

اللَّهُ الرَّحْمَنُ الرَّحِيمُ



دانشکده : برق و رباتیک

گروه : کنترل

کنترل بهینه غیرخطی روی بازوی ماهر ربات هنرمند دو لینکی

دانشجو : فاطمه سلیمان نوری

استاد راهنما :

دکتر محمد حداد ظریف

استاد مشاور :

دکتر محمد مهدی فاتح

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

ماه و سال انتشار :

بهمن ۱۳۹۲

دانشگاه صنعتی شاهرود

دانشکده : برق و رباتیک

گروه : کنترل

پایان نامه کارشناسی ارشد آقای / خانم

تحت عنوان:

در تاریخ توسط کمیته تخصصی 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

پدر بزرگوار و مادر مهربانم

آن دو فرشته ای که از خواسته هایشان گذشتند ، سختی ها را به جان خریدند و خود را

سپر بلای مشکلات و ناملایمات کردند تا من به جایگاهی که اکنون در آن ایستاده ام

برسم .

سپاس حضرت خداوندگاری

را که به من این فرصت را داد تا به این مرحله از علم رسیده و از هیچ محبتی دریغ نکرد و در تمام مراحل زندگی مرا قوت قلب بود .

نمی توانم معنایی بالاتر از تقدیر و تشکر بر زبانم جاری سازم و سپاس خود را در

وصف استادان خویش

آقای دکتر محمد حداد ظریف

و

آقای دکتر محمد مهدی فاتح

آشکار نمایم ، که هر چه گویم و سرایم ، کم گفته ام .

تعهد نامه

اینجانب فاطمه سلیمان نوری دانشجوی دوره کارشناسی ارشد رشته برق کنترل دانشکده برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه کنترل بهینه غیر خطی روی بازوی ماهر ربات هنرمند دو لینیکی تحت راهنمایی آقای دکتر حداد ظریف .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 .

تاریخ
امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد. 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 پایان نامه مسئله کنترل بهینه‌ی غیرخطی بر روی بازوی ماهر ربات دو لینکی بررسی شده است. بیشترین تفاوت بین ربات‌ها، در قسمت‌های مکانیکی و نحوه کنترل آن‌ها است. حداقل کردن خطای ردگیری و کنترل مسیر از اهداف مهم و قابل توجه درسیستم‌های کنترل است. در این تحقیق، الگوریتم بهینه‌سازی اجتماع پرندگان را برای طراحی کنترل کننده بهینه به کار بردیم. به کاربردن الگوریتم بهینه‌سازی پرندگان درمسائل بهینه‌سازی برای سیستم‌های بسیار پیچیده، مفید است زیرا الگوریتم پرندگان مستقل از مدل است و پیاده‌سازی آن بسیار ساده است. الگوریتم بهینه‌سازی پرندگان براساس بهینه‌سازی پارامترهای کنترل، کنترل بهینه را نتیجه می‌دهد. در این روش مقادیر بهینه پارامترهای کنترل به وسیله مینیمم کردن تابع هزینه تعیین می‌شود، معمولاً انتگرال مجذور خطا به عنوان شاخص عملکرد در نظر گرفته می‌شود. در این تحقیق، عملکرد الگوریتم‌های بهینه‌سازی پرندگان از قبیل الگوریتم بهبود یافته، الگوریتم بهینه‌سازی با استفاده از فازی، الگوریتم ضریب اینرسی کاهش‌ی خطی، الگوریتم ضریب اینرسی کاهش‌ی غیرخطی، الگوریتم ضریب اینرسی دینامیکی از نظر دقت و سرعت همگرایی مقایسه شده است. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم بهبود یافته با استفاده از فازی عملکرد بهتری را در کنترل مسیر بازوی ماهر ربات از بقیه الگوریتم‌های ارائه داده است.

کلمات کلیدی: الگوریتم بهینه‌سازی پرندگان، بازوی ماهر ربات، بهینه‌سازی

فهرست

۱	فصل اول: کلیات	۱
۲	مقدمه	۱-۱
۲	رباتیک	۲-۱
۵	انواع ربات	۱-۲-۱
۱۰	مروری بر کارهای انجام شده	۳-۱
۱۱	اهداف پایان نامه	۴-۱
۱۳	طرح کلی پایان نامه	۵-۱
۱۵	فصل دوم: دینامیک ربات	۱۵
۱۶	معادله دینامیکی بازوی رباتیک	۱-۲
۱۷	انرژی جنبشی برای ربات با n - رابط	۱-۱-۲
۱۸	انرژی پتانسیل برای ربات با n - رابط	۲-۱-۲
۱۹	معادلات حرکت	۲-۲
۲۰	کنترل ربات بر مبنای مدل	۳-۲
۲۰	طراحی کنترل کننده	۴-۲
۲۲	نتایج شبیه سازی	۵-۲

۲۷ فصل سوم: سیستم‌های فازی	
۲۸ سیستم فازی	۱-۳
۲۹ چرا از منطق فازی استفاده می‌کنیم؟	۲-۳
۲۹ مجموعه‌های فازی	۳-۳
۳۰ تابع عضویت	۴-۳
۳۱ انواع استنتاج فازی	۵-۳
۳۱ روش ممدانی	۱-۵-۳
۳۴ روش سوگنو	۲-۵-۳
۳۵ فصل چهارم: الگوریتم بهینه‌سازی اجتماع پرندگان	
۳۶ مطالعه رفتار پرندگان و ایده اولیه بهینه‌سازی اجتماع پرندگان	۱-۴
۳۷ الگوریتم بهینه‌سازی اجتماع پرندگان	۲-۴
۴۰ پیاده‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان	۳-۴
۴۲ مزیت‌های الگوریتم بهینه‌سازی اجتماع پرندگان	۴-۴
۴۲ مقادیر پارامترهای الگوریتم	۵-۴
۴۳ پارامترهای اصلی الگوریتم بهینه‌سازی پرندگان	۶-۴
۴۳ اندازه گروه	۱-۶-۴
۴۴ تعداد تکرارها	۲-۶-۴
۴۴ ضرایب شتاب	۳-۶-۴
۴۵ ضریب اینرسی	۴-۶-۴

۴۶.....	معیار توقف در الگوریتم بهینه‌سازی اجتماع پرندگان	۷-۴
۴۷.....	شاخص عملکرد الگوریتم بهینه‌سازی اجتماع پرندگان	۸-۴
۴۸.....	یافتن فرمول نحوه حرکات ذرات	۹-۴
۴۹.....	تجزیه و تحلیل الگوریتم	۱۰-۴
۴۹.....	جهت و اندازه گام تصادفی الگوریتم بهینه‌سازی انبوه ذرات	۱-۱۰-۴
۵۰.....	همگرایی الگوریتم بهینه‌سازی اجتماع پرندگان	۲-۱۰-۴
۵۲.....	پایداری الگوریتم بهینه‌سازی انبوه ذرات	۳-۱۰-۴
۵۳.....	بهبود الگوریتم بهینه‌سازی اجتماع پرندگان	۱۱-۴
۵۵.....	الگوریتم بهینه‌سازی اجتماع پرندگان ترکیبی	۱۲-۴
۵۷.....	فصل پنجم: بهینه‌سازی توسط الگوریتم بهینه‌سازی پرندگان	
۵۸.....	بهینه‌سازی	۱-۵
۵۹.....	انواع مسائل بهینه‌سازی	۲-۵
۵۹.....	بهینه‌سازی با سعی و خطا و بهینه‌سازی روی تابع	۱-۲-۵
۵۹.....	بهینه‌سازی تک بعدی و بهینه‌سازی چند بعدی	۲-۲-۵
۵۹.....	بهینه‌سازی پویا و بهینه‌سازی ایستا	۳-۲-۵
۶۰.....	بهینه‌سازی گسسته و بهینه‌سازی پیوسته	۴-۲-۵
۶۰.....	بهینه‌سازی کمینه‌جو و بهینه‌سازی تصادفی	۵-۲-۵
۶۱.....	الگوریتم‌های بهینه‌سازی	۳-۵
۶۲.....	طراحی کنترل‌کننده	۴-۵

الگوریتم بهینه‌سازی اجتماع پرندگان.....	۵-۵	۶۲
الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی کاهش خطی...۶۳	۱-۵-۵	۶۳
الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی کاهش غیرخطی ۶۳	۲-۵-۵	۶۳
الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی دینامیکی.....۶۴	۳-۵-۵	۶۴
الگوریتم بهینه‌سازی اجتماع پرندگان بهبودیافته.....۶۵	۴-۵-۵	۶۵
الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از روش پیشنهادی.....۶۵	۵-۵-۵	۶۵
فصل ششم: نتایج شبیه‌سازی		
شبیه‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از ضریب اینرسی	۱-۶	۶۷
کاهش خطی.....		۶۹
شبیه‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از ضریب اینرسی	۲-۶	
کاهش غیرخطی.....		۷۳
شبیه‌سازی الگوریتم بهینه‌سازی پرندگان با ضریب اینرسی دینامیکی.....۷۷	۳-۶	۷۷
شبیه‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان بهبود یافته.....۸۱	۴-۶	۸۱
شبیه‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از منطق فازی.....۸۵	۵-۶	۸۵
فازی‌ساز.....	۱-۵-۶	۸۵
قوانین فازی.....	۲-۵-۶	۸۷
استنتاج فازی.....	۳-۵-۶	۸۸
غیرفازی‌ساز.....	۴-۵-۶	۸۸
مقایسه نتایج.....	۶-۶	۹۳

۹۳.....	مقایسه الگوریتم‌ها	۱-۶-۶
۹۸.....	مقایسه بین سیستم بهینه و غیر بهینه	۲-۶-۶
۱۰۱.....	فصل هفتم: نتایج و پیشنهادها	
۱۰۲.....	نگرش کلی	۱-۷
۱۰۲.....	نتیجه‌گیری	۲-۷
۱۰۳.....	پیشنهادات	۳-۷
۱۰۵.....	مراجع	

فهرست شکل‌ها

- شکل ۱-۱: ربات بازوی صنعتی ۵
- شکل ۲-۱: ربات متحرک از نوع چرخ‌دار ۶
- شکل ۳-۱: ربات متحرک از نوع پادار ۷
- شکل ۴-۱: ربات متحرک از نوع پرنده ۷
- شکل ۵-۱: ربات متحرک از نوع هایبرید ۸
- شکل ۱-۲: بلوک دیاگرام شبیه‌سازی بازوی ماهر ربات ۲۴
- شکل ۲-۲: خطاهای ردیابی ۲۵
- شکل ۳-۲: گشتاورهای کنترل ۲۵
- شکل ۱-۳: ساختار اصلی سیستم‌های فازی با فازی‌ساز و غیرفازی‌ساز [۳۷] ۳۲
- شکل ۱-۴: فرایند جستجو ۴۰
- شکل ۲-۴: فلوچارت الگوریتم بهینه‌سازی پرندگان ۴۱
- شکل ۱-۶: بلوک دیاگرام شبیه‌سازی بازوی ماهر بهینه شده ۶۸
- شکل ۲-۶: پارامتر Λ_1 در الگوریتم LDW-PSO ۶۹
- شکل ۳-۶: پارامتر Γ_1 در الگوریتم LDW-PSO ۷۰
- شکل ۴-۶: پارامتر Γ_2 در الگوریتم LDW-PSO ۷۰
- شکل ۵-۶: پارامتر Γ_3 در الگوریتم LDW-PSO ۷۱
- شکل ۶-۶: پارامتر Γ_4 در الگوریتم LDW-PSO ۷۱
- شکل ۷-۶: پارامتر K_1 در الگوریتم LDW-PSO ۷۲
- شکل ۸-۶: تابع هزینه در الگوریتم LDW-PSO ۷۲

۷۳	شکل ۹-۶: پارامتر Λ_1 در الگوریتم NDW-PSO
۷۴	شکل ۱۰-۶: پارامتر Γ_1 در الگوریتم NDW-PSO
۷۴	شکل ۱۱-۶: پارامتر Γ_2 در الگوریتم NDW-PSO
۷۵	شکل ۱۲-۶: پارامتر Γ_3 در الگوریتم NDW-PSO
۷۵	شکل ۱۳-۶: پارامتر Γ_4 در الگوریتم NDW-PSO
۷۶	شکل ۱۴-۶: پارامتر K_1 در الگوریتم NDW-PSO
۷۶	شکل ۱۵-۶: تابع هزینه در الگوریتم NDW-PSO
۷۷	شکل ۱۶-۶: پارامتر Λ_1 در الگوریتم DIW-PSO
۷۸	شکل ۱۷-۶: پارامتر Γ_1 در الگوریتم DIW-PSO
۷۸	شکل ۱۸-۶: پارامتر Γ_2 در الگوریتم DIW-PSO
۷۹	شکل ۱۹-۶: پارامتر Γ_3 در الگوریتم DIW-PSO
۷۹	شکل ۲۰-۶: پارامتر Γ_4 در الگوریتم DIW-PSO
۸۰	شکل ۲۱-۶: پارامتر K_1 در الگوریتم DIW-PSO
۸۰	شکل ۲۲-۶: تابع هزینه در الگوریتم DIW-PSO
۸۱	شکل ۲۳-۶: پارامتر Λ_1 در الگوریتم IPSO
۸۲	شکل ۲۴-۶: پارامتر Γ_1 در الگوریتم IPSO
۸۲	شکل ۲۵-۶: پارامتر Γ_2 در الگوریتم IPSO
۸۳	شکل ۲۶-۶: پارامتر Γ_3 در الگوریتم IPSO
۸۳	شکل ۲۷-۶: پارامتر Γ_4 در الگوریتم IPSO
۸۴	شکل ۲۸-۶: پارامتر K_1 در الگوریتم IPSO
۸۴	شکل ۲۹-۶: تابع هزینه در الگوریتم IPSO
۸۵	شکل ۳۰-۶: سیستم فازی

- شکل ۳۱-۶: توابع عضویت مثلثی ورودی ۸۶
- شکل ۳۲-۶: توابع عضویت مثلثی خروجی ۸۶
- شکل ۳۳-۶: سطح سه بعدی دو ورودی همراه یکی از خروجی ها ۸۷
- شکل ۳۴-۶: پارامتر Λ_1 در الگوریتم Fuzzy-PSO ۸۹
- شکل ۳۵-۶: پارامتر Γ_1 در الگوریتم Fuzzy-PSO ۹۰
- شکل ۳۶-۶: پارامتر Γ_2 در الگوریتم Fuzzy-PSO ۹۰
- شکل ۳۷-۳: پارامتر Γ_3 در الگوریتم Fuzzy-PSO ۹۱
- شکل ۳۸-۶: پارامتر Γ_4 در الگوریتم Fuzzy-PSO ۹۱
- شکل ۳۹-۶: پارامتر K_1 در الگوریتم Fuzzy-PSO ۹۲
- شکل ۴۰-۶: تابع هزینه در الگوریتم Fuzzy-PSO ۹۲
- شکل ۴۱-۶: پارامتر Λ_1 در الگوریتم های مختلف ۹۳
- شکل ۴۲-۶: پارامتر Γ_1 در الگوریتم های مختلف ۹۴
- شکل ۴۳-۶: پارامتر Γ_2 در الگوریتم های مختلف ۹۴
- شکل ۴۴-۶: پارامتر Γ_3 در الگوریتم های مختلف ۹۵
- شکل ۴۵-۶: پارامتر Γ_4 در الگوریتم های مختلف ۹۵
- شکل ۴۶-۶: پارامتر K_1 در الگوریتم های مختلف ۹۶
- شکل ۴۷-۶: تابع هزینه در الگوریتم های مختلف ۹۷
- شکل ۴۸-۶: تابع هزینه ۹۸
- شکل ۴۹-۶: سیگنال کنترلی ۹۹
- شکل ۵۰-۶: سیگنال کنترلی ۹۹

فهرست جدول‌ها

- جدول ۱-۲: مقادیر پارامترهای معادلات دینامیکی ۲۳
- جدول ۱-۵: پارامترهای کنترل برای بهینه‌سازی ۶۲
- جدول ۱-۶: نتایج مقایسه پارامترهای کنترل بین الگوریتم‌های بهینه‌سازی پرندگان ۹۶

فصل اول:

کلیات

۱-۱ مقدمه

در این فصل ابتدا تاریخچه رباتیک^۱ و به دنبال آن توضیحاتی در خصوص ربات و معایب و مزایای آن ارائه می‌شود و پس از آن مروری بر کارهای انجام شده انجام خواهد شد سپس اهداف پایان‌نامه بیان خواهد شد و در پایان طرح کلی پایان‌نامه ارائه خواهد شد.

۱-۲ رباتیک

اولین بار با صنعتی شدن دو کشور فرانسه و ایتالیا، در کارخانجات خود از ماشین‌هایی استفاده کردند که قدرت مکانیکی برای انجام کارها داشتند ولی هیچ نشانی از هوش در آنها وجود نداشت.

سال‌های بعد شاهد ترقی بسیاری در اتوماسیون صنعتی بودند. یک جرثقیل دورانی که به یک چنگک مجهز شده بود در سال ۱۸۹۲ به دست بابیت^۲ ساخته شد. این جرثقیل برای برداشتن شمش‌های داغ از درون کوره مورد استفاده قرار می‌گرفت. در سال ۱۹۳۸ پلارد^۳ یک بازوی مکانیکی برای پاشیدن رنگ ابداع کرد. اولین بازوی مکانیکی که از راه دور کنترل می‌شد توسط گوئرتز^۴ ساخته شد. این وسیله برای کار با مواد رادیواکتیو طراحی شده بود.

بعدها در سال ۱۹۴۷ وسیله مشابهی ساخته شد که مجهز به موتورهای الکتریکی و حسگرهای خاصی بود.

حرکات دست شخص کنترل‌کننده توسط حسگرها اندازه‌گیری می‌شد و برای راه اندازی موتورهای ربات مورد استفاده قرار می‌گرفت. با تمام این اختراعات باز هم کنترل این ماشین‌های مکانیکی مستلزم داشتن یک سیستم هوشمند می‌باشد و به همین دلیل در این دوران اولین کامپیوتر چند منظوره در دانشگاه پنسیلوانیا ساخته شد. با ظهور کامپیوتر امکان کنترل عددی ماشین فراهم

^۱Robotic system

^۲Babbit

^۳Pollard

^۴Goertz

شد. کنترل عددی اولین بار در سال ۱۹۵۲ در دانشگاه ام ای تی^۱ مورد استفاده قرار گرفت و به تولید اولین زبان برنامه‌نویسی برای ماشین‌ها انجامید. رباتیک بر پایه دو فناوری آغاز شده است: بازوهای مکانیکی^۲ و توانایی کنترل‌های عددی ماشین‌ها.

پس از تمام این دوران‌های طی شده باید به فکر ایجاد یک سری ربات‌های هوشیار و دارای حافظه می‌بودند که در سال ۱۹۵۴ اولین بازوی رباتیک حافظه‌دار تولید شد. این ماشین می‌توانست حرکات نقطه به نقطه قابل تکرار انجام دهد. پنج سال بعد، اولین ربات صنعتی حافظه‌دار که از کنترل عددی و تکنولوژی عملگرهای الکتریکی بهره می‌گرفت، ساخته شد. توسعه ربات‌های متحرک براساس تمایل به حمل و نقل اتوماتیک در خطوط تولید صورت پذیرفت و در نهایت منجر به ساخت سیستم‌های حمل و نقل بدون راننده شد که در اواخر دهه ۷۰ میلادی برای نقل و انتقال محصولات در کارخانه‌ها مورد استفاده قرار می‌گرفتند.

بدین صورت جهان صنعتی در چند دهه اخیر، توجه خود را به بکارگیری ربات‌ها در صنایع معطوف داشته است و به همین دلیل تحقیقات گسترده در زمینه‌های مختلف کاربردهای ربات، انجام شده است که برای این تحقیقات و درک پیچیدگی ربات‌ها و کاربرد آن‌ها نیاز به دانش مهندسی برق، مهندسی مکانیک، مهندسی صنایع، علوم کامپیوتر، اقتصاد و ریاضیات دارد [۴۰].

دلایل سرمایه‌گذاری روز افزون در زمینه رباتیک که می‌توان به صورت زیر دسته بندی کرد:

۱- نقش موثر ربات در صنایع پیشرفته، چون انسان‌ها قادر به انجام کارهای سنگین نیستند و به قدرت بدنی زیادی نیاز دارند و همچنین نمی‌توانند در هر محیط و شرایط آب و هوایی سخت، فعالیت کنند به همین دلیل ربات‌ها جایگزین انسان‌ها می‌گردند [۴۰].

^۱MIT

^۲Manipulator

۲- نقش ویژه آن در محیط‌های خطرناک نظیر محیط‌های رادیو اکتیو ، شیمیایی، فضا، اعماق دریا و زمین که استفاده از انسان‌ها در آنجا غیر عملی و نامطلوب است [۴۰].

۳-اندام‌های مصنوعی مثل دست وپای مصنوعی که نیاز به روش‌های تحلیل و طراحی شبیه بازوی ماهر صنعتی دارند.

یک ربات دارای سه مشخصه زیر است:

۱- دارای حرکت و دینامیک است.

۲- قابلیت برنامه‌ریزی جهت انجام کارهای مختلف را دارد.

۳- بعد از این که برنامه‌ریزی شد، قابلیت انجام وظایفش را به صورت خودکار دارد.

ربات‌ها از قسمت‌هایی تشکیل شده‌اند که عبارتند از:

۱- قسمت‌های مکانیکی مفصل بندی شده مانند بازوهای یک ربات

۲- عملگرها^۱: دستگاه‌هایی که باعث حرکت می شوند.

۳- سیستم انتقال نیرو

۴- حس‌گرها

۵- کنترلرها

۶- منبع تغذیه

شاید یکی از دلایلی که باعث شد ربات‌ها به دنیای ما انسان‌ها راه یابد انعطاف‌پذیری^۲ ربات‌ها

¹Operator

²Flexibility

در انجام کارهای متفاوت است. یعنی توانایی آن در انجام گستره‌ای وسیع از اعمالی که بدان سپرده می‌شود. این قابلیت، یک ویژگی ارزشمند به شمار می‌آید.

۱-۲-۱ انواع ربات

از لحاظ حرکتی ربات‌ها را می‌توان به دو دسته تقسیم کرد:

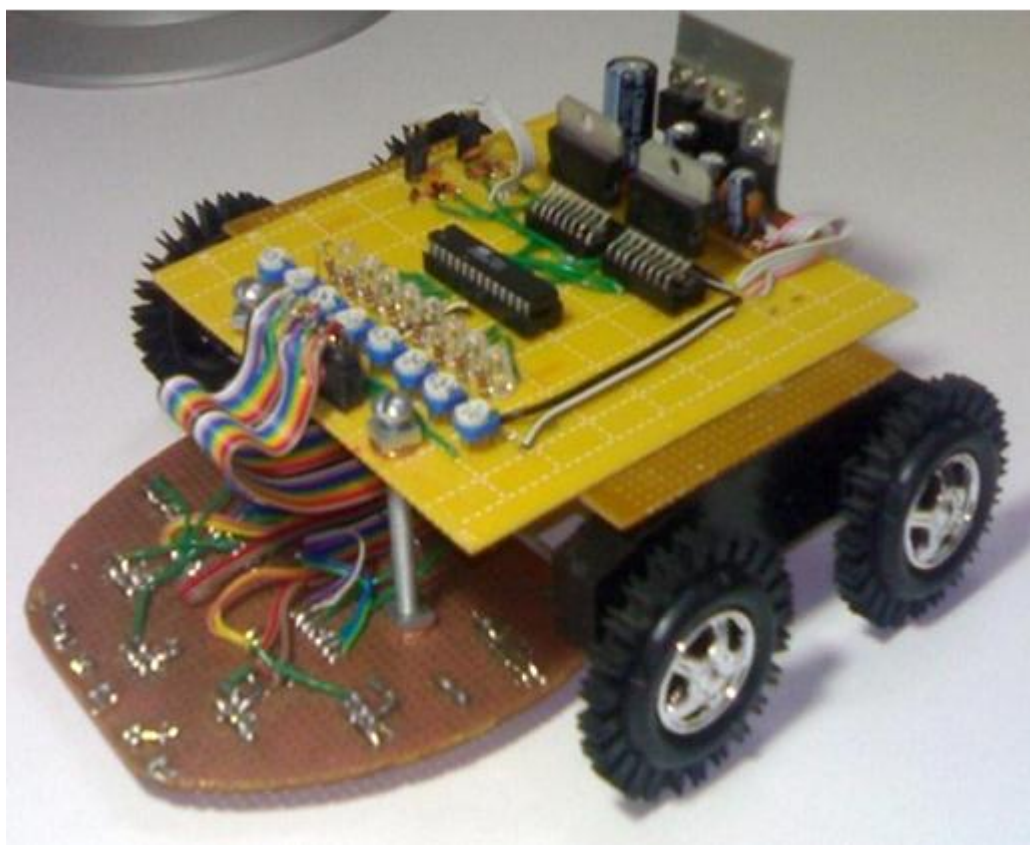
ربات‌های امروزی که شامل قطعات الکترونیکی و مکانیکی هستند در ابتدا به صورت بازوهای مکانیکی برای جابجایی قطعات و یا کارهای ساده و تکراری که موجب خستگی و عدم تمرکز کارگر و افت بازده می‌شد بوجود آمدند. این‌گونه ربات‌ها جابجاگر یا بازوی صنعتی نام دارند. بازوهای ربات‌های صنعتی معمولاً در نقطه ثابت و در فضای کاملاً کنترل شده در کارخانه نصب می‌شوند و به غیر از وظیفه‌ای که به خاطر آن طراحی شده‌اند قادر به انجام کار دیگری نیستند. این وظیفه می‌تواند در حد بسته‌بندی تولیدات، کنترل کیفیت و جدا کردن تولیدات بی کیفیت، و یا کارهای پیچیده‌تری همچون جوشکاری و رنگزنی با دقت بالا باشد.



شکل ۱-۱: ربات بازوی صنعتی

نوع دیگر ربات‌ها که امروزه مورد توجه بیشتری است ربات‌های متحرک هستند که مانند بازوهای ربات‌های صنعتی در محیط ثابت و شرایط کنترل شده کار نمی‌کنند. بلکه همانند موجودات زنده در دنیای واقعی و با شرایط واقعی زندگی می‌کنند و سیر اتفاقاتی که ربات باید با آن‌ها روبرو شود از قبل مشخص نیست. در این نوع ربات‌هاست که تکنیک‌های هوش مصنوعی می‌بایست در کنترل ربات (مغز ربات) به کار گرفته شود. ربات‌های متحرک به دسته‌های زیر تقسیم‌بندی می‌شوند:

- ربات‌های چرخ‌دار



شکل ۱-۲: ربات متحرک از نوع چرخ‌دار

- ربات‌های پادار



شکل ۱-۳: ربات متحرک از نوع پادار

- ربات‌های پرنده



شکل ۱-۴: ربات متحرک از نوع پرنده

- ربات‌های چندگانه (هایبرید)^۱ که ترکیبی از ربات‌های بالا یا ترکیب با جایجاگرها هستند.



شکل ۱-۵: ربات متحرک از نوع هایبرید

- در میان مزایایی که در معرفی ربات‌ها بیان می‌شوند می‌توان به موارد ذیل اشاره کرد:
- ◀ رباتیک و اتوماسیون در بسیاری از موارد می‌توانند ایمنی، میزان تولید بهره و کیفیت محصولات را افزایش دهند.
 - ◀ ربات‌ها می‌توانند در موقعیت‌های خطرناک کار کنند و با این کار جان هزاران انسان را نجات دهند.
 - ◀ ربات‌ها به راحتی محیط اطراف خود توجه ندارند و نیازهای انسان برای آن‌ها مفهومی ندارند. ربات‌ها هیچ‌گاه خسته نمی‌شوند.
 - ◀ دقت ربات‌ها خیلی بیشتر از انسان‌هاست، آن‌ها در حد میلی یا میکرو اینچ دقت دارند.
 - ◀ بسیاری از عملیات طاقت‌فرسا و غیرقابل انجام توسط متصدیان را انجام می‌دهند.
 - ◀ ربات‌ها می‌توانند در یک لحظه چند کار را با هم انجام دهند ولی انسان‌ها در یک لحظه

¹Hybrid

تنها یک کار را انجام می‌دهند.

◀ آن‌ها بر خلاف عامل انسانی یعنی متصدی خط تولید، توانایی پرداختن به سه شیفت را دارند.

◀ هزینه‌های مربوط به جلوگیری از آلودگی صوتی، تعدیل هوا و فراهم کردن روشنایی لازم برای خط تولید دیگر بر واحد تولید تحمیل نخواهد شد.

◀ برای اضافه‌کاری این ربات‌ها، هزینه اضافی پرداخت نمی‌شود، حق بیمه، حق مسکن و هزینه ایاب وذهاب پرداخت نمی‌شود، احتیاج به افزایش حقوق ندارند و هزینه‌ای نیز از بابت بهداشت و درمان به واحد تولیدی تحمیل نمی‌کنند.

◀ مدیر کارخانه با هیچ اتحادیه کارگری و ... روبه‌رو نیست.

مزایای ذکر شده باعث می‌شوند که سهم هزینه کار مستقیم نیروی انسانی در هزینه محصولات تولیدی، واحدهای تولیدی کاهش پیدا کنند. حضور ربات‌ها سبب می‌گردد که انسان‌ها مجبور باشند کمتر به کارهای فیزیکی پرداخته و از نیروی کار به سمت نیروی کار ذهنی متمایل می‌گردد.

معایب ربات‌ها عبارتند از:

◀ ربات‌ها در موقعیت‌های اضطراری توانایی پاسخ‌گویی مناسب را ندارند که این مطلب می‌تواند بسیار خطرناک باشد.

◀ ربات‌ها هزینه‌بر هستند.

◀ ربات‌ها قابلیت محدود دارند یعنی فقط توانایی انجام کاری را دارند که برای آن ساخته شده‌اند.

◀ ربات‌ها از دنیای احساسی خارج هستند و نمی‌توانند همانند انسان احساسی باشند و قوه ادراک آن‌ها محدود به مکان است. اما انسان این‌گونه نیست.

◀ در صورتی که در برنامه‌ریزی آن‌ها خطایی رخ دهد، کل محصول کارخانه خراب می‌شود.

◀ محیط زیست را آلوده می‌کند.

◀ برای برنامه‌ریزی آن‌ها نیاز به شخص طراح است.

◀ باعث بیکار شدن انسان‌ها می‌شوند.

۱-۳ مروری بر کارهای انجام شده

یکی از موضوعات قابل توجه در زمینه کنترل طراحی کنترل‌کننده مقاوم^۱ برای بازوی ماهر ربات است. تا کنون روش‌های مختلفی برای کنترل بازوی ربات از جمله کنترل‌کننده تناسبی انتگرال-گیر مشتق‌گیری [۱]، کنترل تطبیقی [۲،۳]، کنترل تطبیقی مود لغزشی [۴]، کنترل بهینه [۵،۶] و روش‌های هوشمند [۷].

در [۸]، طراحی کنترل موقعیت بهینه غیرخطی برای ربات با استفاده از روش تقریبی، سیستم غیر خطی را به مجموعه‌ای از سیستم‌های متغیرهای زمان خطی می‌توان تقسیم‌بندی کرد و برای طراحی کنترل‌کننده بهینه، از روش تنظیم‌کننده درجه دو خطی^۲ استفاده شده است که برای شاخص عملکرد آن باید ماتریس‌های L ، Q ، R وزن‌دهی شود. در [۹]، برای کنترل مسیر بازوی ربات از کنترل‌کننده بهینه هوشمند استفاده شده است در کنترل بهینه هوشمند از کنترل‌کننده عصبی-فازی برای یادگیری یک تابع غیرخطی در قانون کنترل بهینه استفاده شده است. در [۱۰]، برای شرایط پایداری موقعیت / نیرو بازوی ماهر ربات، یک روش لیاپونوفی پیشنهاد شده است در حالی که پارامترهای کنترل‌کننده با الگوریتم بهینه‌سازی اجتماع پرندگان بهینه شده است. میانگین ریشه مربعات خطا به عنوان تابع هزینه برای طراحی الگوریتم بهینه‌سازی پرندگان در نظر گرفته شده است.

^۱Robust controller

^۲Linear quadratic regulator

در طراحی بسیاری از سیستم‌ها و حل بسیاری از مسائل نیاز داریم که از بین مجموعه وسیع از جواب‌های ممکن یک جواب را به عنوان پاسخ بهینه انتخاب نماییم. اما به علت وسعت زیاد مجموعه جواب عملاً نمی‌توان پاسخ‌ها را آزمود و باید این آزمایش را به صورت تصادفی و با سعی و خطا انجام داد و این روش بسیار زمان‌بر است. از طرف دیگر این روند تصادفی باید به گونه‌ای انجام شود که به سمت بهترین جواب همگرا گردد. الگوریتم بهینه‌سازی پرندگان به این علت که به راحتی قابل پیاده‌سازی و دارای ساختار ساده و همچنین مستقل از حدس اولیه است و نیازی به گرادیان تابع هدف ندارد، دارای اهمیت ویژه است و برای سیستم‌های پیچیده غیرخطی و چندمتغیره از الگوریتم بهینه‌سازی پرندگان می‌توان استفاده کرد. با بکارگیری الگوریتم بهینه‌سازی پرندگان می‌توان پارامترهای سیستم کنترل را به گونه‌ای تعیین نمود که عملکرد سیستم کنترل بهینه شود.

