





دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه کنترل

کنترل غیر خطی بهینه ربات کروی با بکارگیری

الگوریتم بهینه سازی پرندگان

استاد :

دکتر محمد مهدی فاتح

ارائه دهنده

سعید خراشادی زاده

شهریورماه ۱۳۹۰

تقدیم به پیش‌گام صاحب‌امر؛

حضرت بقیة الله الاعظم (عجل الله تعالی فرجه الشریف)؛

او که صدای گام‌های آمدنش، بر عالم طنین افکن شده

با سپاس فراوان از استاد راهنما

دکتر محمد مهدی فاتح

که درخت دانشم را با علم خود آبیاری کردند.

و با تشکر از پدر و مادر عزیزتر از جانم،

این دو گنجینه آسمانی بر زمین

که بهترین مشوقم در مسیر تحصیل بوده‌اند،

و با تشکر از همسر مهربانم.

تعهدنامه

اینجانب سعید خراشادی زاده دانشجوی دوره کارشناسی ارشد رشته مهندسی برق-گرایش کنترل دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه کنترل بهینه غیرخطی بازوی ماهر کروی با استفاده از الگوریتم بهینه سازی پرندگان تحت راهنمایی دکتر محمد مهدی فاتح متعهد می شوم

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

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

این پایان نامه به کنترل غیرخطی مقاوم ربات‌های الکتریکی می‌پردازد و روشی جدید مبتنی بر کنترل ولتاژ برای جبران خطای ردگیری ناشی از اغتشاش خارجی، دینامیک‌های مدل نشده و عدم قطعیت پارامتری ارائه می‌دهد. قانون کنترل روش کنترل ولتاژ در مقایسه با روش مرسوم کنترل گشتاور بسیار ساده‌تر است. زیرا نیازی به مدل غیرخطی پیچیده ربات ندارد. در نتیجه، حجم محاسبات کنترل کننده برای تعیین ولتاژ اعمالی به موتورها به مراتب کمتر از حجم محاسبات برای تعیین گشتاور مفاصل در استراتژی کنترل گشتاور می‌شود. در این پایان‌نامه از الگوریتم بهینه‌سازی پرندگان برای بدست آوردن یک تخمین بهینه از پارامترهای قانون کنترل به منظور دست یافتن به کمترین خطای ردگیری، استفاده می‌شود. پایداری سیستم حلقه بسته با استفاده از روش مستقیم لیپانوف تضمین شده است و عملکرد سیستم کنترل با بکارگیری الگوریتم بهینه‌سازی پرندگان بهبود می‌یابد. قانون کنترل پیشنهادی روی یک بازوی ماهر کروی شبیه‌سازی می‌شود. این ربات توسط موتورهای الکتریکی مغناطیس دائم رانده می‌شود. نتایج شبیه‌سازی‌ها به خوبی توانایی الگوریتم بهینه‌سازی پرندگان را در کاهش دادن خطای ردگیری نشان می‌دهد. همچنین مقایسه نتایج روش‌های کنترل ولتاژ و کنترل گشتاور مزایای روش کنترل ولتاژ را مشخص می‌سازد.

کلمات کلیدی: بازوی ماهر، استراتژی کنترل گشتاور، استراتژی کنترل ولتاژ، کنترل مقاوم غیرخطی،

الگوریتم بهینه‌سازی پرندگان

مقالات مستخرج از پایان نامه

- 1.** Mohammad Mehdi Fateh, Saeed Khorashadizadeh, “ *Optimal Control of Electrically Driven Robots Using Particle Swarm Optimization*”, Accepted in: *International Review of Electrical Engineering*

فهرست مطالب

فصل اول

- مقدمه ۱
- ۱-۱- مرور کارهای گذشته ۲
- ۱-۱-۱- کنترل گشتاور ۲
- ۱-۱-۲- کنترل ولتاژ ۷
- ۱-۳- الگوریتم بهینه سازی پرندگان ۱۰
- ۱-۲- اهداف مورد نظر ۱۳
- ۱-۳- طرح کلی پایان نامه ۱۴

فصل دوم: الگوریتم بهینه سازی پرندگان

- ۱-۲- مقدمه ۱۷
- ۲-۲- چگونگی شکل گیری الگوریتم بهینه سازی پرندگان ۲۰
- ۱-۲-۲- شبیه سازی رفتار اجتماعی برخی موجودات زنده ۲۰
- ۲-۲-۲- عوامل "تطابق سرعت با نزدیک ترین همسایه" و "گیجی" ۲۱
- ۲-۲-۳- جستجوی چند بعدی ۲۴

- ۲۴.....۴-۲-۲- ایجاد شتاب توسط فاصله
- ۲۵.....۵-۲-۲- روابط ساده شده فعلی
- ۲۶.....۳-۲- مقایسه بین الگوریتم ژنتیک و الگوریتم بهینه سازی پرندگان
- ۲۷.....۴-۲- بهینه سازی توابع
- ۳۰.....۵-۲- شناسایی سیستم‌ها
- ۳۲.....۶-۲- کاربرد الگوریتم بهینه‌سازی پرندگان در شناسایی و کنترل سیستم‌ها

فصل سوم: مروری بر مدلسازی سینماتیکی و دینامیکی بازوهای ماهر مکانیکی

- ۳۸.....۱-۳- مقدمه
- ۳۸.....۲-۳- سینماتیک مستقیم
- ۴۱.....۱-۲-۳- ماتریس دوران
- ۴۲.....۲-۲-۳- بردار انتقال
- ۴۴.....۳-۲-۳- الگوریتم دناویت هارتنبرگ
- ۴۷.....۳-۳- سینماتیک وارون
- ۴۸.....۴-۳- ماتریس ژاکوبین
- ۵۱.....۱-۴-۳- وضعیت های تکین

۵-۳-۵- مدلسازی دینامیکی..... ۵۲

۳-۵-۱- انرژی جنبشی..... ۵۲

۳-۵-۲- انرژی پتانسیل..... ۵۳

۳-۵-۳- لاگرانژین..... ۵۳

۳-۵-۴- معادله اویلر-لاگرانژ..... ۵۳

۳-۶-۶- مدل ریاضی بازوی ماهر کروی..... ۵۴

فصل چهارم: کنترل گشتاور سیستم های رباتیک

۴-۱-۱- مقدمه..... ۵۸

۴-۲-۲- دینامیک وارون (گشتاور محاسباتی)..... ۵۸

۴-۳-۳- کنترل غیر خطی مقاوم..... ۶۰

نتایج شبیه سازی ها..... ۶۴

فصل پنجم: کنترل ولتاژ ربات های الکتریکی

۵-۱-۱- مقدمه..... ۶۸

۵-۲-۲- معرفی استراتژی کنترل ولتاژ..... ۶۸

نتایج شبیه سازی ها..... ۷۲

۵-۳-۳- کنترل ولتاژ مقاوم..... ۷۳

نتایج شبیه سازی ها..... ۷۹

فصل ششم: کنترل بهینه غیر خطی

۶-۱- مقدمه ۸۴

۶-۲- کنترل بهینه گشتاور..... ۸۵

۶-۳- کنترل بهینه ولتاژ ۹۲

فصل هفتم

نتیجه گیری..... ۹۸

پیوست ها

پیوست ۱..... ۹۹

مراجع

..... ۱۰۱

فهرست شکل‌ها

فصل دوم

- شکل (۱-۲): نمودار تابع $f(x)$ در (۱۲-۲)..... ۲۸
- شکل (۲-۲): همگرایی ۲۰ پرده به سمت جواب بهینه..... ۲۹
- شکل (۳-۲): همگرایی g_{best} به سمت نقطه بهینه..... ۲۹
- شکل (۴-۲): بلوک دیاگرام شناسایی پارامتر..... ۳۱
- شکل (۵-۲): همگرایی پارامترهای سیستم خطی مرتبه اول..... ۳۴
- شکل (۶-۲): همگرایی پارامترهای سیستم غیر خطی..... ۳۵
- شکل (۷-۲): بلوک دیاگرام سیستم کنترل تعلیق خودرو..... ۳۶
- شکل (۸-۲): مقایسه عملکرد الگوریتم بهینه سازی پرندگان و الگوریتم ژنتیک در طراحی سیستم تعلیق خودرو..... ۳۶

فصل سوم

- شکل (۱-۳): دیاگرام مفصلی ربات هنرمند..... ۳۹
- شکل (۲-۳): دیاگرام مفصلی ربات اسکارا..... ۳۹
- شکل (۳-۳): دیاگرام مفصلی ربات استنفورد..... ۴۰
- شکل (۴-۳): دیاگرام مفصلی ربات کروی..... ۴۰

شکل (۳-۵): محورهای مختصات دوران یافته..... ۴۱

شکل (۳-۶) دستگاه مختصات انتقال یافته..... ۴۲

شکل (۳-۷): اختصاص محورهای مختصات به ربات کروی طبق الگوریتم دناویت هارتنبرگ..... ۴۶

شکل (۳-۸): دیاگرام مفصلی ربات کروی برای انجام سینماتیک وارون..... ۴۸

فصل چهارم

شکل (۴-۱): خطای ردگیری روش خطی سازی فیدبکی بدون اعمال عدم قطعیت..... ۵۹

شکل (۴-۲): سیگنال کنترل (گشتاور) روش خطی سازی فیدبکی بدون اعمال عدم قطعیت..... ۶۰

شکل (۴-۳): عملکرد روش کنترل مقاوم غیر خطی با اعمال عدم قطعیت..... ۶۶

شکل (۴-۴): سیگنال کنترل (گشتاور) روش کنترل مقاوم غیر خطی..... ۶۶

فصل پنجم

شکل (۵-۱): بلوک دیاگرام یک موتور و بار آن..... ۷۰

شکل (۵-۲): سیگنال ولتاژ موتورها در روش کنترل ولتاژ بدون عدم قطعیت..... ۷۲

شکل (۵-۳): خطای ردگیری روش کنترل ولتاژ بدون عدم قطعیت..... ۷۳

شکل (۵-۴): نمودار $\|M\|$ ۸۰

شکل (۵-۵): نمودار $\|H\|$ ۸۰

شکل (۵-۶): خطای ردگیری روش کنترل ولتاژ مقاوم با اعمال عدم قطعیت..... ۸۱

شکل (۵-۷): سیگنال کنترل (ولتاژ) روش کنترل ولتاژ مقاوم با اعمال عدم قطعیت..... ۸۲

فصل ششم

شکل (۶-۱): چگونگی کاهش تابع هزینه مربوط به بهترین..... ۹۰

شکل (۶-۲): تغییرات پارامتر θ_1 بهینه در ۱۰۰ گام شبیه سازی..... ۹۰

شکل (۶-۳): خطای ردگیری جواب بهینه کنترل گشتاور..... ۹۱

شکل (۶-۴): سیگنال کنترل جواب بهینه در کنترل گشتاور..... ۹۲

شکل (۶-۵): خطای ردگیری یکی از جوابهای تصادفی اولین تکرار الگوریتم بهینه سازی پرندگان..... ۹۵

شکل (۶-۶): تغییرات پارامتر θ_1 بهینه در کنترل ولتاژ..... ۹۶

شکل (۶-۷): کاهش تابع هزینه جواب بهینه در کنترل ولتاژ..... ۹۶

فهرست جدول‌ها

فصل دوم

جدول ۱-۲ داده های ورودی خروجی برای شناسایی پارامترهای فنر..... ۳۱

فصل سوم

جدول (۱-۳): جدول دناویت هارتنبرگ برای ربات کروی..... ۴۶

فصل پنجم

جدول (۱-۵): پارامترهای موتور..... ۷۲

فصل ششم

جدول (۱-۶): مقادیر واقعی پارامترهای ربات..... ۸۹

جدول (۲-۶): پارامترهای بهینه در کنترل گشتاور..... ۹۱

جدول (۳-۶): پارامترهای بهینه در کنترل ولتاژ..... ۹۶

فصل اول

مقدمه

❖ مرور کارهای گذشته

❖ اهداف مورد نظر

❖ طرح کلی پایان نامه

۱-۱- مرور کارهای گذشته

۱-۱-۱- کنترل گشتاور

بسیاری از روش‌های ارائه شده برای کنترل سیستم‌ها اعم از خطی و غیرخطی مبتنی بر شناخت دقیق مدل سیستم می‌باشند و موفقیت آنها منوط به در اختیار داشتن مدل دقیق سیستم است. متأسفانه بدست آوردن مدل دقیق سیستم بسیار مشکل، وقت گیر و گاهی غیرممکن می‌باشد. چون ممکن است برخی از دینامیک‌های سیستم مانند اصطکاک، تکرار پذیر نباشند یا نتوان مدل دقیقی برای آنها پیشنهاد داد. علاوه بر این، ممکن است پارامترهای مدل سیستم با گذشت زمان یا تحت تأثیر شرایطی خاص تغییر کند. بنابراین، عدم قطعیت^۱ همواره یکی از مهمترین چالش‌های سیستم‌های کنترل، می‌باشد.

امروزه اتوماسیون نقش مهمی در زندگی بشر به عهده دارد و بسیاری از جنبه‌های زندگی فردی و اجتماعی انسان از قبیل بهداشت، صنعت، کشاورزی، اقتصاد، قانون، سیاست و ... تحت تأثیر آن قرار دارد. معمولاً اتوماسیون به معنی استفاده از رباتها در محیط‌های صنعتی تلقی می‌شود [۱]. در حال حاضر، در صنعت از ربات‌ها برای جوشکاری، ماشین‌کاری و مونتاژکاری خط تولید استفاده می‌شود. بنابراین، بهبود عملکرد سیستم‌های کنترل ربات‌ها تأثیر بسزایی در کیفیت محصولات و افزایش راندمان تولید دارد. به همین دلیل، طراحی سیستم‌های کنترل ربات‌ها توجه بسیاری از محققان را به خود جلب کرده است. و روش‌های بسیاری برای کنترل آنها ارائه شده است که مهم‌ترین آنها در حوزه کنترل غیرخطی عبارتند از: خطی سازی فیدبکی^۲، کنترل مود لغزشی^۳ و طراحی پس گام^۴. در بین این روش‌ها، خطی سازی فیدبکی جذابیت و محبوبیت بیشتری دارد. این روش بسیار کلی است و می‌توان از آن در کنترل انواع سیستم‌ها

¹ Uncertainty

² Feedback linearization

³ Sliding mode control

⁴ Back stepping design

اعم از خطی و غیر خطی و حتی سیستم‌های چند متغیره استفاده کرد. به عنوان مثال، در مقاله [۲] روشی برای کنترل یک سیستم قدرت چند ورودی-چند خروجی^۱ با استفاده از خطی سازی فیدبکی ارائه شده است. خطی سازی فیدبکی توجه بسیاری از محققان را در حوزه کنترل ربات به خود جلب کرده است. با استفاده از این روش می‌توان به راحتی دینامیک غیر خطی پیچیده ربات را که دارای کوپلینگ^۲ بالا نیز می‌باشد، به چند معادله دینامیکی خطی مرتبه دوم تبدیل کرد. این روش، در رباتیک به نام‌های گشتاور محاسباتی^۳، دینامیک وارون^۴ یا کنترل گشتاور مشهور است.

