌ی آخر، تابع هزینه برای متغیر آشوب بهینه می‌شود. در این روش نیز از اینکه عدم قطعیت‌ها چه هستند صحبتی به میان نیامده است ولی کنترل مقاوم، پایداری را در حضور عدم قطعیت تضمین می‌کند.

<sup>۱</sup> Simplex Sliding Mode control

در [۱۱] مروری بر روش‌های بهینه‌سازی توسط الگوریتم‌های تکاملی در حضور عدم قطعیت آمده است. به چهار صورت می‌توان عدم قطعیت را در نظر گرفت: اول اینکه تابع هزینه دارای نویز باشد. حالت دوم این است که پس از بهینه‌سازی، متغیرهای طراحی و پارامترهای محیطی عوض شوند که در نتیجه بهینه‌سازی باید مقاوم باشد. مورد سوم زمانی است که تابع هزینه تقریبی است یعنی دارای خطای تقریب است. در حالت چهارم و آخر، مقدار بهینه در حال تغییر است به این معنا که الگوریتم تکاملی باید مقدار بهینه را پس از یافتن، در گذر زمان ردگیری کند. در همه‌ی این موارد نیاز است که اندازه‌گیری‌های دیگری انجام شود تا الگوریتم تکاملی به طرز رضایت‌بخش، بهینه‌سازی را انجام دهد. مروری بر راهکارها، موارد زیر را بیان می‌کند. در مورد اول، میانگین‌گیری جهت کاهش نویز معرفی شده است. برای مورد دوم، یک قانون کنترل فیدبک جهت کاهش حساسیت جواب به تغییرات پارامترها آمده است. در مورد سوم، روشهای تخمین تابع و نیز کاربردهای موردی<sup>۱</sup> و استفاده از تابع هزینه تقریبی پیشنهاد شده است. در نهایت برای مورد چهارم که شامل محیط دینامیک با مقدار بهینه‌ی در حال تغییر است، از روش‌های بر مبنای حافظه<sup>۲</sup> که اطلاعات مناسب قبلی را برای محیط بعدی ذخیره و استفاده می‌کنند، استفاده می‌شود.

### ۱-۲ مروری بر ساختار پایان‌نامه

موضوع پایان‌نامه‌ی حاضر کنترل بهینه‌ی مد لغزشی یک ربات تحت عدم قطعیت است. ابزار بهینه‌سازی در این تحقیق، الگوریتم پرندگان است که الگوریتمی خارج از خط است. این الگوریتم برای کار خود، نیاز به تابعی قطعی دارد. از طرفی، عدم قطعیت تنها در حالت برخاط قابل تشخیص است و مقدار آن در حالت خارج خط مشخص نیست. در این پژوهش به کمک روشی جدید، ارتباطی بین نتایج بهینه‌سازی الگوریتم

<sup>۱</sup> Ad-hoc

<sup>۲</sup> Memory- based Methods

پرندگان در حالت خارج خط و تخمین عدم قطعیت در حالت برخط ارائه خواهد شد. نوآوری این تحقیق، ارائه‌ی روشی ساده برای تخمین عدم قطعیت جهت استفاده از جدول داده‌ی بهینه شده توسط الگوریتم پرندگان است. سیستم فیزیکی مورد استفاده در این تحقیق، ربات پیوما ۵۶۰ است که در صنعت و محیط آکادمیک، رباتی پرکاربرد است. پارامترها، مدل دینامیک و روابط سینماتیک مستقیم و وارون این ربات نیز در مقالات با دقت بالایی موجود است.

مفاد این پایان‌نامه از این قرار است: در فصل دوم، الگوریتم پرندگان معرفی خواهد شد. فصل سوم درباره‌ی کنترل مد لغزشی برای حالت تک ورودی- تک خروجی و نیز چند ورودی - چند خروجی است. در فصل چهارم به بررسی دینامیک و سینماتیک ربات پیوما ۵۶۰ پرداخته می‌شود. فصل پنجم به ارائه‌ی روشی جدید جهت کنترل بهینه تحت عدم قطعیت اختصاص یافته است. در انتها در فصل شش، نتیجه‌گیری و پیشنهادات آمده است.



فصل دوم

الگوریتم پرندگان

بهینه‌سازی در کارهای تکراری، قدمتی همپای حیات بشر دارد [۱۲]. حتی بدوی‌ترین انسان‌ها نیز از این قاعده پیروی می‌کردند که «زندگی خود و وضعیت‌شان را بهبود دهند». بنابراین استراتژی‌های بسیاری در طبیعت وجود داشته و دارد که می‌توان از آن‌ها برای بهینه‌سازی در شاخه‌های مختلف استفاده کرد. برخی از مدل‌ها و الگوریتم‌هایی که اخیراً بسیار موفق بوده‌اند با استفاده از چنین ایده‌هایی و نیز استفاده‌ی استعاره‌ای از این مفاهیم، توانسته‌اند در بسیاری مسائل مهندسی را یاری دهند.

در بین مسائل بهینه‌سازی می‌توان تفاوتی بین رفتار فردی و رفتار جمعی قائل شد [۱۲]. منظور از رفتار فردی رفتار یک ذره و رفتار جمعی، رفتار جمعی از ذرات است. رفتار جمعی در کل می‌تواند بهتر باشد زیرا مانند رفتار فردی از افتادن در دام یک مینیمم محلی که شاید مینیمم خوبی نباشد، بهتر می‌تواند فرار کند. البته رفتار جمعی نیز ممکن است اشکالاتی داشته باشد و برای نمونه اگر تنها بهبود آنی در نظر گرفته شود، ممکن است به یک ذره که در حال پیشرفت نیست اجازه حرکت داده نشود، حال آنکه با کمی حرکت در فضای جستجو، آن ذره به مینیمم سرتاسری برسد که در نزدیکی نقطه فعلی بوده است. بنابراین، باید اجازه جستجو به ذرات داده شود حتی در مواردی که اوضاع چندان امیدوارکننده نیست.

الگوریتم پرندگان یکی از الگوریتم‌هایی است که از طبیعت و دسته پرندگان، زنبورها، مورچه‌ها و دیگر مخلوقات که با هم زندگی و غذا پیدا می‌کنند، الهام گرفته شده است [۱۲]. در این مخلوقات بهینه‌سازی یعنی یافتن غذا، کوتاهترین مسیر تا یک مقصد، یا هدف‌هایی مشابه، که برای این کار همه آن‌ها فضایی را جستجو می‌کنند و به یکدیگر خبر می‌دهند تا به سرعت بتوانند به هدف برسند. پس نیاز به تعدادی ذره است که اولاً بتوانند فضا را جستجو کنند و ثانیاً با هم در ارتباط باشند.

در نسخه ریاضی الگوریتم پرندگان سوالات بسیاری مطرح می‌شود که البته برخی از آن‌ها مربوط به زندگی این پرندگان یا حشرات می‌شود [۱۲]. به طور مثال:

- ساختار این ارتباط چگونه باشد؟
  - تعداد ذرات چه باشد؟
  - آیا سلسله مراتب در این جا برقرار است؟
  - حرکت بعدی به چه صورت باشد؟ (ذره به کجای فضای جستجو در قدم بعد سر بزند؟)
  - آیا زیر گروه‌هایی نیز در این دسته‌ها تشکیل می‌شود؟ (مثلاً هر ۳ عضو با هم یک ناحیه را بگردند؟)
  - آیا آنچه که در طبیعت است بهترین راه پیاده سازی الگوریتم پرندگان است یا در عالم ریاضیات می‌توان آنرا بهبود بخشید؟
- در ادامه این فصل ابتدا ایده‌های اولیه شکل‌گیری الگوریتم پرندگان بررسی خواهد شد. سپس اولین فرمول‌بندی این الگوریتم خواهد آمد به صورتی که قابل فهم برای رایانه و شبیه‌سازی باشد. در انتها نیز مختصری درباره‌ی مقداردهی اولیه و محدودیت‌ها جهت ممانعت از خروج ذرات از فضا بحث خواهد شد.

## ۲-۱ ایده های اولیه

در ۱۹۲۷ کارل وُن فریش کشف کرد [۱۲] که زنبورها علاوه بر شهد و نکتار، یک مورد دیگر نیز به کندو می‌آورند: /اطلاعات. البته منظور از این اطلاعات در اینجا دو مورد است: موقعیت و کیفیت یک منبع شهد و نکتار که تازه پیدا شده است.

البته وون فریش نتوانست کشف کند که چگونه زنبورها یک منبع جدید را می‌یابند و به سوی آن می‌روند. در هر حال مدل ریاضی باید بگونه‌ای این موارد را مدل کند [۱۲]. همان‌طور که بیان شد، الگوریتم پرندگان الهام گرفته از طبیعت است و در ادامه توضیح داده خواهد شد که چگونه این الهام به مدلی

ریاضی تبدیل می‌شود. البته در این قسمت صرفاً بحث جنبه غیردقیق و به قصد آشنایی است و در قسمت بعد فرمول‌بندی ریاضی بررسی می‌شود.

در بهینه‌سازی با کامپیوتر و الگوریتم‌های تکرار، به طور کلی نیاز به یافتن بهینه داریم که ما را به سمت یافتن یک تابع رهنمون می‌کند [۱۲]. یعنی یک  $f(x)$  می‌خواهیم که به سه طریق یافته می‌شود:

- توسط فرمول ریاضی

- توسط الگوریتم

- توسط خروجی یک فرایند واقعی یا شبیه‌سازی شده

برای سادگی فرض می‌شود در ابتدا هدف یافتن مینیمم یک تابع یا فرمول ریاضی است. یعنی با یک تابع ریاضی روبرو هستیم و نیز این تابع، چند مینیمم ندارد.

قبل از شروع سه مطلب دیگر نیز بهتر است بیان شود [۱۲].

۱- در مورد اطلاعات باید گفت که آنچه هر فرد با خود به کندو می‌آورد در واقع کیفیت و

موقعیت یک محل یا منبع غذای جدید است. اگر موقعیت را با  $x$  نشان دهیم، کیفیت در آن موقعیت را می‌توان با  $f(x)$  نشان داد. یعنی مقدار (کیفیت) در یک نقطه (در آن موقعیت).

۲- بازگشت ذرات به نقطه اولیه مورد مطلوبی نیست به این معنا که مناسب نیست پرندگان یا

زنبورها یا ذرات به یک محل برگردند تا با یکدیگر تبادل اطلاعات کنند. زیرا به معنای این

است که دوباره به جای اول (در نقطه ای که هیچ اطلاعاتی نداشتیم) برگردند. برای رفع این

مشکل فرض می‌شود ذرات با هم در ارتباطند. برای مثال، ذرات تلفن همراه دارند و نیازی به

حضور در کنار یکدیگر برای تبادل اطلاعات ندارند. پس ذرات در اینجا به وسیله ای مثل

موبایل مجهزند [۱۲].

۳- فرمول‌بندی ارائه شده در ادامه، دقیقاً مشابه آن نیست که در طبیعت رخ می‌دهد. زیرا دقیقاً نمی‌دانیم که در طبیعت چه رخ می‌دهد و اساس این ارتباط و انتقال اطلاعات در پرندگان یا زنبورها به چه صورت است. آنچه می‌دانیم فقط بخشی از دانش ما نسبت به طبیعت است و هنوز ناشناخته‌های بسیاری وجود دارد. اما آنچه که باعث می‌شود به این صورت فرمول‌بندی کنیم این است که: /این روش خوب جواب داده است. وگرنه شاید در طبیعت به گونه ای دیگر موارد مشابه انجام شود. پس آنچه در ادامه می‌آید فقط الهام گرفته از طبیعت است و نه مدل‌سازی دقیقی از آنچه واقعاً در بین این موجودات رخ می‌دهد [۱۲].

حال همانگونه که گفته شد باید روشی ریاضی ارائه شود که بتوان آن را فرمول‌بندی کرد و به کامپیوتر داد تا برای تکرار از آن بهره گرفت. قدم به قدم همان گونه که تاکنون گفته شد باید آنچه را رخ می‌دهد به زبان ریاضی بیان کنیم [۱۲].

### ۲-۲ بیان قدم به قدم الگوریتم پرندگان

در ادامه چهار قدم برای پیاده‌سازی الگوریتم معرفی می‌شود [۱۲].

#### ۲-۲-۱ قدم اول: تعریف دسته‌ی ذرات

در ابتدا یک دسته از ذرات تعریف می‌شود. اندازه این دسته می‌تواند هر تعدادی باشد ولی در عمل بین ۲۰ تا ۴۰ ذره معمولاً کفایت می‌کند. در کامپیوتر این ذرات هر کدام نشان دهنده یک نقطه هستند که باید در ابتدا آن‌ها را در فضای جستجو پخش کرد و در هر تکرار به جستجوی بیشتری با آن‌ها پرداخته شود [۱۲].

## ۲-۲-۲ قدم دوم: مقداردهی اولیه ذرات

این ذرات باید در فضا به صورت تصادفی پخش شوند و فضا را بگردند تا به نقاطی برسند که کیفیتشان مناسب است. به این معنا که به  $x$  هایی دست یابیم که  $f(x)$  های آن‌ها کمتر است [۱۲].

## ۳-۲-۲ قدم سوم: تعیین گراف ارتباط ذرات

همانطور که گفته شد، ذرات موبایل دارند و می‌توانند با هم در ارتباط باشند. پس باید به گونه‌ای هر ذره به ذرات مجاور اطلاع دهد. یعنی هر ذره باید به تعدادی ذره اطلاع دهد و تعدادی ذره نیز به او اطلاع دهند.

دربارهٔ تعداد اطلاع دهنده‌ها به هر ذره و تعداد ذره‌هایی که هر ذره باید به آن‌ها اطلاع دهد می‌توان دو مورد زیر را در نظر داشت [۱۲]:

- اگر همهٔ ذرات به یکدیگر خبر دهند، ممکن است رفتار همه بسیار مشابه شود.
- اگر هر ذره فقط به تعداد بسیار کمی ذره اطلاع دهد، ممکن است اطلاعات مفید به گونه‌ای نامطلوب منتقل شود.

پس دو ضابطه را می‌توان در اینجا تشخیص داد:

- ۱ - گستردگی و تنوع (در فضای جستجو)
- ۲ - انتشار (یک خبر)

بطور مثال در مورد اول می‌توان فرض کرد که در عمق بیشتر جستجو می‌شود و در مورد دوم می‌توان فرض کرد که در طول بیشتر جستجو می‌شود (اگر فضای جستجو مانند یک برکه یا چشمه فرض شود) [۱۲].

حال داریم:

$\downarrow$  گستردگی (جستجو)  $\Rightarrow \uparrow$  انتشار اطلاعات  $\Rightarrow \uparrow$  تعداد اطلاع دهنده ها به یک ذره

شکل (۱-۲) ارتباط تعداد اطلاع دهنده‌ها با انتشار اطلاعات و گستردگی [۱۲]

اگر تعداد اطلاع دهنده‌ها کم شود، انتشار کم می‌شود ولی گستردگی بهتر می‌شود و اگر تعداد اطلاع دهنده‌ها زیاد شود، انتشار بهبود می‌یابد ولی همه‌ی فضای جستجو به خوبی واکاوی نشده است. پس به یک مصالحه برمی‌خوریم [۱۲].

برای فائق آمدن بر این مشکل و حل مساله مصالحه روش‌هایی موجود است که در زیر سه روش بیان می‌شود [۱۲]:

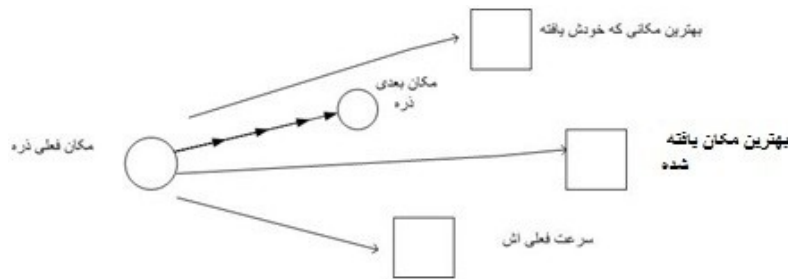
- (۱) انتخاب ۲ یا ۳ گیرنده‌ی اطلاعات برای هر ذره
  - (۲) انتخاب خبردهنده براساس یک قانون (نه برحسب تصادف)، مثلاً براساس یک دیاگرام چرخه-ای براساس انتشار بالا و گستردگی بالا
  - (۳) انتخاب اطلاع دهنده‌ها براساس یک روش هوشمند و خودکار که *I-group* نامیده می‌شوند.
- بحث فعلی تا این نقطه بسیاری از موارد را معلوم کرده است. اما سخت‌ترین موردی که باید مدل شود نقطه‌ی بعدی است که ذره باید برود که همان قدم چهارم است.

## ۴-۲-۲ قدم چهارم: مشخص کردن جا به جایی بعدی ذره

پس از بررسی [۱۲] دیده می‌شود که سه عامل در حرکت هر ذره تاثیر دارد:

- ۱- سرعت فعلی ذره
- ۲- بهترین محلی که قبلاً خود ذره پیدا کرده (که مسلماً به آن سمت می‌رود)
- ۳- بهترین محلی که دسته ذرات پیدا کرده اند (که ذره به آن سمت نیز ممکن است برود)

پس سه عامل مهم است. ولی این عوامل هر کدام تا چه اندازه باید مهم باشند؟  
یعنی اگر بخواهیم وزنی بدهیم، باید چه وزنی به هر کدام از این سه عامل داد؟  
توجه شود که هر کدام از این بردارها وزنی دارند پس با جمع برداری اشتباه نشود (باید وزنها نیز در نظر گرفته شود).



شکل (۲-۲) تعیین مکان بعدی ذره [۱۲]

این سوال ممکن است پیش آید که چرا این روش جواب می‌دهد؟ یکی از دلایل این است که هر ذره می‌تواند به دلخواه خود در فضای جستجو پرسه بزند. یعنی هیچ ذره‌ای آموزش داده نمی‌شود بلکه ذرات آزاد گذاشته می‌شوند تا در فضا آزادانه حرکت کنند.

## ۳-۲ نخستین فرمول بندی مسئله

در این بخش اولین فرمول‌بندی از الگوریتم پرندگان ارائه می‌شود. همانطور که گفته شد، این الگوریتم یک فرایند تکراری است. از طرفی در هر بار تکرار باید برای تعداد ذرات در دسته، تابع هزینه را محاسبه کنیم (زیرا هر ذره در یک  $x_i$  قرار دارد پس  $f(x_i)$ ها باید محاسبه شود که به تعداد ذرات می‌شود) پس دو فاکتور داریم که باید توجه کرد:

۱ - تعداد تکرارهای الگوریتم



۲ - تعداد محاسبات تابع در هر بار تکرار

فاکتور دوم وابسته به تعداد ذرات است. زیرا به تعداد هر ذره باید یک بار  $f(x_i)$  محاسبه شود. ولی سوال این است که کدام یک از این دو عدد مهم تر است؟

مسلماً تعداد محاسبات تابع در هر بار تکرار مهم تر است [۱۲] زیرا پس از هر بار تکرار می توان برآورد کرد که چقدر به جواب اصلی نزدیک شده ایم، نه پس از تعداد مشخصی تکرار (مثلاً ۱۰۰ بار تکرار) زیرا گاهی با چند تکرار به جواب می رسیم، ولی اگر تعداد ذرات زیاد باشد محاسبات بسیار طولانی می شود. پس در اینجا نیز به یک مصالحه نیاز داریم زیرا از یک طرف حجم محاسبات با افزایش تعداد ذرات بالا می رود و این نامطلوب است، ولی از طرفی کاهش بیش از اندازه ذرات نیز باعث می شود که فضا به خوبی جستجو نشود. بنابراین، به مصالحه نیاز داریم که نحوه ی انتخاب آن بسته به مسئله و بازه پارامترها تغییر می کند.

### ۲-۳-۱ یافتن فرمول برای نحوه ی حرکت ذرات

همانگونه که در بخش ۲-۲ توضیح داده شد، برای راه اندازی الگوریتم پرندگان نیاز به ۲ مورد اساسی است:

۱- وضعیت اولیه یا  $x$  ها

۲- وضعیت بعدی یا سرعت ذره ها

توجه شود که سرعت ذره برابر است با:

$$V = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1} \quad (1-2)$$

ولی در هر مرحله اختلاف زمان  $t_2 - t_1$  برابر ۱ است. پس داریم:

$$V = x_2 - x_1 = \Delta x \quad (2-2)$$

بنابراین در واقع سرعت در هر مرحله همان جابجایی را می‌دهد زیرا فرض می‌کنیم عملیات به صورت گسسته رخ می‌دهد.

برای تعیین وضعیت بعدی ذره‌ها، نیاز به تعریف یک معادله حرکت<sup>۱</sup> داریم [۱۲].

۱- **وضعیت اولیه یا  $x$** : در ابتدا می‌توان از یکی از میانگین‌های اقلیدسی استفاده کرد و سپس آن را بهبود بخشید.

۲- **معادله حرکت یا سرعت ذره‌ها (وضعیت بعدی ذره‌ها)**: ابتدا نمادهای مورد نیاز را تعریف می‌کنیم:

(اسکالر)  $d$  → بعد فضای جستجو

(بردار)  $x$  → مکان فعلی ذره

(بردار)  $p$  → بهترین موقعیت یافته‌شده‌ی ذره تا کنون

(بردار)  $g$  → بهترین موقعیت یافته‌شده اطلاع دهنده‌ها تا کنون

$P$  بهترین مکانی است که خود ذره تاکنون پیدا کرده است تا اگر در دفعات بعد از این مقدار کاهش داشتیم حداقل یک محل برای ارائه به عنوان بهترین مکان داشته باشد.  $g$  نیز بهترین مکان توسط اطلاع دهنده‌های این ذره است. به طور مثال اگر چهار ذره دیگر به این ذره اطلاع دهند و یکی از آن‌ها بهترین باشد،  $g$  همان بهترین مقدار است که در تصمیم‌گیری این ذره تاثیر می‌گذارد.

در واقع برای هر ذره چنین برداری داریم: (اگر فضا  $d$  بعدی باشد، یعنی  $x$ ،  $d$  بعدی باشد)

<sup>۱</sup> Equation of Motion

$$X, V, P, G = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} dx_1$$

توجه کنید که در مرحله اول،  $p_0$  همان  $x_0$  است زیرا هنوز ذره فضا را جستجو نکرده است.

حال کافی است همان مسئله‌ای را که در بخش ۲-۱ به صورت غیردقیق گفته شد به زبان ریاضی بیان

شود. گفته شد که ۳ عامل در حرکت بعدی یک ذره سهیم هستند:

۱- سرعت فعلی ذره ( $V$ )

۲- بهترین موقعیتی که تاکنون خود ذره پیدا کرده است ( $p$ )

۳- بهترین موقعیتی که تاکنون اطلاع دهنده‌های ذره پیدا کرده‌اند. ( $g$ )

حال کافی است که ۳ وزن به این عوامل داده و با هم جمع شوند تا بتوان سرعت و جابه‌جایی را یافت.