با توجه به خصوصیات خوب این الگوریتم، در زمینه‌های مختلف به کار برده شده است برای مثال در [۱۱-۱۳]، الگوریتم بهینه‌سازی پرندگان برای تنظیم پارامترهای بهینه کنترل‌کننده تناسبی-انتگرال‌گیر- مشتق‌گیر ارائه شده است. در [۱۴]، استفاده از الگوریتم بهینه‌سازی پرندگان برای آموزش وزن‌های شبکه عصبی پیشنهاد شده است. در [۱۵]، برای جلوگیری از انتخاب پارامترها به صورت سنتی، الگوریتم بهینه‌سازی اجتماع پرندگان برای بهینه‌سازی پارامترهای کنترل‌ر ردیابی وضعیت ماهواره پیشنهاد شده است. در [۱۶]، الگوریتم بهینه‌سازی پرندگان برای حل سیستم‌های با معادلات غیر خطی ارائه شده است. در [۱۷]، با استفاده از ترکیب روش الگوریتم بهینه‌سازی پرندگان و روش برنامه‌ریزی درجه دو متوالی برای حل کنترل بهینه غیرخطی ارائه شده است.

۴-۱ اهداف پایان‌نامه

ربات‌ها سیستم‌های غیرخطی هستند که دارای چند ورودی- چند خروجی^۱، با دامنه‌ی کاری وسیع و دارای عدم قطعیت می‌باشند. معمولاً یک ربات می‌تواند کارهایی که ما انسان‌ها می‌خواهیم

^۱Multi-input Multi-output

انجام دهیم را ارزان تر انجام دهد. علاوه بر این ربات‌ها می‌توانند کارهای خطرناک مانند نظارت بر تأسیسات انرژی هسته‌ای یا کاوش یک آتش‌فشان را انجام دهند. ربات‌ها می‌توانند کارها را دقیق‌تر از انسان‌ها انجام دهند و روند پیشرفت در علم پزشکی و سایر علوم کاربردی را سرعت بخشند. ربات‌ها به ویژه در امور تکراری و خسته کننده مانند ساختن صفحه مدار، ریختن چسب روی قطعات یدکی و... سودمند هستند و در بخش‌های تولید جایگزین کارگران شده‌اند.

با توجه به این که ربات‌ها چند متغیره و به شدت غیرخطی همراه با تزویج^۱ سنگین هستند. نمی‌توان از روش‌های سنتی مثل روش نیوتن یا روش گرادیان بهینه‌سازی بهره جست. کنترل بهینه^۲ سیستم‌های غیرخطی یکی از چالش‌انگیزترین و سخت‌ترین موضوعات در تئوری کنترل است. در چنین شرایطی الگوریتم‌هایی به کار می‌آیند که روش‌های تکاملی یا هوشمند^۳ هستند. از جمله این الگوریتم‌ها، الگوریتم بهینه‌سازی پرندگان است که می‌تواند در شرایطی که روش‌های سنتی از حل مساله عاجزند، برای دستیابی به کنترل بهینه اجرا شود [17].

یکی از مسائلی که در زمینه رباتیک در سال‌های اخیر بطور گسترده مورد توجه و تحقیق قرار گرفته، کنترل مسیر حرکتی بازوهای مکانیکی برای جلوگیری از برخورد به موانع می‌باشد. این وظیفه مهم با پیدا کردن مسیر بهینه بدون برخورد بازوی مکانیکی از حالت اولیه مشخص به یک پیکربندی نهایی حاصل می‌شود. در این پایان‌نامه، کنترل مسیر یک بازوی مکانیکی با ارائه یک کنترل‌کننده تطبیقی^۴ انجام شده است. برای کنترل مسیر بازوی مکانیکی و به منظور دستیابی سریع به جواب-های بهینه در فضای کاری^۵، الگوریتم بهینه‌سازی اجتماع پرندگان مورد استفاده قرار گرفته است.

همچنین به منظور مقایسه، از الگوریتم‌های مختلف بهینه‌سازی پرندگان برای کنترل بهینه ربات استفاده شده است تا پارامترهای بهینه کنترل‌کننده را بدست آورد و شاخص عملکرد یا تابع

¹Coupling

²Optimal control

³Intelligence

⁴Adaptive controller

⁵Task space

هزینه که باید کمینه شود و نشان داده شده است که این الگوریتم‌ها دارای سرعت همگرایی بالا و هم‌چنین دارای قابلیت موثر در اجتناب از کمینه محلی می‌باشند.

۵-۱ طرح کلی پایان‌نامه

فصل دوم: دینامیک ربات را توصیف می‌نماید. در این فصل با استفاده از معادلات اوپلر-لاگرانژ، دینامیک ربات در حالت کلی بیان می‌شود. کنترل ربات بر مبنای مدل را ارائه می‌نماید. در این فصل بر اساس استراتژی گشتاور، کنترلی بر مبنای مدل مطرح شده و برای نشان دادن عملکرد آن، نتایج شبیه‌سازی ارائه خواهد شد.

فصل سوم: سیستم فازی و توضیحی از اصول ابتدایی مربوط به منطق فازی را در بر می‌گیرد را شامل می‌شود.

فصل چهارم: الگوریتم بهینه‌سازی اجتماع پرندگان را توصیف می‌نماید.

فصل پنجم: بهینه‌سازی با الگوریتم بهینه‌سازی اجتماع پرندگان را توصیف می‌نماید. در این فصل الگوریتم بهینه‌سازی پرندگان را روی قانون کنترل گشتاور که در [۲] ارائه شده بکار می‌بریم. الگوریتم بهینه‌سازی پرندگان به منظور طراحی کنترل بهینه بوسیله پیدا کردن پارامترهای طراحی کنترل بهینه بکار برده می‌شود.

فصل ششم: نتایج و شبیه‌سازی کنترل بهینه غیرخطی بازوی ربات ارائه خواهد شد و همچنین در این فصل روش‌های مختلف الگوریتم با یکدیگر مقایسه شده است و سیستم بهینه شده با سیستم غیر بهینه نیز مقایسه شده است.

فصل هفتم: نتایج پایان‌نامه بیان شده و سپس پیشنهادات مطرح می‌شود.

فصل دوم:

دینامیک ربات

این فصل با معادلات اوایلر-لاگرانژ شروع می‌شود که با این رویکرد می‌توان معادلات حرکت را به دست آورد و بعد از آن معادلات دینامیکی بازوی ماهر دو لینکی ارائه می‌شود. بازوهای ماهر ربات به شدت غیرخطی و دارای تزویج شدید و متغیر با زمان هستند. همچنین عدم قطعیت در مدل و پارامترهای سیستم و... همواره وجود دارد که باعث ناپایداری در سیستم‌های ربات می‌شود. در میان روش‌های کنترلی غیرخطی برای ربات، کنترل‌کننده تطبیقی برای ردیابی و غلبه بر نامعینی‌های موجود پیشنهاد شده است. در این فصل همچنین نتایج شبیه‌سازی برای این کنترل‌کننده ارائه شده است.

۱-۲ معادله دینامیکی بازوی رباتیک

برای بیان مدل دینامیکی بازوهای ربات از معادلات لاگرانژ استفاده می‌نماییم. فرم کلی معادلات لاگرانژ یا معادلات اوایلر-لاگرانژ^۱ حرکت بصورت زیر بیان می‌گردد [۴۰].

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, \dots, n \quad (1-2)$$

q : بردار موقعیت مفاصل

\dot{q} : بردار سرعت مفاصل

τ : برداری از گشتاورهای مفاصل

لاگرانژین در معادله (۲-۲) بیانگر تفاضل انرژی جنبشی و انرژی پتانسیل رابطه‌ها می‌باشد [۴۰]:

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q) \quad (2-2)$$

$T(q, \dot{q})$: انرژی جنبشی^۲

^۱Lagrange-Euler equations

^۲Kinetic energy

$U(q)$: انرژی پتانسیل

۱-۱-۲ انرژی جنبشی برای ربات با n -رابط

با در نظر گرفتن یک بازوی ماهر n -رابط ، سرعت‌های زاویه‌ای و سرعت خطی هر نقطه روی هر رابط به سادگی می‌تواند بر حسب ماتریس ژاکوبین و مشتق متغیرهای مفاصل بیان شود. برای ماتریس ژاکوبین $J_{v_{ci}}$ و J_{w_i} داریم [۴۰]:

$$V_{ci} = J_{v_{ci}}(q)\dot{q} \quad (۳-۲)$$

$$W_i = R_i^T(q)J_{w_i}(q)\dot{q} \quad (۴-۲)$$

v_{ci} : سرعت خطی^۱

W_i : سرعت زاویه‌ای^۲

R_i : ماتریس دورانی است که بردارها را از دستگاه جسم به دستگاه مبنا تبدیل می‌کند.

حال در نظر بگیرید که جرم رابط i برابر m_i است و I_i ماتریس اینرسی رابط i در یک دستگاه مختصات موازی با دستگاه i تعیین می‌شود و مبدا آن در مرکز جرم قرار دارد [۴۰].

پس کل انرژی جنبشی جسم به صورت زیر می‌باشد:

$$K = \frac{1}{2}mV_c^T + \frac{1}{2}W^T I W \quad (۵-۲)$$

که در آن:

I : ماتریس لختی $n \times n$ است

این فرمول می‌تواند بدین صورت تفسیر گردد: قسمت اول سمت راست فرمول همان انرژی جنبشی است و قسمت دوم فرمول ، وقتی پدیدار می‌گردد که جسم همانطور که انتقال می‌یابد، می‌-

^۱Liear velocity

^۲Angular velocity

چرخد و این حقیقت را منظور می‌نماید که بخش‌های مختلف جسم در سرعت‌های مختلف حرکت می‌کنند (قسمت دورانی انرژی جنبشی) [۴۰].

بنابراین با قرار دادن (۲-۳ و ۲-۴) در (۲-۵)، کل انرژی جنبشی بازوی ماهر برابر است با:

$$K = T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (۲-۶)$$

ماتریس اینرسی برای یک مجموعه n - رابط ، به صورت زیر بدست می‌آید:

$$M(q) = \sum_{i=1}^n \left(m_i J_{V_{ci}}(q)^T J_{V_{ci}}(q) + J_{W_i}(q)^T R_i(q) I_i R_i(q)^T J_{W_i}(q) \right) \quad (۲-۷)$$

ماتریس اینرسی^۱ یک ماتریس مثبت معین متقارن است که بطور کلی وابسته به موقعیت است [۴۰].

۲-۱-۲ انرژی پتانسیل برای ربات با n - رابط

در بازوی مکانیکی دو عامل باعث ایجاد انرژی پتانسیل^۲ می‌گردند: ۱- نیروی گرانشی (برای رابط‌های صلب و انعطاف پذیر)، ۲- انرژی کنشی (برای رابط‌های انعطاف پذیر) [۴۰].

تنها عامل ایجاد انرژی پتانسیل در رابط‌های صلب انرژی گرانشی ذخیره شده در لینک‌های صلب

می‌باشد که برای هر رابط از رابطه زیر قابل محاسبه است [۴۰]:

$$u_i = g^T r_{ci} m_i \quad (۲-۸)$$

g : بردار معلوم در راستای جاذبه به شکل اینرسی

r_{ci} : بردار مختصات مرکز جرم رابط نام

انرژی پتانسیل کل برای ربات n - رابط ، به صورت زیر می‌باشد:

^۱Inertia matrix

^۲Potential energy

$$U = \sum_{i=1}^n u_i = \sum_{i=1}^n g^T r_{ci} m_i \quad (9-2)$$

از فرمول بالا می‌توان چنین مشاهده کرد که انرژی پتانسیل، یک تابع فقط از مختصات تعمیم داده است و نه مشتقات آن‌ها، یعنی انرژی پتانسیل به موقعیت ربات بستگی دارد و به سرعت بستگی ندارد.

۲-۲ معادلات حرکت

معادله دینامیک ربات شدیداً غیرخطی و چند متغیره و با تزویج سنگین است را می‌توان به شکل زیر نوشت [۴،۲]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (10-2)$$

که در آن:

q : بردار موقعیت مفاصل

\dot{q} : بردار سرعت مفاصل

\ddot{q} : بردار شتاب مفاصل

$M(q)$: ماتریس اینرسی ربات با بعد $n \times n$

$C(q, \dot{q}) \in R^n$: برداری از گشتاورهای مرکز‌گرا^۱ و کوریولیس^۲

$G(q) \in R^n$: برداری از گشتاورهای گرانشی^۳

τ : برداری از گشتاورهای مفاصل

^۱Centrifugal

^۲Coriolis

^۳Gravitational torque

۳-۲ کنترل ربات بر مبنای مدل

بازوی رباتیک، سیستم‌های غیرخطی و چند متغیره هستند، که دارای نامعینی‌های دینامیکی می‌باشند. برخی از روش‌های متعارف برای کنترل مسیر بازوی رباتیک استفاده می‌شوند، کنترل گشتاور محاسبه شده [۱۸] یکی از این روش‌هاست، کنترل تناسبی- مشتقی یکی دیگر از این روش‌ها می‌باشد [۱۹]، ولی برای عملکرد صحیح آن بایستی مدل دینامیکی ربات به طور دقیق مشخص باشد.

۴-۲ طراحی کنترل کننده

معادله دینامیک ذکر شده در (۲-۱۰) می‌تواند به شکل زیر بیان شود [۲]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q})\beta \quad (۱۱-۲)$$

β : بردار ثابت از پارامترهای نامعلوم با بعد $p \times 1$ که خواص جرم جا به جا کننده را توصیف می‌کند.

Y : ماتریس معلوم با بعد $n \times p$ که رگر سور نامیده می‌شود.

قانون کنترل با K که یک ماتریس مثبت معین است و σ که خطای سرعت است به صورت زیر

می‌باشد [۲]:

$$\tau = M(q)\ddot{q}_r + \dot{q}_r + G(q) + K\sigma \quad (۱۲-۲)$$

که در آن \tilde{q} ، $\dot{\tilde{q}}$ ، $\ddot{\tilde{q}}$ به صورت زیر تعریف می‌شود [۲]:

$$\ddot{q}_r = \ddot{q}_d - \Lambda\ddot{\tilde{q}} \quad , \quad \dot{\tilde{q}} = \dot{q} - \dot{q}_d \quad , \quad \tilde{q} = q - q_d \quad , \quad \dot{q}_r = \dot{q}_d - \Lambda\dot{\tilde{q}} \quad (۱۳-۲)$$

که \tilde{q} خطای ردیابی موقعیت، $\dot{\tilde{q}}$ خطای ردیابی سرعت، $\ddot{\tilde{q}}$ سرعت مرجع، همگرایی خطای

ردیابی را تضمین می‌کند، \ddot{q}_r شتاب مرجع، Λ ماتریس مثبت معین است.

σ به صورت زیر محاسبه می‌شود [۲]:

$$\sigma = \dot{q}_r - \dot{q} = \dot{\tilde{q}} + \Lambda \tilde{q} \quad (۱۴-۲)$$

با وجود عدم قطعیت، قانون کنترل زیر به صورت زیر پیشنهاد می‌شود [۲]:

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{G}(q) + K\sigma = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \hat{\beta} + K\sigma \quad (۱۵-۲)$$

که $\hat{M}(q)$ تخمینی از $M(q)$ ، $\hat{C}(q, \dot{q})$ تخمینی از $C(q, \dot{q})$ ، $\hat{G}(q)$ تخمینی از $G(q)$ و همچنین $\hat{\beta}$ تخمینی از β است.

باتوجه به جایگذاری معادله (۱۵-۲) در (۱۲-۲)، خطاهای مدل به صورت زیر نتیجه می-

شود [۲]:

$$\tilde{M} = \hat{M} - M \quad \tilde{C} = \hat{C} - C \quad \tilde{G} = \hat{G} - G \quad (۱۶-۲)$$

به منظور آنالیز پایداری سیستم و به دست آوردن همگرایی خطای ردیابی، تابع لیاپونوفی انتخابی به صورت زیر پیشنهاد می‌شود [۲]:

$$V(t) = \frac{1}{2} [\sigma^T H \sigma + \tilde{\beta}^T \Gamma^{-1} \tilde{\beta}] \quad (۱۷-۲)$$

قانون تطبیق زیر به صورت زیر پیشنهاد می‌شود [۲]:

$$\dot{\tilde{\beta}} = -\Gamma Y^T \sigma \quad (۱۸-۲)$$

با مشتق گیری از رابطه (۱۷-۲) داریم:

$$\dot{V}(t) = -\sigma^T K_D \sigma \leq 0 \quad (۱۹-۲)$$

که این رابطه ایجاب می‌کند که خطای خروجی به سطح $\sigma = 0$ همگرا شود که با داشتن رابطه (۲-)

(۱۷)، نشان می‌دهد هم‌چنان که t به بی نهایت میل می‌کند، \tilde{q} و $\dot{\tilde{q}}$ به 0 میل می‌کنند.

بنابراین، پایداری کلی سیستم و همگرایی خطای ردیابی، هر دو به وسیله کنترل کننده تطبیقی

تضمین می‌شوند.

۲-۵ نتایج شبیه‌سازی

به منظور نشان دادن معادله دینامیکی بازوی ماهر ربات که در قسمت (۲-۲) بیان شد، مولفه-

های ماتریس $M(q)$ ، $C(q, \dot{q})$ آن را می‌توان به صورت زیر نوشت [۴]:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -C\dot{q}_2 & -C(\dot{q}_1 + \dot{q}_2) \\ C\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (۲۰-۲)$$

که در آن:

$$M_{11} = a_1 + 2a_3 \cos q_2 + 2a_4 \sin q_2 \quad (۲۱-۲)$$

$$M_{12} = M_{21} = a_2 + a_3 \cos q_2 + a_4 \sin q_2 \quad (۲۲-۲)$$

$$M_{22} = a_2 \quad (۲۳-۲)$$

$$C = a_3 \sin q_2 - a_4 \cos q_2 \quad (۲۴-۲)$$

$$a_1 = I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \quad (۲۵-۲)$$

$$a_2 = I_e + m_e l_{ce}^2 \quad (۲۶-۲)$$

$$a_3 = m_e l_1 l_{ce} \cos \delta_e \quad (۲۷-۲)$$

$$a_4 = m_e l_1 l_{ce} \sin \delta_e \quad (۲۸-۲)$$

و ماتریس $Y(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)$ را می توان به صورت زیر نوشت:

$$Y_{11} = \ddot{q}_{r1} \quad (29-2)$$

$$Y_{12} = \ddot{q}_{r2} \quad (30-2)$$

$$Y_{21} = 0 \quad (31-2)$$

$$Y_{22} = \ddot{q}_{r1} + \ddot{q}_{r2} \quad (32-2)$$

$$Y_{13} = (2\ddot{q}_{r1} + \ddot{q}_{r2}) \cos q_2 - (\dot{q}_2 \dot{q}_{r1} + \dot{q}_1 \dot{q}_{r2} + \dot{q}_2 \dot{q}_{r2}) \sin q_2 \quad (33-2)$$

$$Y_{14} = (2\ddot{q}_{r1} + \ddot{q}_{r2}) \sin q_2 - (\dot{q}_2 \dot{q}_{r1} + \dot{q}_1 \dot{q}_{r2} + \dot{q}_2 \dot{q}_{r2}) \cos q_2 \quad (34-2)$$

$$Y_{23} = \ddot{q}_{r1} \cos q_2 + \dot{q}_1 \dot{q}_{r1} \sin q_2 \quad (35-2)$$

$$Y_{23} = \ddot{q}_{r1} \sin q_2 - \dot{q}_1 \dot{q}_{r1} \cos q_2 \quad (36-2)$$

در شبیه سازی از مقادیر زیر استفاده می کنیم [۴].

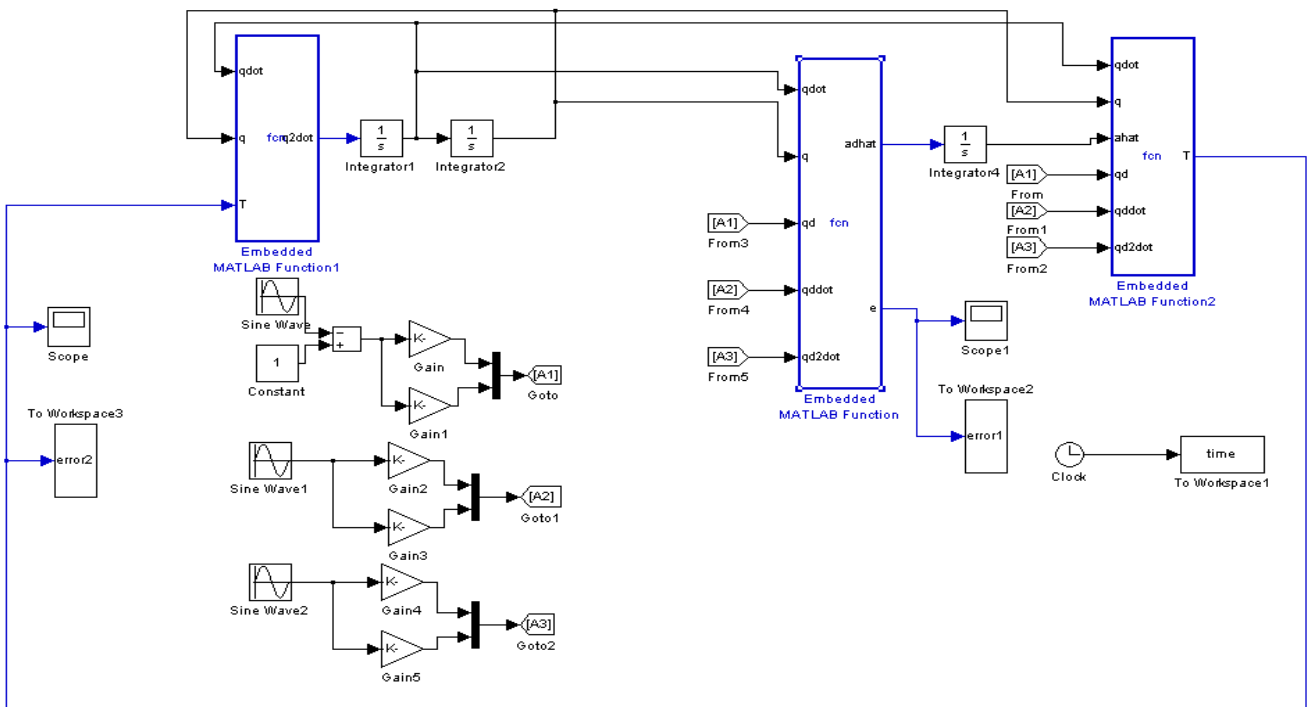
جدول ۱-۲: مقادیر پارامترهای معادلات دینامیکی

m_1	1	l_{ce}	0.6
l_1	1	Γ_1	0.03
m_e	2	Γ_2	0.05
δ_e	$\frac{\pi}{6}$	Γ_3	0.1
I_1	0.12	Γ_4	0.3
l_{c1}	0.5	Λ_1	20
I_e	0.25	K_1	100

به منظور نشان دادن عملکرد قانون کنترل تعدادی شبیه‌سازی ارائه خواهد شد و مسیر مطلوب به صورت زیر در نظر گرفته می‌شود [۴].

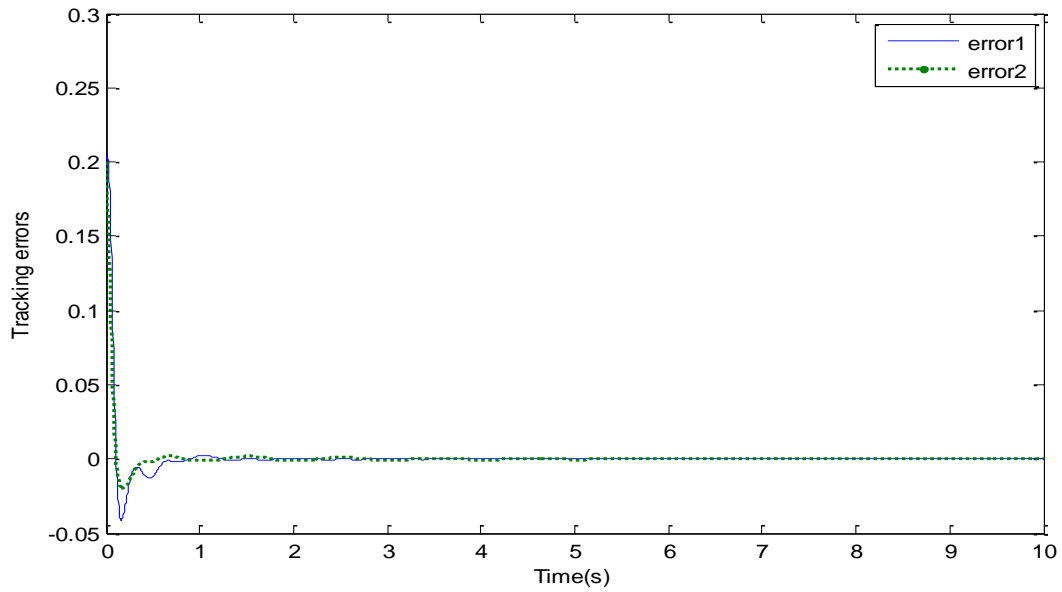
$$q_{d1}(t) = \frac{\pi}{6}(1 - \cos(2\pi t)) \quad (۳۷-۲)$$

$$q_{d2}(t) = \frac{\pi}{4}(1 - \cos(2\pi t)) \quad (۳۸-۲)$$



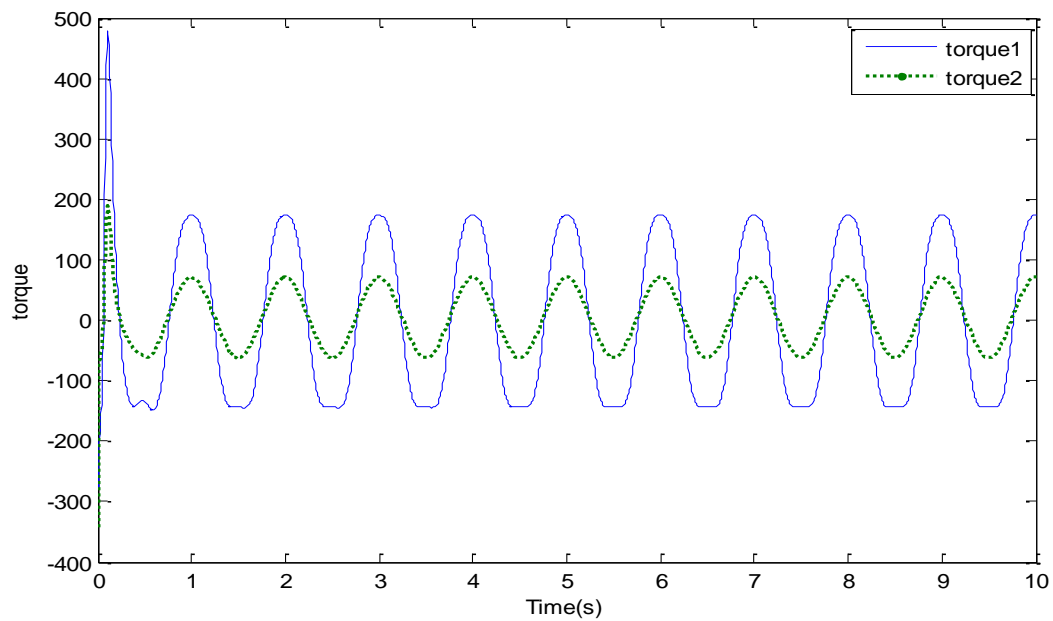
شکل ۱-۲: بلوک دیاگرام شبیه‌سازی بازوی ماهر ربات

شکل (۲-۲) خطاهای ردیابی تحت کنترل تطبیقی را نشان می‌دهد.



شکل ۲-۲: خطاهای ردیابی

شکل (۳-۲) گشتاورهای کنترل تحت کنترل تطبیقی را نشان می‌دهد.



شکل ۳-۲: گشتاورهای کنترل

فصل سوم: سیستم‌های فازی

برای مقابله موثر با پیچیدگی روزافزون در بررسی، مطالعه، مدل سازی و حل مسائل جدید در فیزیک، مهندسی، پزشکی، زیست شناسی و بسیاری از امور گوناگون دیگر، ایجاد و ابداع روش های محاسباتی جدیدی مورد نیاز شده است که بیشتر از پیش به شیوه های تفکر و تعلم خود انسان نزدیک باشد. در جهان واقعیت، مفاهیم نادقیق بسیاری در پیرامون ما وجود دارند که آنها را به صورت روزمره در قالب عبارتهای مختلف بیان می کنیم. هدف اصلی آنست که تا حد امکان، رایانه ها بتوانند مسائل و مشکلات بسیار پیچیده علمی را با همان سهولت و شیوایی بررسی و حل و فصل کنند که ذهن انسان قادر به ادراک و اخذ تصمیمات سریع و مناسب است. منطق فازی فناوری جدیدی است که شیوه هایی را که برای طراحی و مدل سازی یک سیستم نیازمند ریاضیات پیچیده و پیشرفته است، با استفاده از مقادیر زبانی و دانش فرد خبره جایگزین می سازد. در این فصل به بررسی سیستم های فازی جهت آشنایی با آنها می پردازیم.

۱-۳ سیستم فازی

واژه فازی^۱ در فرهنگ لغت به معنای مبهم، گنگ می باشد. سیستم های فازی، سیستم هایی با تعریف دقیق می باشند و کنترل فازی، روش کنترل غیرخطی^۲ نیز با تعریف دقیق می باشد. هدف اصلی منطق فازی شبیه سازی استدلال انسانی است. سیستم فازی، سیستم مبتنی بر قواعد یا مبتنی بر دانش است. مولفه های کلیدی پایگاه دانش سیستم فازی یک مجموعه ای از قوانین اگر-انگاه^۳ به دست آمده از دانش و تجربه بشر می باشد [۳۷،۳۸].

سیستم های فازی نگاشت چند ورودی-چند خروجی از یک بردار حقیقی به یک اسکالر با مقدار حقیقی است [۳۷،۳۸].

^۱Fuzzy

^۲Nonlinear control

^۳If-Then rules

۲-۳ چرا از منطق فازی استفاده می کنیم

اغلب سیستم های فیزیکی غیرخطی هستند و برای مدل سازی آنها، مقدار تقریبی مناسب ضروری است. برای سیستم هایی که پیچیدگی آنها کم و عدم قطعیت نیز ناچیز است می توان با استفاده از معادلات ریاضی ماهیت و رفتار سیستم را به طور دقیق مدل سازی و تحلیل کرد. برای سیستم هایی که پیچیدگی آنها کمی بیشتر است و عدم قطعیت نیز به طور نسبی زیاد است دیگر نمی توان تحلیل دقیق و قطعی از سیستم داشت. در این سیستم ها امکان استفاده از روش های ابتکاری مدرن مانند شبکه عصبی و هوش مصنوعی وجود دارد. در شبکه های عصبی به دلیل قابلیت یادگیری سیستم، عدم قطعیت به مرور کاهش یافته و قابلیت تحلیل موثر سیستم افزایش می یابد. سرانجام برای سیستم هایی با پیچیدگی بالا و عدم قطعیت زیاد که اطلاعات کافی و دقیقی نیز در دسترس نیست رویکرد استدلال تقریبی فازی مطرح می شود که به سیستم های فازی معروف هستند. ورودی سیستم های فازی می توانند اطلاعات نادقیق (فازی) باشند و پردازش های سیستم نیز با بهره گیری از استدلال تقریبی و به طور فازی انجام می شوند [۳۹]. برای بیان اطلاعات مبهم، منطق فازی از متغیرهای زبانی به جای متغیرهای عددی استفاده می کند.

۳-۳ مجموعه های فازی

در مجموعه های کلاسیک، حد و مرز مجموعه به طور قطعی تعریف می شود. در مجموعه های کلاسیک، یک عنصر یا قطعا عضو مجموعه هست یا قطعا عضو مجموعه نیست. لذا فرض اساسی در مجموعه های کلاسیک تعریف دقیق و قطعی حد و مرز مجموعه است و عضویت عناصر در مجموعه نیز دو حالت بیشتر ندارد. یک عنصر یا عضوی از مجموعه هست یا نیست. اما موارد بسیاری در عمل وجود دارد که تعریف حد و مرز دقیق و قطعی برای مجموعه امکان پذیر نیست. به عنوان مثال برای مجموعه افراد چاق نمی توان حد و مرز مشخصی تعریف کرد، زیرا به طور دقیق نمی توان بر چاق بودن و یا نبودن یک شخص قضاوت کرد. لذا نیاز به مجموعه ای داریم که محدوده آن منعطف و به طور

تقریبی (نادقیق) تعریف می شود به این نوع مجموعه، مجموعه فازی^۱ گفته می شود. عضویت عناصر در مجموعه های فازی نیز با درجه عضویت که عددی بین صفر و یک است، بیان می شود [۳۷،۳۸].

۳-۴ تابع عضویت

یک مجموعه فازی با یک تابع تعلق که محدوده مقادیرش از 0 تا 1 است، مشخص می شود. در عبارت ریاضی یک مجموعه فازی A را در فضای جهانی U می توان با یک مجموعه از زوج های مرتب x و مقدار تعلق $\mu_A(x)$ نمایش داد که به صورت زیر است [۳۷،۳۸]:

$$A = \{(X, \mu_A(x)) | x \in U\} \quad (۱-۳)$$

که U پیوسته است. تابع تعلق، یک تابع پیوسته در محدوده $[0,1]$ است. برای تعیین یک تابع تعلق دو راه حل وجود دارد [۳۷،۳۸].

اولین راه حل استفاده از دانش انسان خبره است [۳۷،۳۸].

دومین راه حل از مجموعه های جمع آوری شده (مشاهدات) برای تعیین توابع استفاده می کنیم. توابع تعلق می تواند هم خطی و هم غیرخطی باشد. انتخاب تابع تعلق برای عملکرد سیستم منطق فازی مهم است چون تابع تعلق، همه اطلاعات مربوط به یک فازی را تعیین می کند [۳۷،۳۸].

قانون فازی بر اساس قوانین اگر- آنگاه نقطه شروع ساخت یک سیستم فازی به دست آوردن مجموعه ای از قواعد اگر- آنگاه فازی از دانش افراد خبره یا دانش حوزه مورد بررسی می باشد. از متغیرهای زبانی و قواعد اگر- آنگاه برای فرموله کردن دانش بشری استفاده می شود. پایگاه قواعد فازی مجموعه ای از قواعد اگر- آنگاه فازی را نشان می دهد. قواعد اگر- آنگاه فازی را در بسیاری از فرایند صنعتی و تولیدات می توان استفاده کرد.