خطی سازی فیدبکی روشی مبتنی بر مدل می‌باشد. به همین دلیل محققان تلاش‌های فراوانی جهت مقاوم کردن این روش و غلبه بر عدم قطعیت‌ها انجام داده‌اند. به طور کلی می‌توان این تحقیقات را به سه حوزه کنترل تطبیقی، کنترل مقاوم و کنترل فازی تقسیم کرد که در این فصل مروری مختصر بر آنها خواهیم داشت.

تحقیقات گسترده‌ای برای طراحی سیستم‌های کنترل تطبیقی ربات‌های صلب^۵ به منظور تضمین پایداری سیستم کنترل و محدود ماندن سیگنال‌های داخلی انجام شده است. طبقه‌بندی جامعی از روش‌های تطبیقی ارائه شده است [۳] که آنها را به دو گروه عمده تقسیم می‌کند: روش‌های مبتنی بر دینامیک وارون و روش‌های مبتنی بر غیرفعال بودن^۶. دینامیک وارون تطبیقی را می‌توان با فرض‌های مختلفی پیاده سازی کرد. به عنوان مثال، این روش با فرض اندازه‌گیری شتاب و کراندار بودن معکوس ماتریس اینرسی^۷ ارائه شده است. همچنین روشی ارائه شده است که به فرض کراندار بودن معکوس

¹ Multi-input multi-output

² Coupling

³ Computed torque

⁴ Inverse dynamics

⁵ Rigid

⁶ Passivity

⁷ Inertia matrix

ماتریس اینرسی نیاز ندارد. اما می‌توان دینامیک وارون تطبیقی را طوری پیاده سازی کرد که به فرض اندازه گیری شتاب نیاز نداشته باشد. قوانین تطبیق^۱ در دینامیک وارون تطبیقی را می‌توان به دو صورت استفاده از خطای ردگیری^۲ یا خطای پیش بینی^۳ (خطای بین پارامترهای تخمین زده شده و پارامترهای واقعی) طراحی نمود. باید توجه داشت که در تمامی روش‌های فوق فقط عدم قطعیت پارامتری لحاظ شده است. علاوه بر این، تنظیم پارامترهای قوانین تطبیق ممکن است کار ساده ای نباشد. نکته مهم دیگر در مورد روش‌های تطبیقی، مسئله تحریک پایا^۴ است. در این روش‌ها، چون در حال شناسایی پارامترهای سیستم هستیم تحریک پایا بودن سیگنال‌های تحریک ضروری است [۳].

کنترل غیر خطی مقاوم^۵ بازوهای رباتیک [۴-۷] نیز تلاشی دیگر برای بهبود عملکرد روش خطی سازی فیدبکی جهت غلبه بر عدم قطعیت است. در این روش طراحی قانون کنترل با توجه به مدل نامی^۶ سیستم (مدلی که ما برای سیستم تخمین زده‌ایم و ساختار آن همانند ساختار مدل واقعی است فقط مقادیر پارامترهای آنها تفاوت دارد) صورت می‌گیرد. پایداری سیستم با استفاده از روش مستقیم لیاپانوف اثبات می‌شود. برای جبران عدم قطعیت‌ها یک جمله به قانون کنترلی که خطی‌سازی فیدبکی را انجام می‌دهد، اضافه می‌شود. مقدار این جمله با توجه به باند عدم قطعیت سیستم و شرط منفی بودن مشتق زمانی تابع لیاپانوف بدست می‌آید. در مقایسه با روش‌های تطبیقی که به آنها اشاره شد، این تکنیک قابلیت‌های بیشتری دارد، چون می‌تواند علاوه بر عدم قطعیت پارامتری، عدم قطعیت ناشی از دینامیک‌های مدل نشده^۷ و اغتشاش خارجی^۸ را نیز جبران کند. مزیت دیگر این روش نسبت به روش‌های

¹ Adaptation law

² Tracking error

³ Prediction error

⁴ Persistency of excitation

⁵ Robust nonlinear control

⁶ Nominal model

⁷ Un-modeled dynamics

⁸ External disturbance

روش‌های تطبیقی آن است که نیازی به شناسایی پارامترهای سیستم ندارد. بنابراین، تحریک پایا نبودن سیگنال‌های تحریک مشکلی بوجود نمی‌آورد. در مقایسه با روش‌های فازی، این روش بسیار ساده تر است، چون نیازی به مدل‌سازی سیستم یا دانش کنترلی توسط قوانین فازی ندارد. در نتیجه، حجم محاسبات آن کمتر می‌باشد.

با ظهور منطق فازی^۱ به عنوان یک ابزار توانمند در کنترل سیستم‌های همراه با عدم قطعیت و پیچیده، تحول شگرفی در مهندسی کنترل بوجود آمد. چون به کمک قوانین فازی می‌توان سیستم‌هایی را توصیف کرد که مدل ریاضی دقیقی از آنها در اختیار نیست. روش فازی تطبیقی غیر مستقیم^۲ از این ایده استفاده می‌کند [۸،۹]. ویژگی دیگر منطق فازی، مدل‌سازی دانش و توانایی انسان به منظور کنترل سیستم‌های پیچیده می‌باشد [۱۰] که روش فازی تطبیقی مستقیم^۳ [۱۱] این امکان را فراهم می‌آورد. علاوه بر این، می‌توان روش‌های فازی تطبیقی مستقیم و غیر مستقیم را با هم ترکیب نمود و روشی بدست آورد که عملکرد بهتری داشته باشد [۱۲]. کنترل نظارتی^۴ [۱۳] نیز یکی دیگر از تکنیک‌های ارائه شده در حوزه کنترل فازی می‌باشد. به همین دلیل در سال‌های اخیر، محققان تمرکز بیشتری روی کنترل فازی داشته‌اند و تلاش‌های فراوانی برای کنترل مقاوم ربات با استفاده از کنترل فازی صورت گرفته است [۱۴-۲۶] اکثر روش‌های فازی ارائه شده، مبتنی بر کنترل گشتاور می‌باشند و قانون کنترل، گشتاور ورودی مفاصل را محاسبه می‌کند. اما پیشنهاد قانون کنترل منحصر به روش خطی سازی فیدبکی نمی‌باشد و از سایر روش‌های کنترل غیر خطی مانند طراحی پس گام [۲۷]، کنترل مود لغزشی [۲۸-۳۱] نیز بسیار استفاده شده است. به عنوان مثال، می‌توان به کنترلر فازی تطبیقی مود لغزشی^۵ [۳۱] اشاره کرد. این روش، برای طراحی قانون کنترل و تضمین پایداری سیستم از کنترل مود لغزشی بهره می‌برد و

¹ Fuzzy Logic

² Indirect adaptive fuzzy control

³ Direct adaptive fuzzy control

⁴ Supervisory control

⁵ Adaptive fuzzy sliding mode controller

به منظور جبران خطای ناشی از عدم قطعیت پارامتری از روشهای فازی تطبیقی استفاده می‌کند. اما قانون کنترل این روش پیچیده می‌باشد و خطای ردگیری نیز نسبتاً قابل ملاحظه است. در [۳۲] روش فازی تطبیقی مدل مرجع ارائه شده است که تلاش دارد توسط یک کنترلر فازی عوامل غیر خطی سیستم را با خطی سازی فیدبکی تطبیقی حذف کند و سپس قطب‌های سیستم را در مکان‌هایی قرار دهد که از نظر مشخصات عملکردی (فراجهش^۱، زمان خیز^۲، زمان نشست^۳ و ...) مطلوب می‌باشد. اما قوانین تطبیق این روش محاسبات زیادی در بر دارد.

شبکه‌های عصبی نیز قابلیت‌های بسیاری در کنترل ربات دارند [۳۳-۳۶]. در [۳۷] با استفاده از شبکه‌های عصبی به کنترل امپدانس^۴ ربات پرداخته شده است. کنترل امپدانس یکی از مؤثرترین روش‌های ارائه شده برای کنترل ربات‌هایی است که به محیط اطراف خود نیرو وارد می‌کنند. این کاربرد از ربات‌ها مستلزم کنترل توامان موقعیت و نیرو می‌باشد. کنترل امپدانس می‌تواند رفتار دینامیکی مجری نهایی را با تنظیم پارامترهای دینامیکی همچون اینرسی، سرعت، سختی و... به طور مطلوب کنترل کند. اما به دلیل عدم قطعیت‌هایی که در مورد ربات و محیط داریم، طراحی این پارامترها بر اساس کار موردنظر و شرایط محیط بسیار مشکل است. اما شبکه‌های عصبی توانایی بسیار خوبی در یادگیری رفتار دینامیکی سیستم‌های پیچیده از خود نشان داده‌اند و می‌توانیم تنظیم پارامترهای امپدانس ربات را با شبکه‌های عصبی انجام دهیم [۳۷].

اکثر روش‌های ارائه شده برای کنترل ربات، مبتنی بر کنترل گشتاور می‌باشند و کنترل کننده گشتاور ورودی به مفاصل را محاسبه می‌کند. باید توجه داشت که این گشتاور نمی‌تواند مستقیماً به سیستم اعمال شود، چون باید توسط محرک‌های سیستم فراهم شود و ابتدا باید محرک‌های سیستم

¹ Overshoot

² Rise time

³ Settling time

⁴ Impedance control

طوری تحریک شوند تا گشتاور مطلوب را تولید کنند. حال این سؤال مطرح می‌شود که به ورودی محرک چه سیگنالی اعمال کنیم. علاوه بر این، بسیاری از روش‌های کنترل گشتاور مانند روش کنترل مقاوم غیر خطی، مبتنی بر مدل نامی سیستم هستند. پیچیدگی مدل نامی و مدل واقعی یکسان است. زیرا ساختار آنها مشابه یکدیگر می‌باشد و فقط مقادیر پارامترهای آنها تفاوت دارد. بنابراین، حجم محاسبات کنترل کننده در این روش‌ها زیاد بوده و ممکن است در پیاده سازی عملی با مشکلاتی از قبیل حجم حافظه مورد نیاز و زمان نمونه برداری مواجه باشیم.

نکته مهم دیگر در مورد روش‌های کنترل گشتاور، فیدبک‌های مورد نیاز برای پیاده سازی قانون کنترل است. در اکثر این روش‌ها، علاوه بر فیدبک موقعیت، فیدبک‌های سرعت و گاهی شتاب نیز مورد نیاز است [۳]. این سیگنال‌ها معمولاً آغشته به نویز بوده و عملکرد سیستم کنترل را تضعیف می‌نمایند. بنابراین، ابداع روش جدید برای کنترل ربات که بتواند بر مشکلات مذکور فائق آید ضروری به نظر می‌رسد.

۱-۱-۲- کنترل ولتاژ

در سال‌های اخیر، استراتژی کنترل ولتاژ^۱ [۳۸] ربات‌ها مطرح شده است که نه تنها مشکلات روش‌های کنترل گشتاور را ندارد بلکه دقت آن نیز به مراتب بهتر است. در این استراتژی از موتورهای الکتریکی به عنوان محرک استفاده می‌شود و بازوی ماهر رباتیک به عنوان بار این موتورها محسوب می‌شود که توسط موتورها به حرکت در می‌آید. ورودی موتور سیگنال ولتاژ است و خروجی آن موقعیت زاویه‌ای موتور است. بنابراین در این استراتژی با کنترل موتور سر و کار داریم و گشتاور مورد نیاز برای

¹ Voltage control strategy

حرکت دادن مفاصل ربات به صورت گشتاور بار موتور در معادلات ظاهر می‌شود. به عبارت دیگر، این استراتژی مستقل از دینامیک پیچیده غیر خطی ربات است [۳۸]. زیرا از طریق کنترل موقعیت زاویه‌ای موتور به کنترل موقعیت مفاصل ربات می‌پردازیم. کنترل ولتاژ مشکلات کنترل گشتاور را ندارد. زیرا

- در این استراتژی دینامیک سیستم محرک (موتور) لحاظ شده است و قانون کنترل ولتاژ اعمالی به سیستم را محاسبه می‌کند.

- قانون کنترل در این استراتژی مستقل از مدل ربات است و بنابراین بار محاسباتی کنترل کننده بسیار کمتر است. زیرا در این روش با مدل موتور سر و کار داریم که بسیار ساده تر از مدل ربات است.

- برای پیاده سازی قانون کنترل این استراتژی فقط به فیدبک‌های جریان موتور و متغیرهای مفاصل ربات نیاز داریم که اندازه گیری آنها بسیار راحت تر از اندازه گیری سرعت و شتاب است.

استراتژی کنترل ولتاژ قابلیت‌های بسیاری دارد و می‌تواند به سادگی با سایر روش های کنترلی موجود از قبیل کنترل مقاوم، تطبیقی و فازی به کار گرفته شود. علاوه بر این، می‌توان از تمام تکنیک‌هایی که جهت تحلیل و طراحی سیستم‌ها در کنترل خطی ارائه شده است برای کنترل ربات که یک سیستم غیرخطی است، استفاده نمود. چون همان طور گفتیم در این استراتژی هدف، کنترل موتور است که یک سیستم خطی می‌باشد.

مقاوم نمودن این استراتژی در برابر عدم قطعیت‌ها یک میدان تحقیقاتی وسیع و جدید است. در سال‌های اخیر پیشرفت‌هایی در زمینه اندازه گیری شتاب با استفاده از کدگذار^۱ های نوری حاصل شده است. بر مبنای این تکنولوژی جدید، یک روش مقاوم مبتنی بر کنترل ولتاژ با استفاده از فیدبک شتاب

¹ Encoder

ارائه شده است [۳۹]. روش کنترل مقاوم غیر خطی [۴] نیز می‌تواند برای مقاوم نمودن استراتژی کنترل ولتاژ به کار گرفته شود. مقاله [۴۰] از این روش بهره می‌برد و روش‌هایی برای کاهش تأثیر عدم قطعیت در سیستم حلقه بسته پیشنهاد می‌دهد. در [۴۱] یک روش مقاوم فازی برای کنترل ربات‌های الکتریکی^۱ ارائه شده است. تحلیل پایداری^۲ کنترل فازی بازوهای رباتیک با توجه به پیچیدگی‌های مدل ربات بسیار مشکل است. اما در استراتژی کنترل ولتاژ می‌توان با پیشنهاد دادن یک تابع لیاپانوف ساده پایداری سیستم کنترل فازی ربات را اثبات نمود [۴۱].