ضریب‌ها را  $C_1$ ،  $C_2$  و  $C_3$  می‌نامیم [۱۲].

معادله حرکت در تکرار  $i$  ام به صورت زیر در می‌آید:

(معادله را برای هر بُعد به صورت جداگانه با اندیس  $d$  می‌نویسیم)

$$\begin{cases} v_d^{(i+1)} \leftarrow c_1 * v_d^{(i)} + c_2 (p_d^{(i)} - x_d^{(i)}) + c_3 (g_d^{(i)} - x_d^{(i)}) \\ v_d^{(i+1)} \leftarrow x_d^{(i)} + v_d^{(i)} \end{cases} \quad (3-2)$$

$C_1$ : ثابت است و نشانگر اطمینان ذره به مسیر قبلی اش است.

$C_2$  و  $C_3$  به صورت تصادفی از این بازه انتخاب می‌شوند:  $c_2, c_3 = c_{max} * rand(0, 1)$

دوباره می‌نویسیم:

$$\begin{cases} v_d^{(i+1)} \leftarrow c_1 * v_d^{(i)} + Cmax * rand(0,1) (p_d^{(i)} - x_d^{(i)}) + Cmax * rand(0,1)(g_d^{(i)} - x_d^{(i)}) \\ v_d^{(i+1)} \leftarrow x_d^{(i)} + v_d^{(i)} \end{cases} \quad (4-2)$$

سوال این است که  $C_1$  و  $Cmax$  را چگونه انتخاب کنیم؟

۱- انتخاب  $C_1$ : باید کمتر از ۱ باشد و نامنفی باشد. اگر بسیار کوچک باشد همگرایی سریع و بدون زمان کافی را نتیجه می‌دهد و همگرایی‌ها پخته و رسیده نیست. اگر بسیار بزرگ باشد همگرایی را بسیار کند می‌کند. در عمل محدوده مطلوب آن را ۰,۷ تا ۰,۸ می‌گیرند.  $C_1$  را می‌توان بیش از ۱ انتخاب کرد ولی باید تدابیری اندیشید که ذره به صورت واگرا سرعتش زیاد نشود [۱۲].

۲- انتخاب  $Cmax$ : کاملاً تجربی است. محدوده ۱,۷ تا ۱,۵ پیشنهاد می‌شود [۱۲].

زوج‌های پیشنهادی که به صورت تجربی و از راه شبیه‌سازی بدست آمده‌اند، در رابطه‌ی (۲-۵) آمده است:

$$(C_1, Cmax) = (0.7, 1.47) \text{ or } (0.8, 1.62) \quad (5-2)$$

بیان دو مطلب ضروری است [۱۲]:

۱- یکی از مسائلی که در ادامه باید به آن پرداخته شود، بحث محدودیت برای  $V$  است زیرا معمولاً یک بازه برای  $x$  تعریف می‌شود که به صورت  $[xmin, xmax]$  است ولی چنین محدودیتی هنوز اعمال نشده است.

۲- مسئله دیگر، نحوه توزیع است که در اینجا به صورت یکنواخت<sup>۱</sup> است. یعنی عددی یکنواخت بین صفر و یک. در راستای بهبود الگوریتم، می‌توان نسخه‌های دیگری برای این انتخاب ارائه داد.

<sup>۱</sup> Uniform

### ۲-۳-۲ دو خطای رایج در معادله حرکت

۱- خطای اول مرتبط با نحوه توزیع است. یعنی به جای توزیع یکنواخت از توزیع های دیگر مثل

گوسی می توان استفاده کرد تا ذرات به نحو مطلوبتری توزیع شوند [۱۲].

۲- خطای بعدی این است که معمولاً از یک عدد تصادفی برای  $(p-x)$  و  $(g-x)$  استفاده می کنیم که

باید از دو عدد تصادفی استفاده شود.

برای بررسی بیشتر، معادله حرکت را به صورت زیر بازنویسی می کنیم:

$$v_d^{(i+1)} \leftarrow c_1 * v_d^{(i)} + rand(\cdot, cmax) (p_d^{(i)} - x_d^{(i)} + g_d^{(i)} - x_d^{(i)}) \quad (6-2)$$

و یا:

$$v_d^{(i+1)} \leftarrow c_1 * v_d^{(i)} + rand(\cdot, 2 * cmax) \left( \frac{p_d^{(i)} + g_d^{(i)}}{2} - x_d^{(i)} \right) \quad (7-2)$$

معادله (۷-۲) دو ایراد دارد [۱۲]:

۱- معادله (۷-۲) به معنای این است که همواره وسط بازه انتخاب می شود.

۲- توزیع ذرات بستگی به دستگاه مختصات دارد و تغییر در آن تاثیراتی در کاهش سرعت و یا شتاب

دادن به همگرایی دارد، ولی هر دو تغییر در یک سیر غیرقابل پیش بینی اعمال می شوند. البته

این مورد به دلیل اینکه  $f$ ها معمولاً توابع مقارنی هستند از دید پنهان می ماند.

برای رفع این ۲ مشکل می توان روش های زیر را بررسی و استفاده کرد [۱۲]:

۱- مرکز را به بخش  $(g)$  نزدیک تر کرد تا  $(p)$  و در بخش  $(p-g)$  قرار داد.

۲- از دستگاهی مختصات متقارن مثل کروی استفاده شود و مکان  $x$ ،  $p$  و  $g$  را در آن دستگاه برده و معادله حرکت را پیاده کرد.

در انتهای این بخش، پارامترها مجدداً در جدول (۱-۲) آورده شده‌اند:

جدول (۱-۲) خلاصه‌ی پارامترهای استفاده شده در الگوریتم پرندگان استاندارد [۱۲]

پارامتر	عنوان و ماهیت	قانون تجربی برای انتخاب و ملاحظات
$C_1$	اطمینان ذره به حرکت خود / عدد حقیقی	بین (۱ و ۰) / مقدار پیشنهادی ۰,۷
$C_{max}$	اطمینان ذره به اطلاع بقیه / عدد حقیقی	حدود ۱,۵ / مقدار پیشنهادی: ۱,۴۳
$N$	سایز دسته ذرات / عدد طبیعی	بین ۲۰ تا ۴۰ / مقدار پیشنهادی: ۲۰
$K$	سایز گروه اطلاع دهندگان / عدد طبیعی	بین ۳ تا ۵ برای مسائلی که می بینیم عملی درآمدند / مقدار پیشنهادی ۳

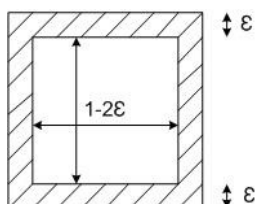
## ۲-۴ درباره محدودیت‌ها؛ خروج ذرات از فضا و مقداردهی اولیه

در این بخش، دو مورد که در تئوری در موردشان صحبت نمی‌شود، آورده خواهد شد. بحث مقداردهی اولیه و بحث اجرای محدودیت‌هایی بر روی ذرات جهت عدم خروج از فضای جستجو.

### ۲-۴-۱ مقداردهی اولیه

مقداردهی اولیه روش‌های بسیاری دارد. در یک روش [۱۲] می‌توان به صورت یکسان ذره‌ها را در بعدهای مختلف پراکنده کرد یا فضا را کاشی‌بندی کرد [۱۳] و هر ذره را در یک بخش قرار داد.

حال فرض شود که یک متوازی السطوح  $D$  بُعدی<sup>۱</sup> به ضلع یک در دسترس است و حاشیه  $\varepsilon=0,01$  از اطراف آن نادیده گرفته می شود. در  $D$  بُعد می خواهیم ببینیم چه اتفاقی می افتد. ابتدا  $D=2$  است.



شکل (۲-۳) متوازی السطوح با حاشیه  $\varepsilon$  از هر ضلع

$$p = 1 - (1 - 2\varepsilon)^D \quad (۲-۸)$$

مساحت ناحیه‌ی هاشور خورده

این نوار نازک که در  $2$  بُعد فقط  $3\%$  مساحت را اشغال می کند، در  $10$  بُعد حدود  $18\%$  و در  $100$  بُعد حدود  $86\%$  حجم را اشغال می کند.

پس روش توزیع یکنواخت، روشی قابل اطمینان برای بعدهای زیاد نیست. روش پیشنهادی و استفاده شده در این پژوهش، [۱۴] است که روش نمونه برداری با توزیع یکنواخت و ظاهری کاملاً تصادفی است که برای نمونه برداری با تکرار نیز کاربرد دارد.

## ۲-۴-۲ جلوگیری از خروج ذرات از فضای جستجو

پس از شبیه سازی دیده می شود که هر ذره ای تمایل دارد از فضا خارج شود که باید تدابیری اندیشیده شود. چند نوع محدودیت در ادامه آورده شده است [۱۵].

### ۱-۲-۴-۲ محدودیت شماره صفر (توقف در مرز)

<sup>۱</sup> D- dimensional

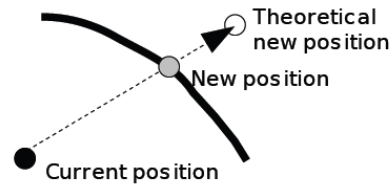
هرگاه ذره به مرز فضا رسید، سرعت را صفر کرده و در مرز متوقف می‌شود.

$$v(t+1) \leftarrow \omega v(t) + rand(0..c)(p(t) - x(t)) + rand(0..c)(p_g(t) - x(t))$$

$$x(t+1) \leftarrow x(t) + v(t+1)$$

$$if x(t+1) > x_{max} \ then \ x(t+1) \leftarrow x_{max}; v(t+1) \leftarrow 0$$

$$if x(t+1) < x_{min} \ then \ x(t+1) \leftarrow x_{min}; v(t+1) \leftarrow 0$$



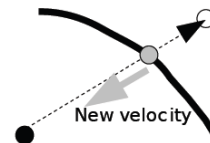
شکل (۳-۲) محدودیت شماره صفر (توقف در مرز) [۱۵]

### ۲-۲-۴-۲ محدودیت شماره یک (برگشت قطعی)

ذره در مرز متوقف شده و سرعت در جهت عکس قرار داده می‌شود تا به درون فضا برگردد.

$$if x(t+1) > x_{max} \ then \ x(t+1) \leftarrow x_{max}; v(t+1) \leftarrow -\gamma v(t+1)$$

$$if x(t+1) < x_{min} \ then \ x(t+1) \leftarrow x_{min}; v(t+1) \leftarrow -\gamma v(t+1)$$



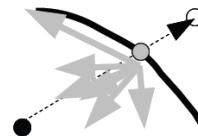
شکل (۴-۲) محدودیت شماره یک (برگشت قطعی) [۱۵]

### ۳-۲-۴-۲ محدودیت شماره دو (بازگشت تصادفی)

ذره به عقب برمی‌گردد ولی نه با ثابت  $\gamma$  بلکه با یک عدد تصادفی در هر بعد که نتیجتاً به صورت شکل (۵-۲) می‌شود:

$$if x(t+1) > x_{max} \ then \ x(t+1) \leftarrow x_{max}; v(t+1) \leftarrow -rand(0..1)v(t+1)$$

$$if x(t+1) < x_{min} \ then \ x(t+1) \leftarrow x_{min}; v(t+1) \leftarrow -rand(0..1)v(t+1)$$

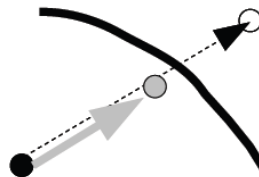


شکل (۵-۲) محدودیت شماره دو (بازگشت تصادفی) [۱۵]

### ۴-۲-۴-۲ محدودیت شماره سه (هایپربولیک)



$$\begin{aligned} \text{if } v(t+1) > 0 \text{ then } v(t+1) &\leftarrow \frac{v(t+1)}{1 + \left| \frac{v(t+1)}{x_{\max} - x(t)} \right|} \\ \text{else } v(t+1) &\leftarrow \frac{v(t+1)}{1 + \left| \frac{v(t+1)}{x(t) - x_{\min}} \right|} \end{aligned}$$



شکل (۶-۲) محدودیت شماره سه (هایپربولیک<sup>۱</sup>) [۱۵]

یعنی اگر  $|x_{\max} - x(t)|$  یا  $|x(t) - x_{\min}|$  که همان جابه‌جایی است و به عبارتی همان سرعت است،  $V_{(t+1)}$ ،  $k$  برابر حد مجاز باشد، در  $\frac{1}{k+1}$  ضرب می‌شود تا هرگز از حدود مجاز خارج نشود.

توجه شود که بدون ترکیب این چند روش، بهترین آن‌ها هایپربولیک است. اگر بتوانیم هیبریدی از چند روش را داشته باشیم، ترکیب هایپربولیک و بازگشت تصادفی بهترین است [۱۵].

<sup>۱</sup> Hyperbolic

فصل سوم

روش کنترل مد لغزشی؛ پیشینه‌ی ریاضی و

معرفی

کنترل سیستم‌های دارای عدم قطعیت همواره از موضوعات جذاب برای مهندسين کنترل بوده است. تقریباً همه سیستم‌ها دارای عدم قطعیت در پارامترهای مدل هستند. برخی از سیستم‌ها شرایط محیطی بسیار متغیری دارند. برای نمونه، سیستم ربات یا سیستم کنترل خودکار یک هواپیما را در نظر بگیرید که دارای پارامترهای غیرقطعی و شرایط کاری و محیطی بسیار متغیر است. تحت چنین شرایطی با روش‌های مرسوم کنترل خطی، کنترل‌کننده را طراحی و استفاده کرد. روش‌های کنترل مقاوم در عین حال که در خارج خط<sup>۱</sup> طراحی می‌شوند، برخط<sup>۲</sup> هستند و می‌توانند سیستم تحت عدم قطعیت را کنترل کنند. از محبوب‌ترین روش‌های کنترل مقاوم می‌توان به کنترل مد لغزشی اشاره کرد. در این روش که روش کنترل ساختار متغیر<sup>۳</sup> است، سیستم کنترل مرتباً بین مقادیر مختلف، عمل سوئیچینگ را انجام می‌دهد و سیستم را به سمت سطحی فرضی با نام سطح لغزش<sup>۴</sup>، که ویژگی‌های مطلوب مانند پایداری و ردگیری را دارد، هدایت می‌کند. در این روش به عمد قانون کنترل به صورت ناپیوسته انتخاب می‌شود تا بتواند به سرعت نسبت به دور شدن سیستم از سطح لغزش عکس العمل نشان دهد و نیز ساختارهای متغیر و متنوعی برای کنترل یک سیستم ارائه شود. در ادامه‌ی این فصل، ابتدا به سیستم‌های دینامیک ناپیوسته اشاره می‌شود که سیستم‌هایی با معادلات ناپیوسته هستند (توجه شود کنترل مد لغزشی نیز به دلیل عمل سوئیچینگ ناگهانی، سیستم را ناپیوسته می‌کند). در قسمت بعد، مختصری درباره‌ی کنترل مد لغزشی ارائه خواهد شد. در پایان این فصل نیز روش کنترل مد لغزشی اسلوتین<sup>۵</sup> بیان خواهد شد.

---

<sup>۱</sup> Offline

<sup>۲</sup> Online

<sup>۳</sup> Variable Structure Control

<sup>۴</sup> Sliding Surface

<sup>۵</sup> J. J. Slotine

### ۳-۱ سیستم های دینامیک ناپیوسته

سیستم های دینامیک ناپیوسته [۱۶]، در بسیاری از کاربردها ظاهر می شوند. در کنترل بهینه بنگ بنگ، سیگنال کنترل بین دو مقدار ناپیوسته سوئیچ می شود. در سیستم های مکانیکی ناصاف<sup>۱</sup>، اجسام صلب در معرض پرش سرعت و نیروی ناپیوسته هستند که در نتیجه اصطکاک و ضربه پدید می آید. در کنترل اجسام توسط ربات، در اثر تماس با اشیاء چه در هوا و چه در زیر آب، نمونه های ناپیوستگی در سیستم در اثر برخورد با محیط وجود دارد.

در اینجا منظور از ناپیوستگی این است که بردار دینامیک سیستم می تواند تابع ناپیوسته ای از حالتها باشد. به طور مشخص، سیستم دینامیک به فرم:

$$\dot{x} = X(t, x(t)) \quad (۱-۳)$$

را که  $d \in N, X: R \times R^d \rightarrow R^d$  را در نظر بگیرید که برای هر  $t \in R$ ، تابع  $x \rightarrow X(t, x)$ ، الزاماً پیوسته نباشد. چنین سیستمی را سیستم دینامیک ناپیوسته<sup>۲</sup> گویند.

دلایل استفاده از سیستم های دینامیک ناپیوسته مختلف است. برای برخی سیستم ها، براساس نتیجه [۱۷-۱۹] درمی یابیم که نمی توان آن ها را توسط فیدبک پیوسته ای وابسته به حالت مثل  $u = -kx$  کنترل و یا پایدارسازی کرد و بنابراین، ناگزیریم که به کنترل فیدبک ناپیوسته یا وابسته به زمان رو بیاوریم. دلیل دیگر براساس نتیجه تئوری میلنر [۲۰] است که بیان می کند که در برخی محیطها که مانع دارند (برای رباتها) باید کنترل ناپیوسته باشد. گاهی نیز در بهینه سازی توابع ناصاف مجبوریم که مجدداً رو به کنترل ناپیوسته بیاوریم. علاوه بر این، به اصطکاک توجه کنید که در همه سیستمها وجود دارد.

<sup>۱</sup> Non-Smooth

<sup>۲</sup> Discontinuous Dynamical Systems

اصطکاک همواره در خلاف جهت حرکت است. برای مدل کردن خلاف جهت حرکت باید از تابع علامت<sup>۱</sup> ضرب در یک علامت منفی استفاده کرد که این خود ناپیوستگی را به دلیل وجود تابع علامت، وارد معادلات دیفرانسیل سیستم می‌کند. بنابراین طیف بسیار وسیعی از سیستم‌ها در دسته سیستم‌های دینامیک ناپیوسته جای می‌گیرند.

در برخورد با این سیستم‌ها، سوال اصلی این است که ایده جواب چه باشد. با مرور مقالات در می‌یابیم که راه حل یکتایی وجود ندارد و روش‌های مختلفی را می‌توان استفاده کرد. نشریات روس در این زمینه بسیار غنی هستند. خلاصه‌ای از ایده‌های مختلف و روش‌های مختلف در جدول (۱-۳) آمده است.

جدول (۱-۳) خلاصه‌ی روش‌های حل معادلات دیفرانسیل ناپیوسته [۱۶]

نام روش	مرجع
Caratheodory	۲۱
Filippov	۲۱
Krasovski	۲۲
Euler	۲۱
Sample & Hold	۲۳
Hermes	۲۵ و ۲۴
Sentis	۲۷ و ۲۶
Ambrosio	۲۸

<sup>۱</sup> Sign Function

اولین و طبیعی‌ترین روش در نظر اول این است که روش کلاسیک و مفهوم آن را تعمیم دهیم و در اینجا نیز با نادیده گرفتن نقاط ناپیوستگی همان روش سیستم پیوسته را اعمال کنیم. روش کارائودوری [۲۰] همین روش است. در این روش به طور تقریبی اجازه داده می‌شود که جواب‌های کلاسیک در برخی نقاط محدود، از جهت بردار میدان تبعیت نکند. این روش در بسیاری از کاربردها استفاده نمی‌شود زیرا کار با ناپیوستگی‌ها، به دلیل تمرکز بر روی مقدار بردار میدان در هر نقطه مشخص توسط این روش، مشکل است.

دو روش فیلیپف و کراسوفسکی [۲۰] در عوض، از ایده شمول دیفرانسیلی یا عضویت دیفرانسیلی استفاده می‌کنند. جهت درک مفهوم شمول دیفرانسیلی، باید مفهوم نگاشت با مقدار مجموعه‌ای<sup>۱</sup> را دریابیم [۱۶]. نگاشت با مقدار مجموعه‌ای در واقع همان نگاشت معمولی است که یک نقطه را به مجموعه‌ای از نقاط در فضای دیگر نگاشت می‌کند. یعنی به یک نقطه، مجموعه‌ای از نقاط تخصیص داده می‌شود. توجه کنید که در این روش برای بیان تساوی، می‌توان یک نقطه را تحت نگاشت به نقطه‌ای دیگر نگاشت کرد که چون این نقطه، عضو این مجموعه تک عضوی است در اینجا به تساوی می‌رسیم. انعطاف‌پذیری و داشتن شرایطی ریاضی برای بحث‌هایی چون یکتایی جواب، وابستگی پیوسته به شرایط اولیه، پایداری و همگرایی مجانبی، این روش را بسیار محبوب ساخته است [۱۶].

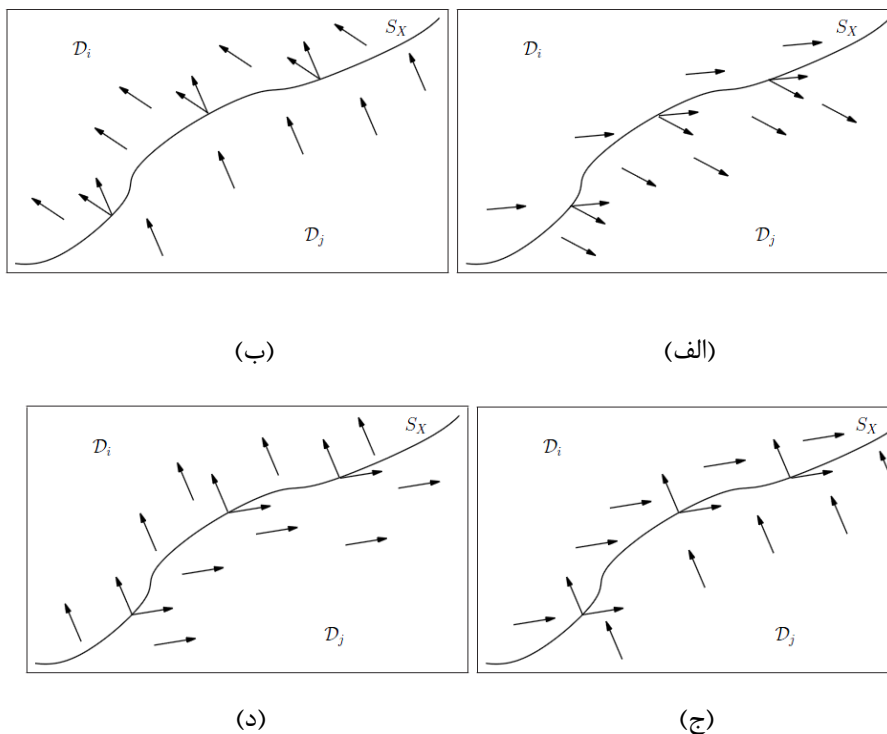
روش‌های دیگری نیز وجود دارد که به طور مثال روش اوپلر [۲۴]، روشی است که بسیار شبیه حالتی است که  $X$  پیوسته است و برای مشخص کردن ویژگی‌های ریاضی پایه‌ای در سیستم‌های دینامیک، مفید است. نکته اساسی در انتخاب روش، مفهوم فیزیکی آن مسئله است که برای برخی روشها معنادارتر از بقیه هستند. در ادامه درباره روش فیلیپف به طور مختصر، کمی بیشتر صحبت خواهد شد.