^۱Fuzzy set

۳-۵ انواع استنتاج فازی

بسیاری از روش های استنتاجی وجود دارد که استنتاج فازی با روش هایی از قبیل روش ممدانی ، روش تاگاکي-سوگنو-کانگ سروکار دارد. روش های ممدانی^۱ و تاگاکي-سوگنو^۲ از مهمترین و گسترده ترین روش های مورد استفاده در کنترل کننده های فازی هستند.

۳-۵-۱ روش ممدانی

این روش استنتاج فازی از مرسوم ترین روش هاست. در سال ۱۹۷۴ توسط پروفیسور ابراهیم ممدانی اولین سیستم کنترلی که مربوط به تنظیم یک موتور بخار بود و بر اساس منطق فازی کنترل می شد پیاده سازی گردید. ممدانی مجموعه ای از قوانین به دست آمده از دانش فرد خبره را، استفاده می کند [۳۸]. روش ممدانی سیستم استنتاج فازی، با چهار مرحله زیر انجام می شود:

۱. فازی سازی متغیرهای ورودی

۲. ارزیابی قانون

۳. اجتماع خروجی قانون

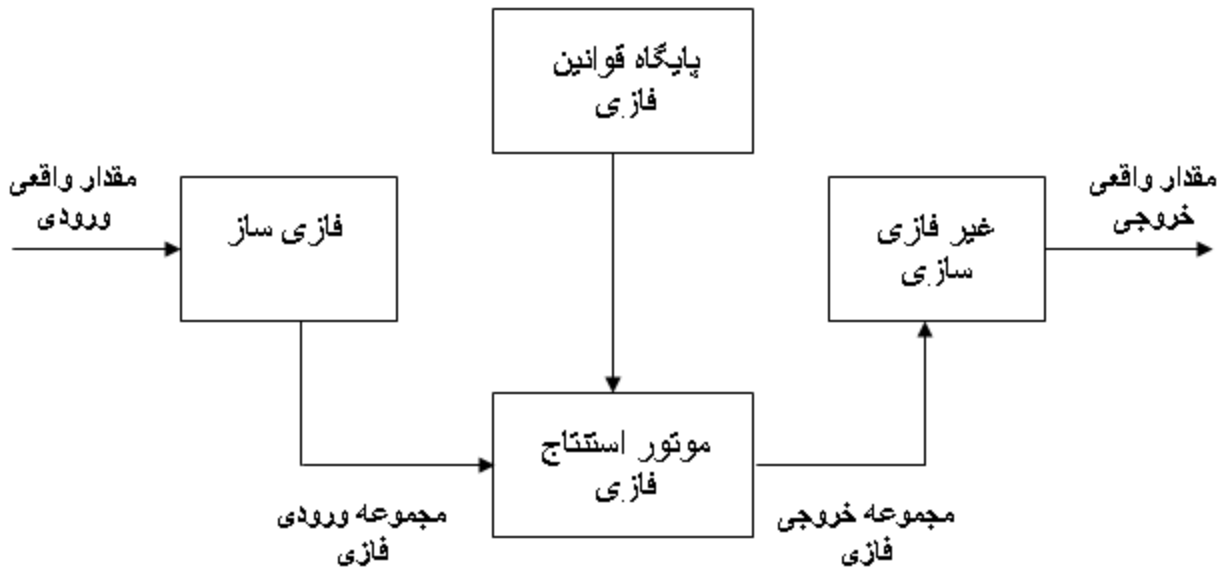
۴. دفازی یا غیرفازی ساز^۳

ساختار اصلی سیستم های فازی با فازی ساز و غیرفازی ساز در شکل (۳-۱) به صورت زیر است:

^۱Mamdani

^۲Tagaki-Sugeno-Kang

^۳Defuzzifier



شکل ۳-۱: ساختار اصلی سیستم های فازی با فازی ساز و غیرفازی ساز [۳۷]

برای بیان استنتاج فازی، هر مرحله با جزئیات بیشتر توضیح داده خواهد شد [۳۷، ۳۸].

◀ قدم اول: فازی سازی ورودی

فازی ساز به عنوان نگاشتی از یک نقطه $X^* \in U$ به یک مجموعه فازی $A' \subset U$ تعریف شده است، که ورودی به موتور استنتاج فازی می باشد. فازی ساز به تعریف معیار مشخصی برای انجام نگاشت نیاز دارد. اولین معیار بیان می کند که ورودی در نقطه X^* قطعی است، نگاشتش در U بدین معنی است که مجموعه فازی A' یک مقدار تعلق برزگی داشته باشد. دومین معیار این است که، اگر ورودی سیستم فازی به وسیله نویز خراب شود، فازی ساز باید بتواند تاثیر نویز را کاهش داده و حذف کند. سومین معیار بیان می کند که فازی ساز باید بتواند در ساده تر کردن محاسبات مربوط به موتور استنتاج فازی نقش داشته باشد.

فازی سازها شامل فازی ساز منفرد، فازی ساز گوسین، فازی ساز مثلثی می باشند.

◀ قدم دوم: ارزیابی قانون

قدم دوم هنگامی که ورودی‌ها فازی شدند، برای قسمت مقدم قوانین فازی بکار برده می‌شود. در صورتی که مقدم (قسمت اگر) داده شده از قوانین، دارای بیش از یک قسمت باشد، عملگرهای فازی (و، یا) را به کار می‌گیریم تا یک عدد را به دست آوریم که نمایانگر حاصل مقدم برای آن قانون است.

◀ قدم سوم: اجتماع تمام خروجی‌ها

از آن جا که تصمیم‌گیری بر اساس بررسی تمام قوانین در سیستم‌های استنتاج فازی صورت می‌گیرد، لذا برای تصمیم‌گیری باید قوانین را به روشی به توان ترکیب کرد. اجتماع (خروجی) روشی است که به واسطه آن تمام مجموعه‌های خروجی هر قانون را به یک مجموعه فازی واحد ترکیب می‌کند.

◀ قدم چهارم: غیرفازی سازی یا دفازی کردن

وظیفه دفازی عملکرد معکوس فازی‌سازی است. غیرفازی‌سازی به عنوان یک نگاشت از مجموعه فازی $B' \subset V$ به یک نقطه قطعی $y^* \in V$ تعریف می‌گردد. به عبارت دیگر، می‌توان گفت که وظیفه فازی‌سازی مشخص کردن نقطه‌ای است که بهترین نماینده مجموعه فازی B' باشد. بدیهی است که انتخاب‌های زیادی برای تعیین این نقطه وجود دارد، اما مناسب‌ترین نقطه می‌تواند با در نظر گرفتن معیار مشخصی تعیین شود. نقطه y^* از نظر شهودی باید نشان دهنده مجموعه فازی B' باشد. علاوه بر این، غیرفازی‌سازی دارای سادگی محاسبات است، این معیار بخصوص برای کنترل‌کننده‌های فازی که در زمان واقعی استفاده می‌شود، مهم است. سر انجام، غیرفازی‌سازی باید پیوسته باشد. از غیرفازی‌سازها می‌توان غیرفازی‌ساز مرکز ثقل، غیر فازی‌ساز میانگین مراکز، غیرفازی‌ساز ماکزیمم اشاره کرد.

۳-۵-۲ روش سوگنو

اکثر کنترل کننده های فازی ، بر اساس تجربه افراد خبره و یا دانش مهندسی کنترل طراحی شده اند. یکی دیگر از روش استنتاج فازی ، روش سوگنو یا تاگاکی - سوگنو - کانگ است. این روش در سال ۱۹۸۵ معرفی شد که در بسیاری از موارد شبیه روش ممدانی است. دو قسمت اول از فرایند استنتاج فازی، که شامل فازی سازی ورودی ها و ارزیابی قانون می شود، کاملاً در دو روش به هم شبیه هستند. تفاوت اصلی روش ممدانی و سوگنو در این است که خروجی سوگنو به صورت تابعی از ریاضی است [۳۸].

نمونه ای از قانون فازی به روش سوگنو به صورت زیر است:

If X is A AND Y is B THEN Z is $f(X,Y)$

که Z, Y, X متغیرهای زبانی هستند، A, B مجموعه های فازی هستند و $f(X, Y)$ تابع ریاضی است.

در این پایان نامه منطق فازی با سیستم استنتاج ممدانی برای تنظیم پارامترهای الگوریتم بهینه سازی اجتماع پرنندگان به منظور بهبود عملکرد جستجو برای جواب بهینه سازی ، استفاده شده است .

فصل چهارم:

الگوریتم بهینه‌سازی اجتماع پرندگان

در این فصل ما الگوریتم بهینه‌سازی اجتماع پرندگان را در نظر گرفته‌ایم که یکی از الگوریتم‌های بهینه‌سازی الهام گرفته شده بیولوژیکی است. در بسیاری از مسایل بهینه‌سازی که می‌خواهیم مینیمم تابع $J(x)$ ، $x \in R^n$ را پیدا کنیم. الگوریتم بهینه‌سازی اجتماع پرندگان یکی از الگوریتم‌های بهینه‌سازی مبتنی بر جمعیت است که می‌تواند برای حل چنین مسائلی استفاده شود. الگوریتم بهینه‌سازی اجتماع پرندگان یک الگوریتم جستجوی مستقیم (بدون گرادیان) است که یک جمعیت از ذرات بطور موازی برای مینیمم کردن یک تابع معلوم در فضای چند بعدی (n -بعدی)، بدون استفاده از اطلاعات گرادیان به جستجو می‌پردازد [۲۲].

۴-۱ مطالعه رفتار پرندگان و ایده اولیه بهینه‌سازی اجتماع پرندگان

الگوریتم بهینه‌سازی اجتماع پرندگان^۱ یک تکنیک بهینه‌سازی مبتنی بر قوانین احتمال است که ایده اولیه آن توسط دکتر راسل سی ابرهارت^۲ (دانشمند علوم کامپیوتر) و جمیز کندی^۳ (روانشناس اجتماعی) در سال ۱۹۹۵ ارائه شد [۲۰].

آن‌ها در ابتدا قصد داشتند که با بهره‌گیری از مدل‌های اجتماعی و روابط اجتماعی نوعی هوش محاسباتی را بوجود بیاورند که به توانایی‌های فردی ویژه نیازی نداشته باشد [۴۱]. بدین ترتیب این الگوریتم، از رفتار اجتماعی پرندگان یا ماهی‌ها در پیدا کردن غذا، الهام گرفته شده است. الگوریتم بهینه‌سازی اجتماع پرندگان ممکن برخی شباهت‌هایی با الگوریتم ژنتیک و الگوریتم مورچه داشته باشد ولی این الگوریتم ساده‌تر است چون از عملگرهای جهش و تلفیق یا فرمون^۴ استفاده نمی‌کند در عوض از اعداد تصادفی و روابط سراسری در میان ذرات گروه استفاده می‌کند [۲۱]. فرض بر این است که گروهی از پرندگان بصورت تصادفی در یک منطقه به دنبال غذا می‌گردند در حالی که در یک

^۱Particle swarm optimization algorithm

^۲Eberhart

^۳Kennedy

^۴Pheromones

منطقه از فضای جستجو^۱، غذا وجود دارد و پرندگان به این علت که از مکان غذا اطلاعاتی ندارند و فقط میزان فاصله خود تا آن مکان را می دانند به دنبال پرنده ای حرکت می کنند که نزدیک ترین فاصله تا غذا را دارد. در الگوریتم بهینه سازی اجتماع پرندگان هر پرنده، یک جواب ممکن در فضای مسئله است که ذره نامیده می شود. هر ذره دارای یک مقدار شایستگی^۲ است که توسط تابع هزینه مسئله محاسبه می شود و پرنده ای که به غذا نزدیک تر است شایستگی بیشتری دارد. به صورت نظری حداقل هر پرنده به عنوان یکی از اعضای گروه از تجربه خود و یافته های سایر اعضا برای یافتن غذا بهره می برد و با استفاده از این اطلاعات مسیر و جهت حرکت خود را به سمت ذره بهینه هدایت می کند. در واقع الگوریتم بهینه سازی اجتماع پرندگان که مقدار کمینه یک تابع را جستجو می کند، همانند دسته ای از پرندگان عمل می کنند که به دنبال غذا می گردند. الگوریتم بهینه سازی اجتماع پرندگان فراتر از مجموعه ذرات^۳ است. هیچ کدام از ذرات به تنهایی قادر به حل مساله نیستند بلکه وقتی می توان به حل مساله امیدوار شد که ذرات با همدیگر در تعامل و در ارتباط باشند و این مشارکت یک مزیت قطعی بر جستجوی رقابتی برای یافتن غذای توزیع شده است. پایه اصلی نظریه الگوریتم بهینه سازی اجتماع پرندگان همین تسهیم اطلاعات بین اعضای یک گروه است [۴۱].

۲-۴ الگوریتم بهینه سازی اجتماع پرندگان

الگوریتم بهینه سازی اجتماع پرندگان با جواب های تصادفی (ذره ها) شروع بکار می کنند، سپس برای یافتن جواب بهینه در فضای مسئله با بروز کردن نسل ها به جستجو می پردازد. هر ذره بصورت چند بعدی با دو مقدار V_{id} و X_{id} که به ترتیب معرف وضعیت مکانی و سرعت مربوط به بعد d از n امین ذره هستند تعریف می شود. در هر مرحله از حرکت جمعیت، هر ذره با دو مقدار بهترین به روز می شود. اولین مقدار، بهترین جواب از لحاظ شایستگی است که تا کنون برای هر ذره بطور جداگانه

¹Search space

²Fitness

³Particles

بدست آمده است (مقدار شایستگی باید ذخیره شود) این مقدار $Pbest$ نامیده می شود. مقدار بهترین دیگری که توسط الگوریتم بهینه سازی اجتماع پرندگان بدست می آید، بهترین مقداری است که تاکنون توسط تمام ذرات در میان جمعیت به دست آمده است این مقدار بهترین کلی است و $Gbest$ نامیده می شود (این مقدار نیز باید ذخیره می شود) [۱۷]. اگر ذره در یک همسایگی مکانی خودش به عنوان ذره ای از جمعیت شرکت کند این مقدار تنها در همان همسایگی محاسبه می شود و بهترین محلی است که $Lbest$ نام دارد.

بعد از یافتن $Pbest$ و $Gbest(Lbest)$ هر ذره سرعت و وضعیت مکانی خود را طبق روابط زیر بروز رسانی می کند [۱۷]:

$$V_i^{k+1} = WV_i^k + C_1 rand_1 \times (Pbest_i^k - X_i^k) + C_2 rand_2 \times (Gbest^k - X_i^k) \quad (۱-۴)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (۲-۴)$$

در این روابط :

W : ضریب اینرسی^۱ که وظیفه بسیار مهمی در همگرایی الگوریتم دارد.

$rand_1$ و $rand_2$: اعداد تصادفی هستند که دارای توزیع یکنواخت در بازه ی (0,1) می باشد.

C_1 و C_2 : عوامل یادگیری یا ضرایب شتاب^۳ نامیده می شود.

برای جلوگیری از واگرایی الگوریتم، مقدار نهایی سرعت هر ذره به یک بازه محدود می شود [۴۱]:

$$V_{id} \in [-V_{min}, V_{max}] \quad (۳-۴)$$

رابطه (۱-۴) دارای سه عبارت است: عبارت اول نسبتی از سرعت فعلی ذره است، عبارت دوم که

^۱Inertia weight

^۲Convergence

^۳Learning factors

متناسب با تفاضل مکان فعلی ذره با بهترین مکان قبلی ذره که برای تغییر سرعت و چرخش ذره به سمت بهترین تجربه شخصی می‌باشد و عبارت سوم که موقعیت فعلی ذره با بهترین جواب در میان کل جمعیت است که سبب هدایت سرعت ذره جدید به سمت جواب بهینه می‌شوند [۱۷].

به عبارت دیگر ذره نام در معادله سرعت شامل سه بخش [۲۲]:

۱. سرعت حرکت

۲. شناختی^۱ (تجربه خودش)

۳. اجتماعی^۲ (تجربه همسایگانش)

در واقع حرکت دسته جمعی ذرات بدون قسمت اول معادله (۲-۴) فرایندی خواهد بود که طی آن فضای جستجو به تدریج کوچک می‌شود و جستجوی محلی بهترین ذره شکل می‌گیرد. اما در مقابل، قسمت اول معادله (۲-۴) باعث حرکت ذرات در مسیر عادی خود خواهد بود تا به دیواره محدود جستجو برسد و به نوعی جستجوی سراسری انجام می‌دهد. با ترکیب این دو عامل سعی شده تا موازنه‌ای بین جستجوی محلی و سراسری برقرار گردد.

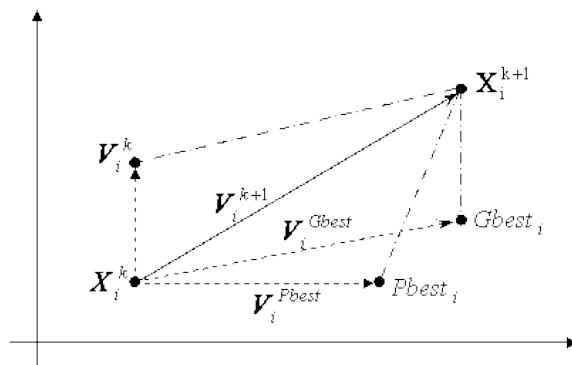
پس می‌توان به این نتیجه رسید که هر ذره به هنگام حرکت موارد زیر را در نظر می‌گیرد [۲۲]:

- جهت حرکت قبلی خود
- بهترین موقعیتی را که در آن قرار داشته است.
- بهترین موقعیتی که به وسیله جمع تجربه شده است.

شکل (۱-۴) یک تصویر مفهومی از فرایند جستجو طبق روابط (۱-۴) و (۲-۴) می‌باشد.

¹Cognitive

²Social



شکل ۴-۱ فرایند جستجو

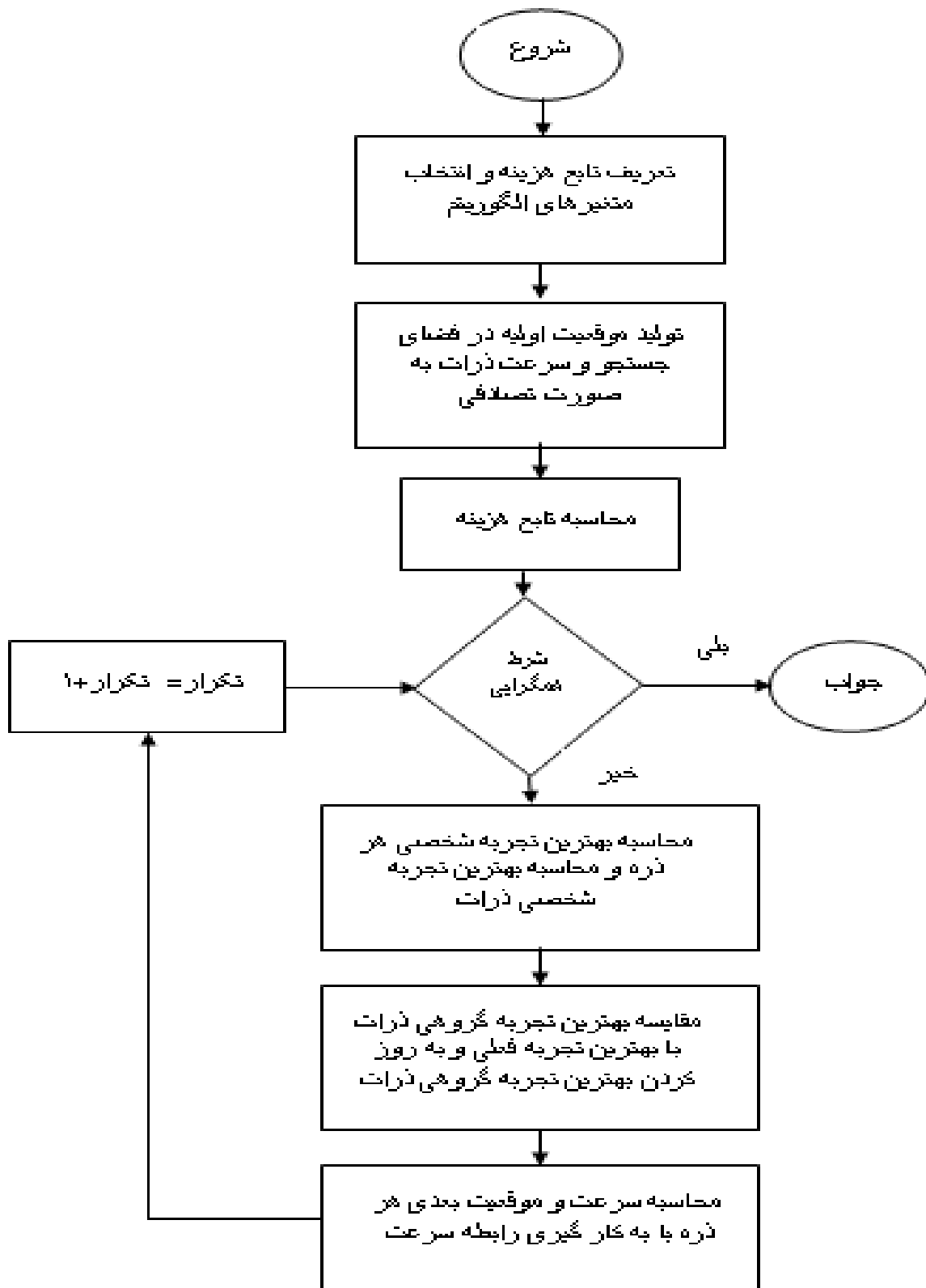
۳-۴ پیاده‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان

فرایند پیاده‌سازی الگوریتم بهینه‌سازی اجتماع پرندگان به وسیله فلوجارت در شکل (۴-۲) قابل

مشاهده است. طبق این شکل مراحل این الگوریتم به صورت زیر قابل بیان است:

- ۱- تعیین متغیرهای کنترلی
- ۲- مقدار دهی اولیه: در طی این مرحله جمعیت بطور تصادفی مقداردهی اولیه می‌شوند و برای تمام ذرات موقعیت و سرعت بطور تصادفی ایجاد می‌شود.
- ۳- ارزیابی جمعیت اولیه: مقدار تابع هزینه به ازای هر ذره ارزیابی می‌گردد و $Gbest$ به عنوان بهترین ذره با بیشترین شایستگی انتخاب می‌شود.
- ۴- حرکت ذرات به سمت موقعیت جدید: جهت سرعت و حرکت هر ذره، براساس بهترین موقعیت خود ذره و بهترین موقعیت تمام ذرات، طبق (۴-۲) و (۴-۳) به‌روز می‌شود.
- ۵- چک کردن موقعیت جدید با محدوده‌های جستجو
- ۶- محاسبه تابع هدف برای موقعیت جدید تمامی ذرات و به‌روز کردن بهترین موقعیت هر ذره و بهترین موقعیت ذرات همسایه
- ۷- بررسی شرط همگرایی: اگر معیار توقف برآورده نشده باشد به مرحله ۴ بر می‌گردد و تکرار می‌شود. در غیر این صورت فرایند بهینه‌سازی پایان می‌یابد (متوقف می‌شود).

۸- نتایج خروجی: بهترین پاسخ بدست آمده توسط این الگوریتم ، خروجی است.



شکل ۴-۲: فلوچارت الگوریتم بهینه سازی پرندگان

۴-۴ مزیت‌های الگوریتم بهینه‌سازی اجتماع پرندگان

اکثر تکنیک‌های تکاملی روند زیر را دنبال می‌کنند:

- ۱- شروع بکار با یک جمعیت تصادفی اولیه
- ۲- محاسبه مقدار شایستگی برای هر جزء
- ۳- تولید مجدد جمعیت بر اساس مقادیر اولیه
- ۴- اگر نیاز برآورده نشده باشد همین روند از مرحله ۲ تکرار می‌شود.

همان‌طور که مشاهده می‌شود این الگوریتم بهینه‌سازی نقاط مشترک زیادی با دیگر الگوریتم‌ها در طی این فرایند مشابه دارد اما وجود یکسری تفاوت‌ها، آن را از سایر روش‌ها متمایز کرده است. مهمترین مساله سادگی الگوریتم بهینه‌سازی اجتماع پرندگان است که از چند نظر قابل بررسی است:

- ◀ پیاده سازی آن بسیار ساده بوده و برنامه آن، از چند خط فراتر نمی‌رود.
- ◀ این سادگی موجب بالا رفتن سرعت در محاسبات و رسیدن سریع به جواب دلخواه با حجم کم حافظه موردنیاز می‌باشد در این الگوریتم تنها $Pbest$ و $Gbest$ یا $Lbest$ اطلاعات را به سایرین می‌دهند و در سیر تکامل، ذره‌ها تحت تاثیر بهترین جواب‌ها حرکت می‌کنند و بنابراین تمام ذره‌ها به سرعت به بهترین جواب در اغلب موارد همگرا می‌شوند.
- ◀ با توجه به پارامترهای کم آن، از لحاظ محاسباتی کم بار است.

۴-۵ مقادیر پارامترهای الگوریتم

W و C_1 و C_2 از پارامترهای الگوریتم بهینه‌سازی اجتماع پرندگان هستند و همگرایی وابسته به مقدار این پارامترهاست. همگرایی شدیداً به مقدار W وابسته است و بهتر است بصورت دینامیک در

بازه ۰.۲ تا ۰.۹ تعریف شوند. C_1 ضریب شتاب مربوط به تجارب شخصی هر ذره است و C_2 ضریب شتاب مربوط به تجارب جمع است، معمولاً در این الگوریتم C_1 و C_2 برابر و عددی بین ۱ تا ۲ قرار می‌دهند.

الگوریتم بهینه‌سازی اجتماع پرندگان که با استفاده از این روابط جمعیت خود را به‌روز کند، الگوریتم بهینه‌سازی پایه یا استاندارد نام دارد.

۴-۶ پارامترهای اصلی الگوریتم بهینه‌سازی پرندگان

الگوریتم پرندگان تحت تاثیر تعدادی از پارامترهای کنترل قرار می‌گیرد که این پارامترهای کنترلی، بعد مسئله نامیده می‌شوند و عبارتند از [27]:

۴-۶-۱ اندازه گروه^۱

همان تعداد ذرات در گروه است. هر چه ذرات بیشتر شود، محدوده اولیه گروه بزرگتر می‌شود که برای مقدار دهی اولیه ذرات استفاده می‌شود. گروه بزرگ، در هر تکرار قسمت‌های بزرگتری را در فضای جستجو تحت پوشش می‌گیرد. هر چند، ذرات بیشتر پیچیدگی محاسبات در هر تکرار را افزایش می‌دهد و جستجو به صورت جستجوی تصادفی موازی را کاهش می‌دهد. در مقایسه با گروه‌های کوچکتر، ذرات بیشتر می‌تواند با تکرارهای کمتر به جواب خوب برسند. در تعدادی از تحقیقات علمی نشان داده شده است که با اندازه‌های کوچک ۱۰ تا ۳۰ ذره در گروه، بهینه‌سازی اجتماع پرندگان قادر به پیدا کردن جواب بهینه می‌باشد. اندازه گروه، وابسته به مسئله است [27].

¹Swarm size

۲-۶-۴ تعداد تکرارها^۱

همچنین تعداد تکرارها برای رسیدن به یک جواب خوب، وابسته به مسئله است. تکرارهای خیلی کم ممکن قبل از موعد، جستجو را خاتمه دهد و تکرارهای بیش از حد، نتیجه‌ای به جز اضافه شدن محاسبات پیچیده‌ای که غیر ضروری هستند را در پی ندارد [27].

۳-۶-۴ ضرایب شتاب^۲

ضرایب شتاب C_1 و C_2 با بردارهای تصادفی $rand_1$ و $rand_2$ تاثیر تصادفی مولفه‌های شناختی و اجتماعی را روی سرعت کلی یک ذره کنترل می‌کند. پارامترهای C_1 و C_2 به عنوان پارامترهای اطمینان نیز نامیده می‌شود. که C_1 اندازه اطمینان ذره به خودش را بیان می‌کند و C_2 اندازه اطمینان ذره به همسایه هایش را بیان می‌کند [27].

اگر $C_1 = C_2 = 0$ باشد، ذرات با سرعت فعلی، پروازشان را ادامه می‌دهند تا به مرز فضای جستجو برخورد کنند [27].

اگر $C_1 > 0$ و $C_2 = 0$ باشد و همچنین موقعیت جدید بهتر باشد، هر ذره بهترین موقعیت در همسایگی اش را در نتیجه‌ی جایگزینی با بهترین موقعیت فعلی به دست می‌آورد. ذرات جستجوی محلی انجام می‌دهند [27].

ذره هیچ‌گونه ارزیابی از جمعیت انجام نمی‌دهد و تنها با حرکت اکتشافی خودش به جستجوی جواب می‌پردازد [27].

اگر $C_1 = 0$ و $C_2 > 0$ ، تمام گروه به یک نقطه جذب می‌شوند. گروه به سمت بهترین ذره تغییر جهت می‌دهند و ذرات از ارزیابی فردی خود استفاده نمی‌کند [27].

¹Number of iterations

²Acceleration coefficient

وقتی که C_1 و C_2 با هم در یک تعادل خوب باشند یعنی $C_1 \approx C_2$ ، ذرات دارای بیشترین تاثیر هستند [27].

اگر $C_1 = C_2$ باشد، ذرات به سمت میانگین $Pbest_i^k$ و $Gbest^k$ جذب می‌شوند که در بیشتر تحقیقات $C_1 = C_2$ استفاده می‌شود و مقدار این ثابت‌ها به مسئله بستگی دارد [27].

اگر $C_2 \gg C_1$ باشد، هر ذره شدیداً به سمت بهترین موقعیت شخصی خودش جذب می‌شود. پس سرگردانی بیش از حد حاصل می‌شود [27].

اگر $C_1 \gg C_2$ باشد، ذرات شدیداً به سمت بهترین موقعیت سراسری جذب می‌شوند که باعث هجوم قبل از موعد ذرات به سمت نقطه بهینه می‌شود [27].

مقادیر زیاد C_1 و C_2 باعث حرکت سریع ذرات و امکان جستجوی فضای بزرگتری را می‌دهد که با کاهش دقت جستجوی فضا همراه خواهد شد، در حالی که مقادیر کم C_1 و C_2 منجر به حرکت آهسته ذرات و افزایش دقت جستجوی فضای می‌گردد [27].

معمولاً C_1 و C_2 مقادیر ثابت هستند و این مقادیر به صورت تجربی به دست آمده است. مقداردهی اولیه اشتباه C_1 و C_2 باعث رفتار تناوبی و واگرایی می‌شود [27].

۴-۶-۴ ضریب اینرسی

همگرایی شدیداً به مقدار W وابسته است در واقع می‌توان به واسطه ی ضریب اینرسی، تاثیر سرعت گذشته را بر سرعت حال کنترل نمود. می‌توان برای برقراری موازنه بین جستجوی سراسری و جستجوی محلی مقدار W را تغییر داد. مقدار زیاد برای W باعث می‌شود که ذرات موجود در الگوریتم به جستجوی مناطق جدیدتر روی بیاورند و یک جستجوی سراسری را انجام دهند در مقابل یک مقدار کم برای W باعث می‌شود که ذرات در منطقه محدودی بمانند و در واقع یک جستجوی

محل^۱ را انجام دهند جستجوی محلی برای دقیق تر کردن جواب های فعلی مناسب است و جستجوی سراسری^۲ برای یافتن جواب های بهتری که احتمالاً در جاهای ناشناخته از فضای جستجو وجود دارند بکار می روند و بهتر است که به صورت دینامیکی تعریف شود. پس با یک مقدار مناسب W می توان باعث ایجاد تعادلی بین جستجوی محلی و جستجوی سراسری شد و در اغلب اوقات باعث کاهش تعداد تکرارهای لازم برای همگرایی به یک جواب مناسب می شود. در الگوریتم بهینه سازی اجتماع پرندگان اولیه، مقدار W ثابت در نظر گرفته می شد. اما نتایج عملی این طور نشان می دهد که بهتر است مقدار W در مراحل ابتدایی، یک مقدار بزرگ در نظر گرفته شود تا جستجوی سراسری و کامل از فضای جستجو صورت گیرد. سپس در طی مراحل اجرای الگوریتم، مقدار W به تدریج کاهش داده می شود تا الگوریتم به مرز همگرایی نزدیک شود و جواب های دقیق تری بدست آورد [۲۳].

۷-۴ معیار توقف^۳ در الگوریتم بهینه سازی اجتماع پرندگان

آخرین مرحله الگوریتم مربوط به شرایط توقف است به عبارت دیگر برای پایان دادن به تکرار جستجو از معیارهایی استفاده می شود. تعیین معیار توقف برای یک فرایند بهینه سازی، یک مسئله مهم در الگوریتم بهینه سازی اجتماع پرندگان است که در سایر الگوریتم ها هم به همان اندازه حائز اهمیت می باشد. برای این منظور معیارهای مختلفی ممکن مورد استفاده قرار بگیرد. شرط توقف را می توان به چند صورت تعریف کرد مثلاً بر روی حداکثر تعداد، یا این که ماکزیمم تغییر در بهترین شایستگی برای تعداد معینی از حرکت جمعیت (S)، کمتر از تلو رانس تعریف شده باشد:

$$|f(p_k^g) - f(p_{k-q}^g)| \leq q \quad \text{where} \quad \varepsilon = 10^{-5} \quad q=1,2,\dots,s \quad (4-4)$$

شرط توقف به مسئله ای که بهینه می شود، وابسته است [۲۹].