کنترل ربات‌ها در فضای کار^۳ پیچیده تر از کنترل در فضای مفصلی^۴ می‌باشد، چون در فضای کار به ماتریس ژاکوبین^۵ نیز نیاز داریم که پارامترهای زیادی دارد. در نتیجه، نسبت به کنترل در فضای مفصلی عدم قطعیت‌های سیستم افزایش می‌یابد. در [۴۲] یک روش مقاوم تطبیقی مبتنی بر کنترل گشتاور برای کنترل ربات در فضای کار ارائه شده است. ترکیب روش‌های کنترلی همچون کنترل مود لغزشی، خطی سازی فیدبکی و طراحی پس گام یکی دیگر از تکنیک‌های ارائه شده جهت کنترل گشتاور ربات در فضای کار می‌باشد [۴۳]. از کنترل فازی نیز برای مقاوم نمودن کنترل ربات در فضای کار استفاده شده است [۴۴]. طراحی قوانین تطبیق جهت شناسایی پارامترهای سینماتیکی و دینامیکی، نیز یکی دیگر از راه حل‌های پیشنهاد شده به منظور غلبه بر عدم قطعیت‌ها برای کنترل ربات در فضای کار می‌باشد [۴۵]. این روش نیز مبتنی بر کنترل گشتاور می‌باشد و تعداد پارامترهایی که باید شناسایی شوند زیاد است. همچنین قانون کنترل آن پیچیده است. در حالی که اگر با استراتژی کنترل ولتاژ به کنترل ربات در فضای کار پردازیم روابط بسیار ساده تر می‌شوند. در مقاله [۴۶] با استفاده از این استراتژی، یک تکنیک ساده و جالب برای جبران عدم قطعیت جهت کنترل ربات در فضای کار ارائه شده است. در این تکنیک

¹ Electrically driven robot manipulators

² Stability analysis

³ Task-space

⁴ Joint-space

⁵ Jacobian

نیازی به شناسایی هیچ پارامتری نداریم و فقط با استفاده از فیدبک گرفتن از ولتاژ خروجی کنترل کننده پس از یک تأخیر زمانی کوچک، سیستم کنترل در مقابل انواع عدم قطعیت‌ها اعم از عدم قطعیت پارامتری، دینامیک مدل نشده و اغتشاش خارجی مقاوم می‌شود. این تکنیک اختصاص به فضای کار ندارد و می‌توان به سادگی از آن در فضای مفصلی استفاده کرد.

در نظر گرفتن انعطاف مفاصل کنترل ربات را بسیار پیچیده تر می‌کند، چون در مقایسه با ربات‌های صلب^۱، تعداد متغیرها دو برابر می‌شود [۱]. علاوه بر این، به دلیل انعطاف، موقعیت رابط نمی‌تواند دقیقاً موقعیت محرک را دنبال کند. بنابراین، قانون کنترل باید بتواند خطای ناشی از انعطاف مفاصل را نیز جبران کند. برای کنترل ربات‌های با مفاصل منعطف^۲، روش‌های بسیاری مبتنی بر کنترل گشتاور ارائه شده است [۴۷-۵۱]. به عنوان مثال، می‌توان به کنترل PD [۵۲]، خطی سازی فیدبکی [۵۳]، کنترل مقاوم [۵۴]، کنترل مود لغزشی [۵۵]، کنترل تطبیقی [۵۶]، و کنترل فازی [۵۷] اشاره کرد. کنترل ربات‌های منعطف با استراتژی کنترل گشتاور به روابط بسیار پیچیده‌ای منجر می‌شود. در حالی که اگر با استراتژی کنترل ولتاژ به این مسئله پردازیم، دشواری‌های آن کاهش چشمگیری می‌یابد [۵۸]. کنترل ربات‌های منعطف با استراتژی کنترل ولتاژ یک زمینه تحقیقاتی بکر است و کارهای اندکی در این حوزه انجام شده است.

۱-۱-۳ الگوریتم بهینه سازی پرندگان^۳

شناسایی سیستم^۴ یکی از مهم‌ترین گام‌های کنترل است. چون معمولاً کنترل کننده بر مبنای مدل طراحی می‌شود. روش حداقل مربعات^۱ یک روش بنیادی در شناسایی سیستم‌ها می‌باشد که با موفقیت

¹ Rigid

² Flexible-joint robot manipulators

³ Particle swarm optimization (PSO)

⁴ System identification

برای شناسایی سیستم های استاتیکی و دینامیکی به کار گرفته شده است. اما این روش برای سیستم‌هایی مناسب است که ساختار مدل آن نسبت به پارامترها خطی باشد [۵۹]. علاوه بر این، یکی از شروط روش حداقل مربعات، مشتق پذیری تابع عملکرد^۲ است [۶۰]. اما تکنیک‌های بهینه سازی تکاملی^۳ مانند الگوریتم بهینه‌سازی پرندگان و الگوریتم ژنتیک^۴، جایگزین‌های مناسبی برای روش حداقل مربعات می‌باشند، زیرا مبتنی بر فرض‌هایی همچون مشتق پذیری، پیوستگی و در اختیار داشتن معادلات سیستم نیستند.

الگوریتم بهینه‌سازی پرندگان در جستجوی نواحی خیلی کوچک عملکرد ضعیفی دارد، زیرا مکانیزمی برای کنترل سرعت ندارد. روش‌های متعددی برای بهبود عملکرد الگوریتم بهینه‌سازی پرندگان با استفاده از ضریب اینرسی ارائه شده است. ضریب اینرسی در چگونگی عملکرد این الگوریتم تأثیر مهمی دارد، زیرا بین جستجوی محلی^۵ و سراسری^۶ تعادل ایجاد می‌کند. اگر ضریب اینرسی بزرگ باشد، جستجوی سراسری راحت تر انجام می‌شود. اما ممکن است به کاهش سرعت همگرایی منجر شود. برعکس، اگر ضریب اینرسی کوچک باشد، سرعت همگرایی بهتر می‌شود اما ممکن است منجر به مینیمم محلی شود. بنابراین، روش‌هایی ارائه شد که در آنها ضریب اینرسی به طور خطی یا غیر خطی کاهش می‌یابد.

چون فرایند جستجو در الگوریتم بهینه سازی پرندگان یک فرایند غیر خطی و پیچیده است، توابع خطی یا غیر خطی کاهشی که برای ضریب اینرسی در نظر گرفته می شود نمی توانند بدون فیدبک

¹ Least squares

² Performance function

³ Evolutionary optimization techniques

⁴ Genetic Algorithm

⁵ Local search

⁶ Global search

گرفتن از تابع برازندگی^۱ منجر به جواب بهینه شوند. در حقیقت اگر تابع برازندگی زیاد باشد به این معنی است که ذرات از نقطه بهینه دور هستند. بنابراین به سرعت های بیشتری نیاز داریم تا فضاهای بیشتری را تحت پوشش قرار دهیم. بنابراین باید ضرایب موجود در رابطه تنظیم سرعت بزرگ باشند. اگر تابع برازندگی کوچک باشد باید این ضرایب کوچک انتخاب شوند. بنابراین می توان این مقادیر را به صورت تابعی از برازندگی مقدار بهینه، تعریف کرد [۶۱]. در [۶۲] برای بهبود عملکرد الگوریتم بهینه سازی پرندگان روشی ارائه شده است که در آن به جای استفاده از یک پرنده به عنوان بهترین پرنده، از چندین پرنده به عنوان بهترین پرندگان استفاده می شود. در الگوریتم بهینه سازی پرندگان ضرایب تنظیم سرعت برای تمام پرندگان یکسان است. اما به نظر می رسد اگر سرعت هر پرنده با توجه به شرایط خاص آن پرنده تنظیم شود، الگوریتم عملکرد بهتری داشته باشد. در [۶۰] روشی مبتنی بر این ایده ارائه شده است.

با توجه به سادگی روابط تازه سازی^۲ پارامترها در الگوریتم بهینه سازی پرندگان و توانایی بسیار خوب آن در حوزه های مختلف بهینه سازی، در سالهای اخیر استفاده از الگوریتم بهینه سازی پرندگان در شناسایی و کنترل سیستم ها افزایش یافته است. به عنوان مثال می توان به کنترل تطبیقی امیدانس سیستم تعلیق [۶۳]، شناسایی سیستم های غیر خطی [۶۴]، طراحی شبکه های عصبی [۶۵] و شناسایی پارامترهای سیستم تعلیق [۶۶] اشاره کرد.

¹ Fitness function

² Update

۱-۲- اهداف مورد نظر

با توجه به مزایای استراتژی کنترل ولتاژ نسبت به استراتژی کنترل گشتاور، در این پایان نامه استراتژی کنترل ولتاژ را برای کنترل موقعیت بازوی ماهر کروی^۱ در فضای مفصلی انتخاب می‌کنیم. بازوی ماهر کروی کاربردهای فراوانی در صنعت دارد. اما با توجه به پیکربندی خاص این ربات، یکی از جالب‌ترین کاربردهای آن را می‌توان در رصدخانه‌ها یافت. برای غلبه بر عدم قطعیت‌ها از بین روش‌های اشاره شده در این فصل، روش کنترل مقاوم غیرخطی [۴] انتخاب شده است. دلیل آن نیز با توجه به مزایایی که برای این روش نسبت به روش‌های فازی و تطبیقی در این فصل ذکر شد، آشکار است. قانون کنترل پارامترهایی دارد که مقدار برخی از آنها به نظر طراح بستگی دارد. مقدار برخی دیگر از پارامترها به مدل نامی سیستم بستگی دارد. بررسی چگونگی تأثیر این پارامترها در خطای ردگیری کار دشواری است. با توجه به قابلیت‌های چشمگیر الگوریتم بهینه سازی پرندگان در حل مسائل مختلف بهینه سازی، در این پایان نامه تعیین مقدار این پارامترها را به عهده این الگوریتم می‌گذاریم.

¹ Spherical robot manipulator

۱-۳- طرح کلی پایان نامه

فصل ۲ به معرفی الگوریتم بهینه سازی پرندگان، چگونگی شکل گیری آن و مقایسه آن با الگوریتم ژنتیک می‌پردازد. سپس با استفاده از الگوریتم بهینه سازی پرندگان پارامترهای چند سیستم ساده شناسایی می‌شود.

فصل ۳ مروری بر مدلسازی سینماتیکی و دینامیکی بازوهای ماهر مکانیکی خواهد داشت و مدل ریاضی بازوی ماهر کروی را ارائه می‌کند.

فصل ۴ روش کنترل مقاوم غیر خطی را توضیح می‌دهد و با استفاده از آن به جبران عدم قطعیت‌ها در کنترل گشتاور ربات می‌پردازد.

فصل ۵ استراتژی کنترل ولتاژ را معرفی می‌کند، سپس از این استراتژی برای کنترل مقاوم ربات کروی استفاده می‌کند.

فصل ۶ به کنترل بهینه غیرخطی ربات با استفاده از الگوریتم بهینه سازی پرندگان می‌پردازد. سپس نتایج کنترل بهینه ولتاژ و کنترل بهینه گشتاور را با هم مقایسه می‌کند.

فصل دوم

الگوریتم بهینه‌سازی پرندگان

- ❖ مقدمه
- ❖ چگونگی شکل‌گیری الگوریتم بهینه‌سازی پرندگان
- ❖ مقایسه بین الگوریتم بهینه‌سازی پرندگان و الگوریتم ژنتیک
- ❖ بهینه‌سازی توابع
- ❖ شناسایی سیستم‌ها
- ❖ کاربرد الگوریتم بهینه‌سازی پرندگان در شناسایی و کنترل سیستمها

۲-۱- مقدمه

مسائل بهینه‌سازی در حوزه‌های مختلف علوم کاربردهای فراوانی دارند و طبقه‌بندی‌های متفاوتی از آنها ارائه شده است. در یک تقسیم‌بندی کلی می‌توان مسائل بهینه‌سازی را به دو گروه خطی و غیر خطی تقسیم نمود. معمولاً هر مسئله بهینه‌سازی شامل اجزای زیر می‌باشد:

- متغیرهای تصمیم‌گیری
- قیود (محدودیت‌ها)
- تابع هزینه

هدف مسائل بهینه‌سازی تعیین متغیرهای تصمیم‌گیری به منظور حداقل یا حداکثر نمودن تابع هزینه با توجه به محدودیت‌ها می‌باشد. در مسائل بهینه‌سازی خطی قیود و تابع هزینه رابطه خطی با متغیرهای تصمیم‌گیری دارند. این نوع از مسائل بهینه‌سازی کاربردهای فراوانی در علوم مختلف مانند اقتصاد دارند. به عنوان مثال می‌توان به حداقل نمودن هزینه‌های تولید، حداکثر نمودن سود، برنامه‌ریزی ترکیب تولید، تعیین تعداد پرسنل و کاهش هزینه‌های حمل و نقل اشاره کرد. الگوریتم سیمپلکس یکی از بهترین روشهای حل مسائل بهینه‌سازی خطی است و کاربردهای فراوانی دارد.

روشهای متعددی برای حل مسائل بهینه‌سازی ارائه شده است. به طور کلی می‌توان این روشها را به دو گروه عمده روشهای مبتنی بر مشتق^۱ و روشهای بدون مشتق^۲ طبقه‌بندی کرد. الگوریتمهای فرود^۳ مانند گرادیان نزولی و شدیدترین فرود^۴ و روش نیوتون^۵ مشهورترین روشهای مبتنی بر مشتق می‌باشند

¹ Gradient-based

² Gradient-free

³ Descent methods

⁴ Steepest descent

⁵ Newton's method

که کاربردهای فراوانی دارند. اما این روشها محدودیت هایی دارند. مهمترین محدودیت این روشها مینیمم ها یا ماکزیمم های محلی است، زیرا این روشها مکانیزم خوبی برای فرار از جوابهای بهینه محلی ندارند. علاوه بر این برخی از مسائل بهینه‌سازی بسیار پیچیده هستند و قیودی دارند که موجب می شود پیدا کردن جواب بهینه دشوار شود. به عنوان مثال می‌توان به مسئله بسیار مهم پخش بار اقتصادی^۱ در سیستم‌های قدرت اشاره کرد. هدف این مسئله بهینه‌سازی غیر خطی حداقل نمودن هزینه تولید برق در نیروگاه ها می باشد. در این مسئله قیود مهمی باید رعایت شود. به عنوان مثال می توان به تعادل تولید و مصرف در هر لحظه، حداقل و حداکثر توان تولیدی هر واحد، شیب بارگیری^۲ ژنراتورها، حداقل زمان روشن بودن^۳ و حداقل زمان خاموش بودن^۴ ژنراتورها و حداکثر توان عبوری از خطوط انتقال اشاره کرد. با توجه این قیود و تعداد زیاد واحدها جواب های بهینه محلی فراوانی برای این مسئله وجود دارد و روشهای مبتنی بر مشتق عملکرد خوبی در مسئله پخش بار اقتصادی ندارند. به همین دلیل برای حل این مسئله معمولاً از روشهای بدون مشتق استفاده می شود [۶۷-۶۹].

روش‌های بدون مشتق مانند الگوریتم ژنتیک و الگوریتم بهینه سازی پرندگان به سادگی می‌توانند برای حل بسیاری از مسائل بهینه سازی به کار گرفته شوند. تصادفی بودن مهمترین ویژگی این روشها است، زیرا معمولاً در این روشها فرایند جستجو با یک مجموعه از جواب های تصادفی آغاز می شود. ویژگی تصادفی بودن، این روشها را قادر به جستجوی جواب بهینه در سراسر فضای مسئله می‌سازد [۷۰]. به همین دلیل در مقایسه با روشهای مبتنی بر مشتق، احتمال اینکه این روشها منجر به جواب بهینه محلی شوند بسیار کمتر است.