---

<sup>۱</sup> Set-Valued Map

همانطور که گفتیم در روش کارائودوری [۲۰]، در اطراف ناپیوستگی فرض بر این است که سیستم ناپیوسته نیست. این فرض تولید جواب می کند ولی به دلیل شرایط بسیار متفاوت در اطراف ناحیه ناپیوستگی، چنین فرضی ممکن است کوتاه نظری باشد. چه می شود اگر به جای اینکه بر روی مقدار بردار میدان در هر نقطه تمرکز کنیم، این مطلب را که بردار میدان در اطراف هر نقطه چگونه به نظر می آید مورد توجه قرار دهیم؟

ایده نگاه کردن به همسایگی هر نقطه، هسته‌ی مفهوم روش فیلیپف است. توجه کنید روش کراسوفسکی بسیار به این مفهوم نزدیک است که پرداختن به آن در این تحقیق نمی گنجد. حال اجازه دهید بررسی شود که در یک سطح ناپیوستگی چه رخ می دهد. فرض کنید که سیستم در  $D_i$  ,  $D_j$  به صورت تکه‌ای پیوسته و در سطح  $S_X$  ناپیوسته است. ۴ حالت همانند شکل (۱-۳) رخ می دهد:



شکل (۱-۳) جهت بردارها در اطراف سطح ناپیوستگی (الف) و (ب) عبور. (ج) حرکت لغزشی. (د) سطح شروع دور شونده

در ۲ حالت (الف) و (ب)، در سطح ناپیوستگی، جهت بردارها در دو سوی سطح ناپیوستگی، یکسان است. به این حالت، حالت عبور گفته می‌شود [۱۶]. گویی بردارها از سطح ناپیوستگی عبور کرده‌اند. این دو حالت بسیار کم در سیستم‌ها رخ می‌دهد. در شکل (۳-۱) (د)، بردارها در دو سوی سطح ناپیوستگی در حال دور شدن از سطح ناپیوستگی‌اند. به این حالت نقطه شروع دورشونده می‌گویند [۱۶]. این حالت در ابتدای حرکت سیستم‌ها رخ می‌دهد. در حالت (ج)، دیده می‌شود که بردارها در دو سوی سطح ناپیوستگی، به سوی سطح هستند و در امتداد آن به حرکت خود ادامه می‌دهند. به این حالت، حرکت لغزشی در امتداد سطح می‌گویند [۲۹]. یعنی پس از نزدیکی به این سطح، بردار میدان در امتداد آن می‌لغزند و در نزدیکی آن حرکت می‌کند. تئوری کنترل مد لغزشی بر روی وجود چنین سطحی بنا نهاده شده است. چنین کنترل‌رهایی، سطوح لغزشی را فراهم می‌کنند که خواص مناسب و دلخواه را دارد که در نهایت سیستم حلقه بسته، پایدار با خصوصیات مطلوب خواهد بود [۲۹].

### ۲-۳ مختصری درباره کنترل مد لغزشی

کنترل مد لغزشی، یک کنترل ساختار متغیر و پس‌خور با استفاده از سوئیچینگ سرعت بالاست [۲۹]. این روش، ابزاری مفید و مقاوم را برای کنترل طرح‌های<sup>۱</sup> غیرخطی فراهم می‌آورد. ریشه‌های این روش را می‌توان در کنترل رله‌ای و روش کنترل بنگ بنگ جستجو کرد. پیشرفت‌ها در مدارات سوئیچ و فرکانس بالا و نیز تکنولوژی کامپیوترها، پیاده‌سازی عملی این روش و نیز علاقه مهندسين کنترل به این تکنیک را افزایش داده است.

کنترل مد لغزشی، ماهیتاً یک قانون کنترل سوئیچینگ سرعت بالاست [۲۹]. این قانون بر روی صفحه‌ای که در فضای حالت است و به سطح لغزش معروف است اعمال می‌شود. این روش، مسیر حالت را از نقطه

<sup>۱</sup> Plant



شروع به این سطح و سپس از این سطح به مسیر دلخواه نهایی می‌برد. در اطراف این سطح لغزش، بهره کنترل‌کننده متفاوت است. با طراحی مناسب سطح لغزش، می‌توان اهداف مرسوم سیستم‌های کنترلی، نظیر پایداری و ردگیری را بدست آورد. انتخاب‌های متفاوت سطح لغزش، سیستم‌هایی بغایت متفاوت را نتیجه می‌دهد که می‌توان در صورت نیاز، در زمان‌های مختلف از ساختارهای مختلفی استفاده کرد.

پس از برخورد مسیر حالت تا سطح، همانطور که در بخش ۳-۱ گفته شد، مسیر حالت در امتداد این سطح می‌لغزد. این خاصیت، یعنی ماندن بر روی سطح سوئیچینگ پس از برخورد را، مدل‌لغزش گویند [۲۹]. گوییم مدل‌لغزش وجود دارد هرگاه در یک همسایگی سطح لغزش، بردار سرعت حالت (مشتق بردار حالت)، به سمت سطح لغزش باشد. با انتخاب گین مناسب برای فیدبک حالت و انتخاب مناسب سطح لغزش می‌توان تضمین کرد که همواره مسیر حالت به سمت سطح لغزش حرکت می‌کند. بخش بعدی کار اطمینان از مدل‌لغزش بر روی سطح لغزش است. پس روند کار، دو بخش دارد: اول انتخاب سطح لغزش مناسب جهت حرکت سیستم به سمت آن، و دوم تعیین قانون کنترل به گونه‌ای که بتواند مدل‌لغزش را در سطح لغزش اجرا کند.

در انتخاب سطوح لغزش، باید گفت که بسیار مختلف است [۲۹]. در برخی موارد از مفهوم خطا استفاده می‌کنند ولی در کل باید سطح لغزش  $\sigma(x) = 0$  رفتارهای مطلوب مثل پایداری و ردگیری را برآورده کند. درباره‌ی وجود مد لغزش نیز باید گفت که در این حالت، نیازمند پایداری مسیر حالت به سطح لغزش  $(\sigma(x) = 0)$ ، و یا حداقل یک همسایگی از  $\{x | \sigma(x) = 0\}$  است. یعنی اینکه نقاط بیان‌کننده باید حداقل به صورت نمایی به سمت سطح بروند که بزرگترین محدوده با چنین خاصیتی را "ناحیه جذب" گویند [۲۹]. به لحاظ هندسی، بردار تانژانت یا مشتق زمانی بردار حالت در ناحیه جذب باید به سمت سطح لغزش باشد. مسئله وجود مدل‌لغزش، یک مسئله پایداری تعمیم یافته است که بنابراین از روش دوم لیاپانوف در اثبات آن بهره می‌جویند [۲۹].

یکی از پدیده‌هایی که در کنترل مد لغزش با آن روبرو خواهیم شد، پدیده لرزش سیگنال کنترل است [۲۹]. همانطور که گفته شد، برای پیاده سازی این روش نیاز به سوئیچینگ فرکانس بالا است که در عمل چنین سرعت بالایی به دلیل مواردی چون تاخیر رله ها، هسیتریزیس و ... وجود ندارد. از طرفی عمل سوئیچینگ می تواند دینامیک های مدل نشده مرتبه بالای سیستم را تحریک کند و باعث ناپایداری شود که برای سیستم‌ها بسیار خطرناک است.

مرور مقالات در راستای حذف یا بهبود پدیده‌ی لرزش، روش‌های بسیاری را پیشنهاد می‌کند. در برخی از این روشها [۳۰] به کاهش این پدیده با بهره سوئیچینگ متغیر بازمان پرداخته شده است که نیازی به طراحی دینامیک جدیدی ندارد. روش دیگر [۳۱]، استفاده از کنترل مد لغزشی چند فازی یا چند مرحله‌ای است. در [۳۲] با استفاده از یک مشاهده‌گر به حذف این پدیده پرداخته شده است که این روش تا حد بسیاری شبیه مفهوم کنترل معادل [۲۹] است. روش دیگر، استفاده از کنترل مد لغزش مراتب بالاتر است.

در عین حفظ همه‌ی ویژگی‌های مثبت کنترلر مد لغزشی سنتی، کنترل مد لغزشی مراتب بالا [۳۳-۳۵]، ایده‌ی جدیدی است که می تواند این پدیده‌ی نامطلوب را کاملاً حذف کند. در این روش به جای استفاده از مشتق اول سطح لغزش، از مشتق مرتبه  $i$  آن استفاده می شود که  $i$ ، درجه نسبی سیستم است. این روش همچنین دقت بالاتری را با توجه به زمان نمونه‌برداری گسسته در مقایسه با روش سنتی مد لغزشی، ارائه می‌دهد. برای سیستم‌های تک ورودی-تک خروجی مشکلی وجود ندارد ولی پیاده سازی این روش برای مراتب دلخواه سیستم‌های چند ورودی-چند خروجی، فقط در برخی مقالات آمده است. در [۳۶-۳۷] از کنترلر مد لغزشی مرتبه دو سخن به میان آمده است، ولی طراحی قاعده‌مند کنترلر مرتبه بالا با درجه نسبی دلخواه برای یک سیستم چند ورودی-چند خروجی، در [۳۸] آمده است.

در ادامه، روش اسلوتین [۳۹] برای یک سیستم تک ورودی - تک خروجی و سپس برای یک سیستم چند ورودی - چند خروجی ارائه خواهد شد.

### ۳-۳ کنترل مد لغزشی، روش اسلوتین

روش اسلوتین با جزئیات کامل در [۳۹] آمده است. در ادامه به بیان یک مثال ساده برای حالت تک ورودی تک خروجی و نیز بیان یک مثال برای حالت چندورودی-چندخروجی می پردازیم.

#### ۳-۳-۱ یک مثال ساده برای حالت تک ورودی تک خروجی

سیستم ساده‌ی زیر را در نظر بگیرید [۳۹]:

$$\ddot{x} + \dot{x} - c_1 x^2 = c_2 u + d(t) \quad (2-3)$$

$$c_2 \in [2,3], \quad c_1 \in [1,2], \quad |d(t)| \leq 0.5$$

در این سیستم  $c_2$  و  $c_1$  پارامترهای دارای عدم قطعیت هستند.  $d(t)$  اختلال<sup>۱</sup> است. باید سیستم را کنترل کرد و تضمین پایداری داشت. سیستم دارای عدم قطعیت است بنابراین، می توان به روش مد لغزشی رجوع کرد.

داریم:

$$\dot{x} = f + bu \quad (3-3)$$

که:

$$f = (c_1 x^2 - \dot{x} + d(t)) , \quad b = c_2 \quad (4-3)$$

<sup>۱</sup> Disturbance

فرض می‌کنیم که:

$$\begin{cases} |f - \hat{f}| \leq F \\ b_{max} \geq b \geq b_{min} \end{cases} \quad (5-3)$$

و انتخاب می‌کنیم:

$$\begin{cases} \hat{b} = \sqrt{b_{min} * b_{max}} \\ \hat{c}_r = \frac{c_{r,max} + c_{r,min}}{2} \end{cases} \quad (6-3)$$

که در اینجا برابر است با:

$$\begin{cases} \hat{b} = \sqrt{2 * 3} = \sqrt{6} \\ \hat{c}_r = \frac{1 + 2}{2} = 1.5 \end{cases} \quad (7-3)$$

و برای  $\hat{f}$ :

$$\Rightarrow \hat{f} = 1.5x^r - \dot{x} \quad (8-3)$$

با توجه به (7-3)

و برای  $F$ :

$$F \geq |f - \hat{f}| = |(c_r x^r - \dot{x} + d(t)) - 1.5x^r - \dot{x}| = |(c_r - 1.5) * x^r| + |d(t)|$$

$$F \geq |c_r - 1.5| |x^r| + 0.5$$

$$Max(F) = |c_{r,max} - 1.5| |x^r| + 0.5 \Rightarrow$$

$$F = 0.5(|x^r| + 1) \quad (9-3)$$

بازنویسی می کنیم:

$$\begin{cases} \hat{f} = \hat{c}_r x^r - \dot{x} \\ \hat{b} = \sqrt{b_{min} * b_{max}} = \sqrt{c_{r,min} * c_{r,max}} \\ \hat{c}_r = \frac{c_{r,max} + c_{r,min}}{2} \\ F = |c_{r,max} - \hat{c}_r| * |x^r| + |d(t)| \end{cases} \quad (10-3)$$

که مقادیر عددی به این صورت است:

$$\begin{cases} \hat{f} = \hat{c}_r x^r - \dot{x} = 1.5x^r - \dot{x} \\ \hat{b} = \sqrt{b_{min} * b_{max}} = \sqrt{c_{r,min} * c_{r,max}} = \sqrt{2 * 3} = \sqrt{6} \\ \hat{c}_r = \frac{c_{r,max} + c_{r,min}}{2} = \frac{1 + 2}{2} = 1.5 \\ F = |c_{r,max} - \hat{c}_r| * |x^r| + |d(t)| = 0.5(|x^r| + 1) \end{cases} \quad (11-3)$$

حال به سراغ سیستم می رویم و آن را کنترل می کنیم.

ابتدا سطح لغزش را تعریف می کنیم:

$$s = \dot{\tilde{x}} + \lambda \tilde{x} \quad , \tilde{x} = e = x - x_{desired} \quad (12-3)$$

با جای گذاری از معادله حالت داریم:

$$\begin{aligned} \dot{s} &= \ddot{\tilde{x}} + \lambda \dot{\tilde{x}} = (\ddot{x} - \ddot{x}_d + \lambda(\dot{x} - \dot{x}_d)) \Rightarrow \\ \dot{s} &= \ddot{x} + (\lambda(\dot{x} - \dot{x}_d) - \ddot{x}_d) \end{aligned} \quad (13-3)$$

که  $\lambda$  پارامتر طراحی است. حال با استفاده از رابطه (3-3) داریم:

$$\dot{s} = f + bu + \lambda \dot{\tilde{x}} - \ddot{x}_d \quad (14-3)$$

بر روی سطح لغزش مقادیر صفر است. پس حدود  $u$  را می‌توان بدست آورد ولی باید توجه کنیم که مقادیر  $u$ ,  $b$ ,  $f$  را نداریم. ولی تخمین آن‌ها را داریم:

$\dot{s} = 0 \Rightarrow$  (with estimated values:)

$$\hat{f} + \hat{b}\hat{u} + \lambda\dot{\hat{x}} - \dot{x}_d = 0 \Rightarrow$$

$$\hat{u} = \frac{1}{\hat{b}} * (\dot{x}_d - \lambda\dot{\hat{x}} - \hat{f}) \quad (15-3)$$

توجه کنید که برای بخش دارای عدم قطعیت، باید تدبیری اندیشید. می‌توان برای این کار، در  $u$ ، بخش دیگری علاوه بر  $\hat{u}$  قرار داد که ما آن را با  $u_{unc}$  نشان می‌دهیم:

$$u = \hat{u} + u_{unc} \quad (16-3)$$

حال به سراغ شرط لغزش می‌رویم:

$$\frac{1}{2} \frac{d}{dt}(s^2) \leq -\eta * sgn(s) \Rightarrow$$

$$\dot{s} \leq -\eta * sgn(s) \quad (17-3)$$

که  $\eta > 0$  پارامتر طراحی و در دست طراح است.

توجه شود که با انتخاب  $V = 0.5s^2$  و با مشتق‌گیری از  $V$  و برقرار بودن (17-3) به سادگی پایداری لیاپانوفی تضمین می‌شود [39]. زیرا ضرب هر عدد در تابع علامتش همواره مثبت و بنابراین ضرب مثبتی از قرینه آن همواره کوچکتر مساوی صفر است ( $-\eta$ ). بنابراین، کافی است (17-3) برقرار باشد که در ادامه به آن خواهیم پرداخت.

توجه کنید که برای  $\dot{s}$  باید مقدار اصلی و دارای عدم قطعیت را بگذاریم:

$$f + b(\hat{u} + u_{unc}) + \lambda\dot{\hat{x}} - \ddot{x}_d \leq -\eta * \text{sgn}(s) \Rightarrow$$

$$\left(f - \frac{b}{\hat{b}}\hat{f}\right) + \left(1 - \frac{b}{\hat{b}}\right) * (\lambda\dot{\hat{x}} - \ddot{x}_d) + b * u_{unc} \leq -\eta * \text{sgn}(s) \quad (18-3)$$

به دلیل وجود  $\text{Sgn}(s)$  در سمت راست نامساوی می‌توان در ساختار  $U_{unc}$  نیز یک  $\text{Sgn}$  قرار داد. به دلیل اینکه می‌خواهیم نامساوی برقرار شود، ضریبی مثل  $k'$  به آن بدهیم و سپس ادامه دهیم تا نامساوی برقرار شود. البته حالت‌های دیگر نیز می‌توان قائل شد ولی این ساختار ساده‌تر از همه است.

$$u_{unc} = -k' \text{sgn}(s) \quad (19-3)$$

در مورد رابطه‌ی (18-3) داریم:

$$f - \frac{b}{\hat{b}}\hat{f} = (f - \hat{f}) + \left(1 - \frac{b}{\hat{b}}\right)\hat{f} \quad (20-3)$$

پس داریم:

$$(f - \hat{f}) + \left(1 - \frac{b}{\hat{b}}\right)\hat{f} + \left(1 - \frac{b}{\hat{b}}\right) * (\lambda\dot{\hat{x}} - \ddot{x}_d) + b * (-k' \text{sgn}(s)) \leq -\eta * \text{sgn}(s) \Rightarrow$$

$$(f - \hat{f}) + \left(1 - \frac{b}{\hat{b}}\right) * (\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}) \leq (bk' - \eta) * \text{sgn}(s) \quad (21-3)$$

برای سمت چپ نامساوی مقدار ماکزیمش را می‌گذاریم:

$$(f - \hat{f}) \leq F, \quad \lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f} \leq |\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}| \quad (22-3)$$

و نیز  $b > 0$ ، برای  $\left(1 - \frac{b}{\hat{b}}\right)$  نیز داریم:

$$\left(1 - \frac{b}{\hat{b}}\right) \leq \text{Max} \left\{ \left|1 - \frac{b_{min}}{\hat{b}}\right|, \left|1 - \frac{b_{max}}{\hat{b}}\right| \right\} = \text{Max}_{\frac{1}{\hat{b}}} \quad (23-3)$$

سپس داریم:

$$(bk' - \eta) * \text{sgn}(s) \geq F + \text{Max}_{\frac{1}{\hat{b}}} * |\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}| \quad (24-3)$$

با انتخاب:

$$bk' = \eta + F + \text{Max}_{\frac{1}{\hat{b}}} * |\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}| \quad (25-3)$$

می توان به مقصود رسید فقط اینکه b باید مقدار مینیمم باشد پس داریم:

$$k' = \frac{1}{b_{min}} (\eta + F + \text{Max}_{\frac{1}{\hat{b}}} * |\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}|) \quad (26-3)$$

و برای یکسان سازی با فرمول u می توانیم تعریف کنیم:

$$k = \hat{b}k' \Rightarrow k' = \frac{1}{\hat{b}}k \Rightarrow$$

$$u_{unc} = -\frac{1}{\hat{b}}k * \text{sgn}(s) \quad (27-3)$$

و سپس داریم:

$$u = \frac{1}{\hat{b}} * (\ddot{x}_d - \lambda\dot{\hat{x}} - \hat{f} - k * \text{sgn}(s)) \quad (28-3)$$

که در این رابطه داریم:

$$k = \frac{\hat{b}}{b_{min}} (\eta + F + \text{Max}_{\frac{1}{\hat{b}}} * |\lambda\dot{\hat{x}} - \ddot{x}_d - \hat{f}|) \quad (29-3)$$



و بدین ترتیب سیگنال کنترل از روابط (۳-۲۸) و (۳-۲۹) بدست می‌آید.

### ۳-۲-۳ قانون کنترل مد لغزشی اسلوتین برای حالت چندورودی چندخروجی

در اینجا نیز برای بیان روش به بیان یک مثال می‌پردازیم. برای سادگی، مثال را برای معادلات حالتی که

شبه معادلات حالت ربات پیوما ۵۶۰ است، بررسی و کنترل می‌کنیم [۳۹].

معادلات حالت را ببینید:

$$\begin{cases} \dot{x}_1 = f_1 + b_{11}u_1 + b_{12}u_2 + b_{13}u_3 + d_1(t) \\ \dot{x}_2 = f_2 + b_{21}u_1 + b_{22}u_2 + b_{23}u_3 + d_2(t) \\ \dot{x}_3 = f_3 + b_{31}u_1 + b_{32}u_2 + b_{33}u_3 + d_3(t) \end{cases} \quad (۳-۳۰)$$

و یا

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} f_1 + d_1(t) \\ f_2 + d_2(t) \\ f_3 + d_3(t) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} * \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (۳-۳۱)$$

سیستم فوق، سیستمی مربعی است. یعنی تعداد ورودی و خروجی اش برابر است. طبق [۳۹] دو فرض

داریم:

۱- فرض اول اینکه عدم قطعیت در محدوده‌ی فضای ماتریس B است که به سادگی

یعنی B معکوس پذیر است.

۲- فرض دوم اینکه  $\hat{B}$  نیز معکوس پذیر است و در نبود عدم قطعیت داریم:  $B = \hat{B}$

(توجه کنید که برای ربات چنین شرایطی برقرار است).

$$B = (I + \Delta)\hat{B} \quad (۳-۳۲)$$

که  $\hat{B}$  مقادیر پارامترها، بدون عدم قطعیت است و  $\Delta_{ij}$  نیز همان بخش دارای عدم قطعیت است که برای آن داریم:

$$D_{ij} \geq |\Delta_{ij}| \quad (33-3)$$

و نیز:

$$F_i \geq |f_i - \hat{f}_i| \quad (34-3)$$

متغیر لغزش را به صورت  $s_i$  در نظر می‌گیریم و داریم:

$$s_i = \left( \frac{d}{dt} + \lambda_i \right) \tilde{x}_i = \dot{\tilde{x}}_i + \lambda_i \tilde{x}_i, \quad i = 1, 2, 3 \quad (35-3)$$

که داریم:

$$\tilde{x}_i = e_i = x_i - x_{d,i} \quad (36-3)$$

و شرط لغزش که به صورت

$$\frac{1}{\gamma} \frac{d}{dt} (s_i^2) \leq -\eta * \text{sgn}(s_i), \quad i = 1, 2, 3 \Rightarrow \quad (37-3)$$

$$\text{شرط لغزش: } \dot{s}_i \leq -\eta_i * \text{sgn}(s_i), \quad i = 1, 2, 3$$

است. باتوجه به (30-3) و (35-3) داریم:

$$\dot{s}_i = \dot{\tilde{x}}_i - \dot{\tilde{x}}_{i,d} + \lambda_i \dot{\tilde{x}}_i, \quad i = 1, 2, 3$$

$$\dot{s}_i = f_i + \sum_{j=1}^3 b_{ij} u_j + d_i(t) - \dot{\tilde{x}}_{i,d} + \lambda_i \dot{\tilde{x}}_i, \quad i = 1, 2, 3 \quad (38-3)$$

حال برای یافتن  $U_{eq}$  باید (۳-۳۸) را برابر صفر قرار داد ولی همه‌ی این موارد را به دلیل داشتن عدم قطعیت نمی‌دانیم. بنابراین باید مقدار تخمین زده شده‌ی آن‌ها را بگذاریم و یک بخش نیز در  $u$  قرار دهیم تا با عدم قطعیت‌ها برخورد کند:

$$u = \hat{u} + u_{unc} \quad (۳۹-۳)$$

از اینجا به بعد فرض می‌کنیم که اختلال در  $f$  وارد شده و یک نماد برای مجموعشان بکار می‌بریم.