¹Local search

²Global search

³Stopping criterion

۴-۸ شاخص عملکرد الگوریتم بهینه سازی اجتماع پرندگان

تابع هدف مطرح شده مبنی بر معیار خطا است. عملکرد یک کنترل کننده بر حسب معیار خطا مورد ارزیابی قرار می گیرد. تعدادی از این معیارها در دسترس هستند و در این جا به صورت زیر ارائه شده است [۱۱]:

- معیار انتگرال قدر مطلق خطا^۱: که بدین صورت به دست می آید:

$$I_{IAE} = \int_0^T |e(t)| dt \quad (۵-۴)$$

- معیار انتگرال مجذور خطا^۲: معیار خطا به صورت معادله زیر داده می شود:

$$I_{ISE} = \int_0^T e^2(t) dt \quad (۶-۴)$$

- معیار انتگرال حاصل ضرب زمان در قدر مطلق خطا^۳: معیار خطا به صورت زیر تعریف می شود:

$$I_{ITAE} = \int_0^T t|e(t)| dt \quad (۷-۴)$$

- معیار میانگین مجذور خطا^۴: معیار خطا به صورت زیر به دست می آید:

$$MSE = E[e^2(t)] \quad (۸-۴)$$

بین چهار شاخص عملکرد فوق کاربرد تابع انتگرال مجذور خطا متداول تر است. چون:

- ✓ این تابع برخی مواقع معیاری از انرژی کنترلی است.
- ✓ مجذور خطا باعث بزرگ شدن مقدار تابع عملکرد می شود.

^۱Integral of Absolute Errors

^۲Integral Square of Errors

^۳Integral of Time multiplied by Absolute Errors

^۴Mean square Error

✓ عملیات ریاضی با این تابع ساده تر است.

۹-۴ یافتن فرمول نحوه حرکات ذرات

برای راه اندازی الگوریتم بهینه سازی اجتماع پرندگان نیاز به دو مورد اساسی داریم [۴۲]:

- وضعیت اولیه

- وضعیت بعدی یا سرعت ذره ها

با استفاده از قوانین مکانیک نیوتن می توان الگوریتم بهینه سازی اجتماع پرندگان را به صورت

زیر بیان نمود:

هر ذره i با جرم m در فضای D بعدی دارای جابجایی $x_i(t)$ می باشد. سرعت و شتاب هر ذره

به صورت زیر بدست می آید:

$$V = \frac{dx}{dt} \quad (۹-۴)$$

$$a = \frac{dV}{dt} = \frac{F}{m} \quad (۱۰-۴)$$

طبق قانون دوم نیوتن، F نیروی وارد بر هر ذره را مشخص می کند. با فرض واحد بودن جرم هر

ذره و Δt برای بکارگیری روابط سرعت و شتاب در فرایند تکراری می توان روابط را بصورت زیر در

آورد:

$$V_t = \frac{X(t) - X(t-1)}{\Delta t} \quad \rightarrow \quad X(t) = X(t-1) + V_t \quad (۱۱-۴)$$

$$\frac{F(t-1)}{m} = \frac{V(t) - V(t-1)}{\Delta t} \quad \rightarrow \quad V(t) = V(t-1) + F(t-1) \quad (۱۲-۴)$$

حال باید نیروی وارد بر ذره را در لحظه ی تکرار $(t-1)$ محاسبه کرد.

۱۰-۴ تجزیه و تحلیل الگوریتم

۱-۱۰-۴ جهت و اندازه گام تصادفی^۱ الگوریتم بهینه سازی اجتماع پرندگان

با توجه به معادلات سرعت و حرکت ذره [۲۴]:

$$V_i^{k+1} = WV_i^k + \frac{C_1 rand_1 (Pbest_i^k - X_i^k)}{\Delta t} + \frac{C_2 rand_2 (Gbest^k - X_i^k)}{\Delta t} \quad (۱۳-۴)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \Delta t \quad (۱۴-۴)$$

با جایگزینی معادله به روز رسانی سرعت (۱۳-۴) در معادله (۱۴-۴)، عبارت زیر را بدست می-

آوریم [۲۴]:

$$X_i^{k+1} = X_i^k + (WV_i^k + \frac{C_1 rand_1 (Pbest_i^k - X_i^k)}{\Delta t} + \frac{C_2 rand_2 (Gbest^k - X_i^k)}{\Delta t}) \Delta t \quad (۱۵-۴)$$

با فاکتورگیری $C_1 rand_1$ و $C_2 rand_2$ از معادله بالا، معادله کلی به شرح ذیل را بدست می-

آوریم [۲۴]:

$$X_i^{k+1} = X_i^k + Wv_i^k \Delta t + (C_1 rand_1 + C_2 rand_2) \left(\frac{C_1 rand_1 Pbest_i^k + C_2 rand_2 Gbest^k}{C_1 rand_1 + C_2 rand_2} - X_i^k \right) \quad (۱۶-۴)$$

شیب کلی جستجو خطی به شکل زیر است [۲۴]:

$$X_i^{k+1} = \hat{X}_i^k + \alpha p^{-k} \quad (۱۷-۴)$$

به طوری که:

$$\hat{X}_i^k = X_i^k + WV_i^k \Delta t \quad (۱۸-۴)$$

$$\alpha = C_1 rand_1 + C_2 rand_2 \quad (۱۹-۴)$$

^۱Stochastic step size

$$p^{-k} = \frac{C_1 rand_1 Pbest + C_2 rand_2 Gbest}{C_1 rand_1 + C_2 rand_2} - X_i^k \quad (20-4)$$

α : می توان به عنوان اندازه گام تصادفی در نظر گرفت.

p^{-k} : می توان به عنوان جهت جستجوی تصادفی در نظر گرفت.

اندازه گام تصادفی با انتخاب پارامترهای اجتماعی و شناختی تعیین خواهد شد.

با دانستن $rand_1 + rand_2 \in [0,1]$ ، مقدار α در بازه $\alpha \in [0, C_1 + C_2]$ قرار می گیرد

و مقدار میانگین آن برابر است با [24]:

$$\frac{[C_1 + C_2]}{2} \quad (21-4)$$

همانند α ، مقدار p^{-k} در بازه:

$$p^{-k} \in \left[-X_i^k, \frac{C_1 Pbest_i^k + C_2 Gbest^k}{C_1 + C_2} - X_i^k \right] \quad (22-4)$$

معادله (20-4) ، نه تنها به پارامترهای شناختی و اجتماعی وابسته است بلکه به مقدار $Pbest$

و $Gbest$ نیز بستگی دارد.

۲-۱۰-۴ همگرایی الگوریتم بهینه سازی اجتماع پرندگان

با جایگزینی معادله (4-13) در معادله (4-14) و آرایش دوباره رابطه موقعیت، شکل کلی

معادله موقعیت برای i امین ذره در $k + 1$ امین تکرار، به صورت زیر است [24]:

$$X_i^{k+1} = X_i^k (1 - C_1 rand_1 - C_2 rand_2) + WV_i^k \Delta t + C_1 rand_1 Pbest_i^k + C_2 rand_2 Gbest^k \quad (23-4)$$

و معادله سرعت ذره نیز بدین است [24]:

$$V_i^{k+1} = -X_i^k \frac{C_1 rand_1 + C_2 rand_2}{\Delta t} + W V_i^k + C_1 rand_1 \frac{Pbest_i^k}{\Delta t} + C_2 rand_2 \frac{Gbest^k}{\Delta t} \quad (24-4)$$

معادلات (۲۳-۴) و (۲۴-۴) با هم ترکیب شده و به شکل ماتریس زیر نوشته می شود [۲۴]:

$$\begin{bmatrix} X_i^{k+1} \\ V_i^{k+1} \end{bmatrix} = \begin{bmatrix} 1 - C_1 rand_1 - C_2 rand_2 & W \Delta t \\ -\frac{(C_1 rand_1 + C_2 rand_2)}{\Delta t} & W \end{bmatrix} \begin{bmatrix} X_i^k \\ V_i^k \end{bmatrix} + \begin{bmatrix} C_1 rand_1 & C_2 rand_2 \\ \frac{C_1 rand_1}{\Delta t} & \frac{C_2 rand_2}{\Delta t} \end{bmatrix} \begin{bmatrix} Pbest_i^k \\ Gbest^k \end{bmatrix} \quad (25-4)$$

این معادله می تواند به عنوان نمایش دینامیک گسسته برای الگوریتم بهینه سازی اجتماع پرندگان در نظر گرفته شود که $[X_i, v_i]^T$ حالت مورد نظر از ورودی خارجی $[Pbest_i, Gbest]^T$ است و ماتریس های اول و دوم به ترتیب متناظر با ماتریس های دینامیک و ورودی می باشند. با فرض این که هیچ تحریک کننده خارجی در دینامیک سیستم وجود ندارد بنابراین $[Pbest_i, Gbest]^T$ ثابت است (ذرات دیگر موقعیت های بهتری می توانند پیدا کنند) پس رفتار همگرایی می تواند برقرار شود [۲۴].

اگر در چنین موردی، هنگامی که $k \rightarrow \infty$ پس $[X_i^{k+1}, V_i^{k+1}]^T = [X_i^k, V_i^k]^T$ که شکل

استاندارد دینامیک به صورت زیر ساده می شود [۲۴]:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 - C_1 rand_1 - C_2 rand_2 & W \Delta t \\ -\frac{(C_1 rand_1 + C_2 rand_2)}{\Delta t} & W \end{bmatrix} \begin{bmatrix} X_i^k \\ V_i^k \end{bmatrix} + \begin{bmatrix} C_1 rand_1 & C_2 rand_2 \\ \frac{C_1 rand_1}{\Delta t} & \frac{C_2 rand_2}{\Delta t} \end{bmatrix} \begin{bmatrix} Pbest_i^k \\ Gbest^k \end{bmatrix} \quad (26-4)$$

تنها زمانی درست است که $V_i^k = 0$ و هم $X_i^k, Pbest_i^k$ با $Gbest^k$ منطبق شوند. بدین صورت چنین نقطه ای لزوماً یک مینیمم محلی و سراسری نیست، جایگزین یک نقطه تعادل شده است. اگر تحریک کننده خارجی در دینامیک سیستم وجود داشته باشد چنین نقطه ای نزدیک بهینه قرار خواهد گرفت بنابراین بهتر است که $Pbest_i$ و $Gbest$ از فرایند بهینه سازی پیدا شوند [۲۴].

۳-۱۰-۴ پایداری الگوریتم بهینه سازی اجتماع پرندگان

مقادیر ویژه ماتریس از معادله (۴-۲۶) بدست آمده است که دید وسیعی از رفتار دینامیکی و پایداری سیستم، می تواند فراهم کند. معادله مشخصه ماتریس به صورت زیر بدست آمده است [۲۴]:

$$\lambda^2 - (W - C_1rand_1 - C_2rand_2 + 1)\lambda + W = 0 \quad (۲۷-۴)$$

که مقادیر ویژه به صورت زیر داده شده است:

$$\lambda_{1,2} = \frac{1 + W - C_1rand_1 - C_2rand_2 \pm \sqrt{(1 + W - C_1rand_1 - C_2rand_2)^2 - 4W}}{2} \quad (۲۸-۴)$$

شرط لازم و کافی برای پایداری یک سیستم گسسته اینست که همه مقادیر ویژه های (λ) بدست آمده از ماتریس در داخل دایره واحد در صفحه مختلط قرار بگیرد.

بنابراین $|\lambda_i| = 1, \dots, n < 1$. تجزیه و تحلیل مقادیر ویژه بدست آمده از (۴-۲۷) منجر به بدست آوردن مجموعه ای از شرایط پایداری برای این الگوریتم می شود:

$$C_1rand_1 + C_2rand_2 > 0$$

$$\frac{C_1rand_1 + C_2rand_2}{2} - W < 1 \quad (۲۹-۴)$$

$$W < 1$$

با توجه به شرایط ذکر شده در بالا ضریب اینرسی در بازه زیر قرار می گیرد:

$$\frac{C_1rand_1 + C_2rand_2}{2} - 1 < W < 1 \quad (۳۰-۴)$$

و حد بالا برای $C_1rand_1 + C_2rand_2$ می تواند به صورت زیر باشد:

$$C_1rand_1 + C_2rand_2 < 4 \quad (۳۱-۴)$$

با دانستن $rand_1, rand_2 \in [0,1]$ انتخاب پارامتر به صورت زیر باشد:

$$0 < C_1 + C_2 < 4 \quad (۳۲-۴)$$

$$\frac{C_1 + C_2}{2} - 1 < W < 1 \quad (۳۳-۴)$$

و مطابق با شرایط مشخص شده در (۴-۲۹) باشد سیستم، همگرایی به نقطه تعادل را تضمین

کرده است.

۱۱-۴ بهبود الگوریتم بهینه سازی اجتماع پرندگان

الگوریتم بهینه سازی پرندگان و بقیه الگوریتم های جستجوی تصادفی دو مشکل اساسی دارند:

۱. همگرایی زودرس است. در صورت عدم تنظیم مناسب پارامترها، الگوریتم بهینه سازی

پرندگان به سوی بهینه محلی متمایل شده و دچار همگرایی زودرس می شود که علت

آن ثابت ماندن ذره ای است که شامل G_{best} می باشد در نتیجه سایر ذرات به سرعت

به این ذره همگرا می شود [۲۵]. اگر چه الگوریتم بهینه سازی پرندگان جواب های

مناسب را بسیار سریع تر از بقیه الگوریتم های تکاملی پیدا می کند ولی معمولاً نمی تواند

همزمان با افزایش تکرارها کیفیت جواب بدست آمده را بهبود بخشد. هنگامی که

بهینه سازی مسائل الگوریتم پرندگان دچار همگرایی زودرس می شود و این مشکل به

دلایل گوناگونی بروز می کند یک دلیل این است که در حالت همسایگی سراسری

ذرات به یک نقطه همگرا می شوند که این نقطه بین بهترین ذره موجود در گروه و

بهترین موقعیت خود ذره قرار دارد. این نقطه همیشه بهینه نیست.

۲. وجود سرعت بالای اطلاعات بین ذرات است که باعث ایجاد ذرات مشابه می شود و در

نتیجه تنوع را کاهش می دهد و باعث افزایش احتمال قرار گرفتن در بهینه محلی

خواهد شد [۲۶]. جهت جلوگیری از مشکلات ذکر شده، تغییراتی در الگوریتم بهینه-

سازی پرندگان استاندارد ایجاد شده است. این تغییرات معمولاً سعی می کنند با افزایش تنوع و گوناگونی جوابها به حل مساله بپردازند.

بررسی های زیادی جهت همگرایی الگوریتم پرندگان صورت گرفته است از جمله این بررسی ها مطالعه بر روی مقدار W است. مقدار و نقش W در همگرایی الگوریتم بهینه سازی اجتماع پرندگان بسیار اثر گذار است. این وزن برای کنترل سرعت فعلی به کار رفته است. به عنوان یک نکته کلی، محققین دریافته اند که اگر ضریب اینرسی متغیر با زمان انتخاب شود، ضریب اینرسی بطور خطی کاهش یابد عملکرد بهتری را به نسبت یک مقدار ثابت دارد. ضریب اینرسی بزرگتر ضمن تغییرات افزایشی بیشتر سرعت در هر مرحله زمانی، به معنای اکتشاف نواحی جدید جستجو است. ضریب اینرسی کوچکتر به معنای تغییرات کمتر سرعت است که برای جستجوی محلی دقیق، به روز رسانی کندتر می شود و مقدار پایین آن در بهره برداری تاثیر می گذارد.

همان طور که گفتیم همگرایی الگوریتم پرندگان شدیداً به مقدار W وابسته است و می توان برای برقراری موازنه بین جستجوی سراسری و جستجوی محلی مقدار W را تغییر داد. مقدار زیاد برای W باعث می شود که ذرات موجود در الگوریتم به جستجوی مناطق جدیدتر روی بیاورند و یک جستجوی سراسری را انجام دهند در مقابل یک مقدار کم برای W باعث می شود که ذرات در منطقه محدودی بمانند و در واقع یک جستجوی محلی را انجام دهند جستجوی محلی برای دقیق تر کردن جواب های فعلی مناسب است و جستجوی سراسری برای یافتن جواب های بهتری که احتمالاً در جاهای ناشناخته از فضای جستجو وجود دارند بکار می روند.

۱۲-۴ الگوریتم بهینه‌سازی اجتماع پرندگان ترکیبی

روش‌های ترکیبی بهینه‌سازی پرندگان از دو یا چند روش بهینه‌سازی تشکیل شده است. یک روش الگوریتم بهینه‌سازی پرندگان ترکیبی، از الگوریتم پرندگان با یک یا چند روش بهینه‌سازی تشکیل شده است. الگوریتم بهینه‌سازی پرندگان با الگوریتم ژنتیک ترکیب شده است این روش برای بهبود همگرایی پارامترها و بهبود عملکرد الگوریتم پیشنهاد شده است [۲۷].

روش ترکیبی دوم این است که الگوریتم بهینه‌سازی پرندگان با روش برنامه‌ریزی درجه دو متوالی^۱ ترکیب شده است و برای کنترل بهینه غیرخطی مسائل کنترلی استفاده شده است [۱۷].

¹Successive quadratic programming

فصل پنجم:

بهینه‌سازی توسط الگوریتم اجتماع پرندگان

در چند سال گذشته، مطالعه تئوری بهینه‌سازی بکار برده شده در سیستم‌های کنترلی با علاقه زیادی ادامه یافته است و توجه زیادی به بهینه‌سازی رفتار سیستم‌ها شده است. به عنوان مثال، دامنه تغییرات یک موشک را به حداکثر می‌رساند، سود یک شرکت را به حداکثر می‌رساند، خطا در تخمین موقعیت یک هدف را به حداقل می‌رساند، انرژی یا هزینه لازم برای دستیابی به برخی از حالت خروجی مورد نیاز را به حداقل می‌رساند. کنترل زمانی به هدف مطلوبش می‌رسد که، معیار مشخص سیستم را که شامل مسئله اصلی تئوری بهینه‌سازی است را به حداقل یا حد اکثر برساند [۲۸]. یکی از فعال‌ترین موضوعات در مهندسی کنترل، کنترل بهینه سیستم‌های غیرخطی است. چنین مسائلی کنترلی در کاربردهای زیادی به عنوان مثال در اقتصاد، مهندسی شیمی، رباتیک، هوا و فضا پدیدار می‌شود. حل کنترل بهینه چنین سیستم‌هایی دارای چندین نقطه بهینه محلی است و همچنین روش‌های سنتی بهینه‌سازی مانند روش گرادیان و روش نیوتن از حل مسئله عاجزند، می‌توان با روش‌های هوشمند بهینه‌سازی را انجام داد [۱۷]. در سال‌های اخیر، الگوریتم بهینه‌سازی پرندگان را به منظور حل مسائل بهینه سیستم‌های پیچیده که کاملاً غیرخطی و چند متغیره با کوپلینگ شدید هستند، به کار برده شده است. در این فصل به منظور کنترل بهینه بازوی مکانیکی ربات، از الگوریتم بهینه‌سازی پرندگان روی قانون کنترل بیان شده در (۲-۴)، استفاده شده است. پارامترهای طراحی کنترل به منظور مینیمم کردن خطای ردگیری، بهینه می‌شوند.

۱-۵ بهینه‌سازی

بیشتر مسائل بهینه‌سازی به فرم ریاضی به صورت زیر بیان می‌شود [۴۱]:

$$\begin{aligned} \text{minimize } & f_i(x) & i = 1, 2, \dots, i \\ \text{subject to } & \Phi_j(x) = 0 & j = 1, 2, \dots, j \\ & \psi_k(x) \leq 0 & k = 1, 2, \dots, k \end{aligned} \quad (۱-۵)$$

که $f_i(x)$ تابع هدف (یا تابع معیار) است که باید بهینه شود و $\Phi_j(x)$ و $\psi_k(x)$ به ترتیب

قیده‌های مساوی و نامساوی هستند که در آن $x = (x_1, x_2, \dots, x_n) \in R^n$ متغیر اصلی و مستقل مساله است [۴۱].

۵-۲ انواع مسایل بهینه‌سازی

می‌توان مسایل بهینه‌سازی را از دیدگاه مختلف به دسته‌های متعددی تقسیم بندی کرد:

۵-۲-۱ بهینه‌سازی با سعی و خطا و بهینه‌سازی روی تابع

سعی و خطا فرایندی است که در آن متغیرهای ورودی تغییر داده می‌شوند و اطلاعات دقیق در خصوص نحوه تاثیر هر متغیر بر خروجی در دست نیست. در مقابل نوع دیگری از بهینه‌سازی وجود دارد که در آن ماهیت مساله به صورت فرمول دقیق در دست است و می‌توان با روش‌های ریاضی به سراغ این‌گونه مسایل رفت [۴۱].

۵-۲-۲ بهینه‌سازی تک‌بعدی و بهینه‌سازی چندبعدی

اگر فقط یک متغیر در مساله بهینه‌سازی وجود داشته باشد، بهینه‌سازی تک‌بعدی نامیده می‌شود. در مقابل مسایلی که دارای بیش از یک متغیر باشند مسایل بهینه‌سازی چندبعدی خوانده می‌شود [۴۱].

۵-۲-۳ بهینه‌سازی پویا و بهینه‌سازی ایستا

اگر تابع یا فرایندی که مورد بهینه‌سازی واقع می‌شود، تابعی از زمان باشد و با گذشت زمان تغییری روی آن‌ها به وجود نمی‌آورد، بهینه‌سازی ایستا خوانده می‌شود [۴۱].

۴-۲-۵ بهینه‌سازی گسسته و بهینه‌سازی پیوسته

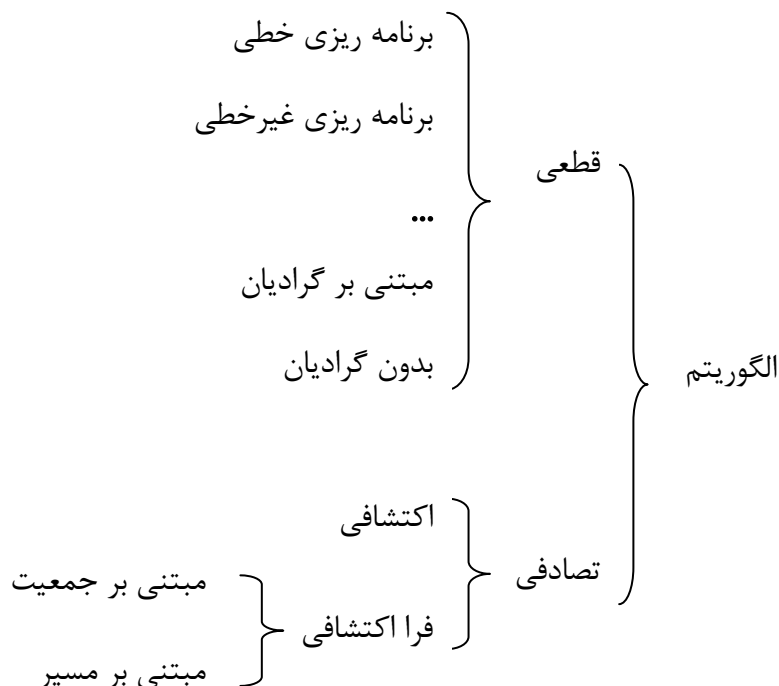
اگر ماهیت متغیرهای مساله بهینه‌سازی پیوسته باشد، آن را پیوسته می‌نامند. در مقابل اگر مقادیری که متغیرهای مساله اختیار می‌کنند، مقادیر محدود و شمارا باشند، مساله را گسسته می‌نامند [۴۱].

۵-۲-۵ بهینه‌سازی کمینه‌جو و بهینه‌سازی تصادفی

برخی از روش‌ها از یک نقطه مشخص در فضای جستجو شروع می‌کنند و با استفاده از قوانینی که در ریاضیات و هندسه دارند، جواب‌های بهتری بدست می‌آورند. این نوع الگوریتم‌ها سرعت بسیار بالایی در هم‌گرایی دارند، اما به راحتی در کمینه‌ها یا بیشینه‌های محلی گرفتار می‌شوند. در این روش‌ها نحوه ایجاد جواب‌های بعدی از روی جواب‌های فعلی، روندی مشخص و معلوم دارد. در مقابل روش‌های تصادفی، از الگوهای احتمالی برای ایجاد جواب‌های بهتر استفاده می‌کنند و به این ترتیب پیش‌بینی عملکرد الگوریتم ساده نیست. سرعت هم-گرایی این نوع از الگوریتم‌ها در مقایسه با الگوریتم‌های کمینه‌جو، کمتر است اما احتمال گرفتار شدن در نقاط بهینه‌ی محلی نیز کمتر می‌شود و امید بیشتری برای یافته شدن نقطه بهینه سراسری وجود دارد [۴۱].

۳-۵ الگوریتم‌های بهینه‌سازی:

بصورت کلی الگوریتم بهینه‌سازی را می‌توان به صورت زیر تقسیم‌بندی کرد [۲۱]:



در این جا بهینه‌سازی هوشمند به دسته‌ی وسیعی از الگوریتم‌های بهینه‌سازی تصادفی مبتنی بر جمعیت اطلاق می‌شود، مانند الگوریتم ژنتیک، الگوریتم بهینه‌سازی اجتماع پرندگان و غیره. اگر مساله‌ی بهینه‌سازی پیچیده، تصادفی، غیرخطی از درجه بالا با بهینه‌ی محلی چندگانه باشد، الگوریتم‌های بهینه‌سازی هوشمند در مقایسه با روش‌های بهینه‌سازی کلاسیک پرمزیت محسوب می‌شوند. از مزایای بهینه‌سازی هوشمندتوازی ذاتیشان، قابلیت حل کردن مسائل عظیم و پیچیده، کمترین نیاز به دانشی خاص در حیطه کار و غیره می‌توان اشاره کرد. الگوریتم‌های بهینه‌سازی کلاسیک ممکن است در دام بهینه محلی بیفتند. ثابت شده است که الگوریتم‌های بهینه‌سازی هوشمند در فرار از بهینه محلی و پیدا کردن بهینه‌ی سراسری در فضاهای جستجوی پیچیده موثر هستند.

در این فصل، الگوریتم بهینه‌سازی اجتماع پرندگان به منظور طراحی کنترل بهینه به

وسیله پیدا کردن پارامترهای کنترل بهینه استفاده می‌شود.

۴-۵ طراحی کنترل کننده

با وجود عدم قطعیت‌ها، قانون کنترل گشتاور به صورت زیر پیشنهاد شده است [۲]:

$$\tau = Y(q, \dot{q}, \ddot{q}_r) \hat{\beta} + K\sigma \quad (۲-۵)$$

قانون تطبیق می‌تواند به صورت زیر بیان شود [۲]:

$$\dot{\hat{\beta}} = -\Gamma Y^T \sigma \quad (۳-۵)$$

مسیر مطلوب به صورت زیر در نظر گرفته خواهد شد [۴]:

$$q_{d1}(t) = \frac{\pi}{6} (1 - \cos(2\pi t)) \quad (۴-۵)$$

$$q_{d2}(t) = \frac{\pi}{4} (1 - \cos(2\pi t)) \quad (۵-۵)$$

پارامترهای قانون کنترل که با استفاده از الگوریتم بهینه‌سازی پرندگان به دست می‌آید به

صورت زیر است.

جدول ۵-۱: پارامترهای کنترل برای بهینه سازی

پارامترهای کنترل یا ذرات					
1	2	3	4	5	6
Λ_1	Γ_1	Γ_2	Γ_3	Γ_4	K_1

۵-۵ الگوریتم بهینه‌سازی اجتماع پرندگان^۱

انواع الگوریتم‌های بهینه‌سازی پرندگان با روش‌های دینامیکی مختلف ضریب اینرسی و ضرایب

یادگیری را می‌توان به صورت زیر تقسیم‌بندی کرد و در این پایان‌نامه از این روش‌ها در بهینه‌سازی

^۱Particle Swarm Optimization (PSO)

پارامترهای کنترل کننده استفاده شده است و به مقایسه این روش‌ها با یکدیگر خواهیم پرداخت.

۵-۵-۱ الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی کاهش

خطی (LDW-PSO)^۱

در این روش ضریب اینرسی الگوریتم بهینه‌سازی اجتماع پرندگان بصورت خطی کاهش می‌یابد. در ابتدا ضریب اینرسی بزرگ، معمولاً ۰.۹ می‌باشد که بصورت خطی کاهش می‌یابد و ضریب اینرسی به مقدار کوچک ۰.۴ می‌رسد [۳۱،۳۰].

معادله‌ای که برای بروزرسانی ضریب اینرسی در الگوریتم بهینه‌سازی استفاده می‌شود بصورت زیر است:

$$W^t = W_{\min} + \frac{\text{iter}_{\max} - t}{\text{iter}_{\max}} \cdot (W_{\max} - W_{\min}) \quad (۶-۵)$$

که iter_{\max} حد اکثر تعداد تکرارها (گام‌های زمانی) برای الگوریتم که اجرا می‌شود، t تکرار فعلی (گام زمانی) الگوریتم، W_{\min} مقدار مینیمم ضریب اینرسی و W_{\max} مقدار ماکزیمم ضریب اینرسی است. W_{\min} و W_{\max} به صورت $0 < W_{\min} < W_{\max} < 1$ انتخاب می‌شوند.

۵-۵-۲ الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی کاهش

غیرخطی (NDW-PSO)^۲

در این روش ضریب اینرسی الگوریتم بهینه‌سازی اجتماع پرندگان بصورت غیر خطی کاهش می‌یابد. این روش به منظور افزایش عملکرد الگوریتم ارائه شده است. در مقایسه با روش کاهش خطی، روش کاهش غیرخطی برای جستجو به زمان کمتری نیاز دارد و با زمان بیشتر جواب‌های بهتری (بهره‌برداری) را پیدا می‌کند. ضریب اینرسی غیرخطی در هر تکرار بصورت زیر تعریف شده است [۳۲].

^۱Linear Inertia Weight PSO

^۲Nonlinear Inertia Weight PSO

$$W^t = W_{\min} + \left(\frac{\text{iter}_{\max} - t}{\text{iter}_{\max}} \right)^n \cdot (W_{\max} - W_{\min}) \quad (7-5)$$

که iter_{\max} حد اکثر تعداد تکرارها (گام‌های زمانی) برای الگوریتم که اجرا می‌شود، t تکرار فعلی (گام زمانی) الگوریتم، W_{\min} مقدار مینیمم ضریب اینرسی و W_{\max} مقدار ماکزیمم ضریب اینرسی است. W_{\min} و W_{\max} به صورت $0 < W_{\min} < W_{\max} < 1$ انتخاب می‌شوند [۳۲].

۳-۵-۵ الگوریتم بهینه‌سازی اجتماع پرندگان با ضریب اینرسی

دینامیکی (DIW-PSO)^۱

در الگوریتم بهینه‌سازی پرندگان استاندارد، اطلاعات بهترین فردی و بهترین سراسری، به ذرات مراحل بعدی داده می‌شود. الگوریتم بهینه‌سازی پرندگان با ضریب اینرسی دینامیکی، روش بهبودیافته الگوریتم بهینه‌سازی پرندگان است. فرض کنید که فضای جستجو، D بعدی و دارای m ذره باشد.

i امین ذره در یک D بعدی به صورت زیر نمایش داده می‌شود:

$$X_i (i = 1, 2, \dots, m) \quad (8-5)$$

این بدین معنی است که i امین ذره در فضای جستجو به صورت زیر ارائه می‌شود:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad (i = 1, 2, \dots, m) \quad (9-5)$$

شایستگی ذره، با قرار دادن موقعیتش در تابع هدف حساب می‌شود. زمانی که شایستگی کمترین مقدار را دارد، X_i متناظر با آن بهترین مقدار را دارد. معادله‌ای که برای ضریب اینرسی در هر تکرار استفاده می‌شود بدین صورت است [۳۳]:

$$W^t = W * u^{-t} \quad (10-5)$$

به طوری که $W \in [0, 1]$ و $u \in [1.0001, 1.0005]$ می‌باشد.

¹Dynamic Inertia Weight PSO

۴-۵-۵ الگوریتم بهینه‌سازی اجتماع پرندگان بهبودیافته^۱

در این روش در طی جستجو، هنگامی که مقدار شایستگی یک ذره دور از مقدار واقعی بهینه سراسری باشد، مقدار ضریب اینرسی بزرگ خواهد شد که نتیجه‌اش، جستجوی سراسری و پیدا کردن مناطق جدید جستجو می‌باشد. در ضمن، هنگامی که مقدار شایستگی یک ذره نزدیک مقدار واقعی بهینه بدست بیاید، مقدار ضریب اینرسی کوچک خواهد شد، که این کوچک شدن، به نزدیکی بهترین شایستگی مقدار بهینه وابسته است. این مقدار کم ضریب اینرسی باعث همگرایی سریع در مکان بهینه محلی می‌شود. انتخاب مناسب ضرایب یادگیری منجر به تسریع همگرایی الگوریتم و جلوگیری از همگرایی زودرس در بهینه‌های محلی می‌شود.

معادله‌های که برای بروزرسانی ضریب اینرسی و ضرایب یادگیری در هر مرحله از الگوریتم بهینه‌سازی بهبودیافته استفاده می‌شود به صورت زیر است [۳۴]:

$$W^t = \frac{1}{(1 + \exp(-\alpha F(gbest^t)))^n} \quad (11-5)$$

$$C_i = \frac{1}{(1 + \exp(-\alpha F(gbest^t)))^n} \quad i = 1,2 \quad (12-5)$$

$$\alpha = \frac{1}{F(gbest^t)} \quad (13-5)$$

$F(gbest^t)$ شایستگی بهینه سراسری در i امین تکرار است. α و n پارامترهای ثابت هستند.