¹ Economic dispatch

² Ramp rate

³ Minimum up-time

⁴ Minimum down-time

الگوریتم بهینه سازی پرندگان، یک روش بهینه سازی توابع غیر خطی می باشد که از شبیه سازی یک مدل اجتماعی ساده الهام گرفته است. این الگوریتم ارتباط تنگاتنگی با الگوریتم ژنتیک دارد. الگوریتم بهینه سازی پرندگان مانند الگوریتم ژنتیک با یک جمعیت اولیه به عنوان جواب های اولیه شروع می کند. اما برخلاف الگوریتم ژنتیک، عملگرهای تکاملی مانند جهش^۱ و تقاطع^۲ ندارد. در الگوریتم بهینه سازی پرندگان جواب های احتمالی که همان پرندگان هستند، در فضای مسئله به تعقیب جواب بهینه می پردازند. نسبت به الگوریتم ژنتیک، الگوریتم بهینه سازی پرندگان به سادگی پیاده سازی می گردد و پارامترهای کمتری دارد. الگوریتم بهینه سازی پرندگان در زمینه های مختلف مانند بهینه سازی توابع، آموزش شبکه های عصبی، کنترل فازی با موفقیت به کار گرفته شده است. عملگرهای ریاضی آن خیلی ساده و به لحاظ محاسباتی بسیار کم هزینه می باشد.

در این فصل چگونگی شکل گیری الگوریتم بهینه سازی پرندگان را توضیح می دهیم. سپس به مقایسه الگوریتم بهینه سازی پرندگان و الگوریتم ژنتیک خواهیم پرداخت. در ادامه بهینه سازی توابع با استفاده از الگوریتم بهینه سازی پرندگان را توضیح می دهیم. سپس مسئله شناسایی سیستم ها را معرفی می کنیم و در نهایت به نمونه هایی از کاربردهای الگوریتم بهینه سازی پرندگان در شناسایی و کنترل سیستم ها اشاره خواهیم کرد.

¹ Mutation
² Crossover

۲-۲- چگونگی شکل‌گیری الگوریتم بهینه‌سازی پرندگان

۲-۲-۱- شبیه‌سازی رفتار اجتماعی برخی موجودات زنده

برخی از دانشمندان حرکت موجودات زنده در دسته‌های پرندگان و ماهی‌ها را شبیه‌سازی کرده‌اند. به عنوان مثال گرنندر^۱، هپر^۲ و رینولدز^۳ حرکت پرندگان را شبیه‌سازی کرده‌اند. رینولدز مجذوب زیبایی‌های موجود در حرکت دسته‌های پرندگان در آسمان شده بود [۷۱]. هپر به عنوان یک جانورشناس علاقمند به کشف قانون‌هایی بود که دسته‌های بزرگ پرندگان را قادر می‌ساخت با وجود تغییر جهت‌های ناگهانی و پراکندگی اولیه به طور هماهنگ پرواز کنند. هر دو دانشمند معتقد بودند که یک سری پردازش‌های محلی در حرکت دسته‌های پرندگان باید وجود داشته باشد. مدل‌هایی که آنها ارائه کرده بودند بر این حقیقت استوار بود که هماهنگی رفتار اجتماعی پرندگان هنگام پرواز تابعی از تلاش آنها برای حفظ فاصله بهینه میان خود و پرندگان مجاور است.

بعید نیست که قوانین مشابهی بر رفتار اجتماعی حیواناتی که به صورت دسته جمعی زندگی می‌کنند، حکمفرما باشد. ویلسون^۴ به عنوان یک جامعه‌شناس درباره ماهی‌ها چنین می‌نویسد: "اعضای یک دسته از ماهی‌ها هنگام جستجو برای غذا می‌توانند از نتایج جدیدی که ماهی‌های دیگر بدست آورده‌اند و نیز تجربیات گذشته سایر اعضای گروه استفاده کنند". این جمله بیان‌گر این حقیقت است که به اشتراک گذاشتن اطلاعات میان هم‌نوعان یک مزیت تکاملی دارد. این فرضیه در توسعه الگوریتم بهینه‌سازی پرندگان بسیار مهم و اساسی بود.

¹ Grenander

² Hepper

³ Reynolds

⁴ Wilson

شبیه سازی رفتار اجتماعی انسان نیز در شکل گیری این الگوریتم نقش تعیین کننده داشته است [۷۱]. رفتار اجتماعی انسان مطمئناً با رفتار اجتماعی پرندگان و ماهی ها تفاوت دارد. مهم ترین وجه تمایز، انتزاعی بودن رفتار اجتماعی انسان می باشد. پرندگان و ماهی ها حرکت فیزیکی خود را طوری تنظیم می کنند که از دشمن اجتناب کنند ، غذا یا جفت پیدا کنند یا پارامتر های محیطی مانند دما را بهینه کنند. انسان علاوه بر اینکه می تواند حرکت فیزیکی خود را در گروه تنظیم کند قادر به هماهنگ کردن افکار و عقاید نیز می باشد.

تشریح روند شکل گیری الگوریتم بهینه سازی پرندگان بهترین روش برای توضیح دادن آن است. همان طور که اشاره شد ، این الگوریتم با شبیه سازی یک محیط اجتماعی ساده آغاز شد. هدف اولیه شبیه سازی گرافیکی حرکت زیبا اما غیر قابل پیش بینی دسته پرندگان بود.

۲-۲-۲ عوامل "تطابق سرعت با نزدیک ترین همسایه" و "گیجی"

شبیه سازی رضایت بخش اولیه مبتنی بر دو فرضیه تطابق سرعت با نزدیک ترین همسایه و "گیجی" بود. برای یک دسته از پرندگان موقعیت ها و سرعت های تصادفی در راستای محورهای X , Y در نظر گرفته شد. در هر تکرار، یک حلقه در برنامه معین می کرد که همسایه هر پرنده کدام است. سپس سرعت همسایه را به آن پرنده نسبت می داد. این قانون ساده یک همانگی خاصی در حرکت به وجود آورده بود.

متأسفانه همه پرندگان به یک جهت خاص متمایل می شدند. بنابراین یک متغیر تصادفی به نام "گیجی" معرفی شد. بنابراین، در هر تکرار، سرعت پرندگان بر اساس برخی تغییرات تصادفی تنظیم می شدند.

¹ Crazyness

شبه سازی هپر یک ویژگی داشت که موجب می شد ، برخی تغییرات به صورت دینامیکی اعمال شود. در برنامه او پرندگان حول یک لانه جمع می شوند. این کار نیاز به "گنجی" را مرتفع ساخت. ایده در نظر گرفتن لانه بسیار جالب بود و منجر به یک سؤال مهیج تری شد. در برنامه هپر مکان لانه مشخص بود. ولی در زندگی واقعی ، پرندگان روی درخت یا سیم تلفن فرود می آیند که نیاز فوری آنها را برآورده کند. مهمتر از این، پرندگان جایی فرود می آیند که غذا وجود داشته باشد. آنها چگونه غذا را پیدا می کنند؟

همه می دانند که اگر مقداری غذای پرنده بیرون بگذاریم ، بعد از چند ساعت تعداد زیادی پرنده حول آن جمع می شوند. هرچند که هیچ دانش قبلی در مورد مکان یا شکل آن نداشته اند. احتمالاً یک ویژگی از چگونگی پرواز دسته پرندگان آنها را قادر می سازد تا از اطلاعات یکدیگر استفاده کنند.

فرض کنید یک لانه در مختصات (۱۰۰،۱۰۰) قرار داشته باشد و پرندگان می خواهند به سمت آن نقطه بهینه) پرواز کنند. بنابراین موقعیت فعلی هر پرنده توسط معادله زیر ارزیابی می شود.

$$Eval = \sqrt{(x - 100)^2 + (y - 100)^2} \quad (1-2)$$

هر پرنده بهترین مقدار $Eval$ و مختصات X, Y آن را به خاطر می سپارد. آن بهترین مقدار p_{best} نامیده شد و مختصات آن p_{bestx}, p_{besty} نامیده شد. در حقیقت p_{bestx}, p_{besty} بردارهایی هستند که بعد آن برابر با تعداد پرنده هاست. سرعت هر پرنده به کمک یک الگوریتم ساده تنظیم می شد. اگر پرنده، سمت راست p_{bestx} خودش بود آن گاه سرعت در راستای X آن یک مقدار منفی تصادفی تعیین می شد که با یک پارامتر سیستم وزن دهی می شد.

$$v_x = v_x - rand * p_increment \quad (2-2)$$

اگر پرنده سمت چپ p_{bestx} بود $rand() * p_increment$ به v_x اضافه می‌شد. v_y نیز به طور مشابه تغییر می‌کرد. علاوه بر این، هر پرنده بهترین موقعیت در کل دسته به نام g_{best} را به خاطر می‌سپرد. این اطلاعات برای تمام اعضای گروه قابل دسترسی بود. سپس v_y, v_x هر عضو به صورت زیر تنظیم شد که در آن $g_increment$ یک پارامتر سیستم است.

$$\text{If } p_x > g_{bestx} \text{ then } v_x = v_x - rand * g_increment \quad (۳-۲)$$

$$\text{If } p_x < g_{bestx} \text{ then } v_x = v_x + rand * g_increment \quad (۴-۲)$$

$$\text{If } p_y > g_{besty} \text{ then } v_y = v_y - rand * g_increment \quad (۵-۲)$$

$$\text{If } p_y < g_{besty} \text{ then } v_y = v_y + rand * g_increment \quad (۶-۲)$$

در جواب ، نقطه (۱۰۰ ، ۱۰۰) توسط یک دایره مشخص شده بود و هر پرنده با یک نقطه رنگی نمایش داده می‌شد. بنابراین یک بیننده می‌توانست چگونگی حرکت پرندگان را مشاهده کند . نتایج تعجب آور بود. اگر $p_increment$ و $g_increment$ نسبتاً زیاد انتخاب می‌شد این طور به نظر می‌رسید که دسته پرندگان ناگهان به طرف مقصد کشیده می‌شوند. طی چند تکرار ، کل دسته پرندگان داخل دایره کوچکی حول هدف جمع می‌شدند. اگر برای $g_increment$ و $p_increment$ مقدار کوچکی انتخاب می‌شد، دسته پرندگان مانند گرداب به طرف هدف می‌چرخیدند و به آن نزدیک می‌شدند و نهایتاً روی هدف فرود می‌آمدند.

۲-۲-۳- جستجوی چند بعدی

بسیاری از مسایل جالب بهینه سازی دو متغیره نیستند. بنابراین باید الگوریتم را برای مدلسازی رفتار اجتماعی چند بعدی آماده کنیم. به سادگی می توان موقعیت و سرعت هر پرنده را از آرایه های یک بعدی به ماتریس های $D \times N$ بعدی تغییر داد که D تعداد ابعاد و N تعداد پرندگان است.

آزمایشات و شبیه سازی های چند بعدی نیز با استفاده از مسایل چند متغیره غیر خطی انجام شد. به عنوان مثال به تنظیم وزن های یک شبکه عصبی می توان اشاره کرد. یکی از آزمایشات ، آموزش یک شبکه عصبی ۳ لایه برای حل مسئله XOR بود. شبکه مورد استفاده یک شبکه ۱-۳-۲ بود که ۱۳ پارامتر دارد. الگوریتم برای این مسئله به خوبی اجرا شد. مجموع مربعات خطا کمتر از ۰,۵ بود و میانگین تکرارهای مورد نیاز ۳۰ تکرار برای ۲۰ پرنده بود. الگوریتم برای شبکه های پیچیده تر نیز به خوبی اجرا شد.

۲-۲-۴- ایجاد شتاب توسط فاصله

اگر چه الگوریتم خوب کار کرد، اما برخی معایبی وجود داشت. چون تنظیم سرعت بر اساس نامساوی های

$$(if p_x > g_{bestx} \rightarrow decrease\ velocity) \quad (7-2)$$

$$(if p_x < g_{bestx} \rightarrow increase\ velocity) \quad (8-2)$$

بود. برخی آزمایشات این حقیقت را آشکار کرد که اصلاحات بیشتر می تواند عملکرد الگوریتم را بهبود بخشد. بجای بررسی کردن علامت نامساوی ها ، سرعت ها بر اساس اختلافشان با بهترین موقعیت تنظیم شدند.

$$v_x = v_x + rand * p_increment * (g_{bestx} - p_x) \quad (9-2)$$

توجه داشته باشید که v_x و p_x ماتریس‌های دو بعدی $D \times N$ هستند.

۲-۲-۵- روابط ساده شده فعلی

کدامیک از ضرایب $p_increment$ یا $g_increment$ باید بزرگتر باشد؟ از آنجا که جوابی برای این سؤال پیدا نشد، این جملات از الگوریتم حذف شدند. ضریب عدد تصادفی در دو ضرب شد تا متوسط آن یک شود. این نسخه عملکرد بهتری از معادلات داشت. تحقیقات بیشتری باید انجام شود تا مقدار بهینه‌ای برای ضرایب c_1 و c_2 در (۲-۱۱)، بدست آید. معمولاً به این ضرایب مقدار دو اختصاص می‌دهیم. شاید بهتر باشد مقدار آنها با توجه به شرایط مسئله تعیین شود. روابط فعلی الگوریتم بهینه سازی پرندگان به صورت زیر است [۷۱]:

$$v^{k+1} = wv^k + c_1 * rand * (p_{best} - p) + c_2 * rand * (g_{best} - p) \quad (10-2)$$

$$p^{k+1} = p^k + v^{k+1} \quad (11-2)$$

که در آن v^k در حالت کلی ماتریس سرعت در تکرار k ام الگوریتم می‌باشد و ابعاد آن $D \times N$ است. w ضریب اینرسی نام دارد و نقش مهمی در عملکرد الگوریتم دارد. معمولاً این ضریب در طی تکرارهای الگوریتم به صورت خطی از ۰,۹ تا ۰,۴ کاهش می‌یابد. p^k ماتریس موقعیت پرندگان در تکرار k ام است و بعد آن $D \times N$ می‌باشد.

شبهه کد الگوریتم فعلی را نیز می توان به صورت زیر نوشت:

Do

For each particle

Calculate fitness value

If the fitness value is better than the best fitness value (pbest) in history

set current value as the new pbest

End

Choose the particle with the best fitness value of all the particles as the gBest

For each particle

Update particle position

End

While maximum iterations or minimum error criteria is not attained

۲-۳- مقایسه بین الگوریتم ژنتیک و الگوریتم بهینه سازی پرندگان

اکثر الگوریتم های تکاملی مراحل زیر را شامل می شوند :

۱. جمعیت اولیه تصادفی
۲. محاسبه مقدار بهینگی برای هر ذره
۳. تولید جمعیت های بعدی با توجه به تابع برازندگی
۴. اگر شرط توقف برقرار شده ، توقف کنیم در غیر این صورت به گام دو می رویم.