(به صورت برداری می‌نویسیم):

$$\begin{aligned} \cdot &= \hat{f} + \hat{B}(\hat{u} + u_{unc}) - \ddot{X}_d + \lambda \dot{\tilde{X}} \Rightarrow \\ \hat{u} &= \hat{B}^{-1} (\ddot{X}_d - \lambda \dot{\tilde{X}} - \hat{f}) - u_{unc} \end{aligned} \quad (۴۰-۳)$$

برای راحتی فرض می‌کنیم  $u_{unc} = \hat{B}^{-1} u'_{unc}$  ، پس داریم:

$$\hat{u} = \hat{B}^{-1} (\ddot{X}_d - \lambda \dot{\tilde{X}} - \hat{f} - u'_{unc}) \quad (۴۱-۳)$$

و سپس برای  $\dot{S}$  داریم:

$$\begin{aligned} \dot{s} &= f + Bu - \ddot{X}_d + \lambda \dot{\tilde{X}} \Rightarrow \\ \dot{s} &= f + (I + \Delta)\hat{B}\hat{B}^{-1} (\ddot{X}_d - \lambda \dot{\tilde{X}} - \hat{f} - u'_{unc}) - \ddot{X}_d + \lambda \dot{\tilde{X}} \Rightarrow \\ \dot{s} &= (f - \hat{f}) - (I + \Delta)u'_{unc} + \Delta(\ddot{X}_d - \lambda \dot{\tilde{X}} - \hat{f}) \end{aligned} \quad (۴۲-۳)$$

حال برای  $\dot{s}_i$  می‌نویسیم:

$$\dot{s}_i = (f_i - \hat{f}_i) + \left\{ \sum_{j=1}^r \Delta_{ij} (\ddot{x}_{i,d} - \lambda_i \dot{\tilde{x}}_i - \hat{f}_j) \right\} - \left\{ \sum_{i \neq j} \Delta_{ij} u'_{j,unc} \right\} - (1 + \Delta_{ii})u'_{i,unc} \quad (۴۳-۳)$$

حال می‌دانیم که باید شرط لغزش برآورده شود:

$$\dot{s}_i \leq -\eta_i \operatorname{sgn}(s_i) \quad (44-3)$$

پس داریم:

$$\left\{ \sum_{i \neq j} \Delta_{ij} u'_{j,unc} \right\} + (1 + \Delta_{ii}) u'_{i,unc} \geq \eta_i \operatorname{sgn}(s_i) + (f_i - \hat{f}_i) + \left\{ \sum_{j=1}^r \Delta_{ij} (\ddot{x}_{i,d} - \lambda_i \dot{x}_i - \hat{f}_j) \right\} \quad (45-3)$$

برای اینکه  $u'_{i,unc}$  را نیز به صورتی بهتر در نظر بگیریم، باید یک دامنه مثل  $k_i$  برای  $s_i$  در نظر بگیریم و به دلیل وجود  $\eta_i \operatorname{sgn}(s_i)$  بهتر است یک ضریب نیز به آن بدهیم. پس داریم:

$$u'_{i,unc} = k_i \operatorname{sgn}(s_i) \quad (46-3)$$

رابطه‌ی (45-3) را بازنویسی می‌کنیم:

$$\left\{ \sum_{i \neq j} \Delta_{ij} k_j \operatorname{sgn}(s_j) \right\} + (1 + \Delta_{ii}) k_i \operatorname{sgn}(s_i) \geq \eta_i \operatorname{sgn}(s_i) + (f_i - \hat{f}_i) + \left\{ \sum_{j=1}^r \Delta_{ij} (\ddot{x}_{i,d} - \lambda_i \dot{x}_i - \hat{f}_j) \right\} \quad (47-3)$$

برای اینکه این نامساوی برقرار باشد، باید در حالتی که سمت راست، ماکزیمم مقدار خود، و سمت چپ، مینیمم مقدار خود را دارد، برقرار باشد.

پس داریم :

$$\operatorname{Min}(\text{Left Hand Side}) \geq \operatorname{Max}(\text{Right Hand Side})$$

حال می‌دانیم که :

$$D_{ij} \geq \Delta_{ij} \geq -D_{ij} \Rightarrow$$

که نتیجه می‌شود:

$$\Delta_{ij} \geq -D_{ij} \quad (48-3)$$

پس برای سمت راست رابطه‌ی (3-47) به لحاظ قدر مطلق می‌توان نوشت:

$$\text{Min(Left Hand Side)} = (1 - D_{ii})k_i - \left\{ \sum_{i \neq j} D_{ij} k_j \right\} \quad (49-3)$$

برای طرف راست با توجه به (3-33) و (3-34) داریم:

$$\text{Max(Right Hand Side)} = \eta_i + F_i + \left\{ \sum_{j=1}^r D_{ij} |\ddot{x}_{i,d} - \lambda_i \dot{x}_i - \hat{f}_i| \right\} \quad (50-3)$$

پس در کل شرط به صورت زیر درمی‌آید:

$$(1 - D_{ii})k_i - \left\{ \sum_{i \neq j} D_{ij} k_j \right\} \geq \eta_i + F_i + \left\{ \sum_{j=1}^r D_{ij} |\ddot{x}_{i,d} - \lambda_i \dot{x}_i - \hat{f}_i| \right\} \quad (51-3)$$

برای حل در حالت تساوی می‌توان از قضیه فروبیینوس استفاده کرد.

**قضیه فروبیینوس - پرون:** اگر ماتریس مربعی  $A$  دارای عناصر نامنفی باشد، آنگاه:

۱- بزرگترین مقدار ویژه حقیقی  $A$ ،  $\rho_1$ ، نامنفی است.

۲- معادله  $(I - \rho^{-1}A)y = z$ ، به شرط اینکه  $Z$  برداری با

عناصر نامنفی باشد، و  $\rho \geq \rho_1$  باشد، یک جواب یکتا با درایه-

های نامنفی برای  $y$  دارد.

با اعمال این قضیه به ماتریس  $D$  بدست خواهد آمد:

$$(I - D)y = z \quad (52-3)$$

توجه کنید که فرض کردیم  $\hat{B}$  معکوس پذیر است. فرض می‌کنیم که  $\rho_1 \geq 1$  و می‌توان قانون کنترل را با  $k$  های مثبت و یکتا بدست آورد.

شروط را بار دیگر می‌نویسم :

$$(\mathbf{1} - D_{ii})k_i - \left\{ \sum_{i \neq j} D_{ij} k_j \right\} = \eta_i + F_i + \left\{ \sum_{j=1}^r D_{ij} |\ddot{x}_{i,d} - \lambda_i \dot{\hat{x}}_i - \hat{f}_i| \right\} \quad (53-3)$$

و برای یافتن  $k$ ،  $z$  را تعریف می‌کنیم:

$$z = \begin{bmatrix} F_1 + \eta_1 + \sum_{j=1}^r D_{1j} |\ddot{x}_{1,d} - \lambda_1 \dot{\hat{x}}_1 - \hat{f}_j| \\ F_r + \eta_r + \sum_{j=1}^r D_{rj} |\ddot{x}_{r,d} - \lambda_r \dot{\hat{x}}_r - \hat{f}_j| \\ F_r + \eta_r + \sum_{j=1}^r D_{rj} |\ddot{x}_{r,d} - \lambda_r \dot{\hat{x}}_r - \hat{f}_j| \end{bmatrix} \quad (54-3)$$

و سپس داریم:

$$\begin{bmatrix} \mathbf{1} - D_{11} & -D_{12} & -D_{13} \\ -D_{21} & \mathbf{1} - D_{22} & -D_{23} \\ -D_{31} & -D_{32} & \mathbf{1} - D_{33} \end{bmatrix} * \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = z \quad (55-3)$$

که در نتیجه داریم:

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = (I - D)^{-1} * z \quad (56-3)$$

در نهایت سیگنال کنترل به صورت زیر بدست می‌آید:

$$U = \hat{B}^{-1} \left( \ddot{X}_d - \lambda \dot{X} - \hat{F} - K \operatorname{sgn}(s) \right) \quad (57-3)$$

که  $k$  از رابطه (56-3) بدست می آید که  $D=[D_{ij}]$  ,  $Z$  نیز در رابطه ی (54-3) تعریف شده است. ۶ پارامتر نامنفی  $\eta_1, \eta_2, \eta_3, \lambda_1, \lambda_2, \lambda_3$  جهت بهینه سازی در دست طراح است.

در اینجا نیز بحث پایداری لیاپونوف با انتخاب  $V = 0.5s^T s$  به سادگی به دست می آید. پس از مشتق-گیری از  $V$  به شروط لغزش می رسیم که همانند حالت تک ورودی-تک خروجی به صورت کوچکتر مساوی صفر می شود.

فصل چہارم

دینامیک و سینماتیک پیوما ۵۶۰



رباتیک یکی از میدان‌های فناوری جدید است که از مرزهای مهندسی سنتی عبور کرده است. این میدان که اکنون به رشته مستقلی تبدیل شده است، نیاز به دانش مهندسی برق، مکانیک، صنایع، علوم کامپیوتر، اقتصاد و ریاضیات دارد. علاوه بر این، رشته‌های جدیدی چون مهندسی ساخت و تولید و مفاهیمی نظیر بنیایی ماشین و هوش مصنوعی نیز در آن وارد شده است. برای درک رفتار ربات نیاز به دانش در سه بخش سینماتیک، دینامیک و کنترل ربات داریم.

سینماتیک مستقیم<sup>۱</sup> به تعیین محل عملگر نهایی براساس دانستن متغیرهای مفاصل می‌پردازد. منظور از متغیرهای مفاصل، زاویه در مفاصل لولایی، و جابه‌جایی در مفاصل کشویی است و دیگر مفاصل را نیز می‌توان به مفاصلی از این دو نوع با فاصله صفر از هم تبدیل کرد. در واقع در سینماتیک مستقیم برای پیوما ۵۶۰ با دانستن زوایای مفاصل، محل عملگر نهایی<sup>۲</sup> برای ربات باید یافته شود.

در سینماتیک وارون<sup>۳</sup>، برعکس این کار انجام می‌شود. یعنی با دانستن مختصات مجری نهایی به یافتن زوایای بین رابطها پرداخته می‌شود. این مسئله کمی مشکل است زیرا گاهی چند جواب و چند حالت می‌توان یافت که همه، مکان عملگر نهایی را به صورت یکسان درآورند.

بازوی ربات اساساً یک دستگاه کنترل موقعیت است که برای این منظور باید ویژگی‌های دینامیکی آن را داشته باشیم تا نیروی لازم را برای راندن آن اعمال کنیم. نیروی بسیار کم به منزله کند عمل کردن بازوی ماهر و نیروی بسیار زیاد، احتمال برخورد با شیء و یا نوسان پیرامون موقعیت مطلوب را زیاد می‌کند. بدست آوردن معادلات دینامیکی حرکت نیز در کل، کار آسانی نیست زیرا تعداد درجات آزادی معمولاً زیاد و سیستم نیز غیرخطی است.

---

<sup>۱</sup> Forward Kinematics

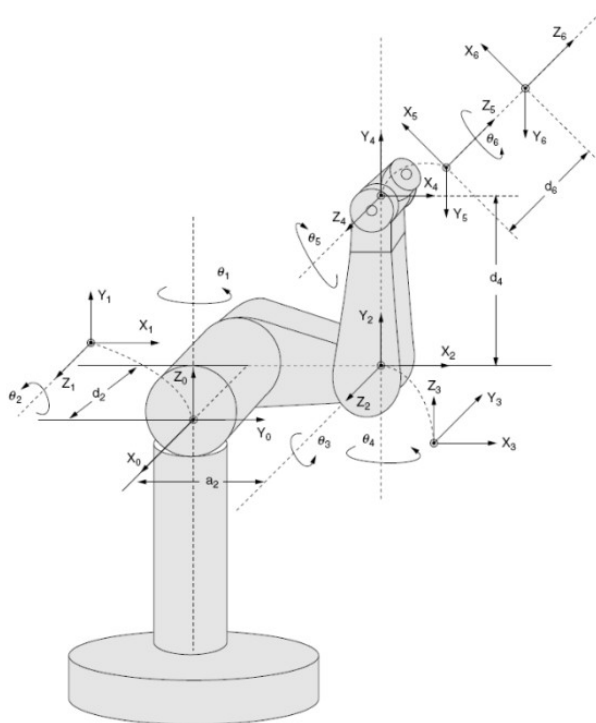
<sup>۲</sup> End-Effector

<sup>۳</sup> Inverse Kinematics

همان طور که گفته شد، بازوی ربات اساساً کنترل کننده موقعیت است [۴۱]. این مسئله کنترل حرکت، از دو مسئله کنترل ردگیری و حذف اثر اغتشاش تشکیل می‌شود. روش‌های کنترل مختلفی جهت کنترل ردگیری و کنترل مقاوم ربات ارائه شده است که در ادامه یکی از این روش‌ها مرور می‌شود. ربات مورد استفاده در این تحقیق، پیوما ۵۶۰ است که هم در صنعت و هم در محیط آکادمیک، بسیار پرکاربرد است [۴۰]. در ادامه‌ی این فصل ابتدا سینماتیک مستقیم و وارون بررسی خواهد شد. سپس دینامیک ربات ارائه می‌شود و در پایان، نتایج کنترل گشتاور این ربات با روش اسلوتین خواهد آمد.

#### ۱-۴ بررسی سینماتیک مستقیم و وارون

سینماتیک مستقیم یافتن محل عملگر نهایی با استفاده از متغیرهای مفاصل است. برای این کار از روش دناویت-هارتنبگ [۴۰] استفاده می‌شود که نمادگذاری آن در شکل (۱-۴) آمده است.



شکل (۱-۴) نمادگذاری دناویت-هارتنبگ برای پیوما ۵۶۰ [۴۰]

پارامترهای دناویت- هارتنبرگ [۴۰]، نیز برای این ربات در جدول (۱-۴) آمده است.

جدول (۱-۴) پارامترهای دناویت-هارتنبرگ برای پیوما ۵۶۰ [۴۰]

Link i	$\theta_i(\text{rad})$	$\alpha_i(\text{rad})$	$a_i(\text{m})$	$d_i(\text{m})$
۱	$\theta_1$	$-\pi/2$	۰	۰
۲	$\theta_2$	۰	۰,۴۳۱۸	۰,۱۴۹۰۹
۳	$\theta_3$	$\pi/2$	۰,۰۲۰۳	۰
۴	$\theta_4$	$-\pi/2$	۰	۰,۴۳۳۰۷
۵	$\theta_5$	$\pi/2$	۰	۰
۶	$\theta_6$	۰	۰	۰,۰۵۶۲۵

روابط سینماتیک مستقیم برای پیوما ۵۶۰ به این صورت است [۴۰]:

$$P_x = \cos(\theta_1) * \{d_4 \sin(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)\} - d_1 \sin(\theta_1) \quad (1-4)$$

$$P_y = \sin(\theta_1) * \{d_4 \sin(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)\} + d_1 \sin(\theta_1)$$

$$P_z = \{d_4 \cos(\theta_2 + \theta_3) - a_2 \sin(\theta_2) - a_3 \sin(\theta_2 + \theta_3)\}$$

که پارامترها در جدول (۱-۴) آمده است.

سینماتیک وارون یافتن زوایا براساس محل خروجی نهایی است. در [۴۲] با استفاده از روش هندسی و

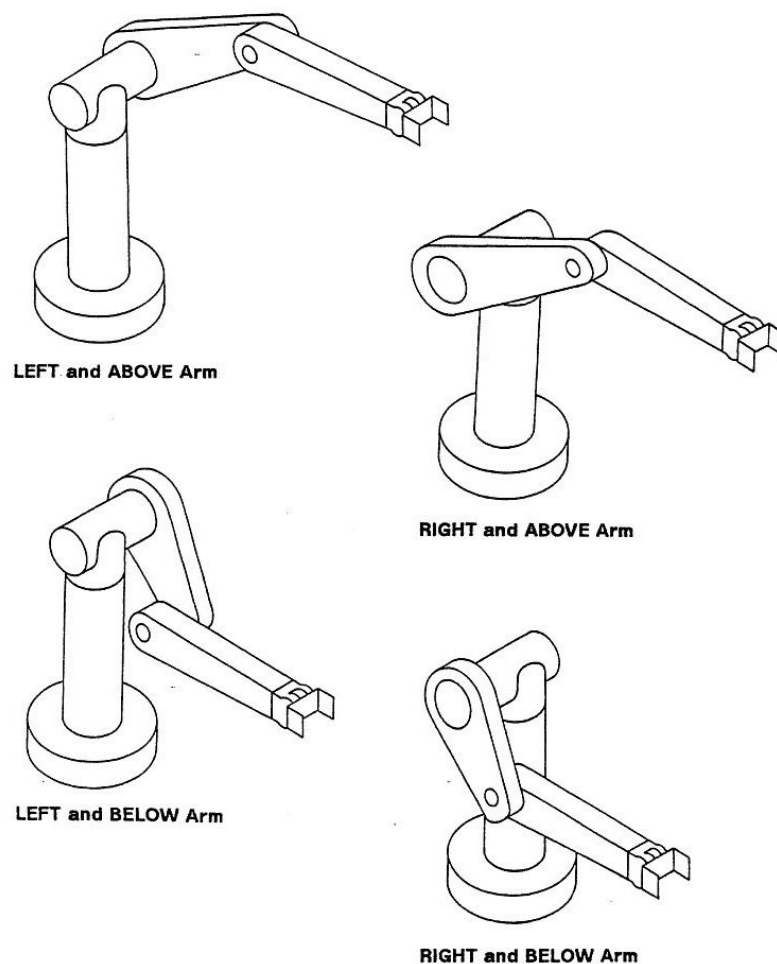
انتخاب‌های کاربر برای پارامترهای آرنج<sup>۱</sup> و بازو<sup>۲</sup> به طور یکتا به تعیین زوایای مفاصل پرداخته شده است.

در پیوست روابط ریاضی آورده شده است.

<sup>۱</sup> Elbow

<sup>۲</sup> Arm

همانطور که گفته شد [۴۲]، دو متغیر بازو و آرنج و نیز متغیر حاصل ضرب آنها، به طور یکتا مقادیر زوایا را مشخص می‌کنند. در شکل (۲-۴) انواع پیکربندی‌ها برای این دو متغیر نمایش داده شده است.



شکل (۲-۴) انواع پیکربندی های پیوما ۵۶۰ [۴۲]

## ۲-۴ دینامیک ربات ۵۶۰ Puma

دینامیک، مطالعه حرکت با توجه به نیروهاست [۴۱]. مدلسازی دینامیک، برای کنترل، طراحی مکانیکی و شبیه سازی، حیاتی است. روش های متعددی برای محاسبه معادلات دینامیک ربات ارائه شده است. این روش ها شامل روش نیوتن- اوپلر، لاگرانژ- اوپلر و روش کین [۴۰] است.

روش نیوتن- اویلر براساس قانون دوم نیوتن است [۴۰]. این روش، رفتار ربات را به صورت لینک به لینک و مفصل به مفصل از پایه تا مجری نهایی توصیف می کند. در روش اویلر- لاگرانژ، محاسبه معادلات دینامیک آسانتر است و روش کین نیز در ۱۹۶۱ توسط پرفسور کین [۴۰] مطرح شد. معادله دینامیک پیوما [۴۰] صورت زیر است:

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau \quad (۲-۴)$$

$\tau$ ، ورودی کنترل و برداری  $۱ \times ۳$ ،  $q$  متغیر زاویه مفاصل،  $\dot{q}$  سرعت مفاصل،  $\ddot{q}$  نیز بیانگر شتاب مفاصل،  $M(q)$  ماتریس اینرسی، ماتریسی  $۳ \times ۳$ ، متقارن و مثبت معین،  $N(q, \dot{q})$  بردار  $۱ \times ۳$  است که عبارت غیر خطی را شامل می باشد که داریم:

$$N(q, \dot{q}) = V(q, \dot{q}) + G(q) \quad (۳-۴)$$

که  $G(q)$  عبارت نیروی گرانش و  $V(q, \dot{q})$  شامل عبارات کوریولیس و گریز از مرکز است. پارامترهای ربات نیز در [۴۰] جداول (۲-۴) و (۳-۴) آمده است، که همه دارای عدم قطعیت هستند.