۵-۵-۵ الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از روش پیشنهادی

(Fuzzy-PSO)

این الگوریتم روش پیشنهادی ما می‌باشد که سیستم فازی برای تنظیم پارامترهای ضرایب

^۱Improved PSO

یادگیری C_1 و C_2 استفاده شده است و برای تنظیم پارامتر W از الگوریتم بهینه‌سازی پرندگان بهبود یافته استفاده شده است. برای این الگوریتم C_1 و C_2 تابع‌های فازی هستند که پارامترهای $NCBPE$ و $Iteration$ به عنوان ورودی‌های تابع، C_1 و C_2 خروجی‌های تابع می‌باشند. مقدار نرمال شده سه مقدار بهترین، بدترین و حالت فعلی مقدار شایستگی تابع می‌باشد که دارای مقداری در بازه $[0,1]$ است و معادله آن به صورت زیر است [۳۵،۳۶]:

$$NCBPE = \frac{CBPE - CBPE_{min}}{CBPE_{max} - CBPE_{min}} \quad (۱۴-۵)$$

که $CBPE$ مقدار شایستگی ذره در حالت فعلی را نشان می‌دهد. $CBPE_{min}$ بهترین شایستگی است ذره تا به حال بدست آورده است. $CBPE_{max}$ بدترین شایستگی است که ذره تا به حال بدست آورده است.

معادله $Iteration$ به عنوان یکی دیگر از ورودی‌های سیستم فازی بصورت زیر نمایش داده می‌شود [۳۶،۳۵]:

$$Iteration = \frac{Iteration}{Iteration_{max}} \quad (۱۵-۵)$$

که $Iteration$ تکرار فعلی الگوریتم و $Iteration_{max}$ ماکزیمم تکرار الگوریتم می‌باشد.

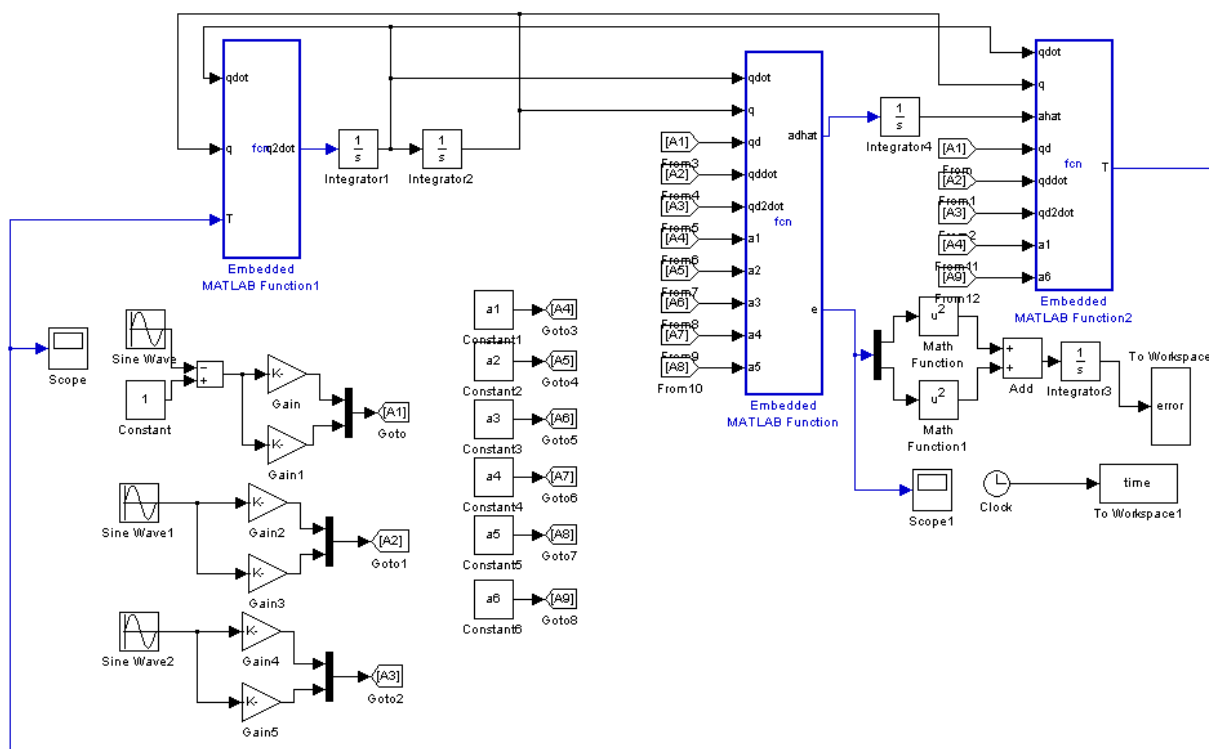
و برای تنظیم ضریب اینرسی W در هر تکرار از معادله زیر استفاده می‌شود [۱۷]:

$$W^t = \frac{1}{(1 + \exp(-\alpha F(gbest^t)))^n} \quad (۱۶-۵)$$

فصل ششم:

نتایج شبیه‌سازی

در این بخش مطالعات و شبیه سازی ها بر روی بازوی ماهر ربات دو لینکی انجام می شود. ابتدا روش الگوریتم بهینه سازی اجتماعی پرندگان با ضریب اینرسی کاهش خطی، سپس الگوریتم بهینه سازی پرندگان با ضریب اینرسی کاهش غیرخطی، بعد از آن الگوریتم بهینه سازی پرندگان با ضریب اینرسی دینامیکی و الگوریتم بهینه سازی بهبود یافته و در نهایت الگوریتم پیشنهادی بهینه سازی پرندگان با استفاده از منطق فازی را برای سرعت و دقت همگرایی پارامترها برای بهینه سازی کنترل کننده بازوی ماهر بکار خواهیم برد. در این فصل این الگوریتم ها را شبیه سازی و نتایج شبیه سازی های به دست آمده را با یکدیگر مقایسه می کنیم.



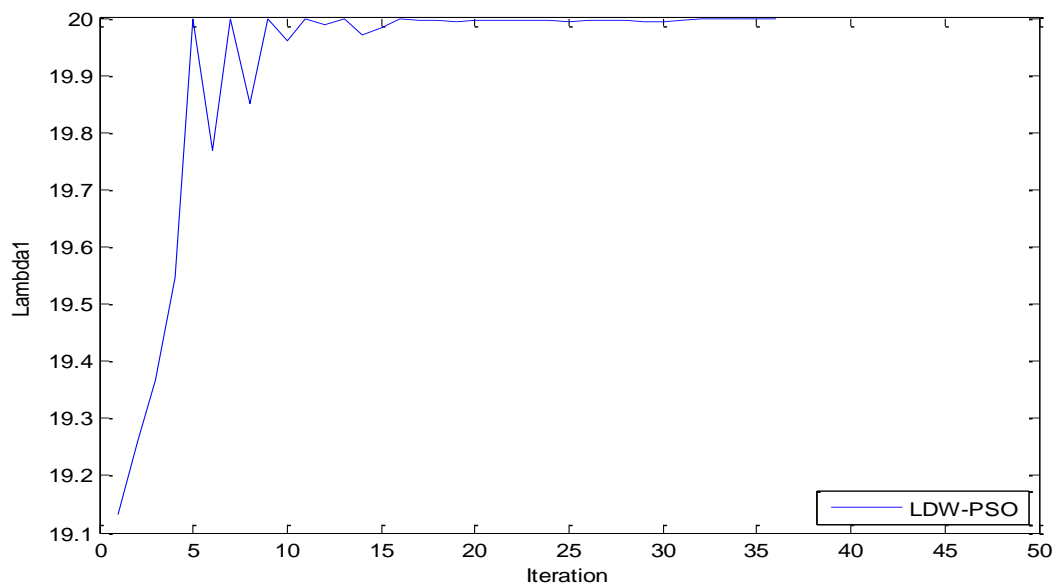
شکل ۶-۱: بلوک دیاگرام شبیه سازی بازوی ربات ماهر بهینه شده

۱-۶ شبیه سازی الگوریتم بهینه سازی اجتماع پرندگان با استفاده از ضریب

اینرسی کاهش خطی (LDW-PSO)

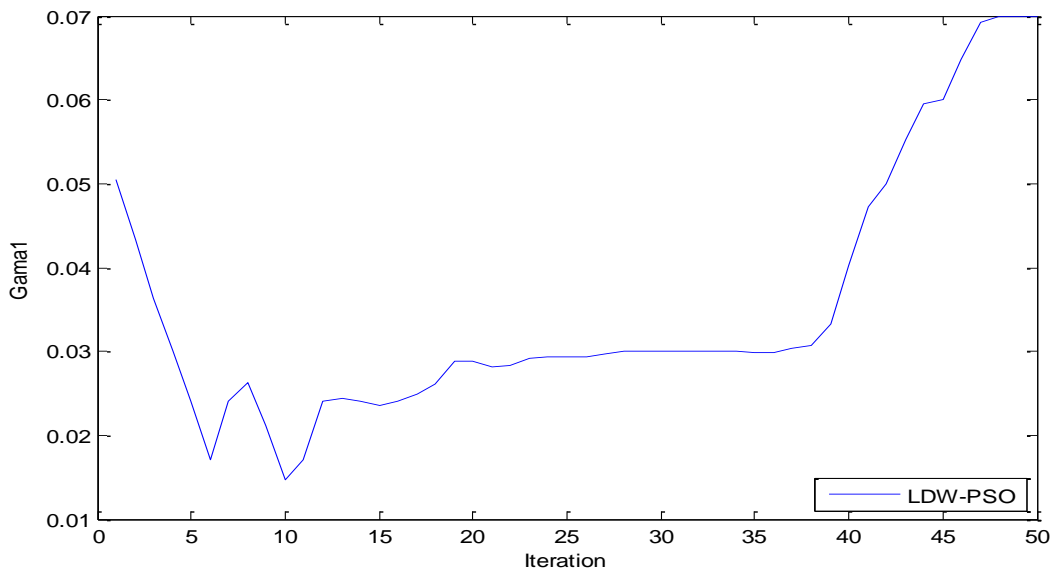
این الگوریتم با استفاده از ضریب اینرسی کاهش خطی برای بهبود عملکرد الگوریتم استاندارد پیشنهاد شده است. در این الگوریتم ضریب اینرسی W به صورت خطی کاهش می یابد که از معادله (۲۶-۵) برای ضریب اینرسی استفاده می شود. این الگوریتم برای بهینه سازی پارامترهای کنترل کننده و مینیمم کردن خطای ردگیری بکار برده شده است. در این الگوریتم برای بهبود عملکرد مسئله بازوی ماهر ربات، ضریب اینرسی ماکزیمم $W_{max} = 0.9$ و ضریب اینرسی مینیمم $W_{min} = 0.4$ ، ضریب یادگیری $C_1 = C_2 = 2$ و تعداد ذرات 10 و تعداد تکرار 50 و هر الگوریتم 25 بار اجرا به کار گرفته شده است.

شکل (۲-۶) روند همگرایی مسیر پارامتر Λ_1 را نشان می دهد.



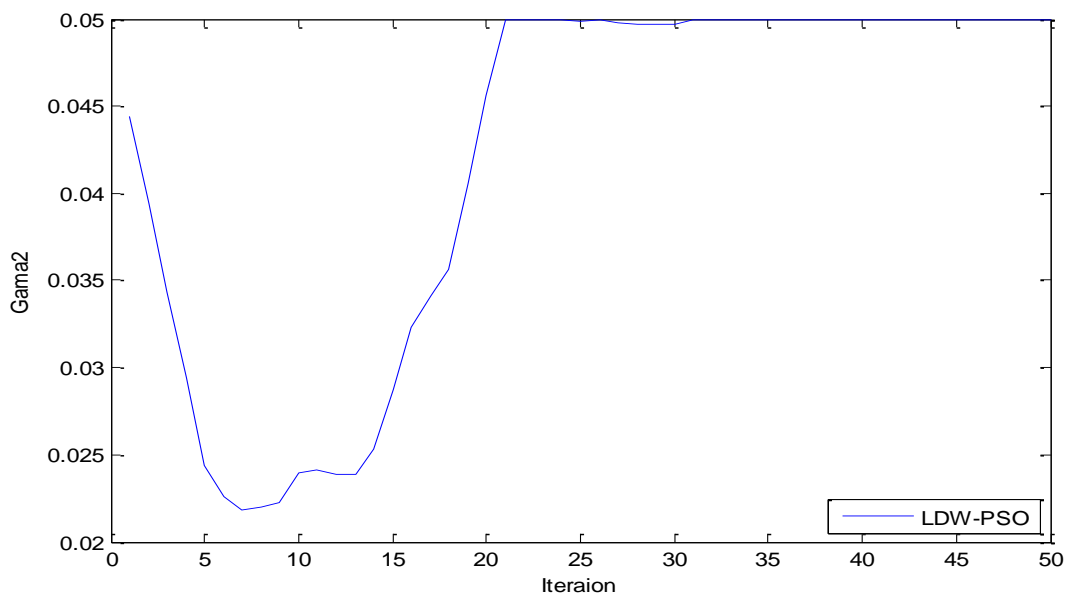
شکل ۲-۶: پارامتر Λ_1 در الگوریتم LDW-PSO

شکل (۳-۶) روند همگرایی مسیر پارامتر Γ_1 را نشان می دهد.



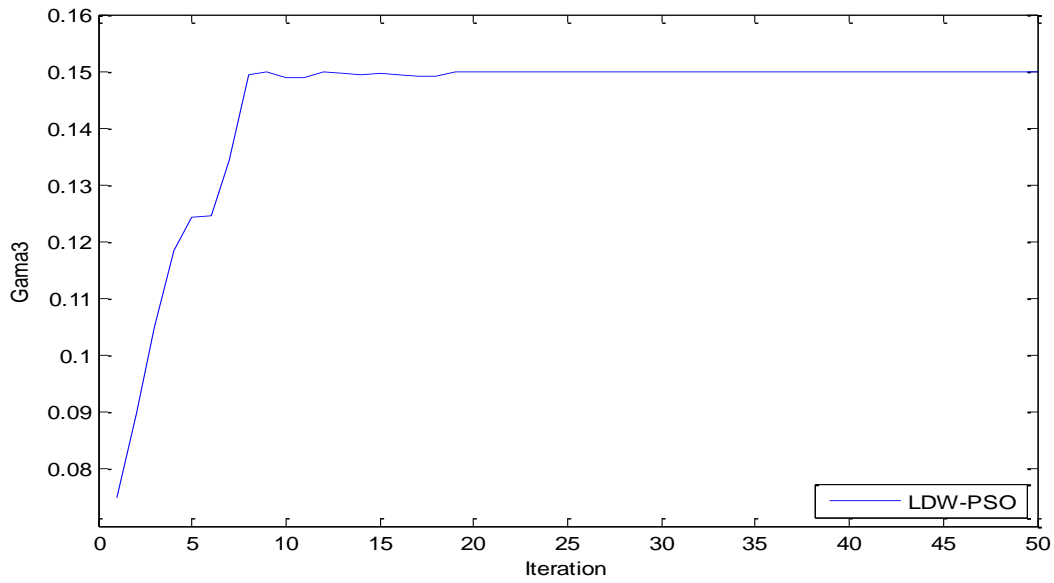
شکل ۳-۶: پارامتر Γ_1 در الگوریتم LDW-PSO

شکل (۴-۶) روند همگرایی مسیر پارامتر Γ_2 را نشان می دهد.



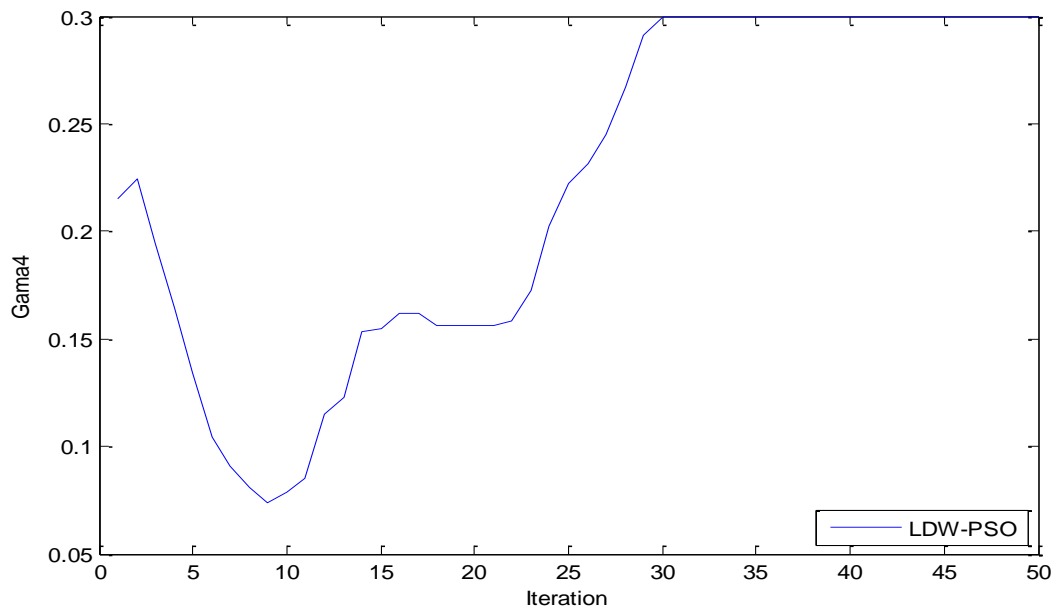
شکل ۴-۶: پارامتر Γ_2 در الگوریتم LDW-PSO

شکل (۵-۶) روند همگرایی مسیر پارامتر Γ_3 را نشان می دهد.



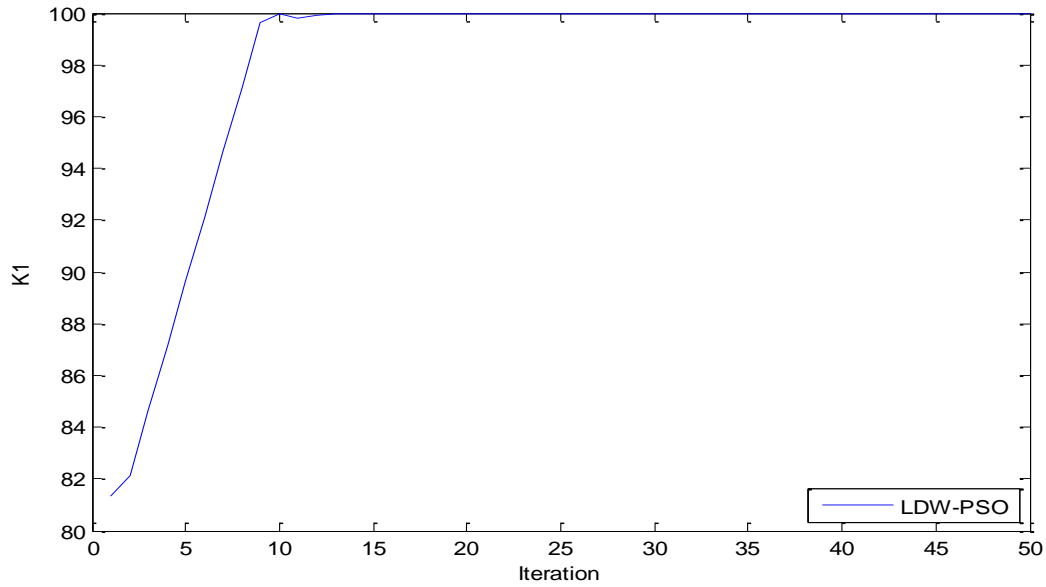
شکل ۵-۶: پارامتر Γ_3 در الگوریتم LDW-PSO

شکل (۶-۶) روند همگرایی مسیر پارامتر Γ_4 را نشان می دهد.



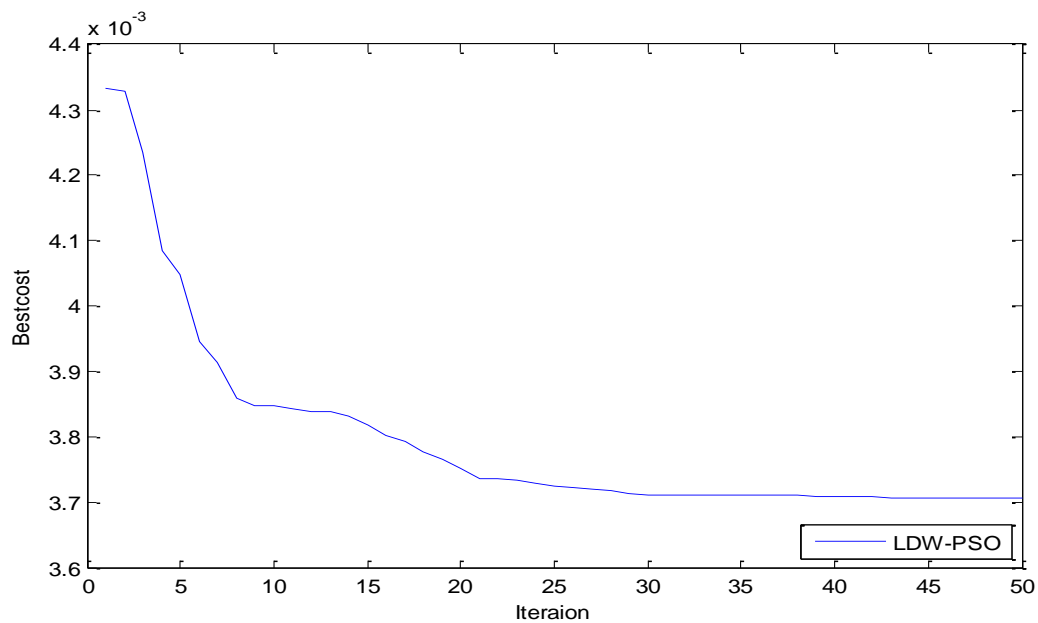
شکل ۶-۶: پارامتر Γ_4 در الگوریتم LDW-PSO

شکل (۶-۷) روند همگرایی مسیر پارامتر K_1 را نشان می دهد.



شکل ۶-۷: پارامتر K_1 در الگوریتم LDW-PSO

شکل (۶-۸) روند همگرایی مسیر تابع هزینه را نشان می دهد.



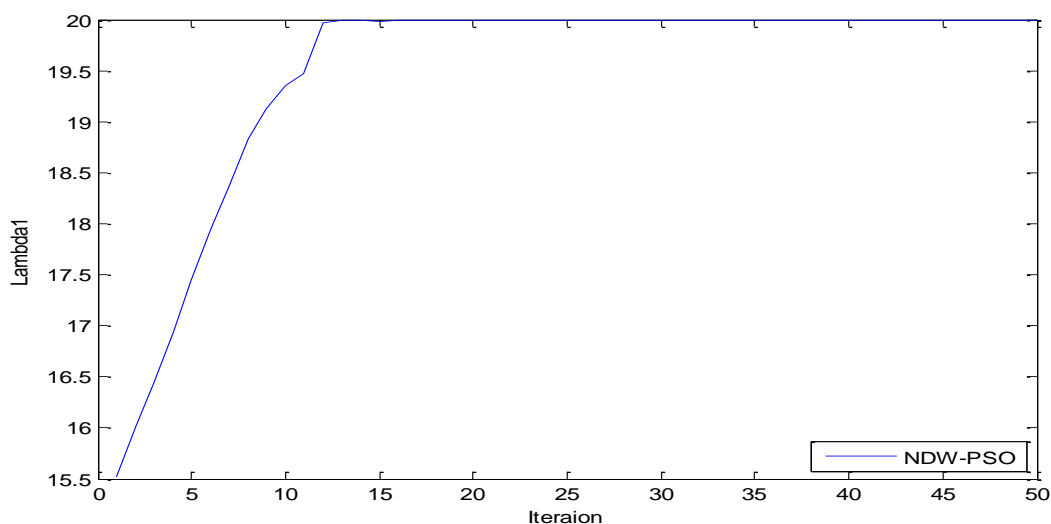
شکل ۶-۸: تابع هزینه در الگوریتم LDW-PSO

۲-۶ شبیه سازی الگوریتم بهینه سازی اجتماع پرندگان با استفاده از ضریب

اینرسی کاهش غیرخطی (NDW-PSO)

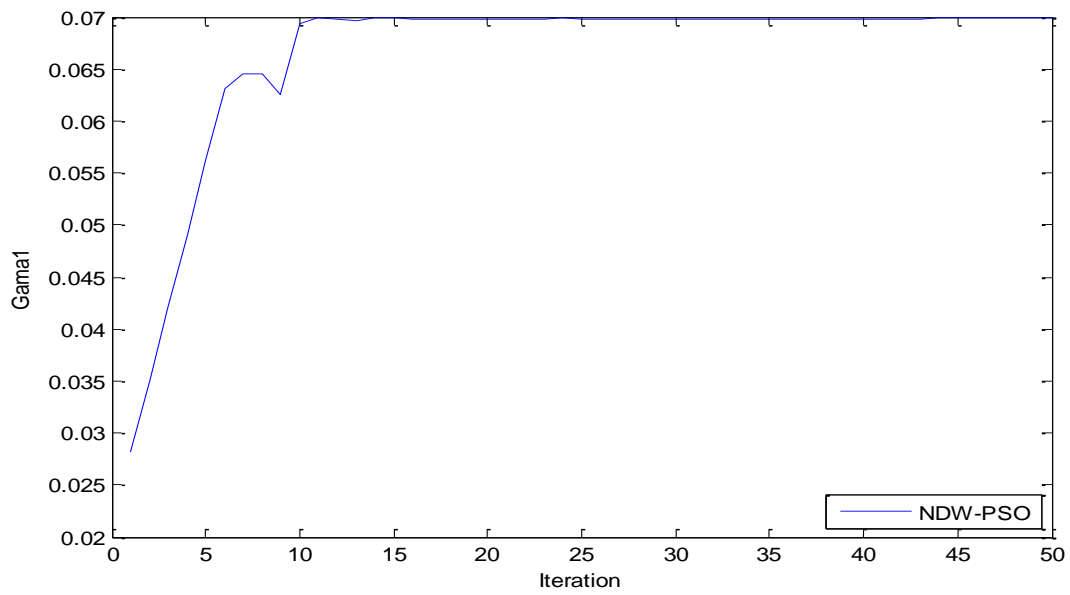
این الگوریتم نیز مانند الگوریتم قبل است با این تفاوت که ضریب اینرسی کاهش غیرخطی است و برای بهبود عملکرد الگوریتم استاندارد پیشنهاد شده است. در این الگوریتم ضریب اینرسی W به صورت غیرخطی کاهش می یابد که از معادله (۵-۲۷) برای ضریب اینرسی استفاده می شود. این الگوریتم برای بهینه سازی پارامترهای کنترل کننده و مینیمم کردن خطای ردگیری بکار برده شده است. در این الگوریتم برای بهبود عملکرد مسئله بازوی ماهر ربات، ضریب اینرسی ماکزیمم $W_{max} = 0.9$ و ضریب اینرسی مینیمم $W_{min} = 0.4$ ، ضریب یادگیری $C_1=C_2 = 2$ ، $n = 1.2$ و تعداد ذرات 10 و تعداد تکرار 50 و هر الگوریتم 25 بار اجرا به کار گرفته شده است.

شکل (۶-۹) روند همگرایی مسیر پارامتر Λ_1 را نشان می دهد.



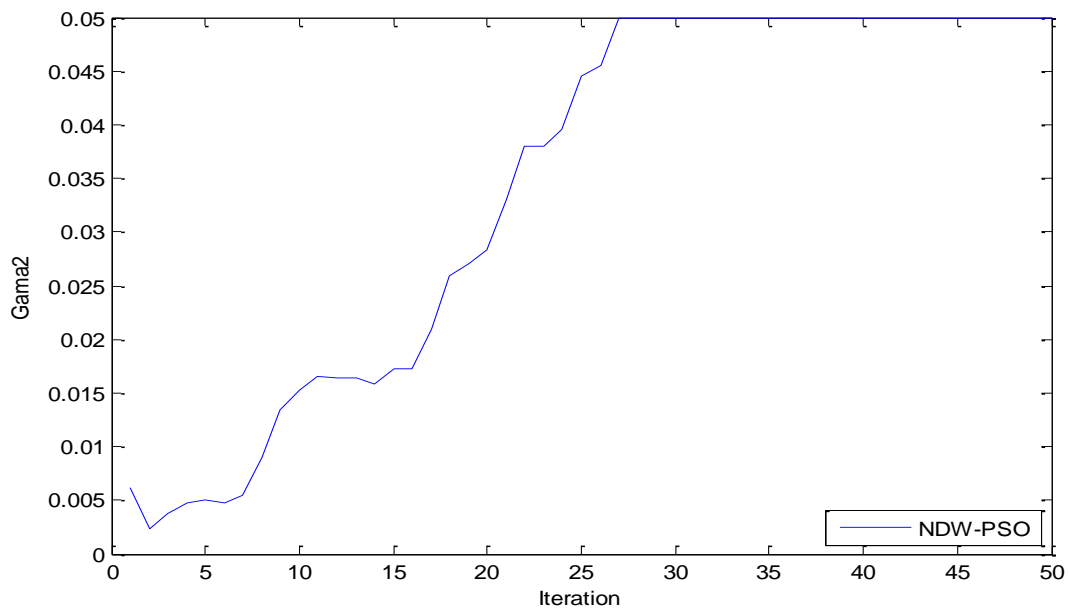
شکل ۶-۹: پارامتر Λ_1 در الگوریتم NDW-PSO

شکل (۶-۱۰) روند همگرایی مسیر پارامتر Γ_1 را نشان می دهد.



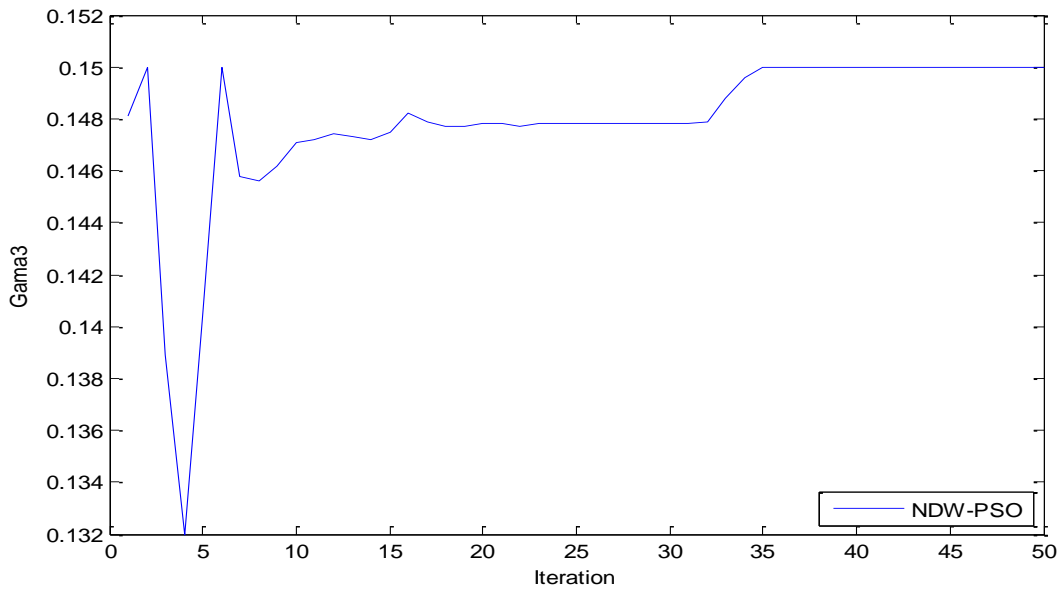
شکل ۶-۱۰: پارامتر Γ_1 در الگوریتم NDW-PSO

شکل (۶-۱۱) روند همگرایی مسیر پارامتر Γ_2 را نشان می دهد.



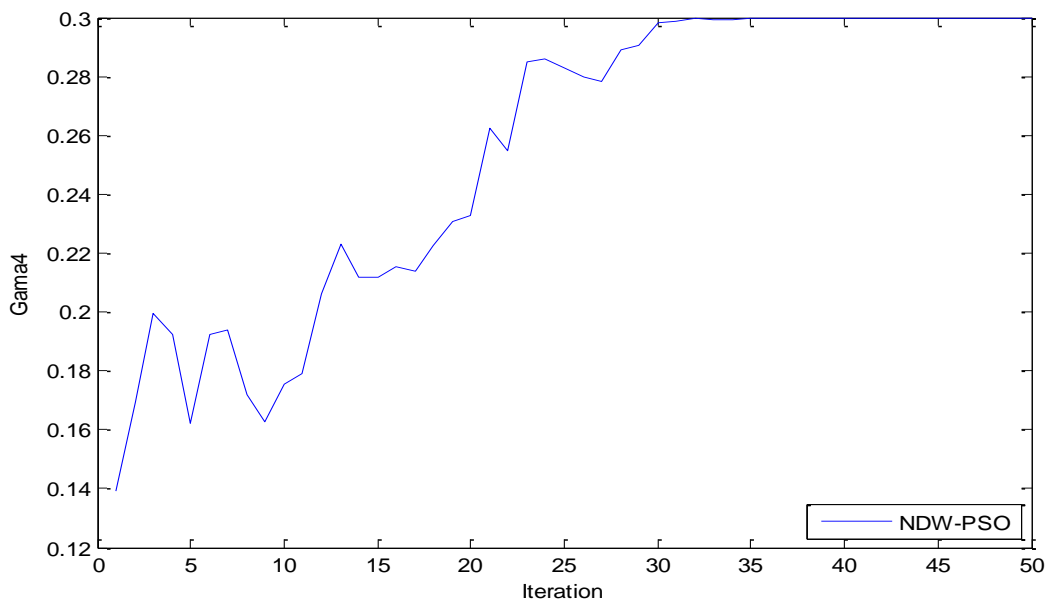
شکل ۶-۱۱: پارامتر Γ_2 در الگوریتم NDW-PSO

شکل (۶-۱۲) روند همگرایی مسیر پارامتر Γ_3 را نشان می دهد.