با توجه به مراحل فوق مشخص می‌شود که دو الگوریتم بهینه‌سازی پرندگان و ژنتیک شباهت‌های بسیاری دارند. هر دو الگوریتم با یک جمعیت اولیه تصادفی شروع می‌کنند و یک کمیت به عنوان برازندگی دارند تا به کمک آن، جمعیت را تغییر دهند. هر دو الگوریتم با تکنیک‌های تصادفی به جستجوی جواب بهینه می‌پردازند. اما الگوریتم بهینه‌سازی پرندگان عملگرهای ژنتیکی مانند جهش و تقاطع ندارد، بلکه هر ذره حافظه دارد و با توجه به مسیر گذشته خود و دیگران حرکت آینده خود را انتخاب می‌کند.

در مقایسه با الگوریتم ژنتیک، نحوه به اشتراک گذاشتن اطلاعات در الگوریتم بهینه‌سازی پرندگان کاملاً متفاوت است. در الگوریتم ژنتیک کوروموزم‌ها، اطلاعات را با یکدیگر به اشتراک می‌گذارند. اما در الگوریتم بهینه‌سازی پرندگان، پرندگانی که برازندگی بهتری دارند به دیگران اطلاعات می‌دهد. به عبارت دیگر تبادل اطلاعات یک طرفه است. بنابراین در الگوریتم بهینه‌سازی پرندگان همه ذرات تمایل دارند به سمت بهترین جواب همگرا شوند. علاوه بر این الگوریتم بهینه‌سازی پرندگان از اعداد حقیقی به عنوان ذرات استفاده می‌کند و برخلاف الگوریتم ژنتیک نیازی به کد کردن جواب‌ها به صورت ۱ و ۰ ندارد. بنابراین پیاده‌سازی آن بسیار راحت‌تر است.

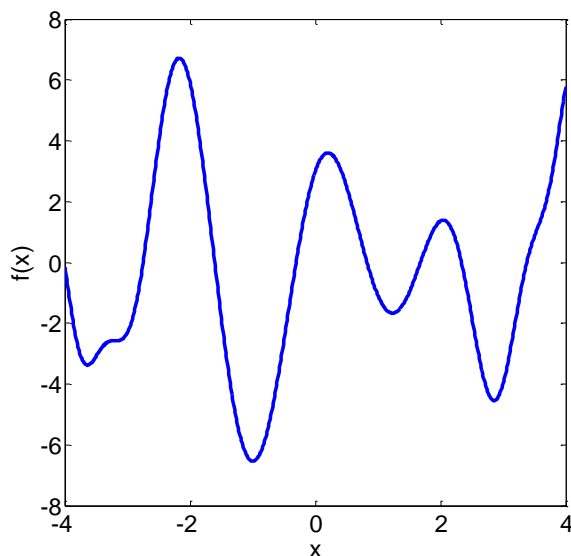
۲-۴- بهینه‌سازی توابع

بهینه‌سازی توابع با استفاده از الگوریتم بهینه‌سازی پرندگان بسیار ساده است. به عنوان مثال تابع اسکالر زیر را در نظر بگیرید:

$$f(x) = 3 \sin(2x) - \sin(x^2) + 3 \cos(3x) \quad (۲-۱۲)$$

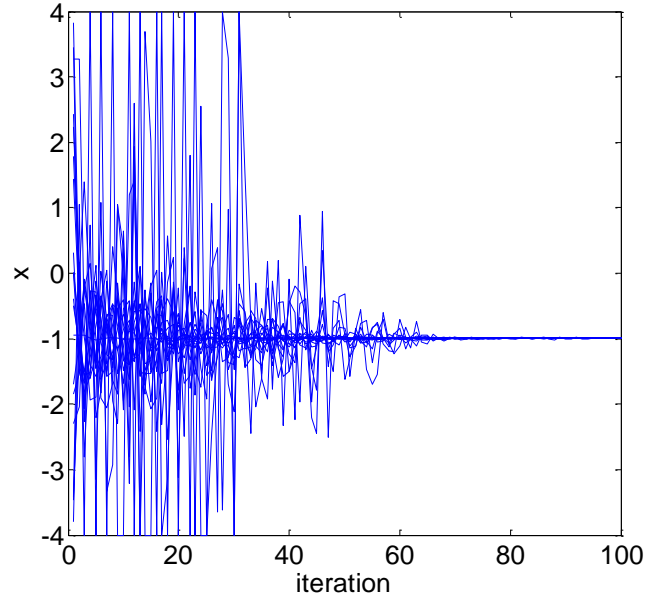
نمودار این تابع در شکل ۲-۱ نشان داده شده است. فرض کنید می‌خواهیم مقدار مینیمم مطلق تابع فوق را در بازه $[-4, 4]$ تعیین کنیم و همچنین مقدار x را در آن نقطه بدست آوریم. الگوریتم بهینه‌سازی

پرنندگان تعدادی جواب تصادفی روی محور x در نظر می گیرند. این جواب‌های تصادفی همان پرنندگان می‌باشند که در فضای مسئله (در اینجا محور x است) به سمت جواب بهینه حرکت می‌کنند.

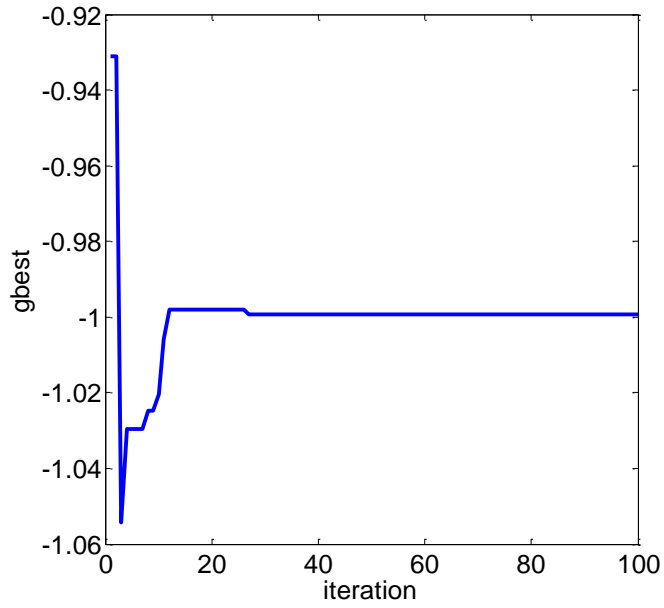


شکل ۱-۲ نمودار تابع $f(x)$ در (۱۲-۲)

الگوریتم بهینه سازی پرنندگان را با روابط (۱۰-۲) و (۱۱-۲) برای حل این مسئله به کار می‌گیریم. بردار p^1 همان جواب های اولیه تصادفی روی محور x است که در بازه $[-4, 4]$ قرار دارند. در طول ۱۰۰ تکرار الگوریتم اگر جوابها خارج از این بازه قرار گیرند، آنها را به ۴ یا -۴ محدود می‌کنیم. همان طور که در شکل ۱-۲ مشاهده می‌شود مینیمم مطلق تابع $f(x)$ در $x = -1$ اتفاق می‌افتد. بنابراین انتظار داریم که همه جوابها به سمت این نقطه همگرا شوند. شکل ۲-۲ چگونگی همگرایی ۲۰ پرنده را به سمت جواب بهینه نشان می‌دهد. تقریباً تمامی جوابها بعد از ۷۰ تکرار به جواب بهینه رسیده‌اند. البته جواب بهینه کل دسته (g_{best}) بعد از ۳۰ تکرار به نقطه بهینه رسیده است. شکل ۳-۲ چگونگی همگرایی g_{best} را به سمت نقطه بهینه نشان می‌دهد.



شکل ۲-۲ همگرایی ۲۰ پرنده به سمت جواب بهینه



شکل ۲-۳ همگرایی g_{best} را به سمت نقطه بهینه

۲-۵- شناسایی سیستم‌ها^۱

تعیین مدل ریاضی یک سیستم ناشناخته با استفاده از اطلاعات ورودی خروجی آن شناسایی سیستم نامیده می‌شود [۷۰]. شناسایی سیستم معمولاً شامل دو گام است:

- شناسایی ساختار^۲: در این گام به دانش و اطلاعات قبلی^۳ در مورد سیستم نیاز داریم تا با استفاده از آن به مجموعه‌ای از مدل‌ها به صورت $y = f(u, \theta)$ دست یابیم. سپس در این مجموعه مناسب‌ترین مدل را جستجو می‌کنیم. y در حالت کلی بردار خروجی سیستم، u بردار ورودی و θ بردار پارامترهای مجهول است. تابع f با توجه به تجربه طراح و قوانین حاکم بر سیستم تعیین می‌شود.

- شناسایی پارامتر^۴: در این گام ساختار مدل شناخته شده است و می‌خواهیم با استفاده از تکنیک‌های بهینه‌سازی، بردار $\hat{\theta}$ را طوری تعیین کنیم که مدل $\hat{y} = f(u, \hat{\theta})$ بتواند مدل واقعی سیستم را با تقریب خوبی توصیف کند. بلوک دیاگرام شناسایی پارامتر در شکل ۲-۴ نشان داده شده است.

مثال: در این مثال با استفاده از روش حداقل مربعات پارامترهای یک فنر را شناسایی می‌کنیم. از قانون هوک می‌دانیم وقتی به فنری که از یک ماده همگن ساخته شده باشد نیرو اعمال کنیم، تغییر طول فنر متناسب با نیروی اعمال شده است. بنابراین، می‌توانیم طول فنر را با رابطه زیر بیان کنیم:

$$l = l_0 + kf \quad (۱۳-۲)$$

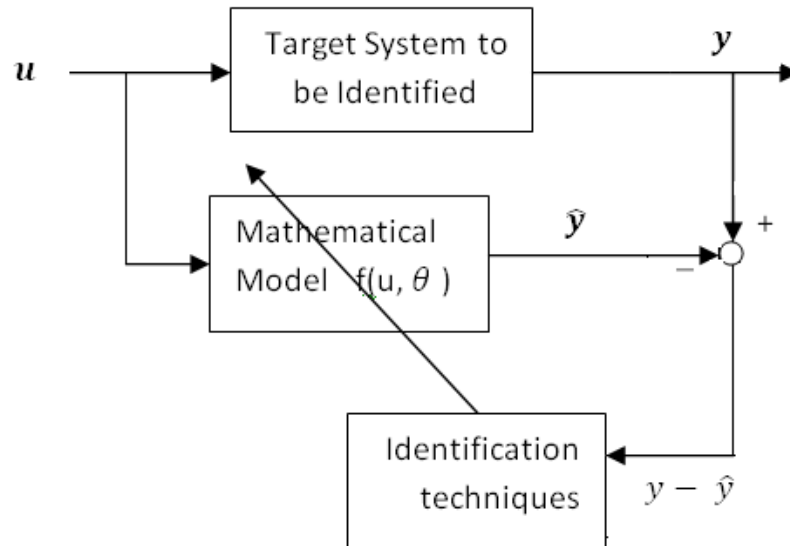
^۱ System identification

^۲ Structure identification

^۳ Priori knowledge

^۴ Parameter identification

که در آن l_0 و l طول فنر قبل و بعد از اعمال نیرو می باشد. k ضریب سختی فنر و f اندازه نیروی اعمال شده است. در این سیستم طول فنر، خروجی و نیروی اعمال شده ورودی آن است. بردار پارامترهای مجهول $\theta = [l_0 \ k]^T$ می باشد. جدول ۱-۲ نتایج چند آزمایش روی این سیستم را نشان می دهد.



شکل ۲-۴ بلوک دیاگرام شناسایی پارامتر

آزمایش	نیرو(نیوتون)	طول فنر(اینچ)
1	1.1	1.5
2	1.9	2.1
3	3.2	2.5
4	4.4	3.3
5	5.9	4.1
6	7.4	4.6
7	9.2	5

جدول ۱-۲ داده های ورودی خروجی برای شناسایی پارامترهای فنر

نتایج جدول فوق را می توانیم به صورت زیر در آوریم [۷۰]:

$$\begin{bmatrix} 1 & 1.1 \\ 1 & 1.9 \\ 1 & 3.2 \\ 1 & 4.4 \\ 1 & 5.9 \\ 1 & 7.4 \\ 1 & 9.2 \end{bmatrix} \begin{bmatrix} l_0 \\ k \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 2.1 \\ 2.5 \\ 3.3 \\ 4.1 \\ 4.6 \\ 5 \end{bmatrix} \quad (۱۴-۲)$$

به عبارت دیگر می توان نوشت:

$$A\theta + E = Y \quad (۱۵-۲)$$

که E خطای اندازه گیری است. بنابراین خواهیم داشت $[۷۰]$:

$$\begin{bmatrix} \hat{l}_0 \\ \hat{k} \end{bmatrix} = (A^T A)^{-1} A^T Y = \begin{bmatrix} 1.2 \\ 0.44 \end{bmatrix} \quad (۱۶-۲)$$

۲-۶- کاربرد الگوریتم بهینه‌سازی پرندگان در شناسایی و کنترل سیستم‌ها

شناسایی سیستم‌ها با استفاده از الگوریتم بهینه‌سازی پرندگان بسیار ساده است. پارامترهای مجهول را به عنوان ذرات یا همان پرندگان در نظر می‌گیریم. سپس همان طور که در شکل ۲-۴ نشان داده شده است، به سیستمی که می‌خواهیم آن را شناسایی کنیم و سیستمی که پارامترهای آن توسط الگوریتم بهینه‌سازی پرندگان تنظیم می‌شود، ورودی یکسان اعمال می‌کنیم. این ورودی باید بتواند تمامی فرکانس‌های موجود در سیستم را تحریک کند. خطا را نیز به صورت اختلاف دو خروجی تعریف می‌کنیم. بنابراین، تابع هزینه‌ای که باید مینیمم شود مجموع مربعات خطا یا انتگرال مجذور خطا می‌باشد [۷۲]. به مثال‌های زیر توجه کنید.