جدول (۲-۴) ثابت‌های جاذبه (N.m) [۴۰]

$g_1 = -37.2 \pm 0.5$	$g_2 = -8.44 \pm 0.20$
$g_3 = 1.02 \pm 0.50$	$g_4 = 0.249 \pm 0.025$
$g_5 = -0.0282 \pm 0.0056$	

جدول (۳-۴) ثابت‌های اینرسی ( $\text{kg.m}^2$ ) ربات پیوما ۵۶۰ [۴۰]

$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.0238 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.000070$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

### ۳-۴ کنترل گشتاور پیوما ۵۶۰

کنترل یک سیستم چند ورودی چند خروجی در بخش ۳-۳-۲ آمد. در این جا نیز همان روابط برای ربات بررسی می‌شود. در روش اسلوتین [۳۹] بحث لرزش سیگنال کنترل وجود دارد که البته با استفاده از تابع اشباع و یا لایه‌ی مرزی می‌توان بر این مشکل فائق آمد. در شکل (۴-۴) نتایج شبیه‌سازی برای ربات با دو نوع تنظیم پارامتر آمده است. در جدول‌های (۳-۴) و (۴-۴) پارامترها به ترتیب برای روش اسلوتین و روش کنترل مراتب بالا، در حالت غیر بهینه و نیز بهینه شده با الگوریتم پرنده‌گان برای سیستم

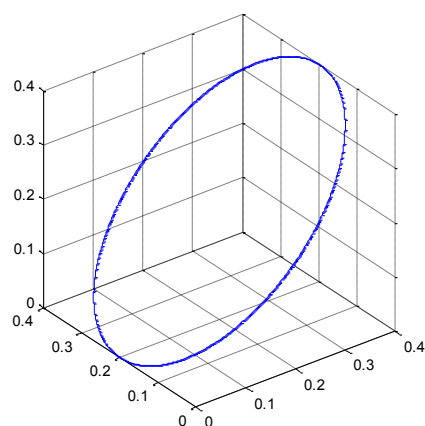
نامی (بدون در نظر گرفتن عدم قطعیت‌ها و اختلال) آمده است. هدف، ردگیری است که مسیر مورد نظر

در رابطه‌ی (۴-۴) و در شکل (۳-۴) آمده است:

$$\begin{cases} p_x = 0.3 + 0.2 * \sin(\pi t / 5) \\ p_y = 0.3 + 0.2 * \cos(\pi t / 5) \\ p_z = 0.3 + 0.2 * \sin(\pi t / 5) \end{cases} \quad (4-4)$$

جدول (۴-۴) پارامترهای روش اسلوتین

کنترلر غیر بهینه	کنترلر بهینه
landa <sub>۱</sub> =۵	landa <sub>۱</sub> =۲,۰۶۰۰
landa <sub>۲</sub> =۵	landa <sub>۲</sub> =۱,۳۲۵۰
landa <sub>۳</sub> =۵	landa <sub>۳</sub> =۱,۹۱۴۸
eta <sub>۱</sub> =۰,۰۰۱	eta <sub>۱</sub> =۰,۰۸۰۰
eta <sub>۲</sub> =۰,۰۰۱	eta <sub>۲</sub> =۰,۰۵۷۲
eta <sub>۳</sub> =۰,۰۰۱	eta <sub>۳</sub> =۰,۰۳۶۴



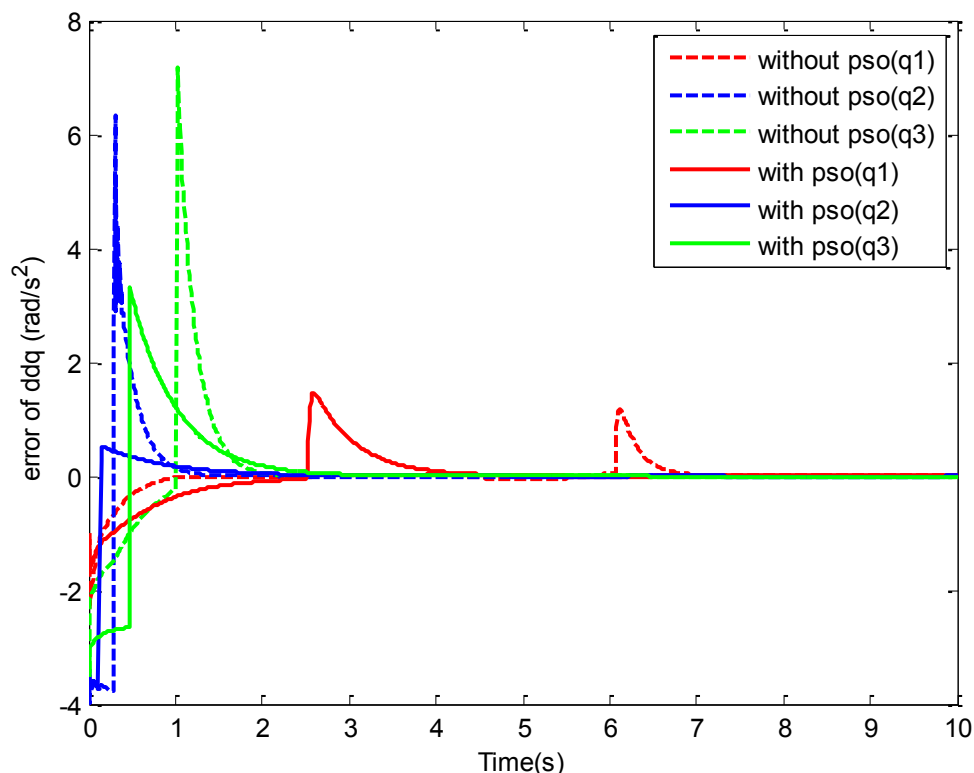
شکل (۳-۴) مسیر مطلوب حرکت بازوی ربات

همانطور که از جدول (۴-۴) نیز مشخص است، در حالت غیر بهینه، هیچگونه اطلاعاتی درباره‌ی نحوه

انتخاب پارامترها در دست نیست. به این دلیل مقادیر به صورت یکسان انتخاب شده است. در حالت بهینه

اما الگوریتم پرندگان بهینه‌سازی را با توجه به تابع هزینه انجام می‌دهد که در نهایت تنظیم پارامترها

مشابه آنچه در جدول (۴-۴) است می‌شود.



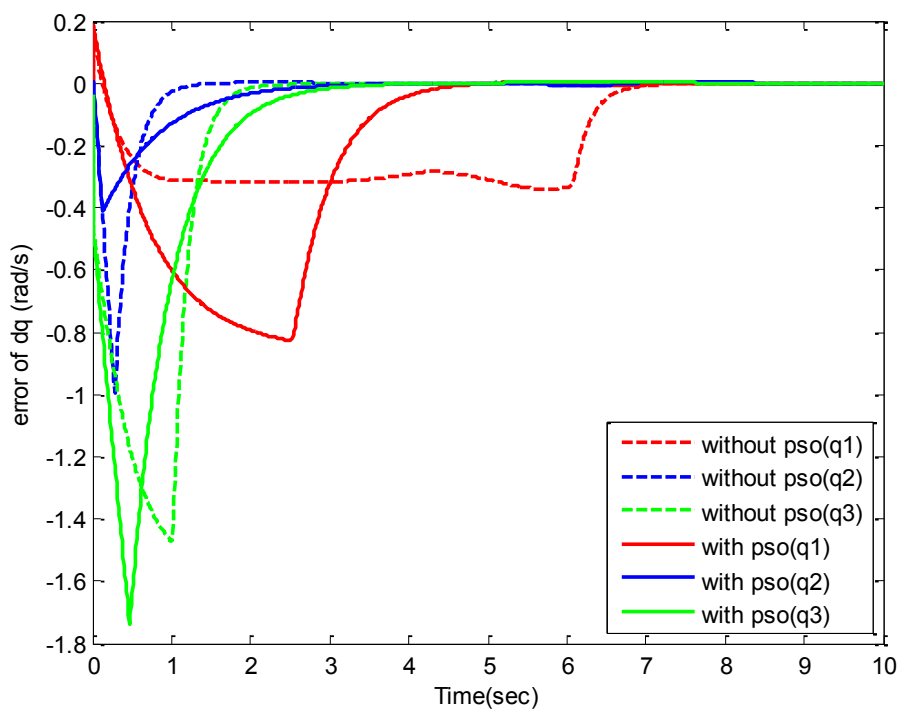
شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین

(الف) خطای  $\ddot{q}$

منظور از حالت بهینه در شکل‌های (۴-۴) این است که مدل نامی که در آن پارامترها دارای عدم قطعیت نیستند، توسط الگوریتم پرندگان بهینه شده‌اند.

در شکل (۴-۴) (الف) خطای  $\ddot{q}$  آورده شده است. با توجه به رابطه‌ی (۴-۲) مقدار این خطا در سیگنال کنترل تاثیر می‌گذارد. همانطور که مشخص است، مقدار ماکزیمم خطا در حالت بهینه‌سازی پارامترها با حالت نامی بهبود داشته‌است و برای خطای مشتق دوم مفاصل اول و سوم به ترتیب از ۷,۵ و ۶,۵ به ۳,۵ و ۰,۷ رادیان بر مجذور ثانیه رسیده است. زمان رسیدن به مقدار پیک نیز به ترتیب از ۶,۴ و ۰,۴ و ۱ ثانیه به ۲,۵، ۰,۱ و ۰,۲ رسیده است که هر دو مورد بیان‌کننده بهبود است.

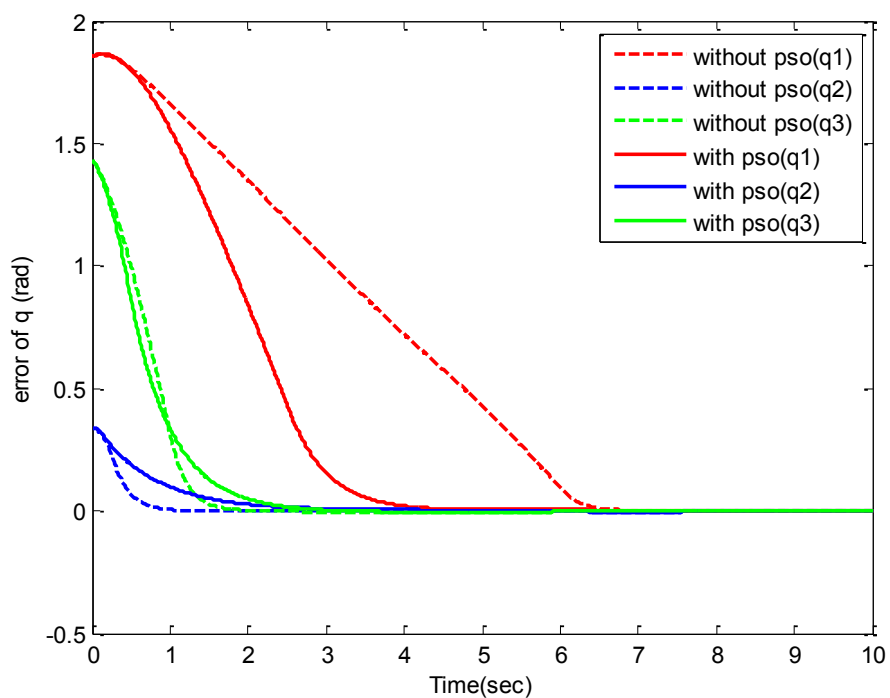




شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین

(ب) خطای  $\dot{q}$

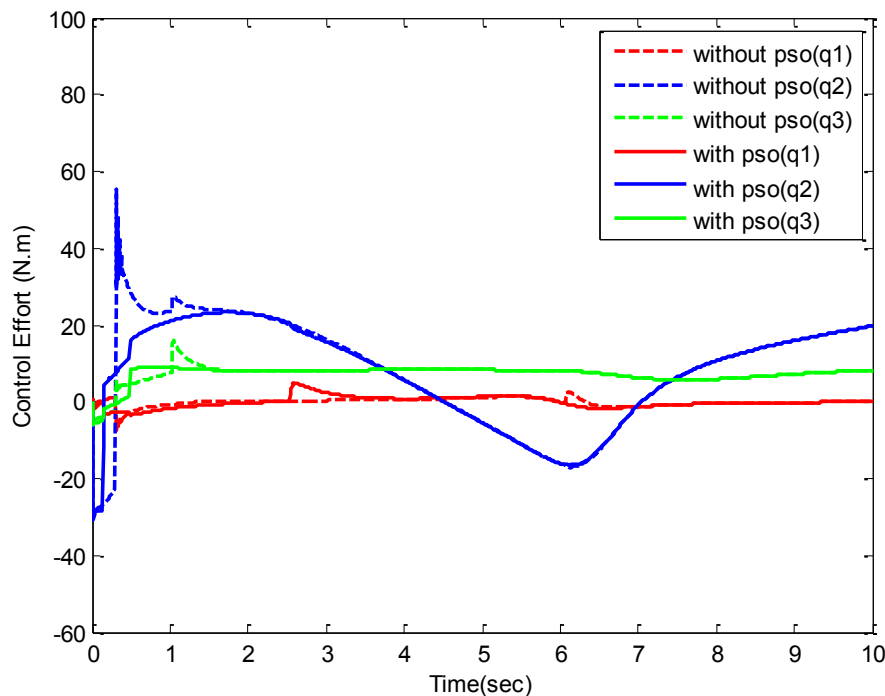
در شکل (۴-۴) (ب) خطای سرعت زاویه‌ای آمده‌است. مقدار فرا نشست برای خطای سرعت مفصل دوم و سوم با افزایش مقدار قدرمطلق از ۰,۳ و ۱,۴ به ۰,۹ و ۱,۷ رادیان بر ثانیه رسیده است که به ظاهر نکته مطلوبی نیست ولی برای مفصل دوم از ۱ به ۰,۴ رسیده است که نشان‌دهنده‌ی پیشرفت است. از طرف دیگر سرعت صفر شدن خطا در مفصل اول از ۶ ثانیه به ۴,۵ ثانیه رسیده است ولی برای مفصل دوم و سوم کمی زمان رسیدن به مقدار نهایی در حدود ۰,۲ ثانیه بیشتر شده است.



شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین

(ج) خطای  $q$

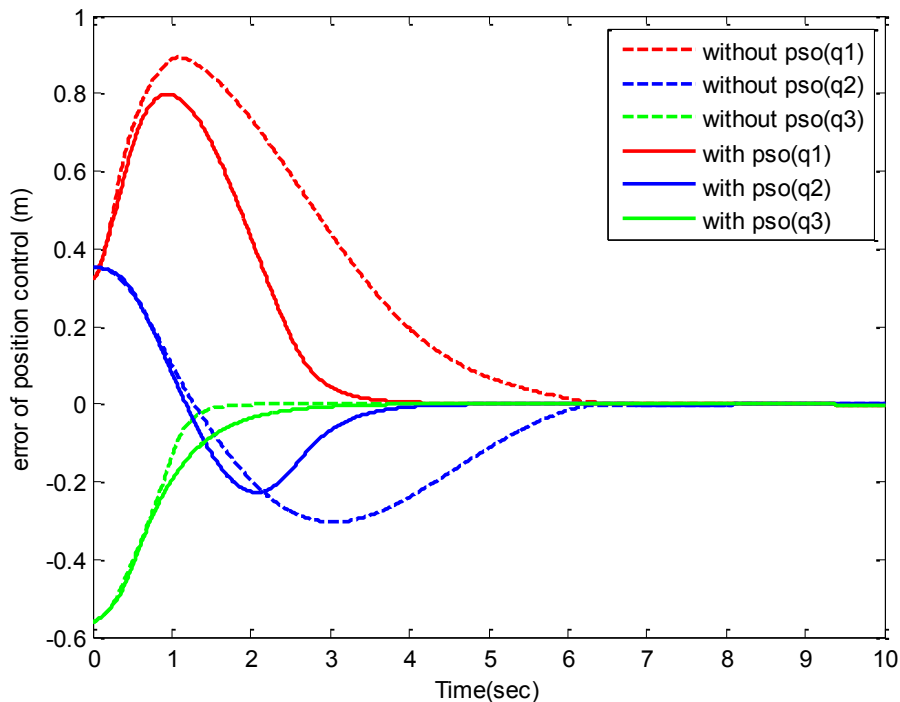
در شکل (۴-۴) (ج) خطای مفاصل آورده شده است. برای مفصل اول، زمان صفر شدن خطا از ۶,۴ به حدود ۴ ثانیه رسیده است که بسیار چشمگیر است ولی برای مفاصل دوم و سوم زمان رسیدن خطا به صفر تغییر چندانی نداشته است. توجه شود که برتری در شکل‌های (۴-۴) (د) و (ه) مشخص است.



شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین

(د) سیگنال کنترل

در شکل (۴-۴) (د)، سیگنال کنترل آمده است. همانطور که مشاهده می‌شود برای گشتاور مفصل اول در ثانیه ۲,۸ یک فرجهش داریم که برای حالت غیر بهینه در ۶ ثانیه رخ داده است که نشان دهنده‌ی سریعتر شدن سیگنال کنترل است. از طرفی برای گشتاور مفصل دوم و سوم به ترتیب یک فرجهش تا حدود مقدار ۵۰ و ۱۵ نیوتن-متر داریم که در حالت بهینه این فرجهش‌ها حذف شده است. پس از حالت گذرا هر دو روش به یک مقدار همگرا می‌شوند که نشان دهنده‌ی طراحی صحیح کنترل‌کننده است.



شکل (۴-۴) نتایج شبیه‌سازی پیوما ۵۶۰ با روش اسلوتین

(ه) خطای کنترل موقعیت

در شکل (۴-۴) (ه) خطای کنترل موقعیت نهایی آمده است. در این شکل می‌توان برتری روش بهینه شده توسط الگوریتم پرندگان در حالت نامی را مشاهده کرد. توجه شود که آنچه که کاربر در عمل مشاهده می‌کند همین مورد است که برای  $p_x, p_y, p_z$  از زمان نشست حدود ۶ ثانیه به حدود ۴ ثانیه رسیده است که بیانگر بهبود است. همچنین مقدار فراجش  $p_x, p_y$  بهبود داشته است و به ترتیب از ۰,۹ و ۰,۳ به ۰,۸ و ۰,۲ ثانیه رسیده است. تنها برای  $p_z$  چندان بهبود نداریم. برای مقایسه بهتر می‌توان به جدول (۵-۴) رجوع کرد. جدول (۵-۴) شاخص کارایی را نشان می‌دهد که به وضوح برای این ربات، بهبود عملکرد قابل مشاهده است.

جدول (۵-۴) مقایسه شاخص‌های کارایی برای ربات پیوما ۵۶۰ برای مدل نامی

performance_index_tau ( $\sum \tau^2$ )	performance_index_e ( $\sum e^2(q)$ )	Perf.index Method
۲,۸۰۳۳e+۰۰۵	۵۹۲,۵۹۶۲	With PSO
۳,۳۲۵۲e+۰۰۵	۹۲۰,۲۹۰۷	Without PSO

همانطور که مشخص است هم برای خطای کلی مفاصل و هم برای مقدار کلی گشتاور بهبود داشته‌ایم و به ترتیب از حدود ۹۲۰ و ۳۳۰۰۰۰ به ۵۹۰ و ۲۸۰۰۰۰ رسیده است که نشان‌دهنده‌ی بهبود است.

## فصل پنج

معرفی یک روش جدید برای کنترل بهینه تحت

عدم قطعیت

هدف پایان‌نامه‌ی حاضر، بهینه‌سازی سیگنال کنترل تحت عدم قطعیت است و همانطور که بیان شد، ابزار بهینه‌سازی، الگوریتم پرنندگان است. این الگوریتم در حالت خارج از خط کار می‌کند و نیاز به اجرای متعدد سیستم در دست بررسی جهت یافتن پارامترهای بهینه دارد. نکته‌ی حائز اهمیت دیگر این است که الگوریتم پرنندگان نمی‌تواند با تابع نامشخص کار کند و تابع هزینه‌ای که مینیمم می‌شود، باید قطعی باشد. از سوی دیگر عدم قطعیت فقط در حالت برخط روی می‌دهد که تابع هزینه سیستم یا فرایند را تا حدی نامعلوم می‌کند. چنین وضعیتی مانع کار بهینه‌سازی شده و مانع از بالارفتن کار می‌شود. البته مانند آنچه در فصل اول اشاره شد [۹-۶]، می‌توان تنها مدل نامی را بهینه کرد اما در این صورت اقدامی در جهت شناخت و تخمین عدم قطعیت صورت نگرفته است. حتی اگر بتوان بگونه‌ای عدم قطعیت را برآورد کرد، الگوریتم پرنندگان زمان کافی برای بهینه‌سازی در حالت برخط را در اختیار ندارد. بنابراین دو مسئله در دست است: اول تخمین عدم قطعیت در حالت برخط و دوم بهینه‌سازی به کمک الگوریتم پرنندگان، برای سیستمی که عدم قطعیت آن تا حدی مشخص شده است. در ادامه‌ی این فصل راه‌حلی برای این دو مسئله ارائه خواهد شد. در ابتدا درباره‌ی عدم قطعیت و الگوریتم پرنندگان صحبت خواهد شد. سپس برای یک سیستم ساده تک ورودی - تک خروجی، نتایج اجرا برای بهینه‌سازی مدل نامی آورده خواهد شد. در بخش سوم یک روش جدید برای مسائل مطرح شده ارائه و با بخش دوم مقایسه می‌شود. در بخش پایانی روش ارائه شده برای ربات پیوما ۵۶۰ پیاده‌سازی و نتایج بررسی خواهد شد.

### ۵-۱ درباره عدم قطعیت و الگوریتم پرنندگان

همانطور که بیان شد، عدم قطعیت در این تحقیق از نوع غیرتصادفی (معین)<sup>۱</sup> است. این بدین معنی است که عدم قطعیت، ساختاری تصادفی<sup>۲</sup> ندارد. اگر عدم قطعیت به صورت معین باشد، می‌توان گفت تابعی

<sup>۱</sup> Deterministic Uncertainty

<sup>۲</sup> Stochastic Uncertainty

مانند  $\varepsilon(x, u, t)$  وجود دارد که می‌تواند بخش عدم قطعیت را فرمول‌بندی کند [۴۳]. تابع  $\varepsilon$  می‌تواند تابع مواردی همانند زمان، حالت‌ها، سیگنال کنترل و حتی شرایط محیطی باشد. اما یافتن چنین تابعی به سادگی امکان‌پذیر نیست. با توجه به اینکه فقط در حالت برخط می‌توان چنین تابعی را تعریف کرد، باید به گونه‌ای آن را در حالت برخط تخمین زد. روش‌های مختلفی برای تخمین در حالت برخط وجود دارد. اگر تابع با ضرایب ثابت باشد، روش مجموع کمترین مربعات گوس که در کنترل تطبیقی استفاده می‌شود، بهترین است. در غیر اینصورت تخمین با سری فوریه نیز یکی از روش‌هاست که تخمین دو متغیره آن در یکی از فرمت‌های تصویری<sup>۱</sup> در کامپیوتر استفاده می‌شود. اما به دلیل وجود عدم قطعیت، نمی‌توان فرض کرد که موارد مورد نیاز ما ثابت ناشناخته<sup>۲</sup> هستند [۴۳].

همانطور که بیان شد، الگوریتم پرندگان برای بهینه‌سازی نیاز به یک تابع ثابت و مشخص دارد. اگر حتی فرض شود بتوان به روشی تابع  $\varepsilon$  را تخمین زد، برای بهینه‌سازی و یافتن مقادیر بهینه توسط الگوریتم پرندگان حداقل چند ثانیه برای شبیه‌سازی نیاز است. حال آنکه با توجه به متغیر بودن مقادیر عدم قطعیت، زمان در حدود ۰/۰۰۱ و یا حداکثر ۰/۰۱ ثانیه است. پس به طور قطع نمی‌توان از الگوریتم پرندگان در حالت برخط به این ترتیب استفاده کرد.

بنابراین، دو مورد است که باید مدنظر قرار داد: مورد اول اینکه به طریقی بتوان از حالت برخط اطلاعاتی درباره عدم قطعیت بدست آورد و مورد دوم اینکه الگوریتم پرندگان نیاز به حداقل چند ثانیه زمان جهت بهینه‌سازی دارد.