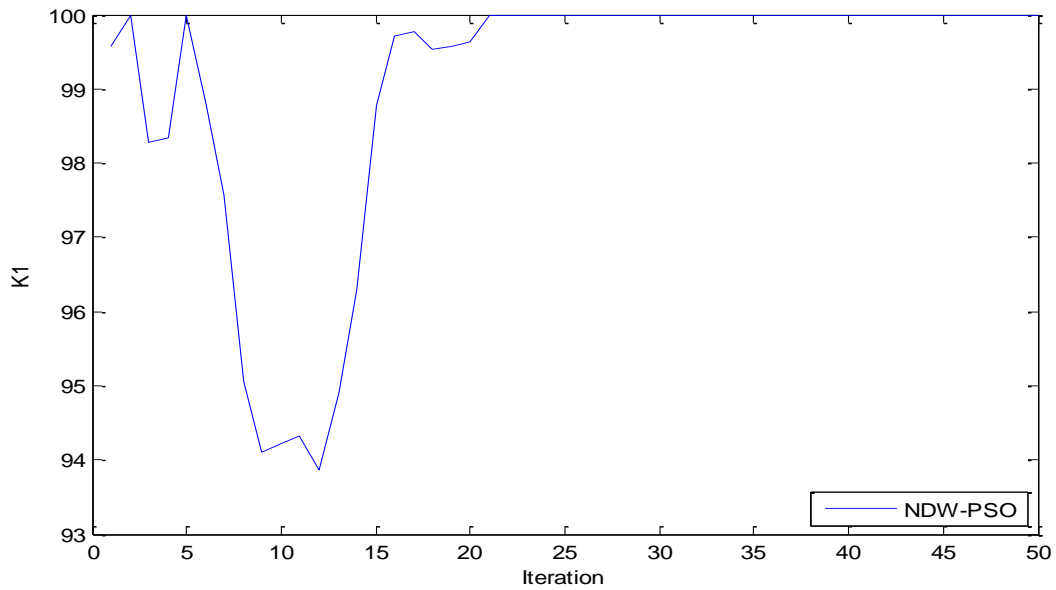
شکل ۶-۱۲: پارامتر Γ_3 در الگوریتم NDW-PSO

شکل (۶-۱۳) روند همگرایی مسیر پارامتر Γ_4 را نشان می دهد.



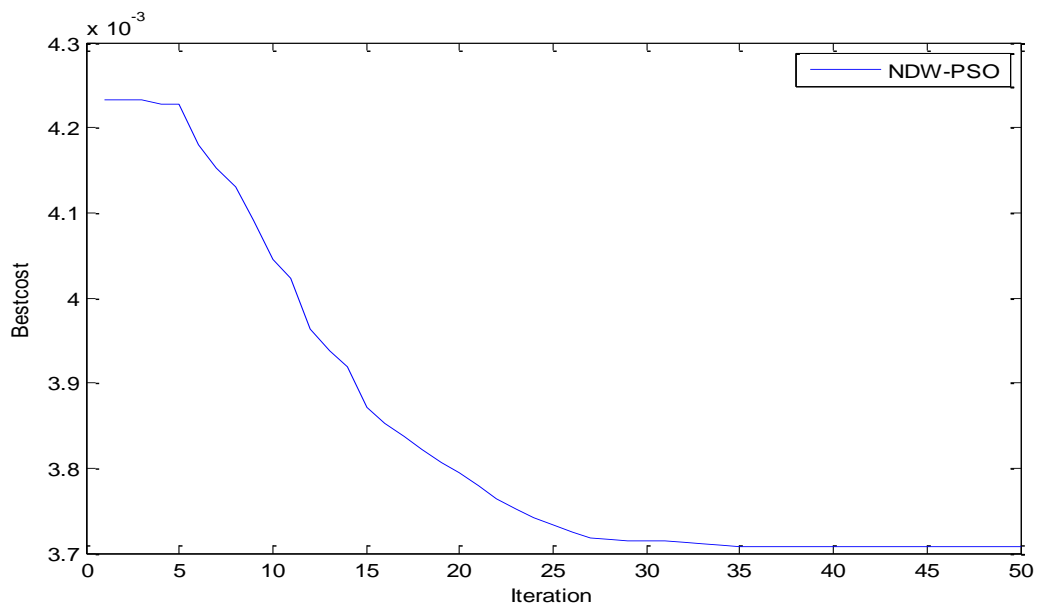
شکل ۶-۱۳: پارامتر Γ_4 در الگوریتم NDW-PSO

شکل (۶-۱۴) روند همگرایی مسیر پارامتر K_1 را نشان می دهد.



شکل ۶-۱۴: پارامتر K_1 در الگوریتم NDW-PSO

شکل (۶-۱۵) روند همگرایی مسیر تابع هزینه را نشان می دهد.



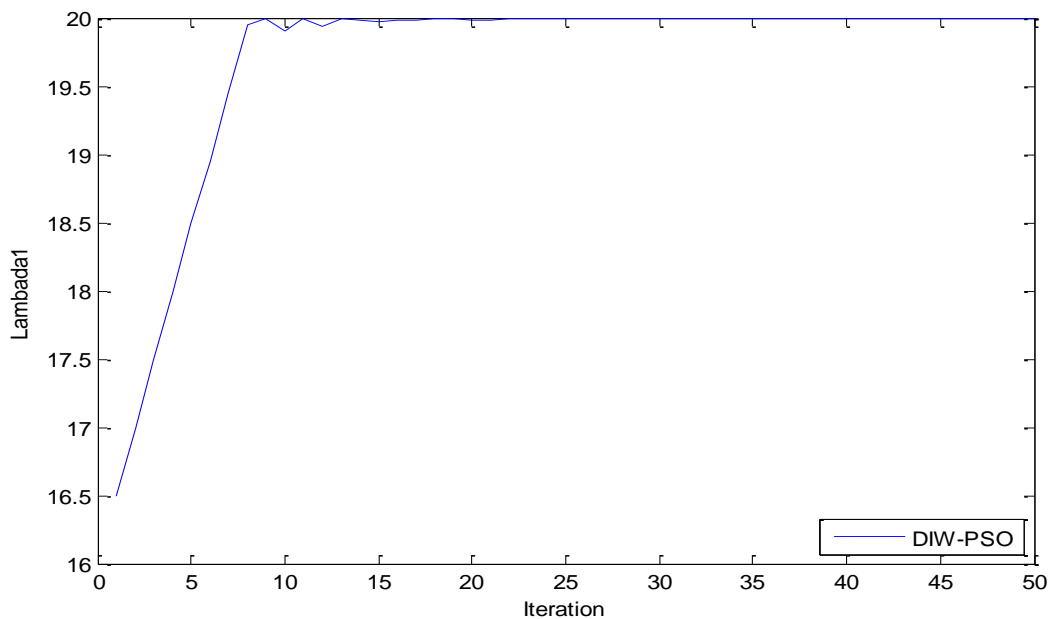
شکل ۶-۱۵: تابع هزینه در الگوریتم NDW-PSO

۳-۶ شبیه سازی الگوریتم بهینه سازی پرندگان با ضریب اینرسی

دینامیکی (DIW-PSO)

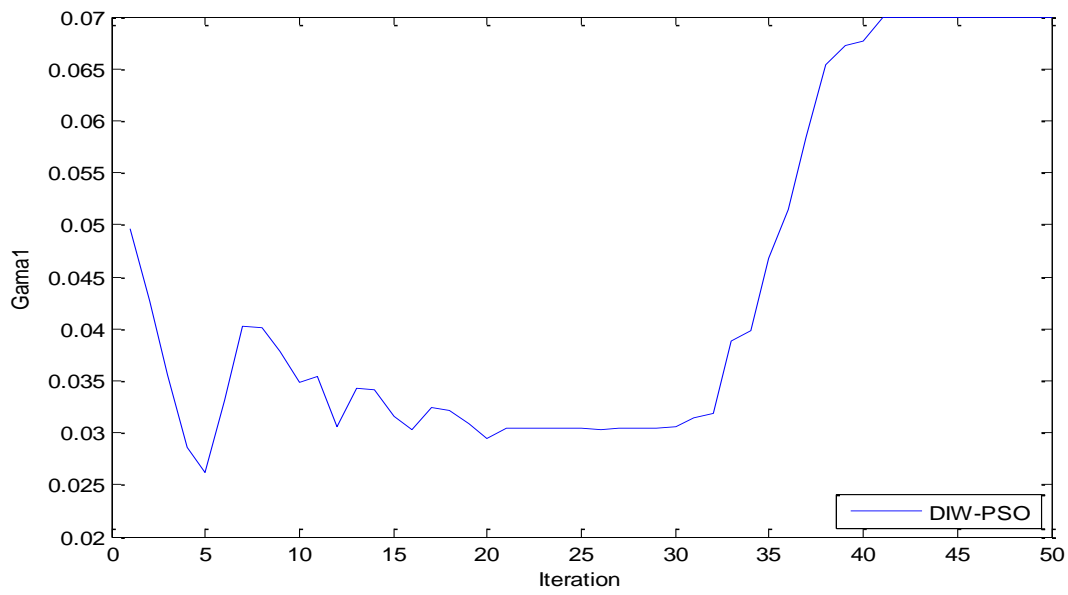
این الگوریتم نیز برای بهبود عملکرد الگوریتم پرندگان استاندارد پیشنهاد شده است. در این الگوریتم ضریب اینرسی W به صورت دینامیکی در معادله (۵-۳۰) تعریف شده است. در این الگوریتم $W = 0.9$ و $u = 1.0002$ می باشد.

شکل (۶-۱۶) روند همگرایی مسیر پارامتر Λ_1 را نشان می دهد.



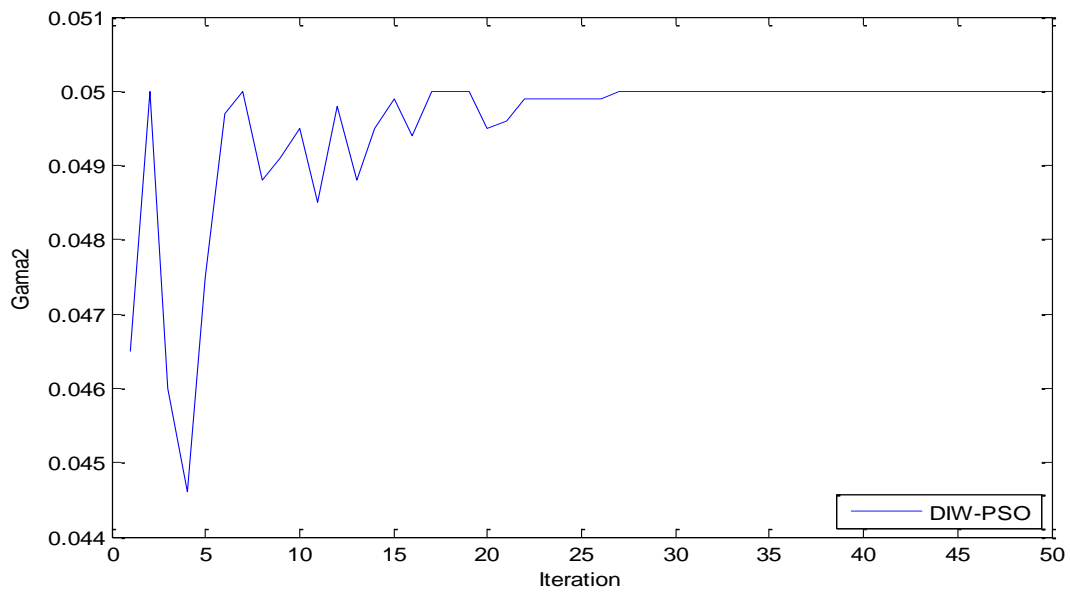
شکل ۶-۱۶: پارامتر Λ_1 در الگوریتم DIW-PSO

شکل (۶-۱۷) روند همگرایی مسیر پارامتر Γ_1 را نشان می دهد.



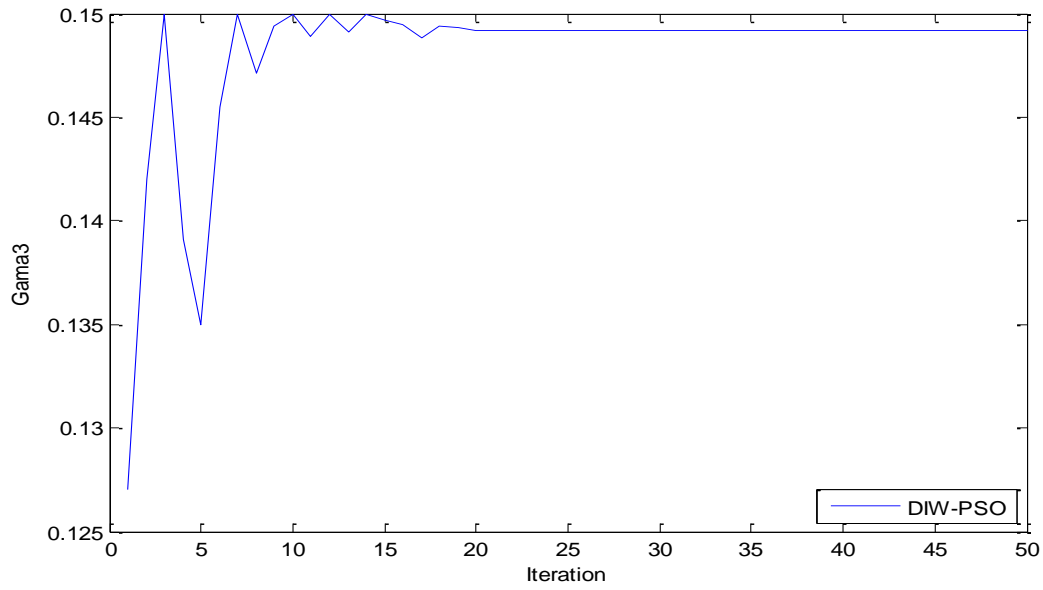
شکل ۶-۱۷: پارامتر Γ_1 در الگوریتم DIW-PSO

شکل (۶-۱۸) روند همگرایی مسیر پارامتر Γ_2 را نشان می دهد.



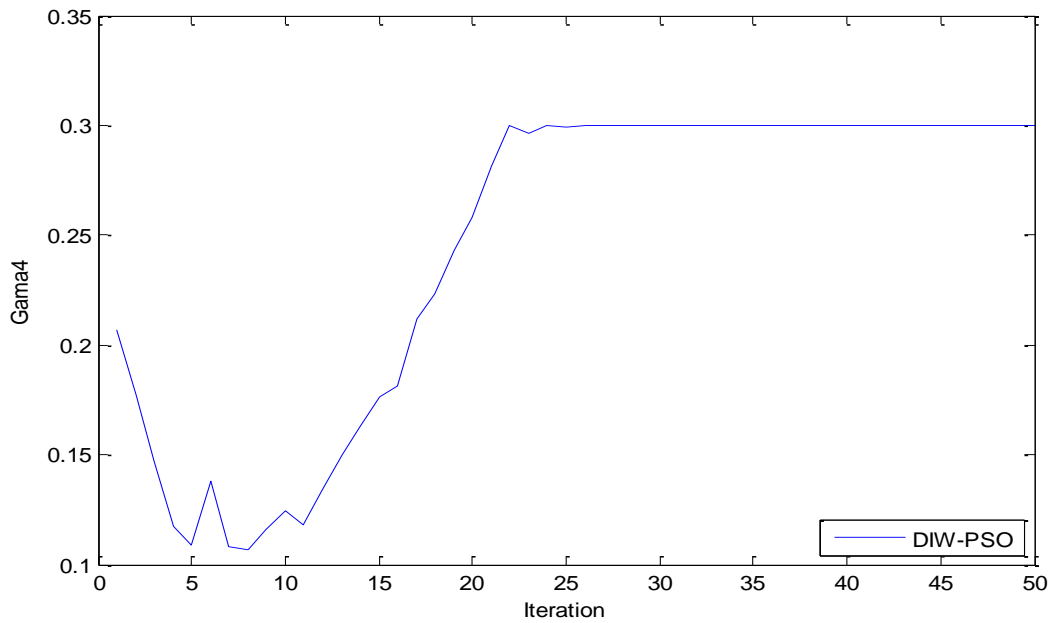
شکل ۶-۱۸: پارامتر Γ_2 در الگوریتم DIW-PSO

شکل (۶-۱۹) روند همگرایی مسیر پارامتر Γ_3 را نشان می دهد.



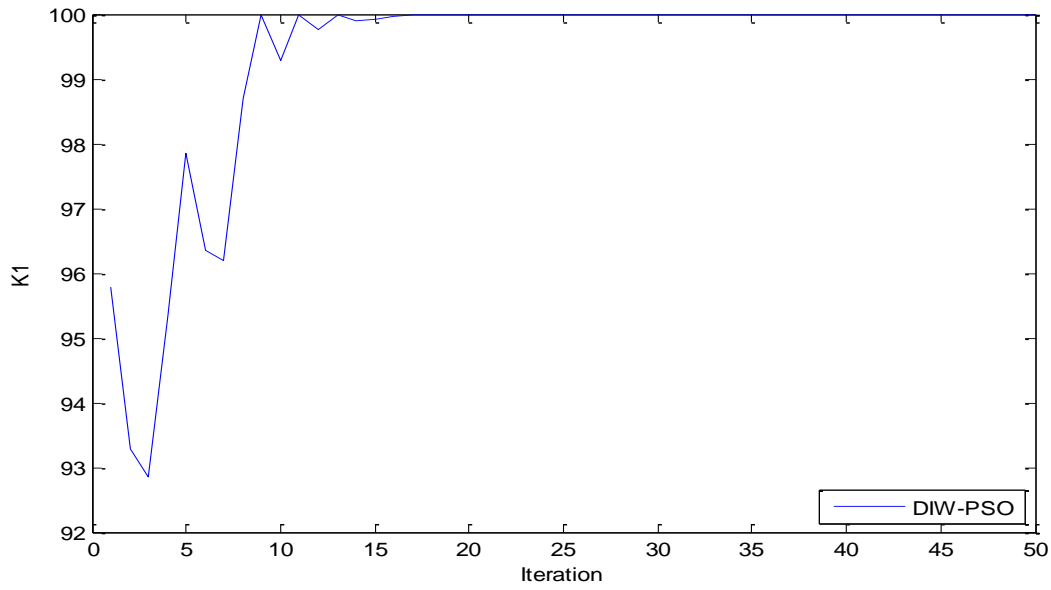
شکل ۶-۱۹: پارامتر Γ_3 در الگوریتم DIW-PSO

شکل (۶-۲۰) روند همگرایی مسیر پارامتر Γ_4 را نشان می دهد.



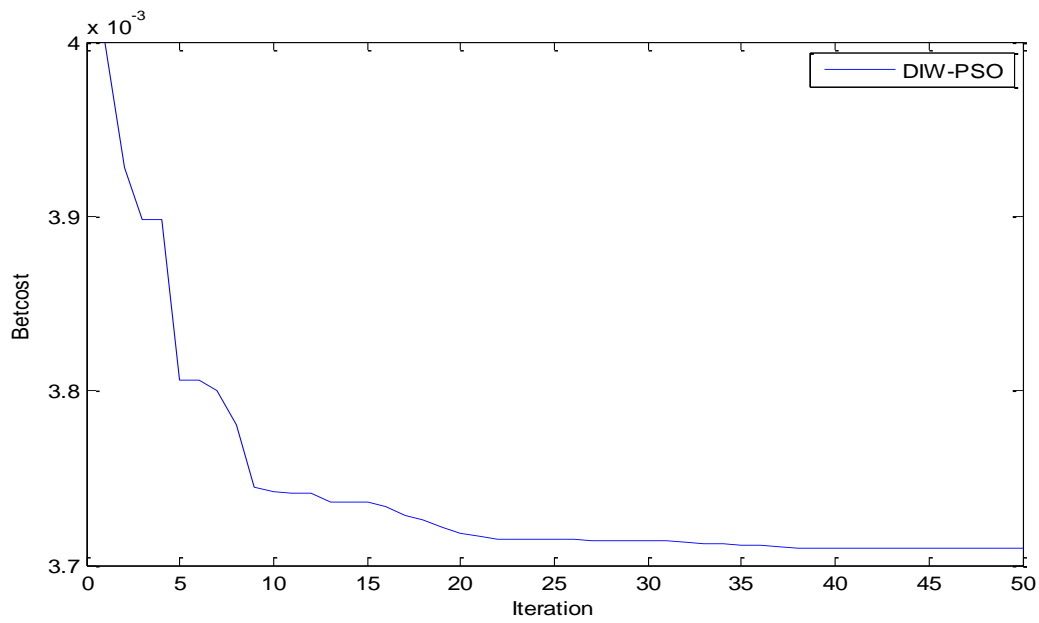
شکل ۶-۲۰: پارامتر Γ_4 در الگوریتم DIW-PSO

شکل (۶-۲۱) روند همگرایی مسیر پارامتر K_1 را نشان می دهد.



شکل ۶-۲۱: پارامتر K_1 در الگوریتم DIW-PSO

شکل (۶-۲۲) روند همگرایی مسیر تابع هزینه را نشان می دهد.



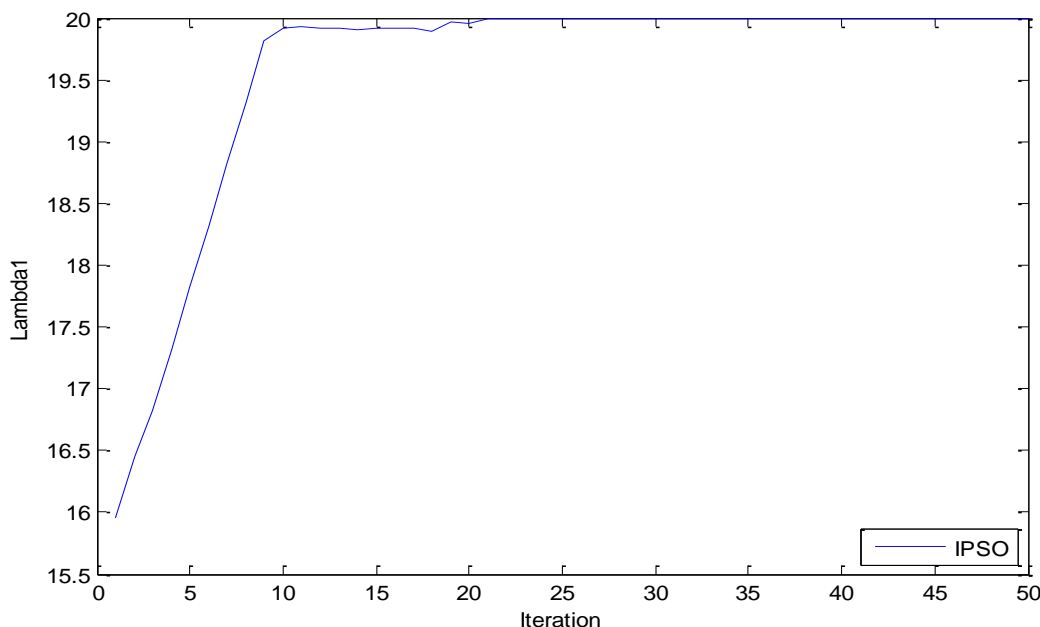
شکل ۶-۲۲: تابع هزینه در الگوریتم DIW-PSO

۴-۶ شبیه سازی الگوریتم بهینه سازی اجتماع پرندگان بهبود

یافته (IPSO)

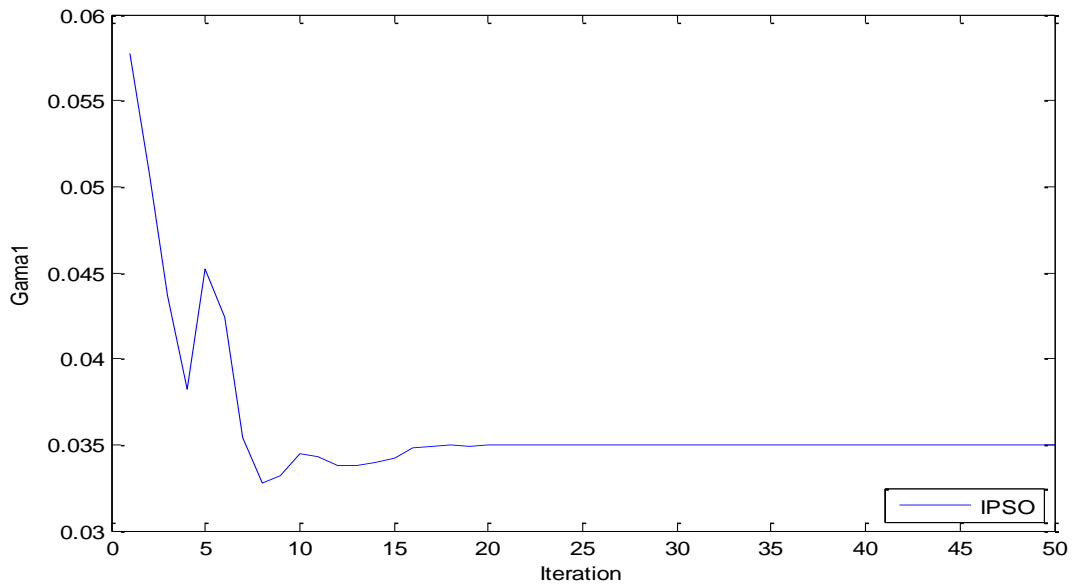
این الگوریتم برای بهبود عملکرد الگوریتم پرندگان استاندارد پیشنهاد شده است. در این الگوریتم ضریب اینرسی W و ضرایب یادگیری C_1 و C_2 طبق معادلات (۳۱-۵) و (۳۲-۵) و (۳۳-۵) استفاده می شود. در این الگوریتم هم چنین $\alpha = rand(0,1)$ و $n = 1.5$ در نظر گرفته می شود. این الگوریتم برای بهینه سازی پارامترهای کنترل کننده و مینیمم کردن خطای ردگیری بکار برده شده است. در این الگوریتم برای بهبود عملکرد مسئله بازوی ماهر ربات، تعداد ذرات 10 و تعداد تکرار 50 و هر الگوریتم 25 بار اجرا به کار گرفته شده است.

شکل (۲۳-۶) روند همگرایی مسیر پارامتر Λ_1 را نشان می دهد.



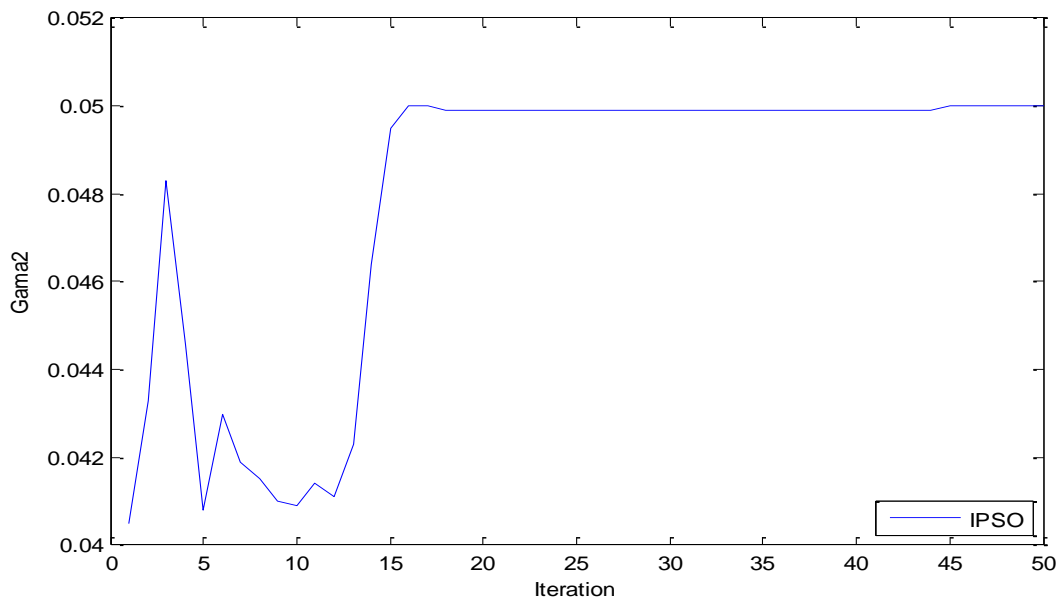
شکل ۲۳-۶: پارامتر Λ_1 در الگوریتم IPSO

شکل (۶-۲۴) روند همگرایی مسیر پارامتر Γ_1 را نشان می دهد.



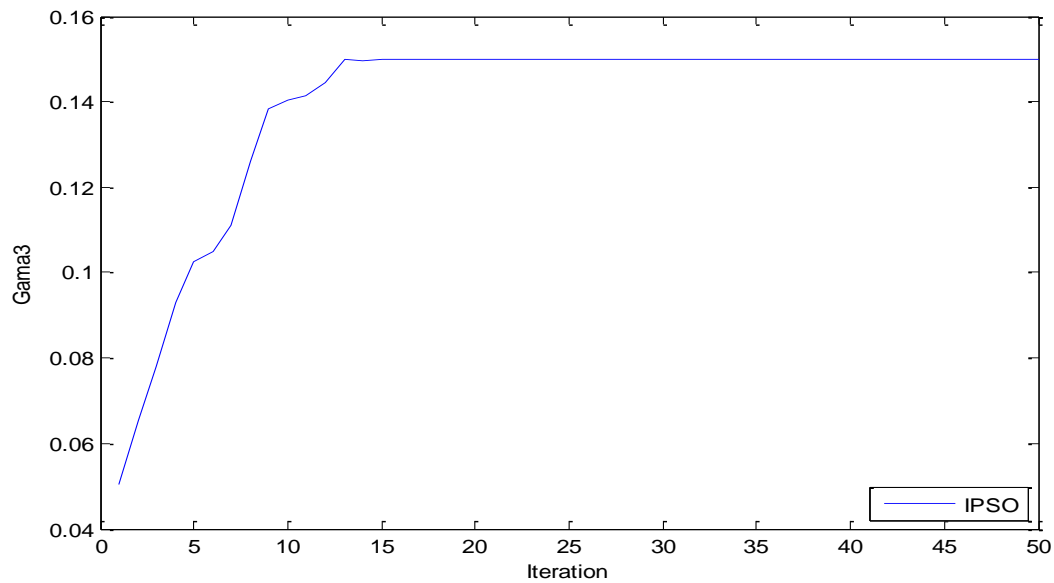
شکل ۶-۲۴: پارامتر Γ_1 در الگوریتم IPSO

شکل (۶-۲۵) روند همگرایی مسیر پارامتر Γ_2 را نشان می دهد.



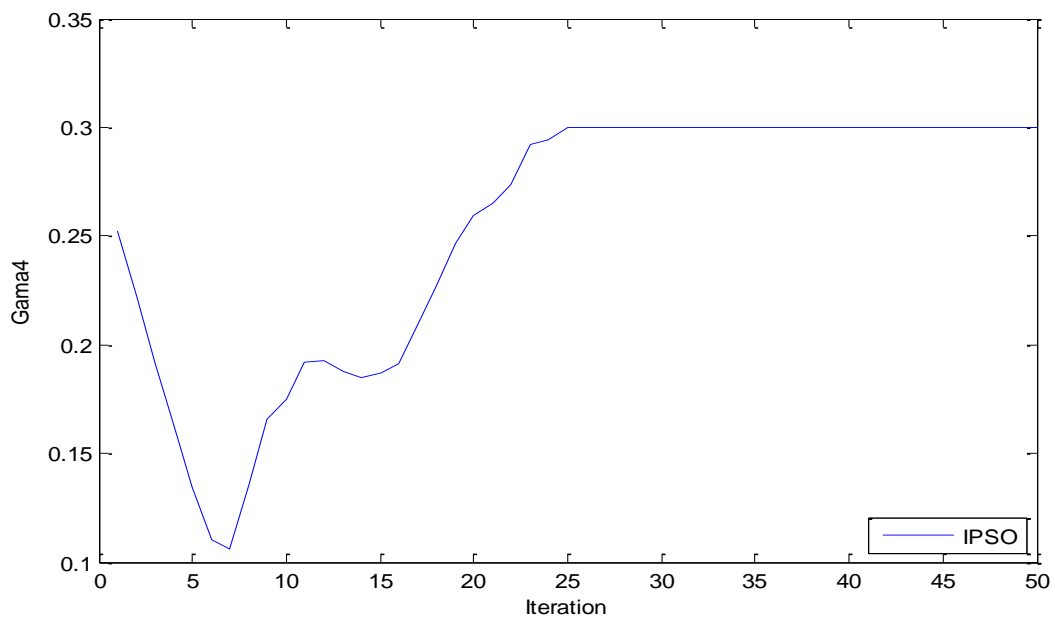
شکل ۶-۲۵: پارامتر Γ_2 در الگوریتم IPSO

شکل (۶-۲۶) روند همگرایی پارامتر Γ_3 را نشان می دهد.



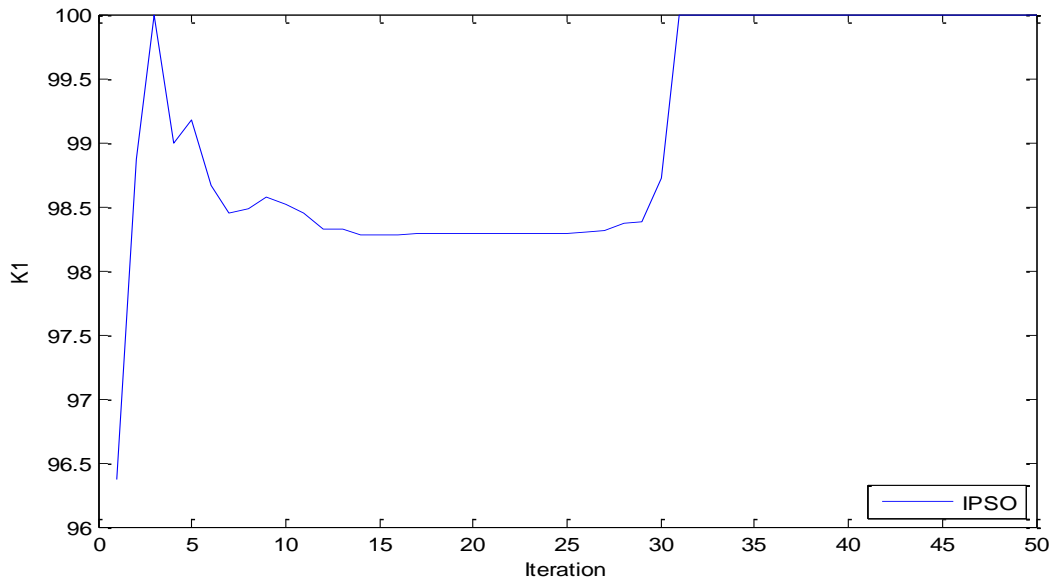
شکل ۶-۲۶: پارامتر Γ_3 در الگوریتم IPSO

شکل (۶-۲۷) روند همگرایی پارامتر Γ_4 را نشان می دهد.