فرض کنید می‌خواهیم پارامترهای سیستم زیر را شناسایی کنیم:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k}{Ts+1} e^{-Ls} \quad (17-2)$$

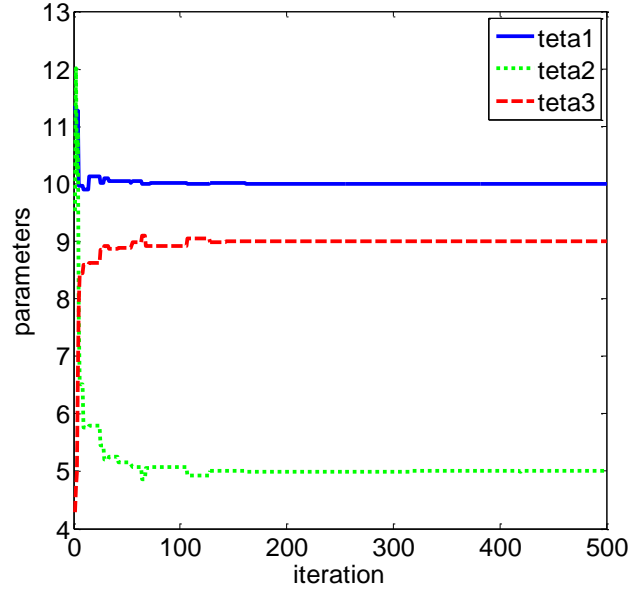
که مقادیر واقعی پارامترهای آن عبارتند از:

$$k = 10 \quad T = 5 \quad L = 9 \quad (18-2)$$

۲۰ پرنده به صورت $\theta = [\theta_1, \theta_2, \theta_3] = [k, T, L]$ در نظر می‌گیریم. تابع هزینه

$$J = \sum_{q=1}^{350} (y(q) - \hat{y}(q))^2 \quad (19-2)$$

را مینیمم می‌کنیم که در آن y خروجی سیستم واقعی و \hat{y} خروجی سیستمی است که پارامترهای آن توسط الگوریتم بهینه‌سازی پرندگان تنظیم می‌شود و q تعداد نمونه‌ها می‌باشد. در هر تکرار برنامه، مقدار تابع هزینه را برای تمام پرندگان بدست می‌آوریم. سپس p_{best} و g_{best} را بدست می‌آوریم و در روابط (۱۰-۲) و (۱۱-۲) جایگذاری می‌کنیم تا موقعیت‌ها و سرعت‌ها را برای تکرار بعدی برنامه بدست آوریم. ضرایب c_1, c_2 را مساوی ۲ قرار می‌دهیم و ضریب اینرسی را بین ۰,۹ و ۰,۴ به صورت کاهشی بین ۵۰۰ تکرار در نظر می‌گیریم. شکل ۲-۵ چگونگی همگرا شدن پارامترها را به سمت مقادیر واقعی نشان می‌دهد. همان‌طور که مشاهده می‌شود تمامی پارامترها به سمت مقادیر واقعی همگرا شده‌اند.



شکل ۲-۵: همگرایی پارامترهای سیستم خطی مرتبه اول

اکنون به شناسایی پارامترهای یک سیستم غیر خطی می پردازیم:

$$x_1(k+1) = a_1 x_1(k) x_2(k), \quad x_1(0) = 1 \quad (20-2)$$

$$x_2(k+1) = a_2 x_1^2(k) + u(k), \quad x_2(0) = 1 \quad (21-2)$$

$$y(k) = a_3 x_2(k) + a_4 x_1^2(k) \quad (22-2)$$

مقادیر واقعی پارامترها عبارتند از:

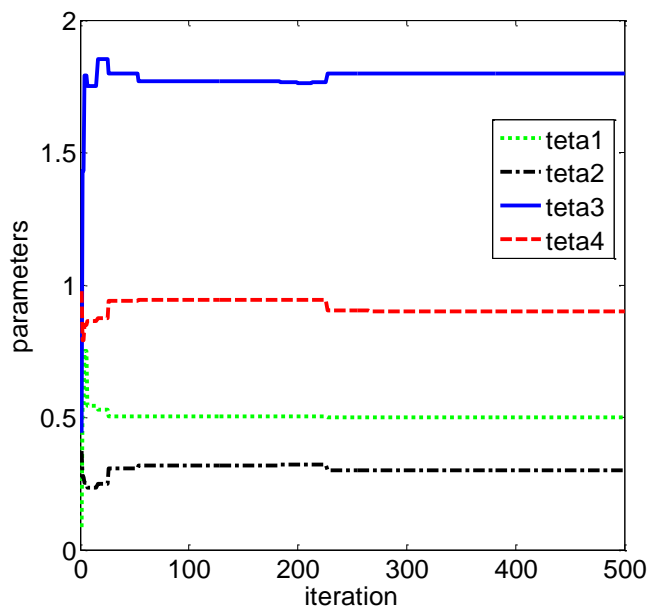
$$a_1 = .5 \quad a_2 = 0.3 \quad a_3 = 1.8 \quad a_4 = 0.9 \quad (23-2)$$

۲۰ پرنده برای $\theta = [\theta_1, \theta_2, \theta_3, \theta_4] = [a_1, a_2, a_3, a_4]$ در نظر می گیریم. ضرایب c_1, c_2 و ضریب

اینرسی را مانند حالت قبل در نظر می گیریم و تابع هزینه زیر را مینیمم می کنیم:

$$J = \sum_{q=1}^8 (y(q) - \hat{y}(q))^2 \quad (24-2)$$

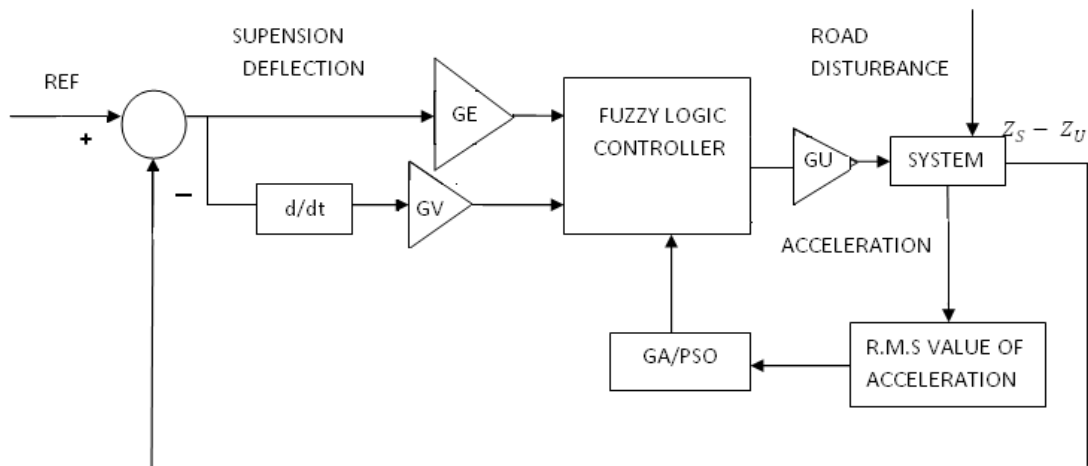
شکل ۲-۶ چگونگی همگرا شدن پارامترها را به سمت مقادیر واقعی نشان می‌دهد.



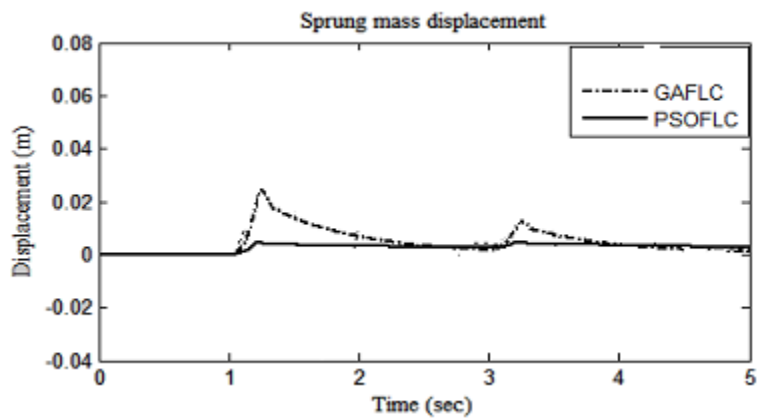
شکل ۲-۶: همگرایی پارامترهای سیستم غیر خطی

همان طور که می‌دانیم تنظیم پارامترهای کنترل کننده‌های فازی معمولاً با سعی و خطا انجام می‌شود. اما ظهور الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت و ترکیب آنها با شبکه‌های عصبی و منطق فازی تحول‌شگرفی در طراحی سیستم‌های هوشمند ایجاد کرد. استفاده از الگوریتم بهینه‌سازی پرندگان برای تنظیم پارامترهای کنترل کننده فازی در سالهای اخیر بسیار متداول شده است. به عنوان مثال، از این الگوریتم در تنظیم پارامترهای کنترل کننده فازی سیستم تعلیق خودرو استفاده شده است. بلوک دیگرام سیستم کنترل در شکل ۲-۷ نشان داده شده است. GV, GU و GU ضرایب تنظیم مقیاس^۱ می‌باشند. در [۷۳] با استفاده از الگوریتم ژنتیک و الگوریتم بهینه‌سازی پرندگان این ضرایب تنظیم شده‌اند. همان طور که در شکل ۲-۸ ملاحظه می‌شود عملکرد الگوریتم بهینه‌سازی پرندگان بسیار مناسب‌تر از الگوریتم ژنتیک می‌باشد.

¹ Scaling factor



شکل ۲-۷: بلوک دیاگرام سیستم کنترل تعلیق خودرو



شکل ۲-۸: مقایسه عملکرد الگوریتم بهینه سازی پرندگان و الگوریتم ژنتیک در طراحی سیستم تعلیق خودرو

فصل سوم

مروری بر مدلسازی سینماتیکی و دینامیکی بازوهای ماهر

مکانیکی

❖ مقدمه

❖ سینماتیک مستقیم

❖ سینماتیک وارون

❖ ماتریس ژاکوبین

❖ مدلسازی دینامیکی

❖ مدل ریاضی بازوی ماهر کروی

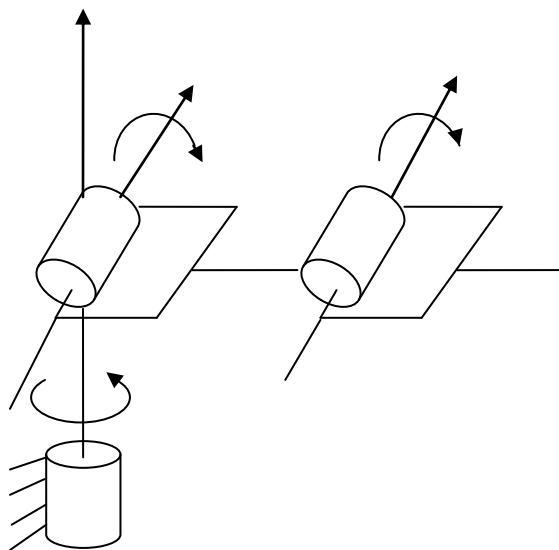
۳-۱- مقدمه

در این فصل چگونگی بدست آوردن مدل ریاضی بازوهای ماهر مکانیکی را تشریح خواهیم کرد. مدل سازی ربات ها شامل دو بخش مدل سازی سینماتیکی و دینامیکی است. در مدل سازی سینماتیکی روابط مربوط به حرکت ربات را بدست می آوریم. به طور کلی می توان گفت منظور از سینماتیک بررسی حرکت بدون حضور نیرو است. در مدل سازی سینماتیکی تشکیل جدول دناویت هارتنبرگ مهمترین گام است. این الگوریتم روشی منظم برای انجام سینماتیک مستقیم ارائه می کند. در مدلسازی دینامیکی روابط انرژی جنبشی و پتانسیل ربات را بدست می آوریم. سپس با استفاده از روش های مکانیک تحلیلی مدل دینامیکی ربات را بدست می آوریم. در پایان، مدل ریاضی ربات کروی ارائه شده است.

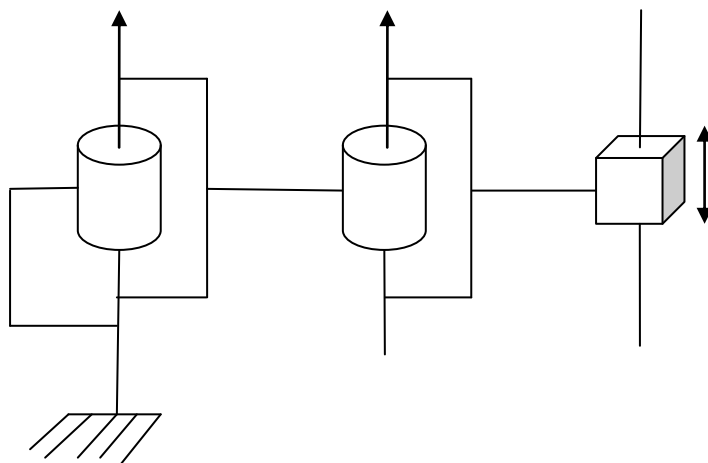
۳-۲- سینماتیک مستقیم

مفاصل رباتها به دو نوع کشویی و لولایی تقسیم می شوند. مفصل لولایی (که به اختصار با R نمایش داده می شود) امکان چرخش نسبی بین دو رابط را فراهم می آورد. مفصل کشویی (که با نماد P نشان داده می شود) اجازه حرکت نسبی طولی بین دو رابط را می دهد. با توجه به چگونگی ترتیب مفاصل رباتها، پیکربندی های مختلفی از آنها وجود دارد. پیکربندی های متداول عبارتند از: هنرمند، استوانه ای، اسکارا، استنفورد و کروی. در پیکربندی هنرمند سه مفصل لولایی وجود دارد. در بازوی ماهر استوانه ای اولین مفصل از نوع لولایی می باشد که اجازه چرخش حول پایه را فراهم می آورد و دو مفصل بعدی کشویی هستند. دیاگرام مفصلی این ربات در شکل ۳-۱ رسم شده است. دیاگرام مفصلی رباتهای اسکارا،

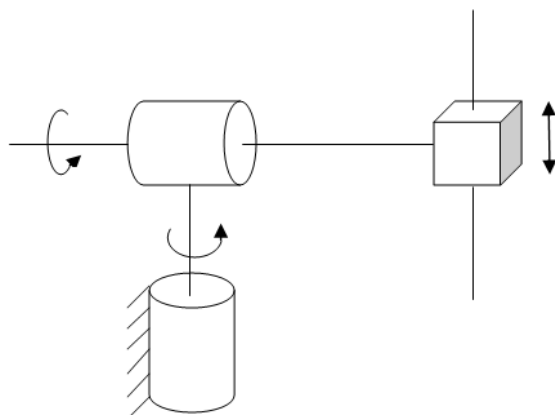
استنفورد و کروی نیز به ترتیب در شکل‌های ۲-۳ تا ۴-۳ رسم شده‌اند. متغیرهای مفاصل در مفصل
 لولایی زاویه بین دو رابط و در مفاصل کشویی طول رابط می‌باشد.