اولین پیشنهادی که می‌توان ارائه کرد این است که باید در حالت خارج از خط توسط الگوریتم پرندگان بهینه‌سازی‌های لازم را انجام داد تا در حالت برخط از آن‌ها بهره گرفت. این مورد بسیار شبیه به جدول-

<sup>۱</sup> JPEG

<sup>۲</sup> Constant Unknown



بندی بهره<sup>۱</sup> است [۴۴]. در اینجا باید به سیستم دارای عدم قطعیت به دید یک دسته سیستم نگاه کرد که این سیستم‌ها، در یک سیستم نامی یکی هستند. اما در حقیقت با یک سیستم نیز روبرو نیستیم بلکه با بینهایت سیستم یا با تعداد بسیاری سیستم روبرو هستیم. هرچند که این سیستم‌ها تغییر بسیار اندکی نسبت به سیستم نامی دارند ولی همین تغییر کم می‌تواند مقدار سیگنال کنترل بهینه را تغییر دهد. در ضمن اختلال نیز سیستم را تغییر می‌دهد. سیستمی که اختلال  $0/1$  در یکی از کانال‌های آن است با همان سیستم و همان پارامترها که اختلال آن  $0/0.5$  در همان کانال است، از نقطه نظر سیگنال کنترل بهینه، متفاوت است. پس باید به طریقی این سیستم‌ها را از هم جدا کرد و بین آن‌ها تفاوت قایل شد.

در اکثر سیستم‌ها، حدود عدم قطعیت ساختاری و پارامتری مشخص است. می‌توان درصدی نیز برای اختلال که در عدم قطعیت ساختاری وارد می‌شود در نظر گرفت. تا اینجا تعدادی پارامتر دارای عدم قطعیت در مدل نامی سیستم و تعدادی متغیر که بیان‌کننده‌ی عدم قطعیت ساختاری و اختلال هستند در دست است. هر کدام از این پارامترها که در کل بیانگر عدم قطعیت‌های پارامتری و ساختاری هستند، می‌توانند هر مقدار دلخواه در بازه‌ی تعریف شده‌ی خود اختیار کنند. برای سادگی و کاهش حجم محاسبات، در هر بازه، تعداد محدودی مقدار برای هر پارامتر، انتخاب می‌شود. توجه شود که هر کدام از این دسته مقادیر، بیانگر تعدادی سیستم هستند. طبق اصل ضرب یا شمارش، تعداد سیستم نامی به تعداد ضرب همه این سیستم‌هایی است که باید برای آن‌ها الگوریتم پرنندگان را اجرا کرد. پس تا اینجا بگونه‌ای وظیفه‌ی الگوریتم پرنندگان در حالت خارج از خط تعریف شد. با اجرای الگوریتم پرنندگان برای این سیستم‌ها، می‌توان پارامترهای کنترلر بهینه را یافت و جدول بهره را برای این سیستم‌ها تشکیل داد.

اکنون اگر به گونه‌ای بتوان حدود عدم قطعیت را یافت، می‌توان مقدار سیگنال کنترل بهینه را از جدول بهره انتخاب کرد. حال باید به سراغ یافتن تخمینی از عدم قطعیت رفت. عدم قطعیت با زمان تغییر می-

<sup>۱</sup> Gain Scheduling

کند ولی سرعت تغییر آن مشخص نیست. ممکن است این سرعت بسیار کم یا بسیار زیاد باشد. در اینجا برای نزدیک شدن به عدم قطعیت فرض می‌شود که در چند لحظه متوالی عدم قطعیت تغییر محسوسی نداشته باشد. مثلاً در فرکانس نمونه برداری ۱ کیلو هرتز، هر نمونه در زمان ۰/۰۰۱ ثانیه می‌آید و می‌توان فرض کرد که:

$$\varepsilon(k) \cong \varepsilon(k-1) \cong \varepsilon(k-2) \quad (1-5)$$

این فرض، فرضی معقول است و با زیاد کردن فرکانس نمونه برداری، این نامساوی به مساوی نزدیکتر می‌شود. حال در این چند لحظه با توجه به معادلات حالت، مقدار سیگنال کنترل و نیز اندازه‌هایی که سنسور داده است، می‌توان چند معادله چند مجهول تشکیل داد که تعداد معادلات و تعداد لحظات باید به اندازه تعداد فاکتورهای دارای عدم قطعیت، چه پارامتری و چه ساختاری باشد. سپس می‌توان با حل این چند معادله چند مجهول، تخمینی از عدم قطعیت‌ها بدست آورد و آنگاه با استفاده از جدول بهره که توسط الگوریتم پرندگان در حالت خارج از خط بهینه شده است، به بهینه سازی سیگنال کنترل پرداخت. در ادامه برای یک سیستم ساده این روش پیاده می‌شود.

## ۵-۲ بهینه‌سازی یک سیستم ساده با الگوریتم پرندگان

سیستم ساده‌ی فصل سه را در نظر بگیرید:

$$\ddot{x} + \dot{x} - c_1 x = c_2 u + d(t)$$

(۲-۵)

$$c_1 \in [2, 3],$$

$$c_2 \in [1, 2]$$

$$|d(t)| \leq 0.5$$

## فصل پنجم: معرفی یک روش جدید برای کنترل بهینه تحت عدم قطعیت

که  $c_2$  و  $c_3$  پارامترهایی دارای عدم قطعیت هستند و  $d(t)$ ، اختلال است. توسط روش کنترل مد لغزشی اسلوتین [۳۹]، این سیستم تک ورودی - تک خروجی را کنترل می‌کنیم. با توجه به روابط بخش ۳-۳-۱ می‌توان نوشت:

$$u = \frac{1}{\hat{b}} * (\ddot{x}_d - \lambda \dot{x} - \hat{f} - k * \text{sgn}(s)) \quad (۳-۵)$$

که  $\text{sgn}$  تابع علامت،  $\lambda > 0$  پارامتر طراحی،  $\hat{b}$  مقدار نامی و انتخابی برای  $b$ ، و برای  $k$  داریم:

$$k = \frac{\hat{b}}{b_{min}} (\eta + F + \left\{ \text{Max} \left\{ \left| 1 - \frac{b_{min}}{\hat{b}} \right|, \left| 1 - \frac{b_{max}}{\hat{b}} \right| \right\} * |\lambda \dot{x} - \ddot{x}_d - \hat{f}| \right) \quad (۴-۵)$$

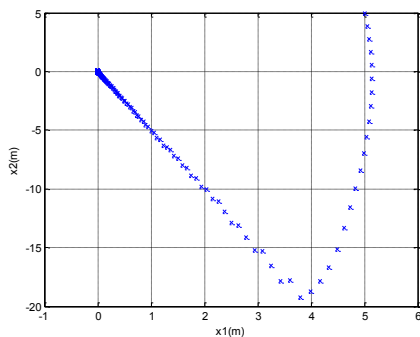
که  $\eta > 0$  متغیر طراحی در شرط لغزش می باشد.

برای پارامترهای نامی داریم:

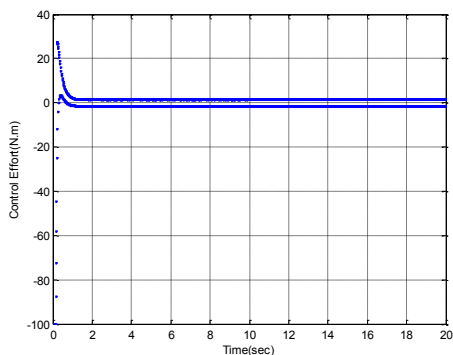
$$\begin{cases} \hat{f} = \hat{c}_r x^r - \dot{x} = 1.5 x^r - \dot{x} \\ \hat{b} = \sqrt{b_{min} * b_{max}} = \sqrt{c_{r,min} * c_{r,max}} = \sqrt{2 * 3} = \sqrt{6} \\ \hat{c}_r = \frac{c_{r,max} + c_{r,min}}{4} = \frac{1 + 2}{4} = 1.5 \\ F = |c_{r,max} - \hat{c}_r| * |x^r| + |d(t)| = 0.5(|x^r| + 1) \end{cases} \quad (۵-۵)$$

برای حالت نامی الگوریتم پرندگان را اجرا کرده که در نتیجه بدست می‌آید:

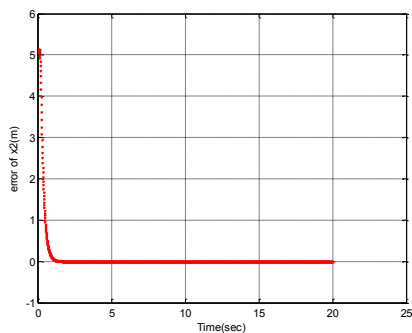
$$\lambda = 2.7106 \quad \eta = 0.0026 \quad (۶-۵)$$



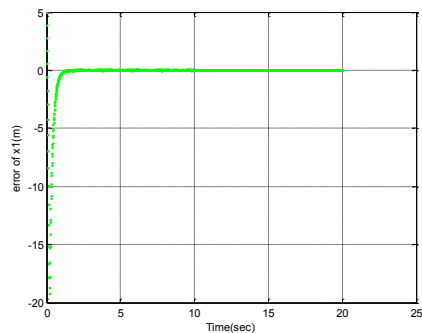
ب) مسیر حالت در صفحه‌ی فاز



الف) سیگنال کنترل



د) خطای  $x$



ج) خطای  $\dot{x}$

شکل (۵-۱) نمودارهای سیستم ساده غیر بهینه

در شکل (۵-۱) الف) سیگنال کنترل آمده است که پدیده‌ی لرزش در آن مشاهده می‌شود. در این شکل به عمد و برای مقایسه با شکل‌های (۵-۲) الف) و (۵-۳) الف) لرزش حذف نشده است. دامنه‌ی نوسان در حدود  $\pm 1.5$  است. در شکل‌های (۵-۱) ب)، (۵-۱) ج) و (۵-۱) د) نیز سیستم کنترل شده و به مقدار مطلوب رسیده است.

### ۵-۳ یک روش جدید برای کنترل بهینه تحت عدم قطعیت

پس از پیاده سازی الگوریتم پرندگان به روش حالت خارج از خط، ابتدا به تولید جدول برای حالت بر خط می پردازیم. جدول برای بازه‌های:

$$c_2 \in [2, 3], \quad (7-5)$$

$$c_2 \in [1, 2]$$

$$d(t) \in [-0.5, 0.5]$$

به ازای ۴ مقدار در هر بازه تولید می‌شود که در واقع ۴ سیستم می‌شود. توجه شود که در حالت خارج

از خط فقط یکی از این سیستم‌ها که همان مقدار نامی است موجود است و حالت‌های دیگر در دسترس

نیست. حال پس از تولید جدول (۱-۵) و یافتن مقادیر  $\lambda$  و  $\eta$  برای این ۶۴ سیستم، بخش بعد تخمین  $c_2$

،  $c_3$  و  $d(t)$  است. برای ۳ لحظه  $k$ ،  $k-1$  و  $k-2$  می‌توان نوشت:

$$\begin{cases} \ddot{x}(k+1) = c_2(k)x^r(k) - \dot{x}(k) + d(k) + c_2(k)u(k) \\ \dot{x}(k) = c_2(k-1)x^r(k-1) - \dot{x}(k-1) + d(k-1) + c_2(k-1)u(k-1) \\ \ddot{x}(k-1) = c_2(k-2)x^r(k-2) - \dot{x}(k-2) + d(k-2) + c_2(k-2)u(k-2) \end{cases} \quad (8-5)$$

حال با فرض:

$$\begin{cases} c_2(k) \cong c_2(k-1) \cong c_2(k-2) \\ c_2(k) \cong c_2(k-1) \cong c_2(k-2) \\ d(k) \cong d(k-1) \cong d(k-2) \end{cases} \quad (9-5)$$

بدست می‌آید:

$$\begin{bmatrix} \ddot{x}(k+1) - \dot{x}(k) \\ \dot{x}(k) - \dot{x}(k-1) \\ \ddot{x}(k-1) - \dot{x}(k-2) \end{bmatrix} = \begin{bmatrix} x^r(k) & u(k) & 1 \\ x^r(k-1) & u(k-1) & 1 \\ x^r(k-2) & u(k-2) & 1 \end{bmatrix} * \begin{bmatrix} c_2 \\ c_2 \\ d \end{bmatrix} \quad (10-5)$$

## فصل پنجم: معرفی یک روش جدید برای کنترل بهینه تحت عدم قطعیت

جدول (۵-۱) جدول بهره بهینه شده با الگوریتم پرندگان برای سیستم ساده

$c_1$	$c_2$	$d(t)$	$\lambda$	$\eta$
۲,۱۲۵۰	۱,۱۲۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۲۵
۲,۱۲۵۰	۱,۱۲۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۹
۲,۱۲۵۰	۱,۱۲۵۰	۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۸
۲,۱۲۵۰	۱,۱۲۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۲۳
۲,۱۲۵۰	۱,۳۷۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۳۷۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۳۷۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۳۷۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۶۲۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۶۲۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۶۲۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۶۲۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۸۷۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۸۷۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۸۷۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۱۲۵۰	۱,۸۷۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۱۲۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۲۵
۲,۳۷۵۰	۱,۱۲۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۵
۲,۳۷۵۰	۱,۱۲۵۰	۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۳۴
۲,۳۷۵۰	۱,۱۲۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۰
۲,۳۷۵۰	۱,۳۷۵۰	-۰,۳۷۵۰	۲,۹۹۹۸	۰,۰۰۳۴
۲,۳۷۵۰	۱,۳۷۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۲
۲,۳۷۵۰	۱,۳۷۵۰	۰,۱۲۵۰	۲,۹۲۲۵	۰,۰۰۴۴
۲,۳۷۵۰	۱,۳۷۵۰	۰,۳۷۵۰	۲,۹۳۱۱	۰,۰۰۹۷
۲,۳۷۵۰	۱,۶۲۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۶۲۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۶۲۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۶۲۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۸۷۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۸۷۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۸۷۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۳۷۵۰	۱,۸۷۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۶۲۵۰	۱,۱۲۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۴
۲,۶۲۵۰	۱,۱۲۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۳۲
۲,۶۲۵۰	۱,۱۲۵۰	۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۳۳
۲,۶۲۵۰	۱,۱۲۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۰
۲,۶۲۵۰	۱,۳۷۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۵
۲,۶۲۵۰	۱,۳۷۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۹
۲,۶۲۵۰	۱,۳۷۵۰	۰,۱۲۵۰	۲,۹۹۹۹	۰,۰۰۲۲
۲,۶۲۵۰	۱,۳۷۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۱
۲,۶۲۵۰	۱,۶۲۵۰	-۰,۳۷۵۰	۲,۸۸۷۹	۰,۰۰۲۸
۲,۶۲۵۰	۱,۶۲۵۰	-۰,۱۲۵۰	۲,۸۱۴۵	۰,۰۰۸۰
۲,۶۲۵۰	۱,۶۲۵۰	۰,۱۲۵۰	۲,۷۸۴۸	۰,۰۰۹۷
۲,۶۲۵۰	۱,۶۲۵۰	۰,۳۷۵۰	۲,۹۷۸۹	۰,۰۰۴۰
۲,۶۲۵۰	۱,۸۷۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۶۲۵۰	۱,۸۷۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۶۲۵۰	۱,۸۷۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۶۲۵۰	۱,۸۷۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۸۷۵۰	۱,۱۲۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۵
۲,۸۷۵۰	۱,۱۲۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۳۳
۲,۸۷۵۰	۱,۱۲۵۰	۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۳۳
۲,۸۷۵۰	۱,۱۲۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۱۷
۲,۸۷۵۰	۱,۳۷۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۵
۲,۸۷۵۰	۱,۳۷۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۵
۲,۸۷۵۰	۱,۳۷۵۰	۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۲۵
۲,۸۷۵۰	۱,۳۷۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۳
۲,۸۷۵۰	۱,۶۲۵۰	-۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۵
۲,۸۷۵۰	۱,۶۲۵۰	-۰,۱۲۵۰	۳,۰۰۰۰	۰,۰۰۱۰
۲,۸۷۵۰	۱,۶۲۵۰	۰,۱۲۵۰	۲,۹۸۳۷	۰,۰۰۴۱
۲,۸۷۵۰	۱,۶۲۵۰	۰,۳۷۵۰	۳,۰۰۰۰	۰,۰۰۳۴
۲,۸۷۵۰	۱,۸۷۵۰	-۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۸۷۵۰	۱,۸۷۵۰	-۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۸۷۵۰	۱,۸۷۵۰	۰,۱۲۵۰	۱,۱۶۶۷	۰,۰۰۵۵
۲,۸۷۵۰	۱,۸۷۵۰	۰,۳۷۵۰	۱,۱۶۶۷	۰,۰۰۵۵

که در نتیجه داریم:

$$\begin{bmatrix} c_2 \\ c_1 \\ d \end{bmatrix} = \begin{bmatrix} x^T(k) & u(k) & 1 \\ x^T(k-1) & u(k-1) & 1 \\ x^T(k-2) & u(k-2) & 1 \end{bmatrix}^{-1} * \begin{bmatrix} \dot{x}(k+1) - \dot{x}(k) \\ \dot{x}(k) - \dot{x}(k-1) \\ \dot{x}(k-1) - \dot{x}(k-2) \end{bmatrix} \quad (11-5)$$

حال با توجه به تخمین بدست آمده و با توجه به جدول بهره (۵-۱) می توان نتایج را برای سیستم واقعی مشاهده کرد. توجه کنید که این رابطه برای هر سه داده ی متوالی بدست می آید و به صورت بازگشتی (که وابسته به تخمین اولیه است) نمی باشد.

### ۵-۳-۱ بررسی نتایج برای سیستم ساده

برای سیستم بخش ۵-۲، الگوریتم پرندگان استاندارد و روش جدید را مقایسه می کنیم. در شبیه سازی از تابع علامت استفاده کردیم و از تابع اشباع<sup>۱</sup> یا همان لایه ی مرزی استفاده نشده است. زمان شبیه سازی ۱۰ ثانیه و هدف پایدارسازی<sup>۲</sup> بوده است. نقطه شروع [۵،۵] برای  $X$  و  $\dot{X}$  بوده است و برای مدل به حالت بر خط قرار داده شد:

$$\begin{aligned} c_2 &= 2.5 + 0.5 * \cos(\pi t) \\ c_1 &= 1.5 - 0.5 * \sin(2\pi t) \\ d(t) &= 0.5 * \sin(3\pi t) \text{ for } t \in [0.5, 5] \end{aligned} \quad (12-5)$$

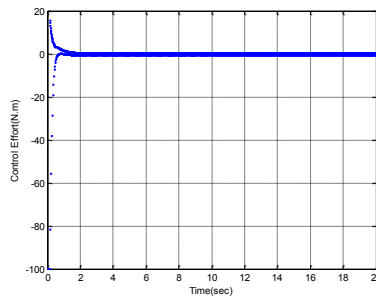
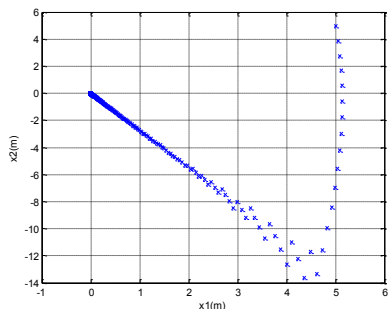
شاخص کارایی را به صورت (۵-۱۳) تعریف می کنیم:

<sup>۱</sup> Saturation

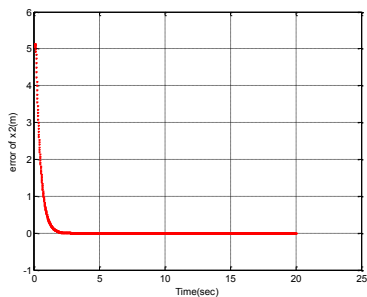
<sup>۲</sup> Stabilization

$$Performance\ Index = \int_{t_{start}}^{t_{final}} (x^T + u^T) dt \quad (5-13)$$

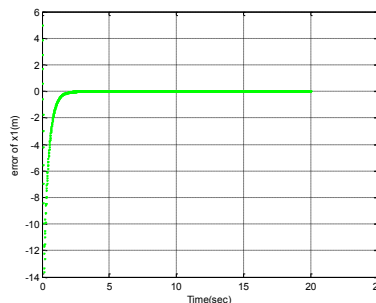
که  $t_s$  زمان شروع و  $t_f$  زمان پایان شبیه‌سازی است.



ب) مسیر حالت در صفحه‌ی فاز



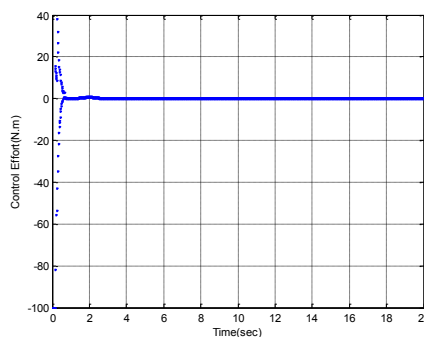
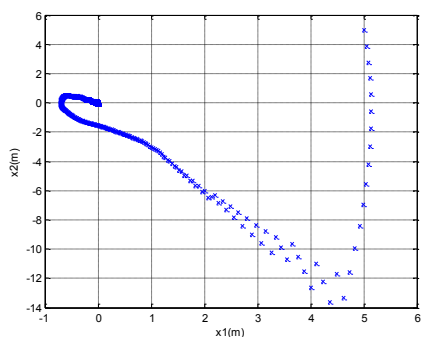
الف) سیگنال کنترل



د) خطای  $x$

ج) خطای  $\dot{x}$

شکل (۲-۵) نمودار های سیستم ساده بهینه شده با الگوریتم پرندهگان در حالت خارج از خط

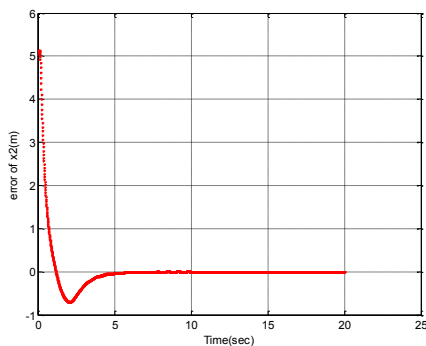


ب) مسیر حالت در صفحه‌ی فاز

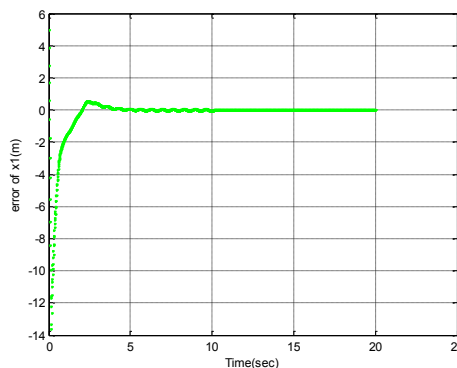
الف) سیگنال کنترل

شکل (۳-۵) نمودار های سیستم ساده بهینه شده با روش جدید





د) خطای  $x$



ج) خطای  $\dot{x}$

ادامه‌ی شکل (۵-۳) نمودار های سیستم ساده بهینه شده با روش جدید

در شکل (۵-۲) (الف) سیگنال کنترل هنوز لرزش دارد که دامنه‌ی آن در حدود  $\pm 0.5$  در حال نوسان است که نسبت به شکل (۵-۱) بهبود داشته است. سرعت نشست نیز در حدود ۱ ثانیه است که نسبت به شکل (۵-۱) (الف) تغییر چندانی نداشته است.