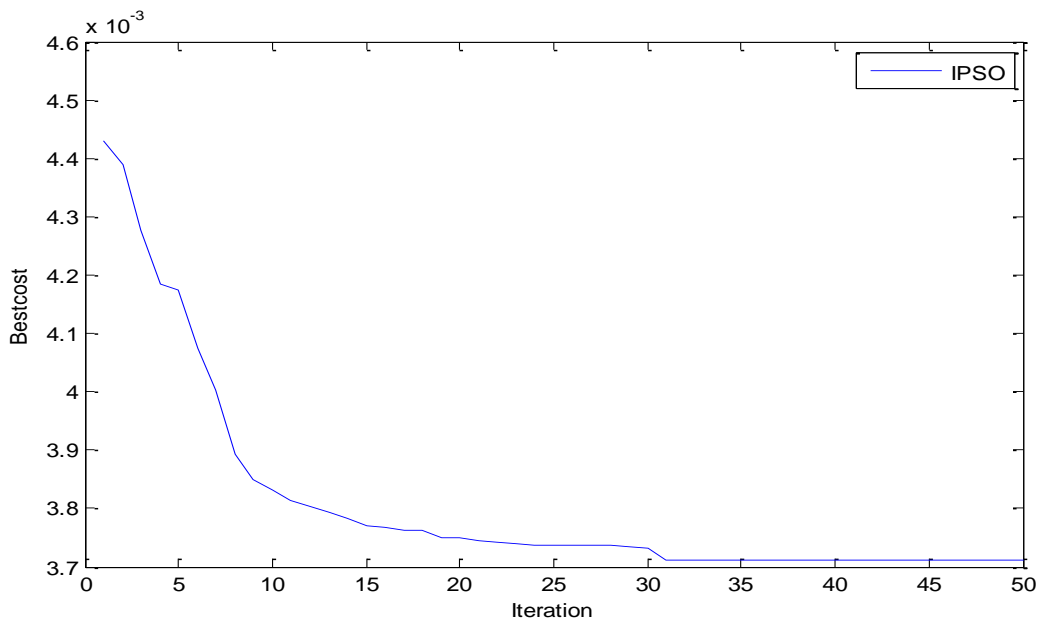
شکل ۶-۲۷: پارامتر Γ_4 در الگوریتم IPSO

شکل (۶-۲۸) روند همگرایی پارامتر K_1 را نشان می دهد.



شکل ۶-۲۸: پارامتر K_1 در الگوریتم IPSO

شکل (۶-۲۹) روند همگرایی مسیر تابع هزینه را نشان می دهد.



شکل ۶-۲۹: تابع هزینه در الگوریتم IPSO

۵-۶ شبیه سازی الگوریتم بهینه سازی اجتماع پرندگان با استفاده از منطق

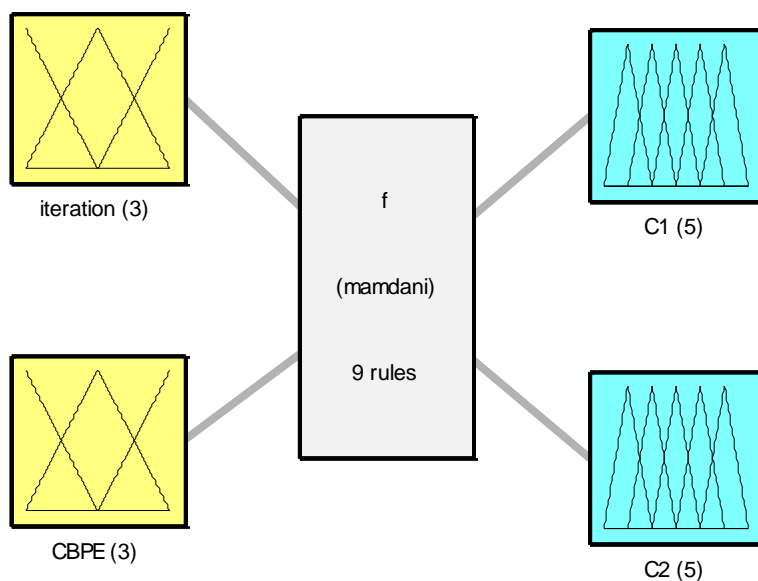
فازی (Fuzzy-PSO)

این روش کنترلی پیشنهادی، ترکیبی از سیستم فازی و الگوریتم بهینه سازی پرندگان

است که ضریب اینرسی براساس معادله (۵-۳۶) تغییر داده می شود و پارامترهای C_1 و C_2

براساس منطق فازی ارائه می شود. سیستم فازی شامل مولفه های اصلی فازی ساز، قوانین فازی،

استنتاج فازی، غیرفازی ساز است که به صورت زیر توصیف می شود:



System f: 2 inputs, 2 outputs, 9 rules

شکل ۶-۳۰: سیستم فازی

۱-۵-۶ فازی ساز

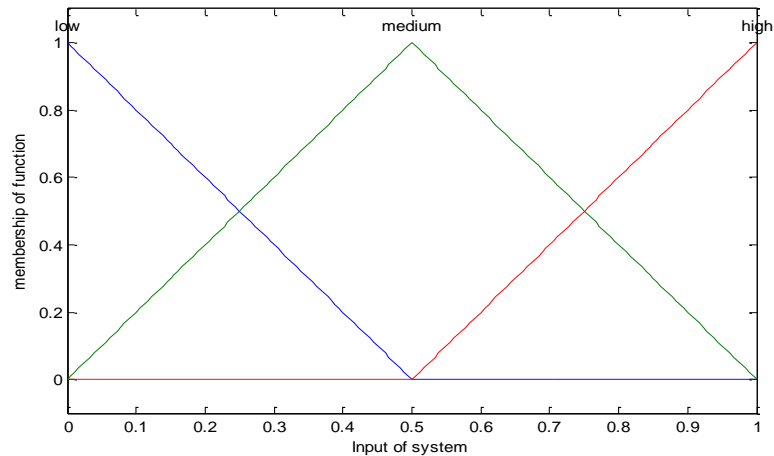
در میان مجموعه ای از توابع عضویت، توابع عضویت مثلث برای هر ورودی و هر خروجی

استفاده

می شود که در شکل های (۳۱-۶) و (۳۲-۶) دیده می شود.

شکل (۳۱-۶) توابع عضویت مثلثی برای ورودی های $CBPE$ و $Iteration$ به سیستم

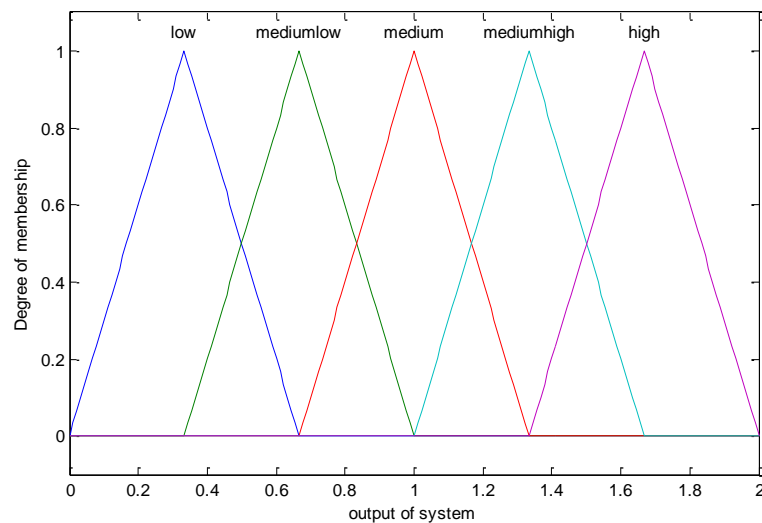
فازی می باشد و مقادیر محدوده برای ورودی، روی $[0,1]$ تعیین می شود.



شکل ۳۱-۶: توابع عضویت مثلثی ورودی

شکل (۳۲-۶) توابع عضویت مثلثی برای خروجی G_1 به سیستم فازی می باشد و مقادیر

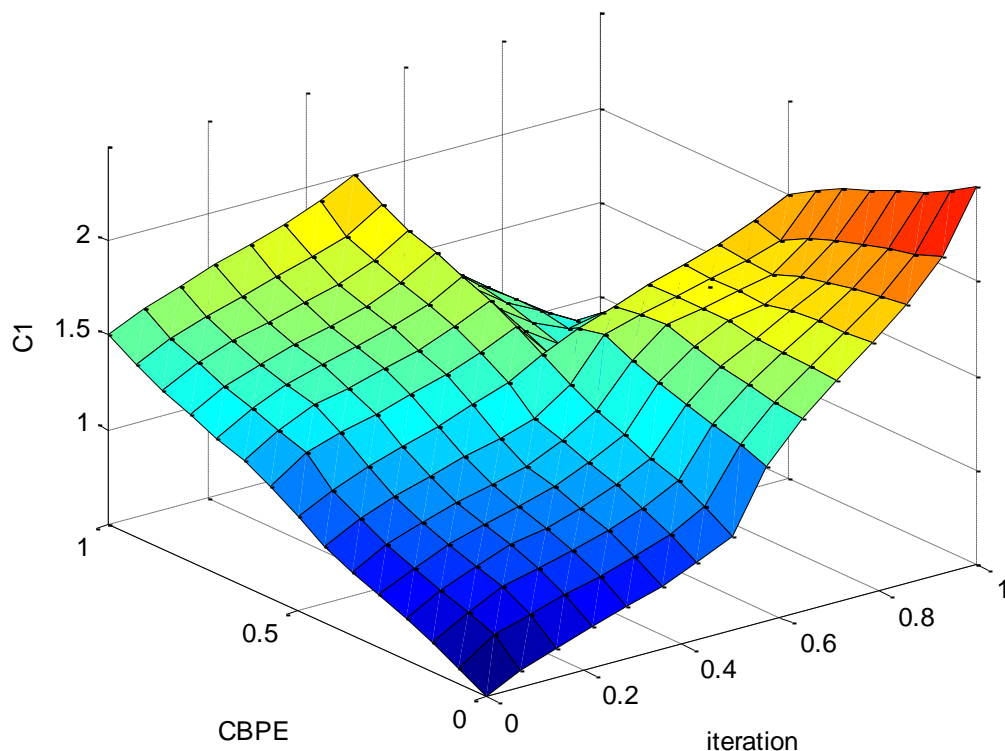
محدوده برای خروجی، روی $[0,2]$ تعیین می شود.



شکل ۳۲-۶: توابع عضویت مثلثی خروجی

شکل (۳۳-۶) یک سه سطح سه بعدی است که دو ورودی را همراه با یکی از خروجی های

سیستم رسم شده است.



شکل ۳۳-۶: سطح سه بعدی دو ورودی همراه یکی از خروجی ها

۲-۵-۶ قوانین فازی

قوانین فازی ممدانی برای فرموله کردن عباراتی که شرایط در برگیرنده منطق فازی را شامل می شوند، به کار می روند. قوانین فازی برای تنظیم ضرایب یادگیری C_1 و C_2 به کار می روند. هر قانون با یک نگاشت، فضای ورودی را به فضای خروجی مرتبط می کند. پایگاه قوانین می تواند به صورت زیر بیان شود:

1. if(iteration is low)and(CPBE is low) then (c1 is low)(c2 is high)
2. if(iteration is low)and(CPBE is medium) then (c1 is medium low)(c2 is medium

high)

3. if(iteration is low)and(CPBE is high) then (c1 is medium)(c2 is medium)

4. if(iteration is medium)and(CPBE is low) then (c1 is medium low)(c2 is high)

5. if(iteration is medium)and(CPBE is medium) then (c1 is medium)(c2 is high)

6. if(iteration is medium)and(CPBE is high) then (c1 is medium high)(c2 is low)

7. if(iteration is high)and(CPBE is low) then (c1 is high)(c2 is low)

8. if(iteration is high)and(CPBE is medium) then (c1 is medium high)(c2 is medium low)

9. if(iteration is high)and(CPBE is high) then (c1 is low)(c2 is medium low)

۳-۵-۶ استنتاج فازی

استراتژی کنترل فازی برای نگاشت از ورودی‌ها به خروجی‌ها به کار برده می‌شود. روش

استنتاج فازی ممدانی در این پایان‌نامه استفاده شده است. عملگر "و" ^۱ برای ترکیب مقادیر عضویت استفاده شده است.

۴-۵-۶ غیر فازی ساز

برای به دست آوردن کنترل قطعی، یک غیر فازی ساز لازم می‌شود. غیر فازی ساز مرکز ثقل

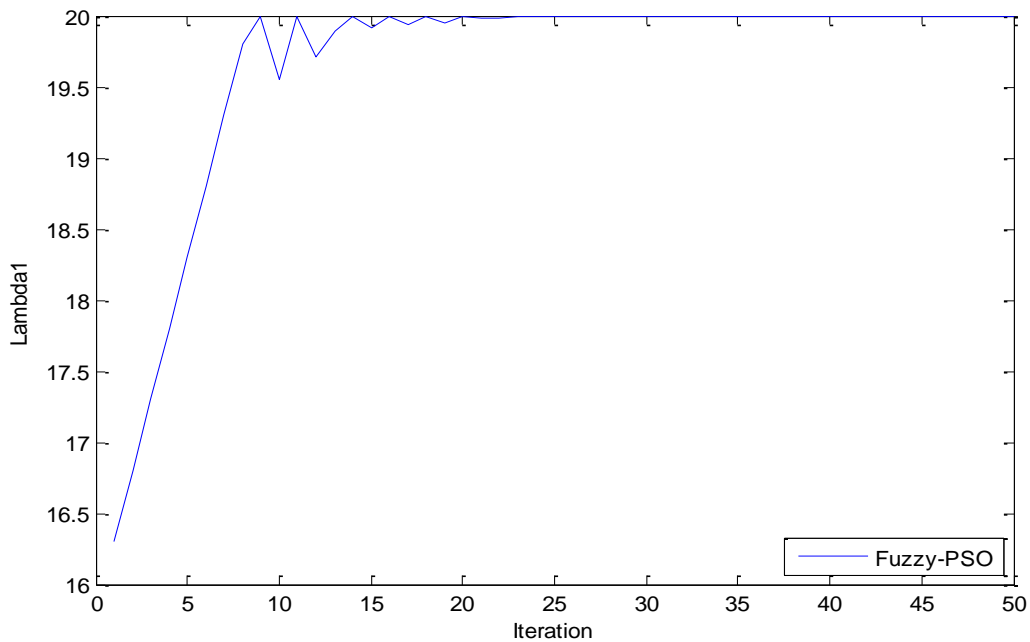
استفاده می‌شود که به صورت زیر نشان داده می‌شود:

^۱AND

$$y = \frac{\int_y \sum_{i=1}^n y \mu_{Bi}(y) dy}{\int_y \sum_{i=1}^n \mu_{Bi}(y) dy} \quad (1-6)$$

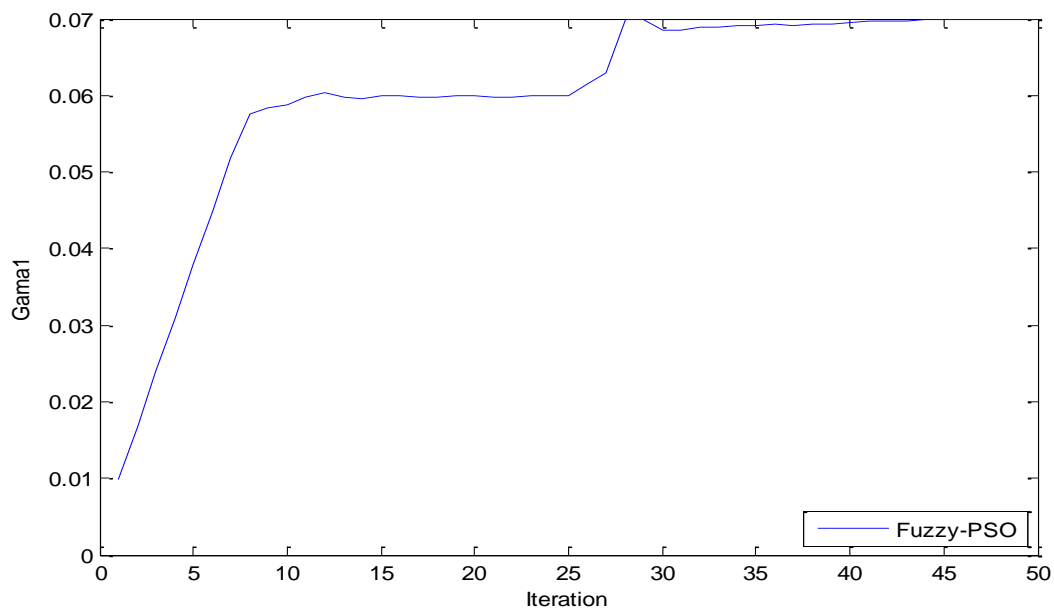
که $\mu_{Bi}(y)$ تابع تعلق است. ورودی‌های غیرفازی‌ساز یک مجموعه فازی از $\mu_{Bi}(y)$ می‌باشد و خروجی آن عدد y می‌باشد.

شکل (۳۴-۶) روند همگرایی مسیر پارامتر Λ_1 را نشان می‌دهد.



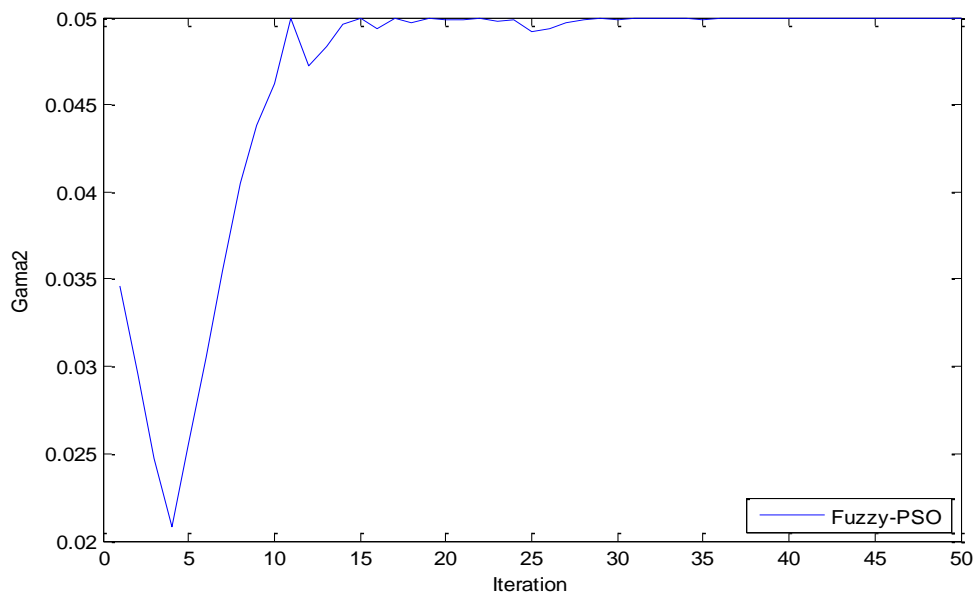
شکل ۳۴-۶: پارامتر Λ_1 در الگوریتم Fuzzy-PSO

شکل (۳۵-۶) روند همگرایی مسیر پارامتر Γ_1 را نشان می دهد.



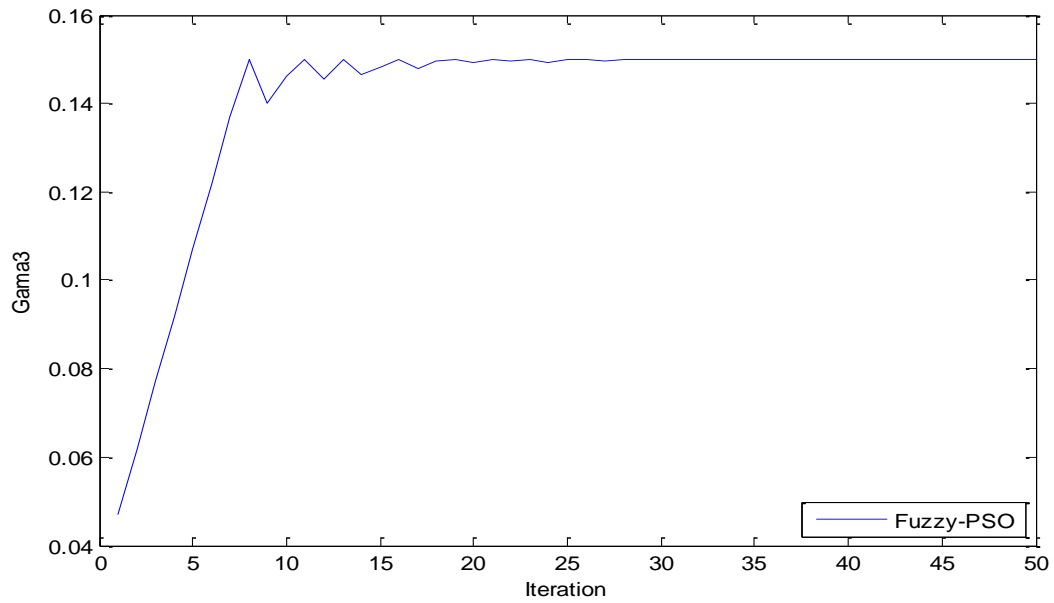
شکل ۳۵-۶: پارامتر Γ_1 در الگوریتم Fuzzy-PSO

شکل (۳۶-۶) روند همگرایی مسیر پارامتر Γ_2 را نشان می دهد.



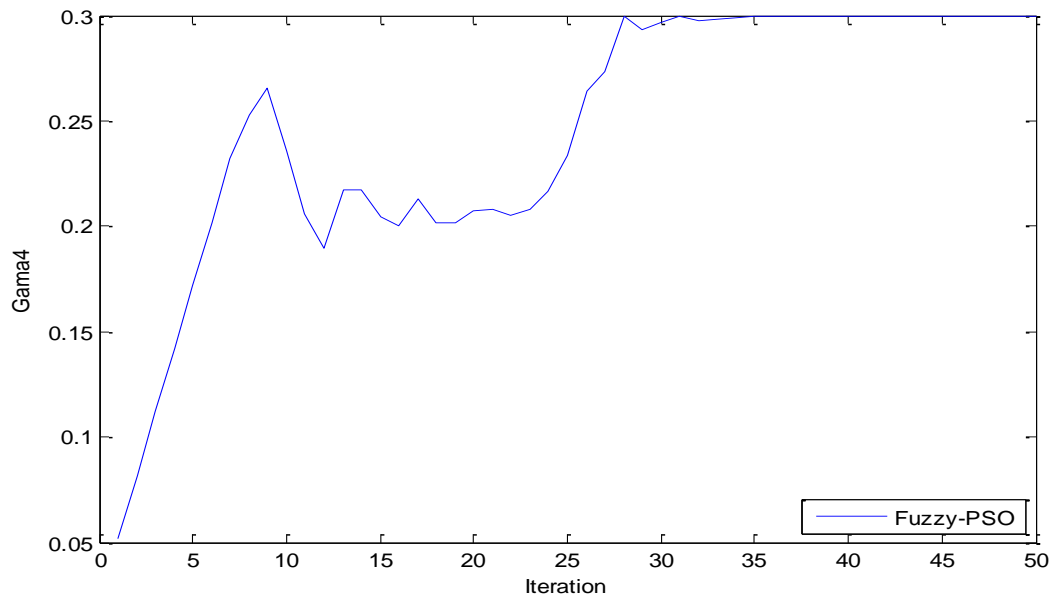
شکل ۳۶-۶: پارامتر Γ_2 در الگوریتم Fuzzy-PSO

شکل (۳۷-۶) روند همگرایی مسیر پارامتر Γ_3 را نشان می دهد.



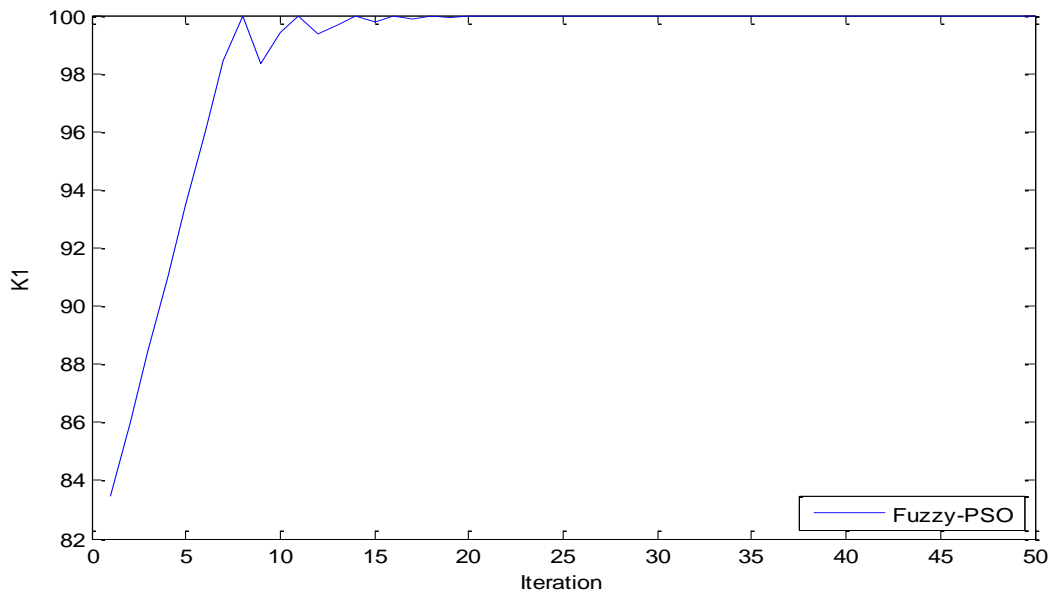
شکل ۳۷-۶: پارامتر Γ_3 در الگوریتم Fuzzy-PSO

شکل (۳۸-۶) روند همگرایی مسیر پارامتر Γ_4 را نشان می دهد.



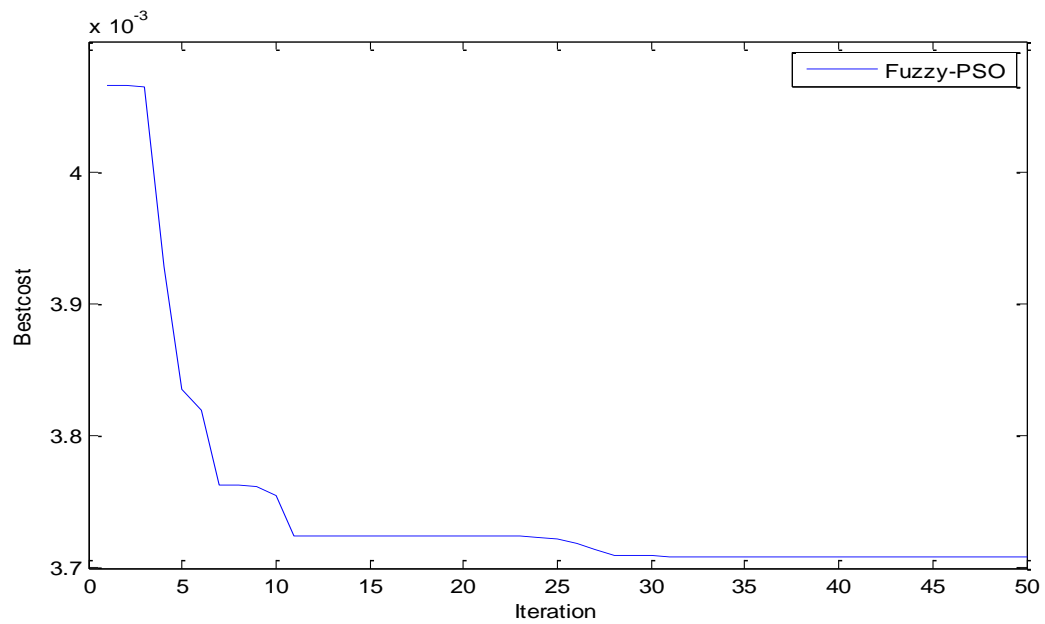
شکل ۳۸-۶: پارامتر Γ_4 در الگوریتم Fuzzy-PSO

شکل (۶-۳۹) روند همگرایی مسیر پارامتر K_1 را نشان می دهد.



شکل ۶-۳۹: پارامتر K_1 در الگوریتم Fuzzy-PSO

شکل (۶-۴۰) روند همگرایی مسیر تابع هزینه را نشان می دهد.



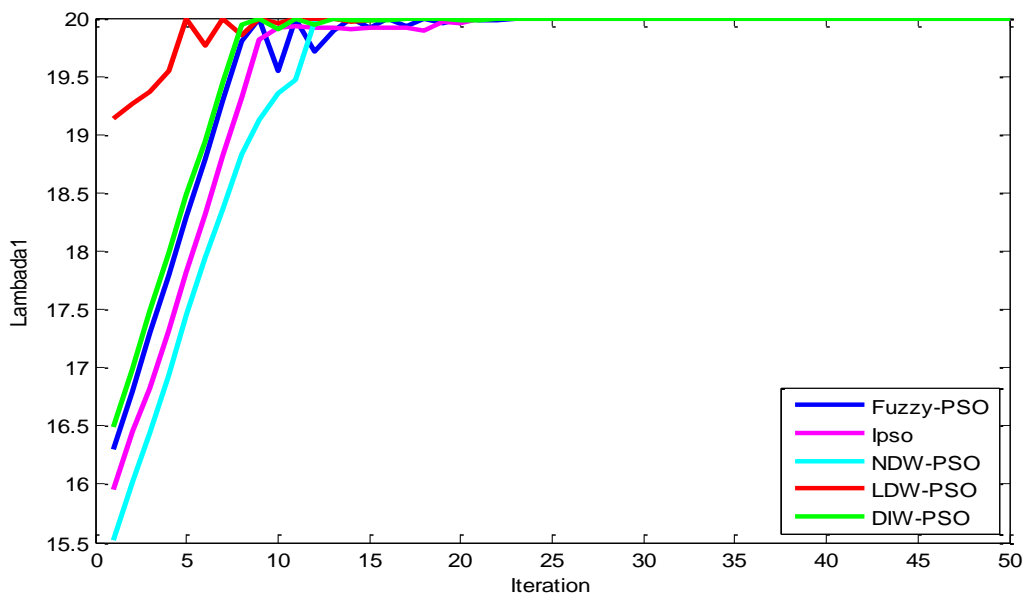
شکل ۶-۴۰: تابع هزینه در الگوریتم Fuzzy-PSO

۶-۶ مقایسه نتایج

۱-۶-۶ مقایسه الگوریتم‌ها

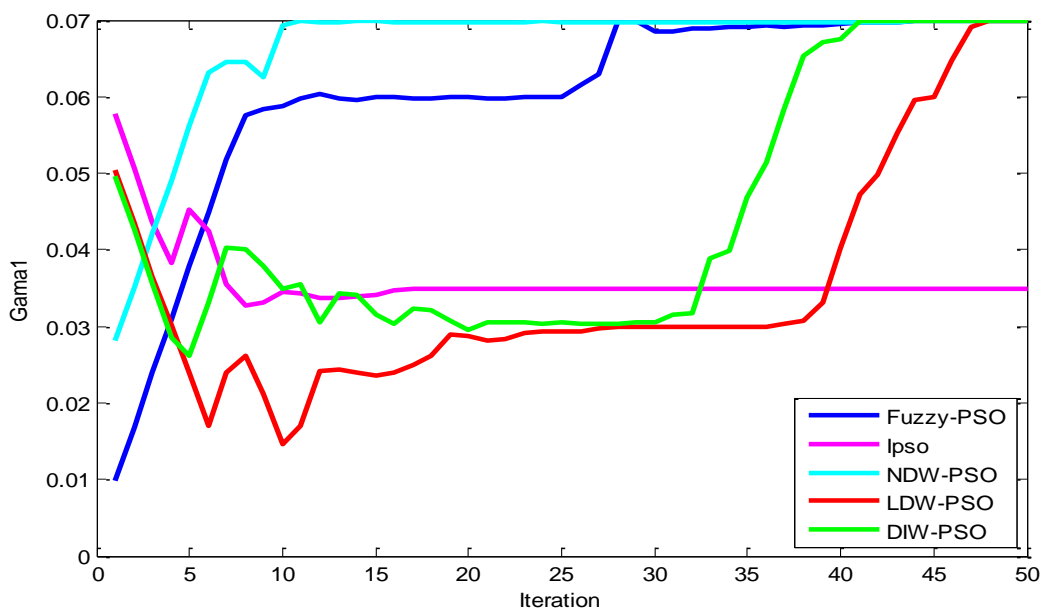
در این قسمت پارامترهای بهینه شده و تابع هزینه بدست آمده توسط الگوریتم‌های مختلف پرندگان با یکدیگر مقایسه می‌شود. هدف از این پیاده‌سازی آشنایی با چگونگی عملکرد الگوریتم‌ها در پارامترهای کنترل کننده می‌باشد.

شکل (۶-۴۱) مقایسه سرعت همگرایی پارامتر Λ_1 را در الگوریتم‌های مختلف نشان می‌دهد.