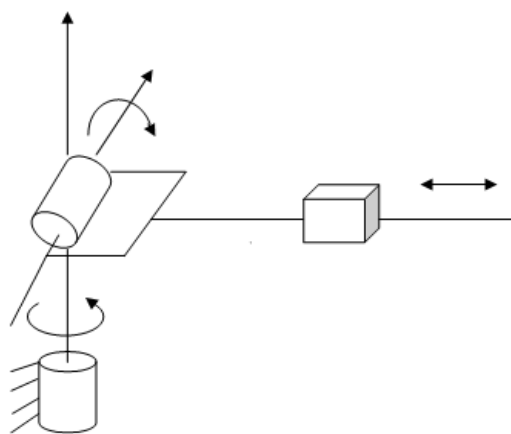
شکل ۱-۳ دیاگرام مفصلی ربات هنرمند



شکل ۲-۳ دیاگرام مفصلی ربات اسکارا



شکل ۳-۳ دیانگرام مفصلی ربات استنفورد



شکل ۴-۳ دیانگرام مفصلی ربات کروی

دستورالعمل دناویت-هارتنبرگ روشی منظم و منسجم برای مدلسازی سینماتیکی انواع رباتها می باشد. آنالیز سینماتیک مستقیم (تعیین موقعیت و جهت مجری نهایی با استفاده از متغیرهای مفاصل) با استفاده از این دستورالعمل انجام می شود. برای انجام این کار ابتدا باید ماتریس های تبدیل و دوران و بردار انتقال را معرفی کنیم.

۳-۲-۱- ماتریس دوران: فرض کنید P_0 نمایش نقطه P در دستگاه مختصات $\{0\}$ با محورهای

$\{x_0, y_0, z_0\}$ باشد و P_1 نمایش همان نقطه در دستگاه مختصات $\{1\}$ با محورهای $\{x_1, y_1, z_1\}$ باشد.

همان طور که در شکل ۳-۵ مشاهده می شود، مبدأ این دو دستگاه نقطه O می باشد، اما محورهای آنها

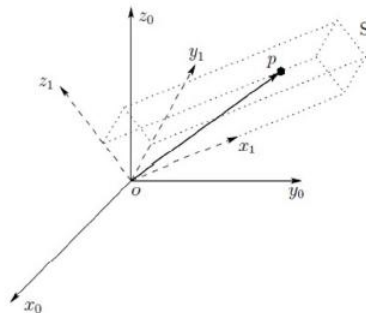
نسبت به هم دوران یافته است. با استفاده از ماتریس دوران و P_0 می توانیم P_1 را بدست آوریم و

بالعکس. به راحتی می توان نشان داد [سپانگ] که R_1^0 (توصیف دستگاه $\{0\}$ در دستگاه $\{1\}$) به صورت

زیر بدست می آید.

$$R_1^0 = \begin{bmatrix} i_0 \cdot i_1 & j_0 \cdot i_1 & k_0 \cdot i_1 \\ i_0 \cdot j_1 & j_0 \cdot j_1 & k_0 \cdot j_1 \\ i_0 \cdot k_1 & j_0 \cdot k_1 & k_0 \cdot k_1 \end{bmatrix} \quad (1-3)$$

$\{i, j, k\}$ بردارهای یکه دستگاه های مختصات می باشند که در هم ضرب نقطه ای می شوند.



شکل ۳-۵ محورهای مختصات دوران یافته

همچنین روابط زیر به سادگی قابل اثبات می باشند [۱].

$$P_1 = R_1^0 P_0 \quad , \quad P_0 = R_0^1 P_1 \quad (2-3)$$

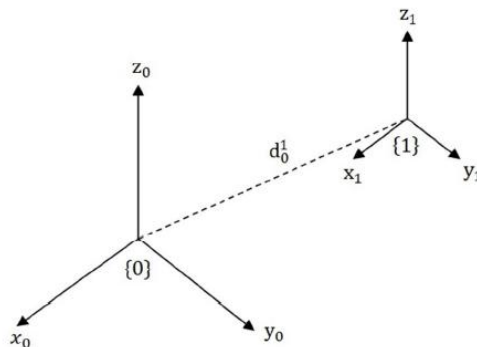
$$R_0^1 = R_1^0 T = R_1^0^{-1} \quad (3-3)$$

به عنوان مثال اگر دستگاه {1} از دوران دستگاه {0} حول محور z_0 به اندازه زاویه θ بوجود آمده باشد، خواهیم داشت:

$$R_0^1 = R_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-3)$$

۳-۲-۲- بردار انتقال: فرض کنید در شکل ۳-۶ محورهای دستگاه‌های {0} و {1} موازی باشند و بردار d_0^1 برداری از مبدأ O_0 به مبدأ O_1 است که در دستگاه {0} بیان می‌شود. مطابق قبل، هر نقطه دو نمایش P_0 و P_1 دارد که به صورت زیر به هم مربوط می‌شوند.

$$P_0 = P_1 + d_0^1 \quad (5-3)$$



شکل ۳-۶ دستگاه مختصات انتقال یافته

بیشترین رابطه کلی بین دستگاه‌های مختصات به صورت ترکیب دوران خالص و انتقال خالص می‌تواند باشد و به عنوان حرکت صلب معرفی می‌شود [۱]. اگر دو حرکت صلب به صورت زیر داشته باشیم:

$$P_0 = R_0^1 P_1 + d_0^1 \quad (6-3)$$

$$P_1 = R_1^2 P_2 + d_1^2 \quad (7-3)$$

ترکیب آنها حرکت صلب سومی را تعریف می‌نماید که می‌توانیم با جایگذاری P_1 از (۷) در (۶) بیان کنیم.

$$P_0 = R_0^1 R_1^2 P_2 + R_0^1 d_1^2 + d_0^1 \quad (۸-۳)$$

چون رابطه بین P_0 و P_2 یک حرکت صلب است می توانیم آن را به صورت زیر بیان کنیم:

$$P_0 = R_0^2 P_2 + d_0^2 \quad (۹-۳)$$

بنابراین، داریم:

$$R_0^2 = R_0^1 R_1^2 \quad (۱۰-۳)$$

$$d_0^2 = R_0^1 d_1^2 + d_0^1 \quad (۱۱-۳)$$

تساوی ماتریسی زیر

$$\begin{bmatrix} R_0^1 & d_0^1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1^2 & d_1^2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0^1 d_1^2 & R_0^1 d_1^2 + d_0^1 \\ 0 & 1 \end{bmatrix} \quad (۱۲-۳)$$

که در آن 0 به جای (000) قرار گرفته است نشان می دهد که حرکت های صلب بوسیله مجموعه ماتریس ها به فرم زیر می تواند نشان داده شود:

$$T = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (۱۳-۳)$$

که به آن ماتریس تبدیل همگن می گوییم. به عنوان مثال، ماتریس تبدیل همگن T که دوران α درجه حول محور X ، سپس انتقال b واحدی در امتداد محور X فعلی و بعد انتقال d واحدی در امتداد محور فعلی Z و بعد دوران θ درجه حول محور فعلی Z دارد به صورت زیر می باشد:

$$T = Rot_{x,\alpha} Trans_{x,b}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & -s_\theta & 0 & 0 \\ s_\theta & c_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۱۴-۳)$$

که در آن در c_θ نماد $\cos(\theta)$ و s_θ نشان دهنده $\sin(\theta)$ است. در دناویت هارتنبرگ نیز هر ماتریس تبدیل A_i به صورت چهار ضرب تبدیل پایه نشان داده می شود:

$$A_i = Rot_{z,\theta_i} \times Trans_{z,d_i} \times Trans_{x,a_i} \times Rot_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\alpha_i}s_{\theta_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -s_{\alpha_i}c_{\theta_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15-3)$$

کمیت های $\theta_i, \alpha_i, d_i, a_i$ در گام γ دستورالعمل دناویت هارتنبرگ تعریف خواهند شد. در ربات هایی با $n+1$ رابط، رابط ها را از صفر تا n شماره گذاری می کنیم از پایه ربات به عنوان رابط صفر استفاده می کنیم و مفاصل را از یک تا n شماره گذاری می کنیم. به انتهای هر رابط یک دستگاه مختصات متصل می کنیم. شماره این دستگاه مختصات همان شماره رابط است.

۳-۲-۳- الگوریتم دناویت هارتنبرگ: با این مقدمات، دستورالعمل دناویت هارتنبرگ را

می توان به صورت زیر خلاصه نمود:

گام ۱: محورهای مفاصل را z_0, \dots, z_{n-1} قرار داده و نامگذاری کنید.

گام ۲: دستگاه پایه را نصب کنید. مبدأ را در هر جای دلخواه روی محور z_0 تنظیم کنید.

محورهای x_0 و y_0 را با در نظر گرفتن دستگاه راستگرد به طور مناسب انتخاب کنید. برای

$i = 1, \dots, n-1$ گام های ۳ تا ۵ را اجرا نمایید.

گام ۳: مبدأ o_i را جایی که عمود مشترک Z_i و Z_{i-1} ، Z_i را قطع می کند قرار دهید. اگر Z_{i-1} و Z_i متقاطع هستند نقطه o_i را در نقطه تقاطع قرار دهید. اگر Z_{i-1} و Z_i موازی هستند o_i را در محل مفصل i قرار دهید.

گام ۴: x_i را در امتداد عمود مشترک بین Z_{i-1} و Z_i و در عبور از o_i قرار دهید. وقتی Z_{i-1} و Z_i متقاطع هستند در جهت عمود به صفحه Z_i و Z_{i-1} قرار دهید.

گام ۵: y_i را با تکمیل دستگاه راستگرد مشخص سازید.

گام ۶: دستگاه مختصات قسمت پایانی $x_n y_n z_n$ را تعیین کنید.

گام ۷: یک جدول از پارامترهای رابط $\theta_i, \alpha_i, d_i, a_i$ درست کنید.

a_i : طول امتداد x_i از o_i تا محل تقاطع محورهای x_i و Z_{i-1} می باشد.

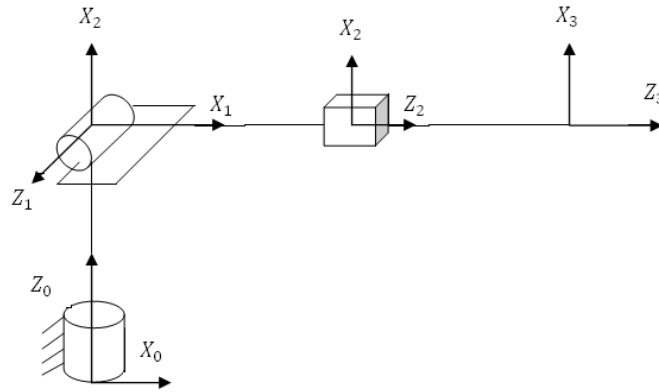
d_i : طول امتداد Z_{i-1} از o_{i-1} تا محل تقاطع محورهای x_i و Z_{i-1} است. هرگاه مفصل i کشویی باشد d_i متغیر است.

α_i : زاویه بین Z_{i-1} و Z_i که حول x_i اندازه گیری می شود.

θ_i : زاویه بین x_{i-1} و x_i که حول Z_{i-1} اندازه گیری می شود. هرگاه مفصل i لولایی باشد θ_i متغیر است.

گام ۸: ماتریس های تبدیل همگن A_i را با جایگذاری پارامترهای بالا در (۱۵) تشکیل دهید.

گام ۹: $T_0^n = A_1 \dots A_n$ را تشکیل دهید. این ماتریس تبدیل موقعیت و جهت دستگاه آخرین دستگاه مختصات را در دستگاه مختصات پایه نشان می دهد. شکل ۳-۷ چگونگی اختصاص دستگاه های مختصات به بازوی ماهر کروی با استفاده از قوانین دناویت هارتنبرگ نشان می دهد.



شکل ۳-۷ اختصاص محورهای مختصات به ربات کروی طبق الگوریتم دناویت هارتنبرگ

جدول دناویت هارتنبرگ برای این ربات به صورت زیر است.

شماره رابط	θ	d	a	α
رابط ۱	θ_1^*	d_1	0	$-\pi/2$
رابط ۲	θ_2^*	0	0	$\pi/2$
رابط ۳	0	d_3^*	0	0

جدول ۳-۱ جدول دناویت هارتنبرگ برای ربات کروی

ماتریس تبدیل هر رابط با استفاده از رابطه (۱۵) و مقادیر داده شده در جدول به ازای آن رابط، به صورت زیر بدست می آید.

$$A_1 = T_0^1 = \begin{bmatrix} c_{\theta_1} & 0 & s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۱۶-۳)$$

$$A_2 = T_1^2 = \begin{bmatrix} c_{\theta_2} & 0 & s_{\theta_2} & 0 \\ s_{\theta_2} & 0 & -c_{\theta_2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17-3)$$

$$A_3 = T_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18-3)$$

$$T_0^3 = T_0^1 T_1^2 T_2^3 \quad (19-3)$$

$$T_0^3 = \begin{bmatrix} c_{\theta_1} c_{\theta_2} & s_{\theta_1} & c_{\theta_1} s_{\theta_2} & d_3 c_{\theta_1} s_{\theta_2} \\ s_{\theta_1} c_{\theta_2} & -c_{\theta_1} & s_{\theta_2} s_{\theta_1} & d_3 s_{\theta_2} s_{\theta_1} \\ s_{\theta_2} & 0 & -c_{\theta_2} & d_1 - d_3 c_{\theta_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20-3)$$

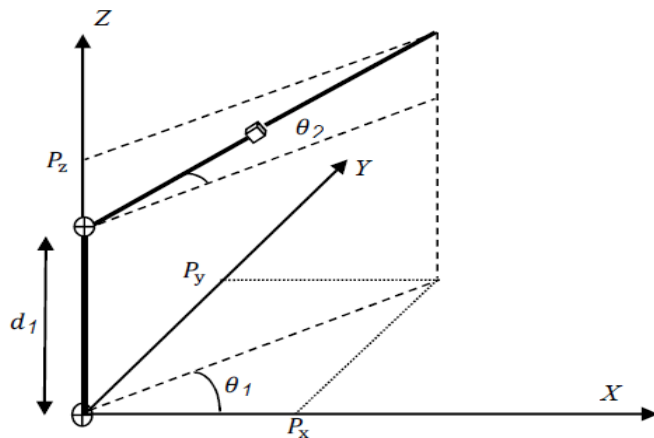
۳-۳- سینماتیک وارون

منظور از سینماتیک وارون یافتن متغیرهای مفاصل به ازای موقعیت و جهت مجری نهایی است. برای سینماتیک وارون الگوریتم مشخصی ارائه نشده است و با توجه به پیچیدگی خاص هر ربات انجام می شود. فرض کنید مختصات مجری نهایی $P = [p_x, p_y, p_z]^T$ معلوم است و می خواهیم متغیرهای مفاصل را بر حسب آن بیان کنیم. با استفاده از دیاگرام مفصلی نشان داده شده در شکل ۳-۸ و استفاده از روابط هندسی و مثلثاتی روابط زیر به سادگی بدست می آیند

$$\theta_1 = \text{atan2}(p_x, p_y) \quad (21-3)$$

$$\theta_2 = \text{atan2}(\sqrt{p_x^2 + p_y^2}, p_z - d_1) \quad (22-3)$$

$$d_3 = \sqrt{(p_z - d_1)^2 + p_x^2 + p_y^2} \quad (23-3)$$



شکل ۳-۸ دیاگرام مفصلی ربات کروی برای انجام سینماتیک وارون

۳-۴- ماتریس ژاکوبین

روابط سرعت ربات بوسیله ماتریس ژاکوبین بدست می‌آیند. این ماتریس یکی از مهمترین کمیت‌های آنالیز و کنترل حرکت ربات محسوب می‌شود. ماتریس ژاکوبین در طراحی مسیره‌های هموار، تعیین ترکیب‌های تکین، تبدیل نیروها و گشتاورها از مجری نهایی به مفاصل به کار می‌رود. در حقیقت این ماتریس را می‌توان به عنوان یک ماتریس تبدیل در نظر گرفت که بردار سرعت مفاصل (\dot{q}) را به بردار سرعت در فضای دکارتی یا همان فضای کار (کل حجم جارو شده توسط مجری نهایی ربات) (\dot{x}) تبدیل می‌کند. رابطه سرعت‌ها در فضای مفصلی و فضای کار به صورت زیر می‌باشد:

$$\dot{x} = J(q)\dot{q} \quad (۳-۲۴)$$

روابط بدست آوردن ژاکوبین به صورت زیر خلاصه می‌شود. (چگونگی بدست آوردن این روابط به تفصیل در [۱] ارائه شده است.)