در شکل (۵-۳) (الف) همانطور که ملاحظه می‌شود، روش جدید توانسته است دامنه‌ی نوسان را به  $\pm 0.05$  برساند که عملاً در حد صفر است. دلیل تفاوت ظاهری مسیر حالت در صفحه‌ی فاز در شکل (۵-۳) (ب) اضافه شدن اختلال است که در رابطه‌ی (۵-۱۲) آمده است و از زمان ۰,۵ تا ۵ ثانیه به سیستم اعمال می‌شود. زمان نشست نیز از حدود ۱ ثانیه به حدود ۰,۴ ثانیه رسیده است. بنابراین، روش جدید هم برای بهبود نوسان در سیگنال کنترل و هم برای زمان نشست در حضور اختلال موثر واقع شده است.

همانطور که توضیح داده شد و نیز در شکل‌های (۵-۱)، (۵-۲) و (۵-۳) مشخص است، هر دو روش الگوریتم پرندهگان در حالت خارج از خط (روش دو) و روش جدید (روش سه) نسبت به روش بدون تنظیم (روش یک) که در آن  $\lambda = 5, \eta = 1$  است، بهینه می‌باشند. برای مقایسه‌ی بهتر می‌توان به سیگنال کنترل دقت کرد. در نگاه اول درمی‌یابیم که روش یک در حالت دائمی دارای نوسان سیگنال

## فصل پنجم: معرفی یک روش جدید برای کنترل بهینه تحت عدم قطعیت

کنترل است که دامنه‌ی آن بین  $\pm 1.5$  در حال سوئیچ است. در روش دو، می‌بینیم که دامنه بین  $\pm 0.5$  و در روش سه دامنه حدود  $\pm 0.05$  در حال نوسان است. یعنی در روش سه، نوسان بسیار ناچیز است و با یک لایه‌ی مرزی [۳۹] می‌توان آن را حذف کرد.

جدول (۲-۵) شاخص کارایی برای سیستم ساده

سیستم شاخص کارایی	سیستم بهینه توسط روش جدید	سیستم بهینه با الگوریتم پزندگان در حالت خارج از خط
$t_{start}=0$	$1,77 \cdot 10^5$	$1,68 \cdot 10^5$
$t_{start}=0,5$	۶۷۴,۹	۱۰۴۴,۶
$t_{start}=1$	۱۷۲,۰	۵۴۳,۵
$t_{start}=2$	۴۷,۴	۴۶۶
$t_{start}=3$	۹,۷	۴۴۰,۱

برای اینکه بهتر این موضوع را بررسی کنیم، در شاخص کارایی رابطه‌ی (۵-۱۳) که در جدول (۵-۱) آمده است، زمان شروع را  $0,5$ ،  $1$ ،  $2$  و  $3$  ثانیه قرار می‌دهیم. در ابتدای کار، سیگنال کنترل مقادیری با دامنه‌ی حدود  $100$  دارد که به دلیل اینکه به توان  $2$  می‌رسد، به  $10000$  نزدیک می‌شود. حال هرچه در انتها مقدار دامنه‌ی سیگنال کنترل کمتر باشد، به هیچ عنوان معلوم نخواهد شد. همانطور که در جدول (۲-۵) مشخص است، اگر چه برای زمان شروع صفر، روش دو بهتر بود، ولی از زمان  $0,5$  ثانیه به بعد، مقدار نوسان سیگنال کنترل در روش جدید بسیار کاهش داشته است که به دلیل اثرات نامطلوب نوسان این روش مطمئن‌تر است. دلیل این برتری را می‌توان در تخمین مناسب‌تر از اطلاعات حالت بر خط

دانست که در واقع به سطح لغزش بسیار نزدیکتر می‌شود. به این ترتیب می‌توان بهتر بودن نتایج را ناشی از این مورد دانست.

## ۴-۵ شبیه سازی و نتایج برای پیوما ۵۶۰

در ادامه ابتدا برای پیوما روابط را استخراج کرده و سپس نتایج شبیه‌سازی آمده است.

### ۱-۴-۵ پیاده سازی روش جدید برای ربات

در این بخش با توجه به ایده بخش ۳-۵ به بهینه سازی برای ربات پیوما ۵۶۰ می‌پردازیم. هدف، کنترل موقعیت عملگر نهایی است که در اینجا کنترل ردگیری است و مسیر مطلوب برای مکان عملگر نهایی به صورت شکل (۳-۴) و رابطه‌ی (۵-۴) است. برای ربات با توجه به روابط (۲-۴) و (۳-۴) داریم:

$$\begin{cases} \ddot{q}_1 = b_{11}u_1 + b_{12}u_2 + b_{13}u_3 + \hat{f}_1 + F_1 \\ \ddot{q}_2 = b_{21}u_1 + b_{22}u_2 + b_{23}u_3 + \hat{f}_2 + F_2 \\ \ddot{q}_3 = b_{31}u_1 + b_{32}u_2 + b_{33}u_3 + \hat{f}_3 + F_3 \end{cases} \quad (14-5)$$

برای لحظات  $k-1, k-2, k-3, k-4$  و نیز برای  $i = 1, 2, 3$  داریم:

$$\begin{cases} \ddot{q}_{i,(k)} = b_{i1,(k-1)}u_{1,(k-1)} + b_{i2,(k-1)}u_{2,(k-1)} + b_{i3,(k-1)}u_{3,(k-1)} + \hat{f}_{i,(k-1)} + F_{i,(k-1)} \\ \ddot{q}_{i,(k-1)} = b_{i1,(k-2)}u_{1,(k-2)} + b_{i2,(k-2)}u_{2,(k-2)} + b_{i3,(k-2)}u_{3,(k-2)} + \hat{f}_{i,(k-2)} + F_{i,(k-2)} \\ \ddot{q}_{i,(k-2)} = b_{i1,(k-3)}u_{1,(k-3)} + b_{i2,(k-3)}u_{2,(k-3)} + b_{i3,(k-3)}u_{3,(k-3)} + \hat{f}_{i,(k-3)} + F_{i,(k-3)} \\ \ddot{q}_{i,(k-3)} = b_{i1,(k-4)}u_{1,(k-4)} + b_{i2,(k-4)}u_{2,(k-4)} + b_{i3,(k-4)}u_{3,(k-4)} + \hat{f}_{i,(k-4)} + F_{i,(k-4)} \end{cases} \quad (15-5)$$

و با فرض:

$$\begin{cases} F_{i,(k-1)} \approx F_{i,(k-2)} \approx F_{i,(k-3)} \approx F_{i,(k-4)} \\ b_{ij,(k-1)} \approx b_{ij,(k-2)} \approx b_{ij,(k-3)} \approx b_{ij,(k-4)} \end{cases}, i = 1, 2, 3 \quad (16-5)$$

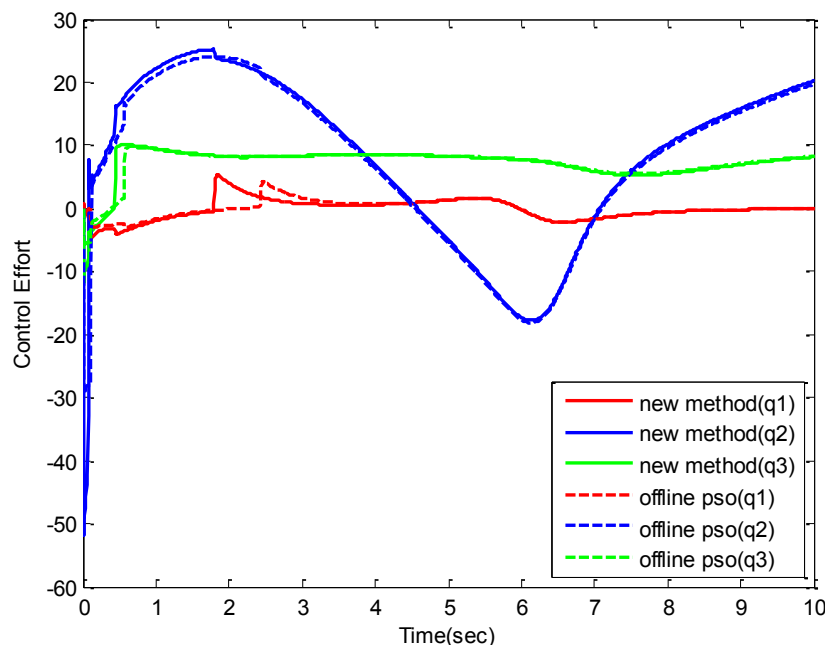
که در نهایت بدست می آید:

$$\begin{bmatrix} \ddot{q}_{i,(k)} - \hat{f}_{i,(k-1)} \\ \ddot{q}_{i,(k-1)} - \hat{f}_{i,(k-2)} \\ \ddot{q}_{i,(k-2)} - \hat{f}_{i,(k-3)} \\ \ddot{q}_{i,(k-3)} - \hat{f}_{i,(k-4)} \end{bmatrix} = \begin{bmatrix} u_{1,(k-1)} & u_{2,(k-1)} & u_{3,(k-1)} & 1 \\ u_{1,(k-2)} & u_{2,(k-2)} & u_{3,(k-2)} & 1 \\ u_{1,(k-3)} & u_{2,(k-3)} & u_{3,(k-3)} & 1 \\ u_{1,(k-4)} & u_{2,(k-4)} & u_{3,(k-4)} & 1 \end{bmatrix} * \begin{bmatrix} b_{ij,(k-1)} \\ b_{ij,(k-2)} \\ b_{ij,(k-3)} \\ b_{ij,(k-4)} \end{bmatrix}, i = 1, 2, 3 \quad (17-5)$$

که از حل این معادلات تخمینی برای حالت برخط بدست می آید. در ادامه نتایج را برای ۲ سیستم بررسی شده است.

## ۵-۴-۲ نتایج شبیه سازی

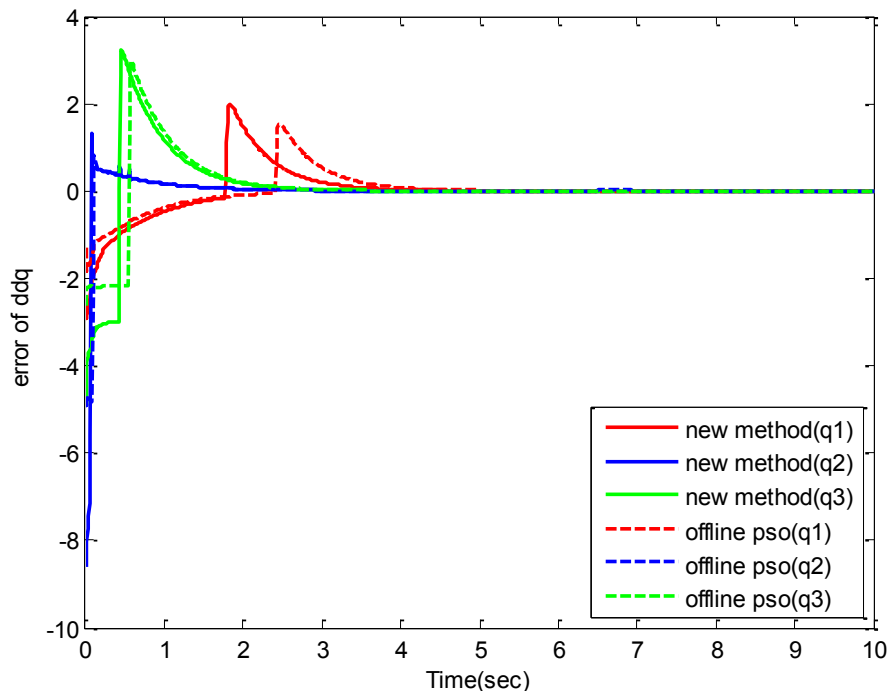
برای مقایسه، دو سیستم در نظر گرفته می شود. سیستم اول با الگوریتم پرندگان در حالت خارج از خط بهینه شده است و سیستم دوم با استفاده از روش جدید بهینه شده است. مقایسه برای سیستم غیر بهینه و بهینه با الگوریتم پرندگان در انتهای فصل سوم در شکل (۴-۴) آورده شد. برای مدل کردن تغییر پارامترها و اختلال برای مشابه سازی با حالت برخط، از توابع Sin استفاده شده است.



شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط

#### (الف) سیگنال کنترل

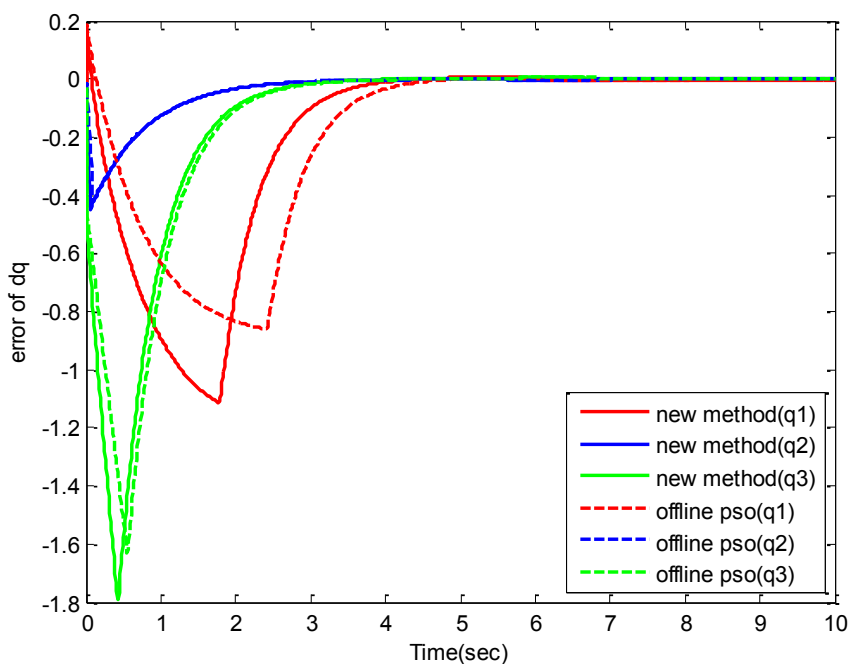
در شکل (۴-۵) (الف) سیگنال کنترل برای روش جدید و الگوریتم پرندگان در حالت خارج خط، مقایسه شده است. همانطور که مشخص است پس از ثانیه ۳، هر دو روش تقریباً عملکرد یکسانی نشان می‌دهند که به دلیل کمتر شدن خطا از یک مقدار مشخص است. برای گشتاور مفصل اول فراجاهش سریعتر رخ داده است و در حدود ۰٫۶ ثانیه زودتر از روش الگوریتم پرندگان خارج از خط به مقدار نهایی رسیده است که نکته‌ی مثبتی محسوب می‌شود. برای گشتاور مفصل دوم و سوم نیز مقدار فراجاهش کمی (در حد یک نیوتن-متر) افزایش داشته است. برای مقایسه باید شاخص کارایی را بررسی کرد که در ادامه خواهد آمد.



شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط

(ب) خطای  $\ddot{q}$

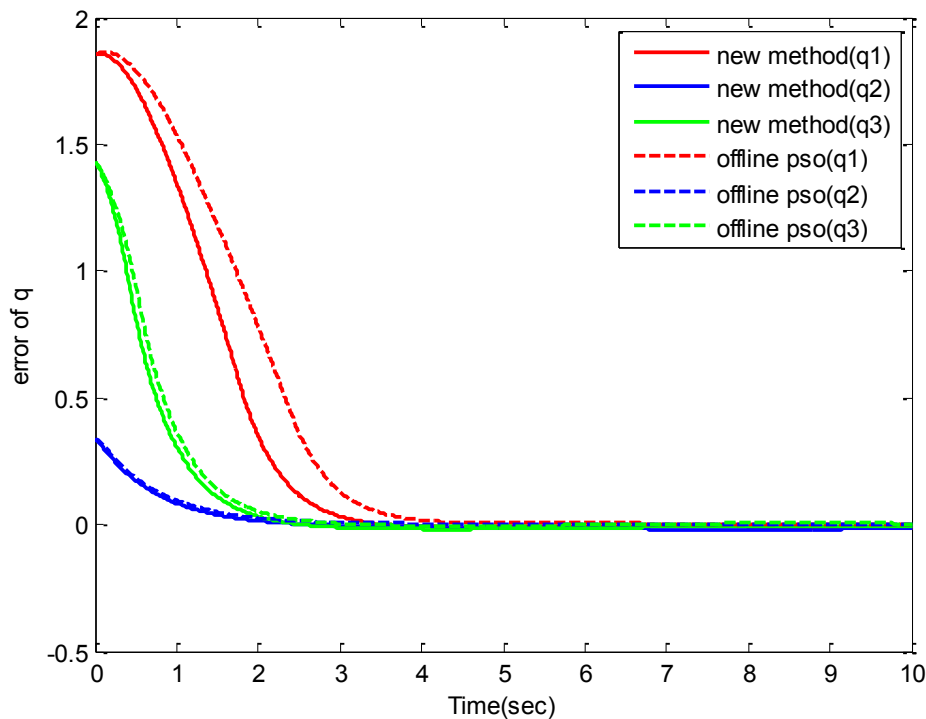
در شکل (۴-۵) (ب) خطای شتاب مفاصل آمده است. همانطور که مشخص است برای مفصل اول خطا سریعتر به مقدار نهایی خود رسیده است و از ۲,۵ ثانیه به حدود ۲ ثانیه رسیده است. برای مفاصل دوم و سوم تغییر چندانی در سرعت قابل مشاهده نیست.



شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط

### (ج) خطای $\dot{q}$

در شکل (۴-۵) (ج) خطای سرعت مفاصل آورده شده است. در اینجا نیز سرعت رسیدن به مقدار نهایی برای مفاصل دوم و سوم تغییر چندانی نداشته است. برای مفصل اول سرعت رسیدن به مقدار نهایی در حدود ۰,۷ ثانیه بهبود داشته است. هرچند که زمان نشست تغییری نداشته است ولی زمان اولین فروجهش کمتر و مقدار آن بیشتر شده است. در اینجا نیز سریعتر شدن حرکت مشهود است ولی مثل شکل‌های (۴-۵) (الف) و (ب) باید شاخص‌های کارایی را بررسی کرد.

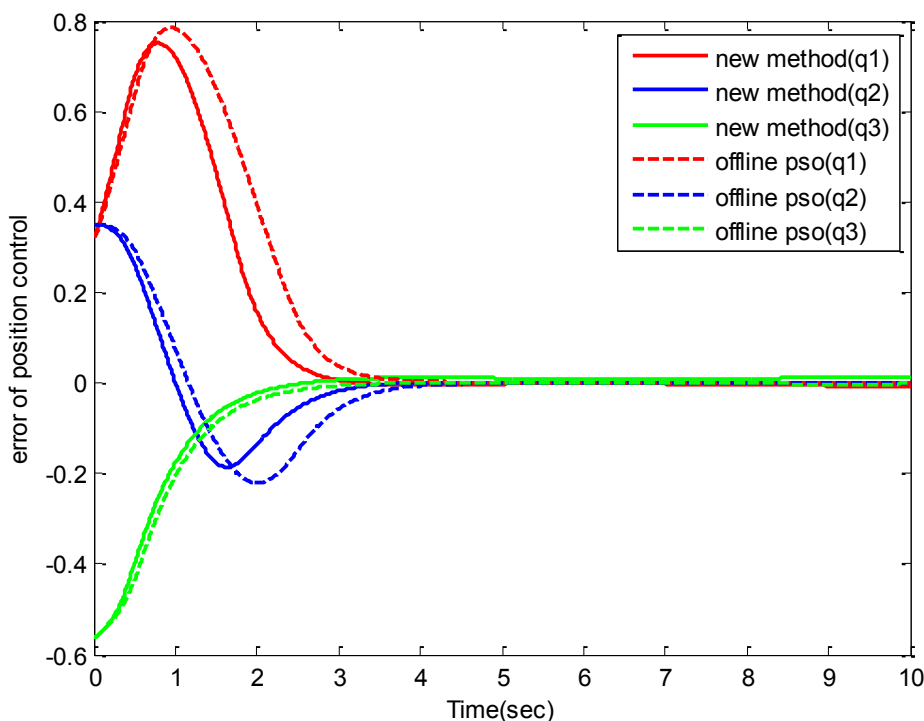


شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط

(د) خطای  $q$

در شکل (۴-۵) (د) خطای موقعیت مفاصل از مقدار مطلوب آمده است. سرعت رسیدن به مقدار مطلوب در حدود ۰,۷ ثانیه برای مفصل اول بهبود داشته است. برای مفاصل دوم و سوم نیز بهبود قابل مشاهده است ولی بسیار ناچیز است. در کل برای بررسی بیشتر باید مجدداً گفت که شاخص‌های کارایی باید بررسی شود که در ادامه خواهد آمد.





شکل (۴-۵) مقایسه روش پیشنهادی با الگوریتم پرندگان در حالت خارج از خط

(ه) خطای کنترل موقعیت مجری نهایی

در شکل (۴-۵) (ه) خطای کنترل موقعیت عملگر نهایی آورده شده است. در اینجا نیز زمان رسیدن به مقدار مطلوب (صفر شدن مقدار خطا) در حدود ۰,۳ ثانیه کاهش داشته است که مورد مطلوبی است. مقدار قدر مطلق فراجاهش و فروجهش نیز به ترتیب برای مفاصل اول و دوم کاهش داشته که مقدار مطلوبی است.

جدول (۳-۵) مقایسه زمان نشست و مجموع خطای ردگیری هر سه مفصل (شکل ۴-۵ د) برای ربات پیوما ۵۶۰

Settling Time (sec)	Tracking Error (m <sup>۲</sup> )	Perf.index Method
۳,۹۲۲	۱۱۷۱,۳ * ۱۰ <sup>-۴</sup>	Standard PSO
۳,۱۵۶	۹۱۵,۲۶۰۴ * ۱۰ <sup>-۴</sup>	New Method

## فصل پنجم: معرفی یک روش جدید برای کنترل بهینه تحت عدم قطعیت

جدول (۴-۵) مقایسه شاخص‌های کارایی برای ربات پیوما ۵۶۰

performance_index_tau ( $\sum \tau^2$ ) (N.m) <sup>۲</sup>	performance_index_e ( $\sum e^2(q)$ ) (rad) <sup>۲</sup>	Perf.index Method
۲,۸۵۰۴e+۰۰۶	۵,۸۷۳۸e+۰۰۳	Standard PSO
۳,۰۳۸۹e+۰۰۶	۴,۷۰۱۱e+۰۰۳	New Method

همانطور که در جدول (۳-۵) مشخص است، زمان نشست از حدود ۳,۹ ثانیه به حدود ۳,۲ ثانیه رسیده است که در کل بهبود داشته است. شاخص کارایی خطای حالت دائم نیز در حدود ۰,۰۲ (مجذور رادیان) بهبود داشته است که به دلیل اینکه به توان دو رسیده این مقدار شده است. دلیل این بهبود را می‌توان در بیشتر شدن ناچیز شاخص کنترل دانست که در جدول (۴-۵) مشخص است.