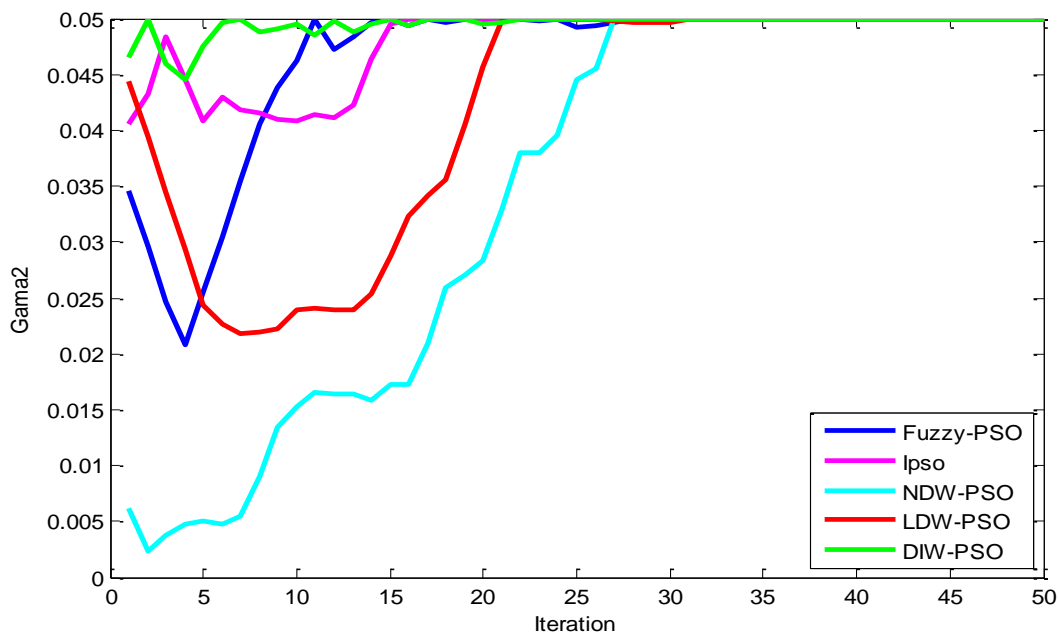
شکل ۶-۴۱: پارامتر Λ_1 در الگوریتم‌های مختلف

شکل (۴۲-۶) مقایسه سرعت همگرایی پارامتر Γ_1 را در الگوریتم‌های مختلف نشان می‌دهد.



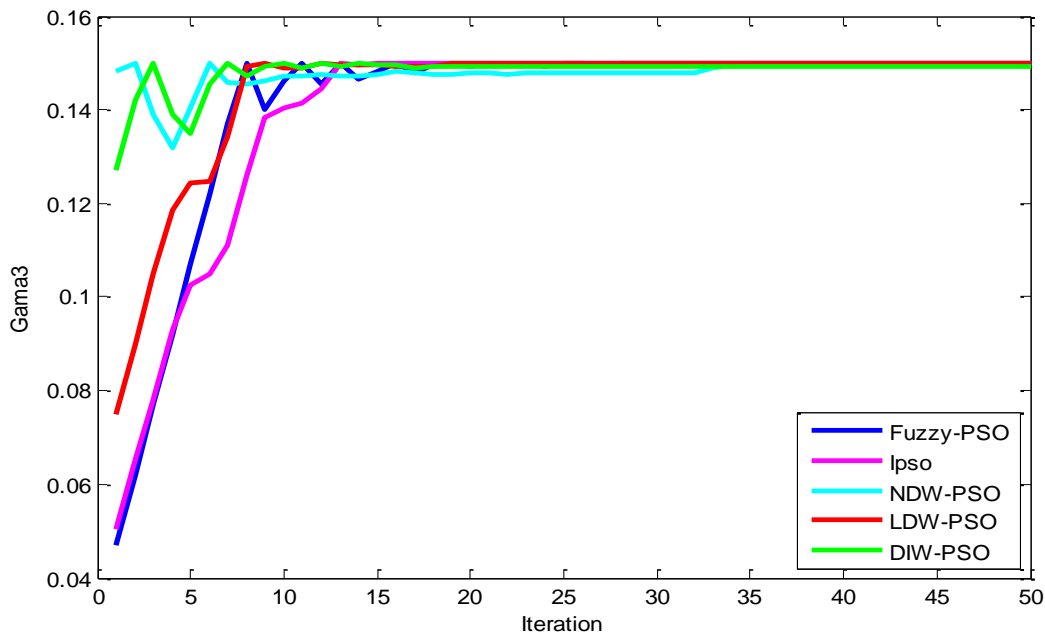
شکل ۴۲-۶: پارامتر Γ_1 در الگوریتم‌های مختلف

شکل (۴۳-۶) مقایسه سرعت همگرایی پارامتر Γ_2 را در الگوریتم‌های مختلف نشان می‌دهد.



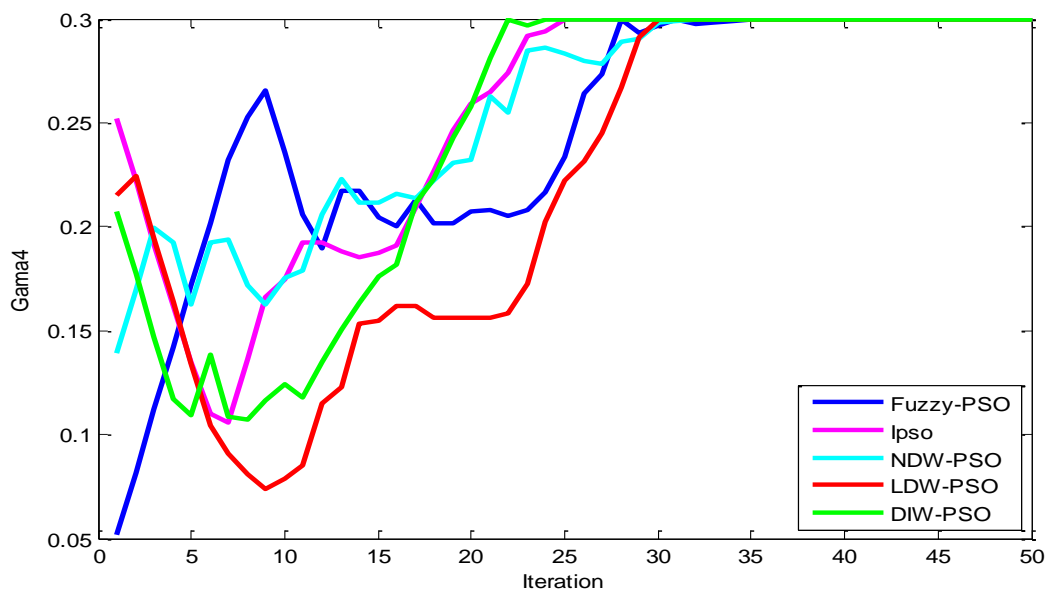
شکل ۴۳-۶: پارامتر Γ_2 در الگوریتم‌های مختلف

شکل (۴۴-۶) مقایسه سرعت همگرایی پارامتر Γ_3 را در الگوریتم‌های مختلف نشان می‌دهد.



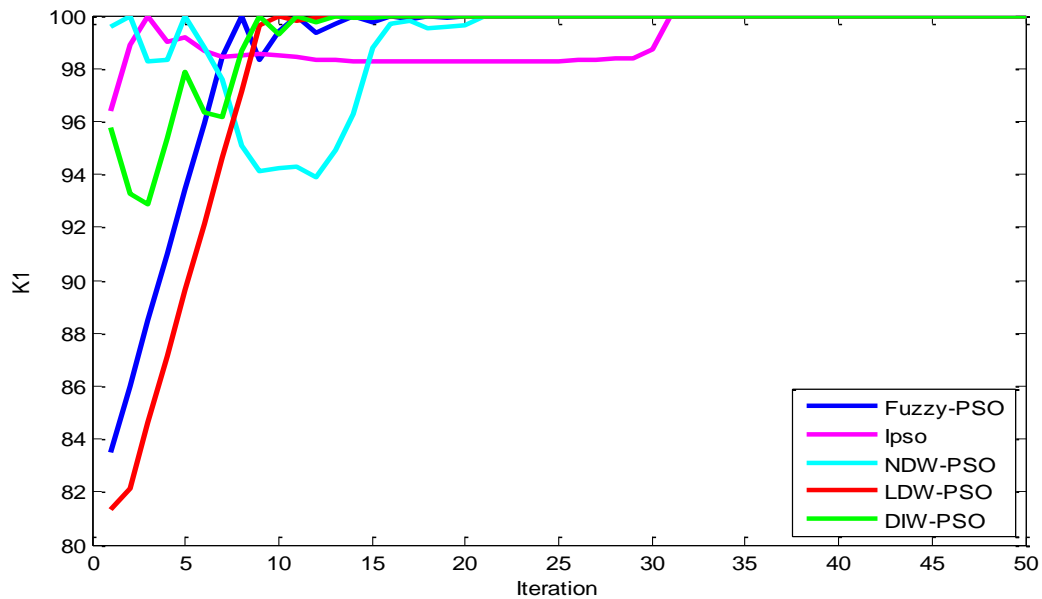
شکل ۴۴-۶: پارامتر Γ_3 در الگوریتم‌های مختلف

شکل (۴۵-۶) مقایسه سرعت همگرایی پارامتر Γ_4 را در الگوریتم‌های مختلف نشان می‌دهد.



شکل ۴۵-۶: پارامتر Γ_4 در الگوریتم‌های مختلف

شکل (۶-۴۶) مقایسه سرعت همگرایی پارامتر K_1 را در الگوریتم‌های مختلف نشان می‌دهد.



شکل ۶-۴۶: پارامتر K_1 در الگوریتم‌های مختلف

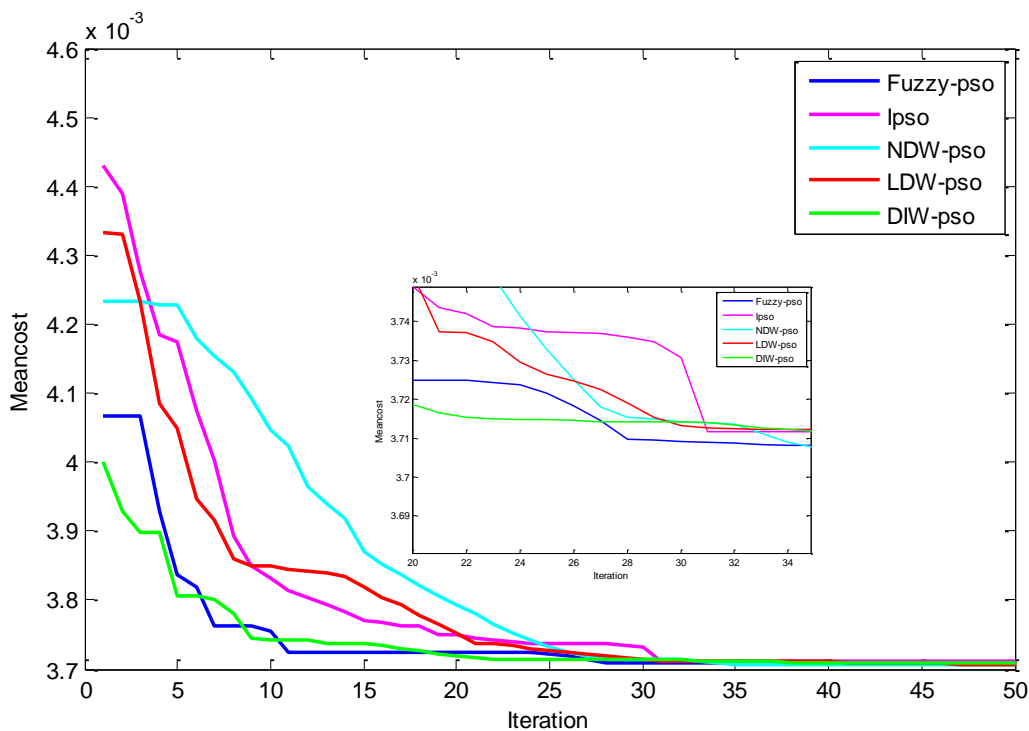
جدول ۶-۱: نتایج مقایسه پارامترهای کنترل بین الگوریتم‌های بهینه سازی پرندگان

پارامترهای کنترل	مقدار واقعی	Mean value LDW-PSO	Mean value NDW-PSO	Mean value DIW-PSO	Mean value IPSO	Mean value Fuzzy-PSO
Γ_1	0.03	0.0595	0.0591	0.0668	0.0420	0.0415
Γ_2	0.05	0.0500	0.0499	0.0500	0.0482	0.0500
Γ_3	0.1	0.1499	0.1499	0.1499	0.1499	0.1500
Γ_4	0.3	0.3000	0.2999	0.2949	0.2388	0.2996
Λ_1	20	19.9996	19.9987	19.9984	19.9986	19.9991
K_1	100	99.9987	99.9981	99.9837	99.9960	99.9869

پارامترهای کنترل کننده بدست آمده از هر یک روش های الگوریتم بهینه سازی پرندگان در جدول (۱-۶) و شکل های (۴۱-۶) تا (۴۶-۶) نشان داده شده است. مطابق جدول (۱-۶) و شکل های (۴۱-۶) تا (۴۶-۶)، کنترل کننده به دست آمده از الگوریتم بهینه سازی پرندگان با استفاده از منطق فازی به کمترین هزینه خطای انتگرال مجذور خطا رسیده است در نتیجه بهترین ردیابی را داشته است.

شکل (۴۷-۶) مقایسه سرعت همگرایی مسیر تابع هزینه را در الگوریتم های مختلف نشان می-

دهد.

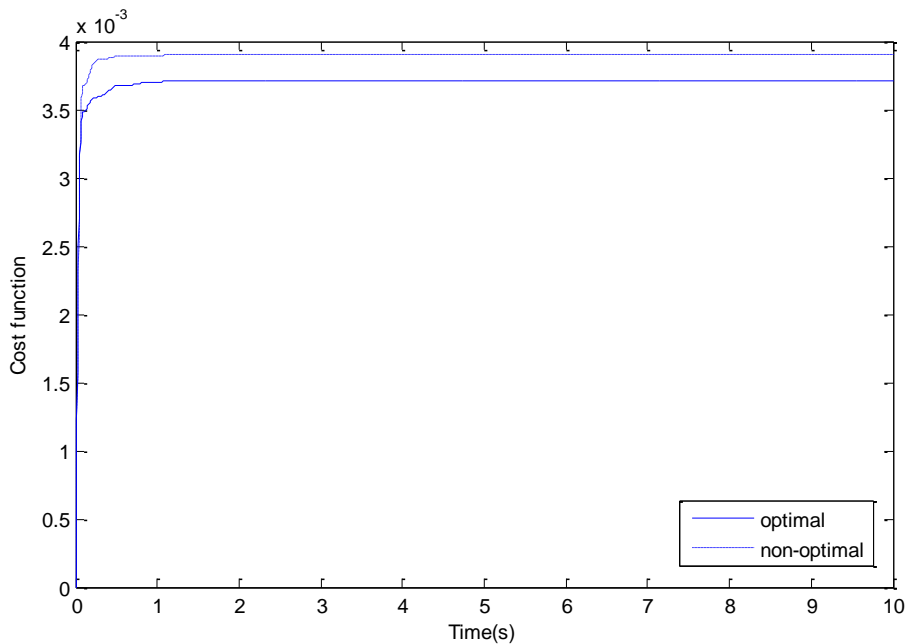


شکل ۴۷-۶: تابع هزینه در الگوریتم های مختلف

هدف طراحی، تعیین ضرایب کنترل کننده به گونه ای بود که انتگرال مجذور خطا کمینه شود. همان طور که مشخص است الگوریتم بهینه سازی پرندگان با استفاده از منطق فازی تابع هزینه کمتری دارد و عملکرد بهتری نسبت به سایر الگوریتم ها دارد.

۲-۶-۶ مقایسه بین سیستم بهینه و غیربهینه

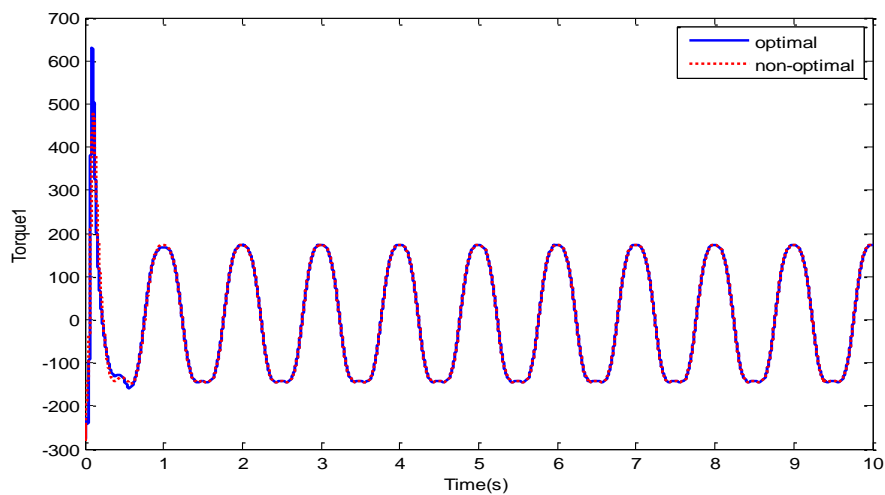
در این قسمت الگوریتم بهینه‌سازی اجتماع پرندگان با استفاده از منطق فازی به مسئله طراحی کنترل کننده بهینه تطبیقی برای سیستم ربات اعمال می‌شود. هدف کنترل مسیر عملگر نهایی است که در این جا کنترل ردگیری است و مسیر مطلوب برای مکان عملگر نهایی به صورت رابطه (۵-۴) و (۵-۵) است. با توجه به این که الگوریتم بهینه‌سازی پرندگان با استفاده از منطق فازی در قسمت (۶-۶-۵) عملکرد بهتری به سایر الگوریتم‌ها دارد. در این قسمت سیستم بهینه شده توسط این روش با سیستم غیر بهینه مقایسه می‌شود.



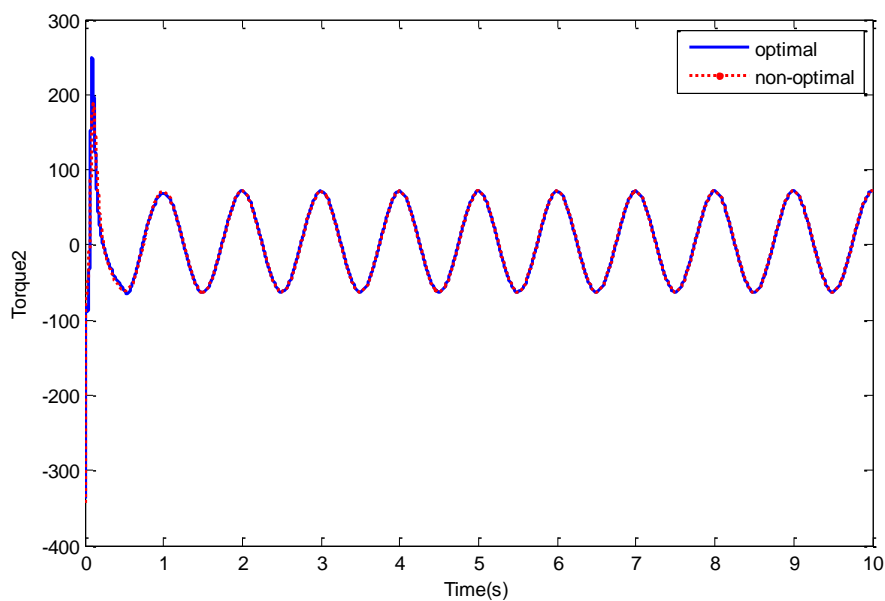
شکل ۶-۴۸: تابع هزینه

شکل (۶-۴۸) تابع هزینه سیستم را برای قبل و بعد بهینه‌سازی نشان می‌دهد و مشخص است

که تابع هزینه سیستم بهینه شده مینیمم شده است و خطای ردگیری کاهش یافته است.



شکل ۶-۴۸: سیگنال کنترلی



شکل ۶-۵۰: سیگنال کنترلی

همان طور که در شکل‌های (۶-۴۹ و ۶-۵۰) مشخص است سیگنال کنترل برای قبل و بعد بهینه سازی مقایسه شده است. همانطور که مشخص است برای گشتاور اول و دوم در موقع بهینه، مقدار فراجهدش کمی افزایش یافته است به دلیل این که شاخص عملکرد دارای انتگرال می باشد. برای سیگنال کنترل هر دو روش مقدارشان بر هم منطبق است.

فصل هفتم:

نتایج و پیشنهادها

۱-۷ نگرش کلی

هدف این پایان نامه توسعه دادن و به کار بردن یک روش بهینه‌سازی بر روی مدل ریاضی بازوی ماهر ربات دو لینکی می‌باشد و هم چنین مقایسه نتایج به دست آمده بین سیستم غیربهینه و سیستم بهینه شده صورت گرفته است. اگر چه همه روش‌ها، نتایج قابل قبولی برای تابع هدف به دست آورده- اند ولی روش پیشنهادی نتایج بهتری داشته است.

۲-۷ نتیجه‌گیری

در این پایان نامه، الگوریتم بهینه‌سازی اجتماع پرندگان برای کنترل بهینه غیرخطی بازوی ربات ارائه گردیده است. الگوریتم بهینه‌سازی اجتماع پرندگان براساس بهینه‌سازی پارامترهای کنترل، کنترل بهینه را نتیجه می‌دهد. استفاده از الگوریتم بهینه‌سازی اجتماع پرندگان برای حل مسئله کنترل بهینه غیرخطی ربات دارای مزایای قابل توجهی می‌باشد. این روش هیچ‌گونه وابستگی به معادلات حالت ندارد. همچنین این روش بر خلاف سایر روش‌های ابتکاری برای حل مسائل کنترل بهینه، به حدس‌های اولیه چندان حساس نیست و قابلیت همگرایی بالایی را دارا می‌باشد. از امتیازات اصلی دیگر این الگوریتم ، سادگی در پیاده‌سازی و کارایی بالا و کم هزینه بودن در پردازش محاسباتی است. الگوریتم بهینه‌سازی اجتماع پرندگان را باید بر روی مدل‌های تکرارپذیر اجرا کنیم.

روش الگوریتم بهینه‌سازی پرندگان ضریب اینرسی به صورت کاهشی خطی، ابتدا در نظر گرفته شده است و دومین الگوریتم ضریب اینرسی به صورت کاهشی غیرخطی، سومین روش الگوریتم بهینه‌سازی با ضریب اینرسی دینامیکی ، چهارمین روش بهبود یافته الگوریتم پرندگان و پنجمین روش الگوریتم پیشنهادی با استفاده از منطق فازی می‌باشد. در الگوریتم‌های بهینه‌سازی اجتماع پرندگان روش اول تا سوم، ضریب اینرسی برای جستجو از مناطق محلی و سراسری، به صورت خطی و غیرخطی کاهش می‌یابد. الگوریتم بهینه‌سازی پرندگان بهبود یافته، برای بهبود پارامترهای الگوریتم

بهینه‌سازی اجتماع پرندگان از قبیل ضریب اینرسی و ضرایب یادگیری استفاده شده است. در الگوریتم پیشنهادی ضریب اینرسی با استفاده از الگوریتم بهبود یافته و ضریب یادگیری با استفاده از منطق فازی، بهبود و توسعه یافته است. انتخاب مناسب این پارامترها منجر به عملکرد بهتر سیستم و مینیمم کردن خطای ردگیری می‌شود. به طور کلی تاثیر، انعطاف‌پذیری و کاربرد الگوریتم بهینه‌سازی اجتماع پرندگان و روش‌های مختلف الگوریتم برای مسئله کنترلی پیشنهادی نشان داده شده است.

۳-۷ پیشنهادات

روش‌های ابتکاری در چند دهه گذشته توسعه پیدا کرده است تا حل مسائل بهینه‌سازی آسان شود زیرا قبلا حل آن‌ها سخت یا غیرممکن بوده است. در این پایان‌نامه ایده مطرح شده باعث عملکرد بهتر سیستم و مینیمم کردن خطای ردگیری شده است، به منظور ارائه راه کار برای محققین بعدی، پیشنهادات زیر ارائه می‌شود:

۱. امکان اجرای الگوریتم به صورت آنلاین ایجاد شود.
۲. استفاده از روش‌های مختلف بهینه‌سازی هوشمند و مقایسه آن‌ها با هم
۳. ترکیب روش‌های هوشمند با یکدیگر برای به دست آوردن کنترل بهینه در برابر اغتشاشات

خارجی و اصطکاک

مراجع

- [1] Alvarez-Ramirez, J., Cervantes, I., Kelly, R. (2000). "PID regulation of robot manipulators". stability and performance. *System & Control Letters*. 41, 73-83.
- [2] Burkan, R., Uzmay, I. (2005). "A model of parameter adaptive law with time varying function for robot control". *Applied Mathematical Modelling*. 29, 361-371.
- [3] Faieghi, M.R., Delavari, H., Baleanu, D. (2012). "A novel adaptive controller for two-degree of freedom polar robot with unknown perturbations". *Commun Nonlinear Sci Numer Simulate*. 17, 1021-1030.
- [4] Zeinali, M., Notash, L. (2010). "Adaptive sliding mode control with uncertainty estimator for robot manipulators". *Mechanism and Machine Theory*. 45, 80-90.
- [5] Choi, Y., Chung, W.K., Youm, Y. (2001). "On the Optimal PID Performance Tuning for Robot Manipulators". *IEEE/RSJ International Conference On Advanced Intelligent Robots and Systems*, Maui, Hawaii, US, 2001.
- [6] Wai, R. J., Tu, C. H., Hsieh, K.Y. (2003). "Design of Intelligent Optimal Tracking Control for Robot Manipulator". *IEEE/ASME International Conference On Advance Intelligence Mechatronics*. 2003.
- [7] Perez P, J., Perez, J. P., Soto, R., Flores, A., Rodriguez, F., Meza, J. L. (2012). "Trajectory Tracking Error Using PID Control Law for Two Link Robot Manipulator Via Adaptive Neural Networks". *Procedia Technology*. 3, 139-146.
- [8] Khan, B., Arif, J., Riaz, S. (2010). "Nonlinear Optimal Position Control Design for 2-Link RR-Robot Manipulator". *6th International Conference on Emerging Technologies*, (2010).
- [9] Wai, R. J., Tu, C.H., Hsieh, K. Y. (2003). "Design of Intelligence Optimal Control for Robot Manipulator". *International Conference on Advanced Intelligence*

Mechatronics. 1, 20-24.

[10] Mehdi, H., Boubaker, O. (2011). “ Position/force control optimized by Particle Swarm intelligence for constrained robotic manipulator”. 11th International Conference on Intelligent System Design and Applications, 190-195.

[11] Girirajkumae, S. M., Jayaraj, D., Kishan, A. R. (2010). “PSO based Tuning of a PID Controller for a High Performance Drilling Machine”. International Journal of Computer Applications. vol. 1, pp. 0975-8887.

[12] Cao, S., Tu, J., Liu, H. (2010). “PSO Algorithm-Based Robust design of PID Controller for PMSM”. Sixth International Conference on natural Computation, 2010.

[13] Chang, W.D., Shih, S.P. (2010). “PID controller design of nonlinear systems using an improved particle swarm optimization approach”. Commun Nonlinear Sci Numer Simulat. 15, 3632-3639.

[14] Sun, S., Zhang, J., Wang, J., Xu, L. (2011). “The Application of New Adaptive PSO in AGC and AFC Combination Control System”. Procedia Engineering.16, 702-707.

[15] Chen, S.M., Dong, Y.F. (2011). “Satellite Attitude Tracking Controller Optimization based on Particle Swarm Optimization”. Procedia Engineering. 15, 526-530.

[16] Jaberipour, M., Khorram, E., Karimi, B. (2011). “Particle swarm algorithm for solving systems of nonlinear equations”. Computers and Mathematics with Application. 62, 566-576.

[17] Modares, H., Naghibi Sistani, M.B. (2011). “Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm”. Engineering Application of Artificial Intelligence. 24, 476-484.

[18] Spong, M.W., Vidyasagar, M. (1989). “Robot dynamics and control”. New York. Wiley.

[19] Kelly, R., Salgado, R. (1994). “PD control with computed feedforward of robot manipulators: a design procedure”. IEEE Transactions on Robotics and Automation. 10(4):566-571.

[20] Kennedy, J., Eberhart, R.c. (1995). “Particle swarm optimization”. IEEE International Conference on Neural Networks. vol. 4, pp. 1942-1948 .

[21] Yang, X. S. (2010). “**Engineering Optimization An Introduction with**

Metaheuristic Applications". New Jersey, John Wiley and Sons, Inc.

[22] Gazi, V., Passino, K. M. (2011). "**Swarm Stability and Optimization**". Springer Science.

[23] Shi, Y., Eberhart, R. (1998). "A Modified Particle Swarm Optimizer". IEEE International Conference On Evolutionary Computation, USA. 255-262.

[24] Perez, R.E., Behdinan, K., (2007). "Particle swarm approach for structural design optimization. Computers and structures". 85, 1579-1588.

[25] Alatas, B., Akin, E., Ozer, A.B. (2009). "Chaos embedded particle swarm optimization algorithms". Chaos, Solitons and fractals. 40(4), 1715-1734.

[26] Riccardo, P., Kennedy, J., Blackwell, T. (2007). "Particle swarm optimization. Swarm Intelligence". (1)1, 33-57.

[27] Engelbrecht, A.P. (2007). "**Computational Intelligence: An Introduction**" (second edition). John Wiley and Sons Ltd.

[28] Sage, Andrew P., White, Chelsea C. (1977). "**OPTIMUM SYSTEMS CONTROL**" (second ed.). New Jersey, Prentice-Hall, Inc.

[29] Hassan, R., Cohanin, B. (2005). "A Comparison Of Particle Swarm Optimization and The Genetic Algorithm". 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamic & Materials Conference. Austin, Texas. 18-21.

[30] Shi, Y., Eberhart, R.C. (1998a). "Parameter selection in particle swarm optimization". Seventh Annual Conference on Evolutionary Programming, New York, pp. 591-600.

[31] Shi, Y., Eberhart, R.C. (1998b). "A modified particle swarm optimizer". Conference on Evolutionary Computation, pp. 69-73.

[32] Chatterjee, A., Siarry, P. (2006). "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization", Computers and Operations research, 33(3), 859-871.

[33] Jiao, B., Lian, z., Gu, X. (2008). "A dynamic inertia weight particle swarm optimization algorithm". Chaos, Solitons and Fractals. 37, 698-705.

[34] Modares, H., Alfi, A., Fateh M.M. (2010). "Parameter identification of chaotic dynamic

systems through an improved particle swarm optimization". Expert Systems with Applications. 37, 3714-3720 .

[35] Shi, Y. (2001). "Fuzzy Adaptive Particle Swarm Optimization". Proceeding of the Congress on Evolutionary computation, vol. 1, pp. 101-106.

[36] Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., Valdez, M. (2013). "Optimal design of fuzzy classification system using pso with dynamic parameter adaptation through fuzzy logic. Expert system with applications". 40, 3196-3206.

[۳۷] وانگ، لی، ۱۳۷۸، "سیستم‌های فازی و کنترل فازی"، (ترجمه، م. تشنه لب،

ن. صفاپور، د. افیونی)، چاپ چهارم، دانشگاه صنعتی خواجه نصیر طوسی، تهران

[۳۸] شعبانی‌نیا، ف.، سعیدنیا، س.، ۱۳۸۸، "مقدمه‌ای بر منطق فازی با استفاده از

MATLAB"، چاپ دوم، انتشارات خانیران، تهران

[۳۹] شوندی، ح.، ۱۳۸۵، "نظریه‌های مجموعه‌های فازی و کاربرد آن در مهندسی

صنایع و مدیریت"، گسترش علوم پایه، تهران

[۴۰] اسپانگ، م. دبلیو، ویدیا ساگار، ام.، ۱۳۸۷، "کنترل و دینامیک ربات"، (ترجمه،

م.م. فاتح)، چاپ دوم، دانشگاه صنعتی شاهرود، شاهرود.

[۴۱] مطیع قادر، ح.، لطفی، ش.، سید اسفهلان، م.، ۱۳۸۹، "مروری بر برخی از روش‌های

بهینه‌سازی هوشمند"، دانشگاه آزاد اسلامی، شبستر

[۴۲] هاشم زاده، م.، ۱۳۹۲، پایان‌نامه کارشناسی ارشد، "کنترل بهینه مود لغزشی یک ربات

در حضور عدم قطعیت با استفاده از الگوریتم پرندگان"، دانشکده برق و رباتیک، دانشگاه شاهرود

Abstract

In this thesis, a nonlinear optimal control problem is investigated on the two-link artist robot manipulator. The maximum difference in between robots are their mechanical parts and how to control them. Minimizing the tracking error and tracking control are important purpose and considerable in control systems. In this study, particle swarm optimization algorithm(PSO) is used to design the optimal controller. Applying the particle swarm optimization algorithm is useful for highly complex systems in optimization problems because the PSO is easy to be implemented and independent of the model. The PSO results the optimal control base on optimization control parameters. In this algorithm, the optimal value of control parameters are determined by minimizing the cost function, and the integral square of errors (ISE) as a performance criteria. In this case, the performance of PSO algorithms such as an improved PSO (IPSO), an improved PSO using fuzzy logic (F-PSO), a linearly decreasing inertia weight of PSO (LWD-PSO) and a nonlinearly decreasing inertia weight of PSO (NDW-PSO) are compared in terms of parameter accuracy and convergence speed. As a result, the simulation results show that the F-PSO approach presents a better performance in the tracking control of robot manipulators than other algorithms.

Key Words: particle swarm optimization (PSO), robot manipulators, optimization



Shahrood University of Technology

Department of Electrical and robotic Engineering

**Nonlinear Optimal Control on Two-linked Articulated Robot
Manipulator**

Fatemeh Soleimannoori

Supervisor:

Dr. Mohammad Haddad Zarif

Advisor:

Dr. Mohammad Mahdi Fateh

2014