در جدول (۴-۵) شاخص کارایی خطای کل بهبود داشته و از حدود ۵۹۰۰ به ۴۷۰۰ (نیوتن-متر به توان دو) رسیده است ولی برای سیگنال کنترل در الگوریتم جدید بهبود چندانی مشاهده نمی‌شود و کمی نیز مقدار شاخص بیشتر است. دلیل این را می‌توان در تلاش سیگنال کنترل برای بهبود زمان نشست دانست که این انرژی بیشتر اولیه برای کاهش زمان نشست بوده است. پس از فاز اولیه همانطور که در شکل (۵-۴) (الف) نیز مشخص است، هر دو روش مقدارشان بر هم منطبق است.

دلیل بهبود شاخص کارایی خطای حالت دائم برای سیستم مثال بخش ۲-۵ در جدول (۴-۵) پس از زمان اولیه، بحث لرزش سیگنال کنترل است. در حالت دایمی، به دلیل اینکه تخمین بهتری از عدم قطعیت داشته‌ایم، محدوده‌ی لرزش کمتر شده‌است. لایه مرزی که در [۳۹] معرفی شده است، در اینجا اندازه‌ی حد ۰/۰۵ دارد ولی برای الگوریتم پرنندگان استاندارد حدود ۰/۵ است که در کل بهبود داشته

است. در واقع در لحظاتی، به عدم قطعیت نزدیکتر شده که به سطح لغزش نیز نزدیکتر می‌شویم و پس از آن، این فاصله‌ی کم تا سطح لغزش حفظ می‌شود.

درباره این روش به دو مطلب باید توجه شود. اول اینکه در همه لحظات شبیه‌سازی نمی‌توان تخمینی مناسب از عدم قطعیت داشت. گاهی  $c_2$ ،  $c_3$  و یا  $d(t)$  خارج از بازه‌ی مجازشان می‌روند که در آنجا می‌توان از مقدار الگوریتم پرنندگان استاندارد (در حالت نامی) استفاده کرد. ولی هرگاه که هر ۳ پارامتر مقدارشان در محدوده<sup>۱</sup> باشد، باید از جدول بهره استفاده کرد.

بحث دیگر اینکه سیستم در حال تغییر است. آیا دادن  $u_{i,opt}$  به سیستم  $s_i$  در بازه  $[T_i, T_{i+1}]$  و سپس، دادن  $u_{i+1,opt}$  به سیستم  $s_{i+1}$  در بازه‌ی  $[T_{i+1}, T_{i+2}]$  می‌تواند در کل سیستم را بهینه کند یا خیر. در پاسخ به چنین سوالی، ابتدا باید به نتایج شبیه‌سازی توجه کرد که در کل، بهبود را نشان می‌دهد ولی تا اثبات ریاضی نباشد نمی‌توان نتیجه‌ی قطعی گرفت. مورد دیگر اینکه، در حالت بر خط هیچ کس نمی‌تواند به طور قطع پیش‌بینی کند که سیستم چه رفتاری خواهد داشت. زیرا اگر اینگونه باشد، می‌توان در حالت خارج از خط تمامی محاسبات را انجام داد و در حالت بر خط به راحتی سیستم را کنترل بهینه کرد. چون ما در حصار زمان هستیم، فرض می‌شود که تا زمانی که سیستم تغییر نکرده، با سیستم  $s_i$  در بازه  $[T_i, T_{i+1}]$ ، روبرو هستیم و هرگاه سیستم تغییر کرد باید به  $s_{i+1}$  برویم و باز فرض می‌شود که برای سیستم  $s_{i+1}$  نیز تا  $\infty$  وضع به همین منوال است. البته با توجه به بحث پایداری تضمین شده توسط لیاپانوف و نیز بحث بهینگی می‌توان این مطلب را آنالیز ریاضی کرد که در کارهای بعدی دنبال خواهد شد.

<sup>۱</sup> In Range

فصل ششم  
نتیجه گیری

## ۶-۱ نتیجه‌گیری و جمع‌بندی

در این پایان‌نامه به مسئله‌ی کنترل مد لغزشی بهینه‌ی بازوی رباتیک در حضور اختلالات خارجی و عدم قطعیت پرداخته شد. همانطور که بیان شد، الگوریتم پرندگان به عنوان ابزار بهینه‌سازی، توانایی بهینه‌سازی در حالت دارای عدم قطعیت را ندارد. این الگوریتم نیاز به تابع هزینه‌ی مشخص و قطعی دارد تا بتواند با اجرای چندباره‌ی سیستم، مقدار بهینه را پیدا کند. در حالت دارای عدم قطعیت باید تدابیری اندیشید تا بتوان از این الگوریتم مجدداً بهره جست. در این راستا، ایده‌ی استفاده از تخمین عدم قطعیت در حالت برخط و ترکیب نتایج بدست آمده در حالت خارج خط مطرح شد. این ایده که برای پیاده‌سازی نیاز به محاسبات پیچیده‌ای ندارد، توانست سیستم دارای عدم قطعیت را بهینه‌سازی کند. نتایج شبیه‌سازی نشان داد که این روش در مقایسه با سیستم غیر بهینه و نیز سیستم بهینه شده با الگوریتم پرندگان در حالت خارج خط که تنها مدل نامی را بهینه می‌کند، برتری دارد. بحث دیگری که مشاهده شد، بحث لرزش سیگنال کنترل است که در سیستم بهینه شده توسط روش پیشنهادی، ابعاد لایه‌ی مرزی نیز کاهش داشته است. دلیل این امر را می‌توان در تخمین مناسب‌تر از عدم قطعیت و نزدیک شدن هرچه بیشتر به سطح لغزش دانست. نزدیک شدن هرچه بیشتر به سطح لغزش، باعث کاهش خطا و نیز کاهش انرژی سیگنال کنترل می‌شود باعث عملکرد بهتر و بهبود شاخص کارایی می‌شود. به عنوان مطالعه‌ی موردی همانطور که بیان شد، روش پیشنهادی برای ربات پیوما ۵۶۰ که رباتی پرکاربرد در صنعت و محیط آکادمیک است، پیاده‌سازی شد. نتایج شبیه‌سازی، حاکی از بهبود عملکرد ربات و کاهش پدیده‌ی لرزش در سیستم ربات است.

## ۲-۶ پیشنهادات

در این با توجه به ایده‌ی مطرح شده در پایان‌نامه، که باعث بهبود شاخص کارایی و تخمین بهتر عدم قطعیت شد، به منظور ارائه‌ی راهکار برای محققین بعدی، پیشنهادات زیر ارائه می‌شود:

- برای یک سیستم دلخواه که در حالت خارج از خط با الگوریتم پرندگان یا دیگر روش‌های بهینه‌سازی، بهینه شده است، روش پیشنهادی آنالیز ریاضی شود تا بررسی شود که آیا این روش برای یک سیستم دلخواه نیز بهینه است.
- با توجه به اینکه در برخی موارد با تغییر اندک در پارامترها، سیستم کنترل دستخوش تغییر می‌شود که گاهی ممکن است این تغییرات ناگهانی باشد، مکانیزمی برای تغییرات نرمتر سیگنال کنترل ارائه شود.
- با ترکیب این روش با دیگر روش‌های هوشمند مانند منطق فازی، مدلی جهت پیش‌بینی و بیان عدم قطعیت بدست آورد. این مدل می‌تواند درک بهتری از عدم قطعیت را فراهم آورد و نیز می‌تواند راهگشای بهتری برای بهینه‌سازی در حالت برخط برای برخی فرایندهای تکرارپذیر باشد.

- [1] Srinivasan, B., Bonvin, D., Visser, E., & Palanki, S. (2003). "Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty". *Computers & Chemical Engineering*, 27(1), 27–44.
- [2] Hansen, N., Niederberger, A. S., Guzzella, L., & Koumoutsakos, P. (2009). "A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion". *Evolutionary Computation, IEEE Transactions on*, 13(1), 180–197.
- [3] Fouad, Allouani, Djamel Boukhetela, and Fares Boudjema(2013). "Decentralized Sliding Mode Controller Based on Genetic algorithm and a Hybrid approach for Interconnected Uncertain Nonlinear Systems." *International Journal of Control and Automation*, Vol.6 , No. 1, 61–86.
- [4] Lin, F., & Brandt, R. D. (1998). "An optimal control approach to robust control of robot manipulators". *Robotics and Automation, IEEE Transactions on*, 14(1), 69–77.
- [5] Lin, F., Zhang, W., & Brandt, R. D. (1999). "Robust hovering control of a PVTOL aircraft". *Control Systems Technology, IEEE Transactions on*, 7(3), 343–351.
- [6] Ker-Wei, Y., & Shang-Chang, H. (2006, October). "An application of AC servo motor by using particle swarm optimization based sliding mode controller". In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on* (Vol. 5, pp. 4146–4150). IEEE.
- [7] Yuhua, X., Chongwei, Z., Wei, B., & Lin, T. (2009, May). "Dynamic Sliding Mode Controller Based on Particle Swarm Optimization for Mobile Robot's Path Following". In *Information Technology and Applications, 2009. IFITA'09. International Forum on* (Vol. 1, pp. 257–260). IEEE.
- [8] Serbencu, A. E., Serbencu, A., Cernega, D. C., & Minzu, V. (2010, July). "Particle swarm optimization for the Sliding Mode controller parameters". In *Control Conference (CCC), 2010, 29th Chinese* (pp. 1859–1864). IEEE.
- [9] Chen, Z., Meng, W., Wang, Z., & Zhang, J. (2008, December). "Sliding Mode Variable Structure Control Based on Particle Swarm Optimization". In *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on* (Vol.1, pp. 692–696). IEEE.
- [10] Lu, Z., Shieh, L. S., Chen, G., & Coleman, N. P. (2005). "Simplex sliding mode control for nonlinear uncertain systems via chaos optimization". *Chaos, Solitons & Fractals*, 23(3), 747–755.

- [11] Jin, Y., & Branke, J. (2005). "Evolutionary optimization in uncertain environments-a survey". *Evolutionary Computation, IEEE Transactions on*, 9(3), 303–317.
- [12] Clerc, M. (2006). "Particle swarm optimization" (Vol. 243). London: Iste.
- [13] Richards, M., & Ventura, D. (2004, July). "Choosing a starting configuration for particle swarm optimization". In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on* (Vol. 3, pp. 2309–2312). IEEE.
- [14] Wong, T. T., Luk, W. S., & Heng, P. A. (1997). "Sampling with Hammersley and Halton points". *Journal of graphics tools*, 2(2), 9–24.
- [15] Clerc, M. (2006). "Confinements and Biases in Particle Swarm Optimisation". Available at: <http://hal.archives-ouvertes.fr/hal-00122799>
- [16] Cortes, J. (2008). Discontinuous dynamical systems. *Control Systems, IEEE*, 28(3), 36–73.
- [17] Brockett R. W. (1983), "Asymptotic stability and feedback stabilization," in *Geometric Control Theory* (R. W. Brockett, R. S. Millman, and H. J. Sussmann, eds.), Birkhäuser Verlag (Boston, MA), pp.181–191.
- [18] Sontag, E. D. (1998). *Mathematical control theory: deterministic finite dimensional systems* (Vol. 6). Springer.
- [19] Bacciotti, A., & Rosier, L. (2005). *Liapunov functions and stability in control theory*. Springer.
- [20] Byrnes, C. I., & Isidori, A. (1991). "Asymptotic stabilization of minimum phase nonlinear systems". *Automatic Control, IEEE Transactions on*, 36(10), 1122–1137.
- [21] Filippov, A. F., & Arscott, F. M. (Eds.). (1988). "Differential Equations with Discontinuous Righthand Sides": *Control Systems* (Vol.18). Springer.
- [22] Krasovskii, N. N. (1963). "Stability of motion: applications of Lyapunov's second method to differential systems and equations with delay". *Stanford university press*.
- [23] Krasovskii N. N. and Subbotin A. I. (1988), "Game-Theoretical Control Problems". New York: Springer Verlag.
- [24] Hermes H. (1967), "Discontinuous vector fields and feedback control," in *Differential Equations and Dynamical Systems*, New York: Academic Press pp.155–165.



- [25] Hájek, O. (1979). "Discontinuous differential equations, I". *Journal of Differential Equations*, 32(2), 149–170.
- [26] Sentis R. (1978), "Equations différentielles `a second membre mesurable," *Boll. Unione Matematica italiana*, vol. 5, no. 15-B, pp. 724–742.
- [27] Bacciotti A. (2004), "Some remarks on generalized solutions of discontinuous differential equations," *International Journal of Pure and Applied Mathematics*, vol. 10, no. 3, pp. 257–266.
- [28] Ambrosio L. (1988), "A lower closure theorem for autonomous orientor fields," *Proc. R. Soc. Edinb*, vol. A110, no. 3/4, pp. 249–254.
- [29] DeCarlo, R. A., Zak, S. H., & Matthews, G. P. (1988). Variable structure control of nonlinear multivariable systems: a tutorial. *Proceedings of the IEEE*, 76(3), 212–232.
- [30] Zinober, A. S. I., Ei-Ghezawi, O. M. E., & Billings, S. A. (1982). "Multivariable variable-structure adaptive model-following control systems". In *Control Theory and Applications, IEE Proceedings D* (Vol.129, No.1, p. 6). IET.
- [31] Lee, H., & Utkin, V. I. (2007). "Chattering suppression methods in sliding mode control systems". *Annual Reviews in Control*, 31(2), 179–188.
- [32] Utkin, V. I. (1992). "Sliding modes in control and optimization" (Vol.116). Berlin: Springer-Verlag.
- [33] Drakunov S. V., Utkin V. I. (1992), "Sliding mode control in dynamic systems", *Int. J. of Control*, 55(4), pp.1029–1037.
- [34] Emel'yanov S. V., Korovin S. V. and Levantovsky L. V. (1993), "Higher order sliding modes in control system", *Differential Equations*, 29(11), pp.1627–1647.
- [35] Levant, A. (2003). "Higher-order sliding modes, differentiation and output-feedback control". *International journal of Control*, 76(9–10), 924–941.
- [36] Bartolini, G., Pisano, A., Punta, E., & Usai, E. (2003). "A survey of applications of second-order sliding mode control to mechanical systems". *International Journal of Control*, 76(9–10), 875–892.
- [37] Floquet, T., Barbot, J. P., & Perruquetti, W. (2003). "Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems". *Automatica*, 39(6), 1077–1083.

[38] Defoort, M., Floquet, T., Kokosy, A., & Perruquetti, W. (2009). "A novel higher order sliding mode control scheme". *Systems & Control Letters*, 58(2), 102–108.

[39] Slotine, J. J. E., & Li, W. (1991). "Applied nonlinear control" (Vol.199, No.1). New Jersey: Prentice hall.

[40] Piltan, F., Emamzadeh, S., Hivand, Z., Shahriyari, F., & Mirzaei, M. (2012). "PUMA-560 Robot Manipulator Position Sliding Mode Control Methods Using MATLAB/SIMULINK and Their Integration into Graduate/Undergraduate Nonlinear Control, Robotics and MATLAB Courses". *International Journal of Robotics and Automation*, 3(3), 106–150.

[۴۱] اسپانگ م. دبلیو. ، ویدیا ساگار ام. ، (۱۳۸۵) ، "کنترل و دینامیک ربات" ، ترجمه: فاتح م.م. ، دانشگاه صنعتی شاهرود، شاهرود.

[42] Lee, C. G., & Ziegler, M. (1984). "Geometric approach in solving inverse kinematics of PUMA robots". *Aerospace and Electronic Systems, IEEE Transactions on*, (6), 695–706.

[43] Rockafellar, R. T., & Wets, R. J. B. (1991). "Scenarios and policy aggregation in optimization under uncertainty". *Mathematics of operations research*, 16(1), 119–147.

[44] Åström, K. J., & Wittenmark, B. (2008). "Adaptive control". Courier Dover Publications.

## پیوست

تابع  $atan^2$  بدین صورت تعریف می‌شود:

$$\theta = atan^2\left(\frac{y}{x}\right) = \begin{cases} 0 \leq \theta \leq 90 & (\text{for } +x \text{ and } +y) \\ 90 \leq \theta \leq 180 & (\text{for } -x \text{ and } +y) \\ -180 \leq \theta \leq -90 & (\text{for } -x \text{ and } -y) \\ -90 \leq \theta \leq 0 & (\text{for } +x \text{ and } -y) \end{cases} \quad (\text{پ-۱})$$

مورد دیگر که باید ذکر شود همان متغیرهای بازو و آرنج است که برای بازوی چپ مقدار ۱- و برای بازوی راست مقدار ۱+ است. برای آرنج نیز اگر بالای بازو باشد ۱+ و اگر پایین بازو باشد ۱- قرار می‌دهیم.

پ-۱-۱ یافتن  $\theta_1$ 

برای یافتن  $\theta_1$  با استفاده از پارامترهای جدول (۱-۴) و رابطه (۱-۴) بدست می‌آید:

$$\theta_1 = atan^2 \left\{ \frac{-ARM \cdot p_y \sqrt{p_x^2 + p_y^2 - d_r^2} - p_x d_r}{-ARM \cdot p_x \sqrt{p_x^2 + p_y^2 - d_r^2} + p_y d_r} \right\} ; -\pi \leq \theta_1 \leq +\pi \quad (\text{پ-۲})$$

پ-۱-۲ یافتن  $\theta_2$ 

ابتدا تعریف می‌کنیم:

$$\sin(\alpha) = -\frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_r^2}} \quad (\text{پ-۳})$$

و نیز بدست می‌آید:

$$\cos(\alpha) = -\frac{ARM \cdot \sqrt{p_x^2 + p_y^2 - d_r^2}}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_r^2}} \quad (\text{پ-۴})$$

تعریف می‌کنیم:

$$\cos(\beta) = \frac{p_x^2 + p_y^2 + p_z^2 + a_r^2 - d_r^2 - a_r^2 - d_f^2}{2a_r \sqrt{p_x^2 + p_y^2 + p_z^2 - d_r^2}} \quad (\text{پ-۵})$$

و نیز مشخص است که:

$$\sin(\beta) = \sqrt{1 - \cos(\beta)^2} \quad (\text{پ-۶})$$

حال با توجه به روابط (پ-۳)، (پ-۴)، (پ-۵) و (پ-۶) و نیز متغیرهای آرنج و بازو داریم:

$$\theta_r = \operatorname{atan} \left\{ \frac{\cos(\beta) \sin(\alpha) + (ARM.ELBOW) \sin(\beta) \cos(\alpha)}{\cos(\alpha) \cos(\beta) - (ARM.ELBOW) \sin(\alpha) \sin(\beta)} \right\} ; -\pi \leq \theta_r \leq +\pi \quad (\text{پ-۷})$$

### پ-۱-۳ یافتن $\theta_3$

در اینجا نیز مثل حالت قبل ابتدا چهار تعریف داریم:

$$\cos(\varphi) = \frac{a_r^2 - a_r^2 + d_f^2 + p_x^2 + p_y^2 + p_z^2 - d_r^2}{2a_r \sqrt{a_r^2 + d_f^2}} \quad (\text{پ-۸})$$

$$\sin(\varphi) = ARM \cdot ELBOW \sqrt{1 - \cos(\varphi)^2} \quad (\text{پ-۹})$$

$$\sin(\gamma) = \frac{d_r}{\sqrt{a_r^2 + d_r^2}} \quad (\text{پ-۱۰})$$

$$\cos(\gamma) = \frac{|a_r|}{\sqrt{a_r^2 + d_r^2}} \quad (\text{پ-۱۱})$$

اکنون داریم:

$$\sin(\theta_r) = \cos(\gamma) \sin(\varphi) - \sin(\gamma) \cos(\varphi) \quad (\text{پ-۱۲})$$

$$\cos(\theta_r) = \cos(\gamma) \cos(\varphi) + \sin(\gamma) \sin(\varphi) \quad (\text{پ-۱۳})$$

و در نهایت بدست می‌آید:

$$\theta_r = \text{atan} \left\{ \frac{\sin(\theta_r)}{\cos(\theta_r)} \right\} ; -\pi \leq \theta_r \leq +\pi \quad (\text{پ-۱۴})$$

برای یافتن زوایا باید توجه کرد که به دلیل محدود بودن اندازه Sin و Cos به  $\pm 1$  در کل دو شرط در

دست داریم:

$$0.7711 = a_r^z + a_r^z + d_r^z + d_r^z \geq p_x^z + p_y^z + p_z^z \geq d_r^z = 0.223 \quad (\text{پ-۱۵})$$

$$p_x^z + p_y^z \geq d_r^z = 0.223$$

توجه شود برای مسئله ردگیری، مقادیر مطلوب عملگر نهایی از قبل معلوم می‌شود. برای این که ربات در ناحیه غیرمجاز نرود، بهتر است که ابتدا با استفاده از سینماتیک وارون، مقادیر مطلوب برای متغیرهای مفاصل را یافت و سپس به جای عملگر نهایی، هدف را ردگیری مقابله مطلوب برای متغیرهای مفاصل قرار داد. در این تحقیق از این روش استفاده شده است.

## **Abstract**

The following thesis scrutinizes the problem of optimal sliding mode control of a robot manipulator under uncertainty. Minimizing the control signal and tracking error are two of the most important goals in control systems, which should be taken into consideration. The novelty of this research is to combine the results of PSO algorithm in an offline mode with the estimated values of uncertainties in order to optimize the control signal parameters. The major advantage of the proposed method over other traditional ones is its online nature. In an online mode, one can have a better estimation of actual model if better bounds of uncertainty would be available. Therefore, the Particle Swarm Optimization technique will be applicable for the actual dynamical system. Another merit of the proposed method is its fewer online computations. This concerning the fact that majority of online computations would be already done in an offline mode and available in a table for the controller. A case study for PUMA 560 is performed. This Robot is nonlinear, multi input-multi output systems with uncertain parameters, which should be control via Robust Control approaches such as Sliding Mode Control. The biggest barrier in the way of optimization of these systems is the existence of disturbances and uncertainties which varies the actual system from the nominal one. This study is taking advantage of both online estimation of uncertainty and a gain table optimized by PSO in an offline mode in order to improve the control signal parameters. Simulation results show improvement and the superiority of the proposed method over the traditional Sliding Mode Control approach.

**Keywords:** Sliding Mode Control, Particle Swarm Optimization, Deterministic Uncertainty, Optimal Control, Robot Manipulators, PUMA 560



**Shahrood University of Technology**

**Faculty of Electrical and Robotic Engineering**

**Optimal Sliding Mode Control of a Robot under Uncertainty  
Using PSO Algorithm**

**S. Mohammad Hashemzadeh**

**Supervisor:**

**Dr. M. Hadad Zarif**

**Advisor:**

**Dr. M. M. Fateh**

Sep 2013