ماتریس ژاکوبین برای مفاصل کشویی:

$$J_i = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \quad (25-3)$$

ماتریس ژاکوبین برای مفاصل لولایی:

$$J_i = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (d_0^n - d_0^{i-1}) \\ z_{i-1} \end{bmatrix} \quad (26-3)$$

ژاکوبین بازوی ماهر کروی به صورت زیر است:

$$J = [J_1 J_2 J_3] = \begin{bmatrix} z_0 \times d_0^3 & z_1 \times (d_0^3 - d_0^1) & z_2 \\ z_0 & z_1 & 0 \end{bmatrix} \quad (27-3)$$

که در آن

$$z_{i-1} = R_0^{i-1} k \quad (28-3)$$

$$k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (29-3)$$

$$d_0^1 = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \quad (30-3)$$

$$d_0^3 = \begin{bmatrix} d_3 c_{\theta_1} s_{\theta_2} \\ d_3 s_{\theta_1} s_{\theta_2} \\ d_1 - d_3 c_{\theta_2} \end{bmatrix} \quad (31-3)$$

$$z_2 = R_0^2 k = \begin{bmatrix} c_{\theta_1} s_{\theta_2} \\ s_{\theta_1} s_{\theta_2} \\ -c_{\theta_2} \end{bmatrix} \quad (32-3)$$

بنابراین ماتریس ژاکوبین به صورت زیر می باشد:

$$J = \begin{bmatrix} -d_3 s_{\theta_1} s_{\theta_2} & -d_1 c_{\theta_1} & c_{\theta_1} s_{\theta_2} \\ d_3 c_{\theta_1} s_{\theta_2} & -d_1 s_{\theta_1} & s_{\theta_1} s_{\theta_2} \\ 0 & 0 & -c_{\theta_2} \\ 0 & s_{\theta_1} & 0 \\ 0 & -c_{\theta_1} & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (33-3)$$

معمولاً به مجری نهایی ربات‌ها یک میج کروی متصل می‌شود که امکان قرار گرفتن در جهت‌های مختلف را فراهم می‌کند. اگر بخواهیم ماتریس ژاکوبین را با در نظر گرفتن میج بدست آوریم باید به روشی که گفته شد عمل کنیم. در غیر این صورت می‌توانیم ژاکوبین را به صورت زیر بدست آوریم:

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} & \frac{\partial P_x}{\partial d_3} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} & \frac{\partial P_y}{\partial d_3} \\ \frac{\partial P_z}{\partial \theta_1} & \frac{\partial P_z}{\partial \theta_2} & \frac{\partial P_z}{\partial d_3} \end{bmatrix} \quad (34-3)$$

P_x ، P_y و P_z مختصات مجری نهایی در دستگاه پایه است:

$$P_x = d_3 c_{\theta_1} s_{\theta_2} \quad (35-3)$$

$$P_y = d_3 s_{\theta_1} s_{\theta_2} \quad (36-3)$$

$$P_z = d_1 - d_3 c_{\theta_2} \quad (37-3)$$

بنابراین، ماتریس ژاکوبین به صورت زیر بدست می‌آید:

$$J = \begin{bmatrix} -d_3 s_{\theta_1} s_{\theta_2} & d_3 c_{\theta_1} c_{\theta_2} & c_{\theta_1} s_{\theta_2} \\ d_3 c_{\theta_1} s_{\theta_2} & d_3 s_{\theta_1} c_{\theta_2} & s_{\theta_1} s_{\theta_2} \\ 0 & d_3 s_{\theta_2} & -c_{\theta_2} \end{bmatrix} \quad (38-3)$$

وضعیت های تکین: به ازای برخی مقادیر متغیرهای ربات، مرتبه ماتریس ژاکوبین کاهش پیدا می کند که به آنها وضعیت های تکین ربات گفته می شود. بنابراین، نقاط تکین ربات کروی از رابطه زیر بدست می آیند:

$$\det(J) = d_3^2 s_{\theta_2} = 0 \Rightarrow \theta_2 = 0, \pi \quad (39-3)$$

وضعیت های تکین به دلایل زیر حائز اهمیت هستند:

- در وضعیت های تکین، برخی جهات معینی از حرکت، قابل اجرا نمی باشد.
- در تکین ها ممکن است سرعت های محدود مجری نهایی به سرعت های نامحدود مفاصل منجر شود.
- نزدیک تکین ها راه حل یگانه ای برای سینماتیک وارون وجود ندارد. ممکن است پاسخ وجود نداشته باشد یا بی شمار پاسخ داشته باشیم.
- در تکین ها ممکن است نیروها و گشتاورهای محدود مجری نهایی به سرعت های نامحدود مفاصل منجر شود.

۳-۵- مدلسازی دینامیکی

برای بدست آوردن مدل دینامیکی ربات انرژی‌های جنبشی و پتانسیل ربات را محاسبه می‌کنیم. سپس لاگرانژین سیستم را تشکیل می‌دهیم. آنگاه با استفاده از معادلات اویلر لاگرانژ معادله دینامیکی ربات بدست می‌آید.

۳-۵-۱- انرژی جنبشی: انرژی جنبشی ربات از مجموع انرژی‌های جنبشی رابط‌های آن بدست می‌آیند و انرژی جنبشی هر رابط از مجموع انرژی‌های جنبشی تمام نقاط آن بدست می‌آید. می‌توان نشان داد که انرژی جنبشی یک ربات با n رابط از رابطه زیر بدست می‌آید.

$$KE = \frac{1}{2} \dot{q}^T \left(\sum_{i=1}^n (m_i J_v^T J_v + J_w^T R_i I_i R_i^T J_w) \right) \dot{q} \quad (40-3)$$

J_v و J_w از روابط (۳-۲۵) و (۳-۲۶) با توجه به نوع مفصل i ام محاسبه می‌شوند و R_i ماتریس دوران دستگاه مختصات i ام در دستگاه مبنا می‌باشد. m_i جرم رابط i ام است. در رابطه بالا I_i تانسور اینرسی یا تانسور لختی می‌باشد که به صورت زیر محاسبه می‌شود:

$$I = \int_A \begin{bmatrix} -(p_z^2 + p_y^2) & p_x p_y & p_x p_z \\ p_x p_y & -(p_x^2 + p_z^2) & p_z p_y \\ p_x p_z & p_z p_y & -(p_x^2 + p_y^2) \end{bmatrix} dm \quad (41-3)$$

اگر تعریف کنیم:

$$D(q) = \sum_{i=1}^n (m_i J_v^T J_v + J_w^T R_i I_i R_i^T J_w) \quad (42-3)$$

خواهیم داشت:

$$KE = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (43-3)$$

$D(q)$ ماتریس اینرسی ربات نام دارد.

۳-۵-۲- انرژی پتانسیل: انرژی پتانسیل گرانشی ربات با قرار دادن مبدأ دستگاه مختصات رابط در

مرکز جرم آن و استفاده از رابطه معروف mgh به صورت زیر محاسبه می‌شود. r_{c_i} مختصات مرکز جرم می‌باشد.

$$PE = g^T r_{c_i} m_i \quad (44-3)$$

$$PE = g^T (m_1 d_0^{c_1} + m_2 d_0^{c_2} + m_3 d_0^{c_3}) \quad (45-3)$$

$$g^T = [0 \quad 0 \quad 9.81] \quad (46-3)$$

۳-۵-۳- لاگرانژین: لاگرانژین عبارتست از اختلاف انرژی جنبشی و پتانسیل. بنابراین می‌توان

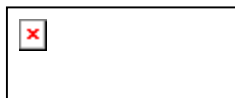
نوشت:

$$L = KE - PE \quad (47-3)$$

$$L = \frac{1}{2} \dot{q}^T D(q) \dot{q} - PE \quad (48-3)$$

۳-۵-۴- معادله اوایلر-لاگرانژ: معادله دینامیکی ربات با قرار دادن لاگرانژین از رابطه (۴۷-۳) در

معادله اوایلر-لاگرانژ که به صورت زیر بیان می‌شود، بدست می‌آید.



(49-3)

با جایگذاری لاگرانژین در رابطه فوق و انجام محاسبات لازم، معادله دینامیکی ربات به صورت زیر بدست می آید.

$$D(q)\ddot{q} + \dot{D}(q)\dot{q} - \frac{1}{2} \frac{\partial}{\partial q} (\dot{q}^T D(q) \dot{q}) + \frac{\partial}{\partial q} PE = \tau \quad (50-3)$$

که معمولاً به فرم زیر بیان می شود:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (51-3)$$

۳-۶- مدل ریاضی بازوی ماهر کروی

در این پایان نامه فرض می کنیم مدل ربات (ماتریس های D, G, C) در اختیار است و از مدل بدست آمده در [۷۴] استفاده می کنیم.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (52-3)$$

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (53-3)$$

$$I_1 = \begin{bmatrix} I_{xx1} & I_{xy1} & I_{xz1} \\ I_{yx1} & I_{yy1} & I_{yz1} \\ I_{zx1} & I_{zy1} & I_{zz1} \end{bmatrix} = \begin{bmatrix} 20.2 & 0 & 0 \\ 0 & 20.25 & 0 \\ 0 & 0 & 32.05 \end{bmatrix} \quad (54-3)$$

$$I_2 = \begin{bmatrix} I_{xx2} & I_{xy2} & I_{xz2} \\ I_{yx2} & I_{yy2} & I_{yz2} \\ I_{zx2} & I_{zy2} & I_{zz2} \end{bmatrix} = \begin{bmatrix} 2.7 & -2.98 & -0.32 \\ -2.98 & 12.61 & 0.1 \\ -0.32 & 0.1 & 13 \end{bmatrix} \quad (\Delta\Delta-3)$$

$$I_3 = \begin{bmatrix} I_{xx3} & I_{xy3} & I_{xz3} \\ I_{yx3} & I_{yy3} & I_{yz3} \\ I_{zx3} & I_{zy3} & I_{zz3} \end{bmatrix} = \begin{bmatrix} 4.42 & 0 & 0 \\ 0 & 4.39 & 0 \\ 0 & 0 & 0.14 \end{bmatrix} \quad (\Delta\Delta-3)$$

$$D(q) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \quad (\Delta\Delta-3)$$

$$d_{11} = I_{yy1} + I_{zz2} + I_{zz3} + (I_{xx2} + I_{xx3} - I_{zz2} - I_{zz3}) \sin^2(q_2) \quad (\Delta\Delta-3)$$

$$-2(I_{xz2} + I_{xz3}) \cos(q_2) \sin(q_2)$$

$$+(m_2 l_{c2}^2 + m_3 l_{c3}^2)$$

$$d_{12} = (I_{xy2} + I_{xy3}) \sin(q_2) - (I_{yz3} + I_{yz2}) \cos(q_2) \quad (\Delta\Delta-3)$$

$$d_{13} = 0 \quad (\Delta\Delta-3)$$

$$d_{21} = d_{12} \quad (\Delta\Delta-3)$$

$$d_{22} = m_2 l_{c2}^2 + m_3 l_{c3}^2 + 2m_3 l_{c3} q_3 + m_3 q_3^2 + I_{yy2} + I_{yy3} \quad (\Delta\Delta-3)$$

$$d_{23} = d_{31} = d_{32} = 0 \quad (\Delta\Delta-3)$$

$$d_{33} = m_3 \quad (\Delta\Delta-3)$$

$$h(q, \dot{q}) = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \quad (65-3)$$

$$h_1 = \dot{q}_2^2 [(I_{xy2} + I_{xy3}) \cos(q_2) + (I_{yz2} + I_{yz3}) \sin(q_2)] \quad (66-3)$$

$$-2\dot{q}_1\dot{q}_2(I_{xz2} + I_{xz3}) \cos(2q_2)$$

$$+\dot{q}_1\dot{q}_2(I_{xx2} + I_{xx3} - I_{zz2} - I_{zz3} + m_3(2l_{c3}q_3 + q_3^2 + l_{c3}^2) + m_2l_{c2}^2) \sin(2q_2)$$

$$+\dot{q}_1\dot{q}_3m_3(l_{c3} + q_3 - (l_{c3} + q_3)\cos(2q_2))$$

$$h_2 = \dot{q}_1^2(I_{xz2} + I_{xz3}) \cos(2q_2) \quad (67-3)$$

$$+\frac{1}{2} \sin(2q_2) [\dot{q}_1^2(I_{zz2} + I_{zz3} - I_{xx2} - I_{xx3} - m_2l_{c2}^2 - m_3l_{c3}^2 - m_3q_3^2)]$$

$$\sin(q_2) g[m_2l_{c2} + m_3l_{c3} + m_3q_3] - \dot{q}_1^2 m_3 l_{c3} q_3 \sin(2q_2)$$

$$+2\dot{q}_2\dot{q}_3(m_3l_{c3} + m_3q_3)$$

$$h_3 = -m_3[\dot{q}_1^2(l_{c3} + q_3 - l_{c3}\cos^2(q_2) - q_3\cos^2(q_2)) + \dot{q}_2^2(l_{c3} + q_3) \quad (68-3)$$

$$+ g \cos(q_2)]$$

در روابط فوق، m_i جرم رابط i ام، I_i تانسور اینرسی رابط i ام، l_{ci} فاصله مرکز جرم رابط i ام از مفصل قبلی می باشد.

$$m_1 = 60.47, m_2 = 88.25, m_3 = 14.32 \quad (Kg) \quad (69-3)$$

$$l_{c1} = 0.5, l_{c2} = 0.11, l_{c3} = .4 \quad (m) \quad (70-3)$$

فصل چهارم

کنترل گشتاور سیستم‌های رباتیک

❖ مقدمه

❖ دینامیک وارون (گشتاور محاسباتی)

❖ کنترل غیرخطی مقاوم

۴-۱- مقدمه

همان طور که در فصل اول اشاره شد، روش خطی سازی فیدبکی (دینامیک وارون) یکی از محبوب ترین و پرکاربردترین روش های ارائه شده برای کنترل ربات می باشد که مبتنی بر کنترل گشتاور می باشد. یکی از اهداف این پایان نامه مقایسه دو استراتژی کنترل گشتاور و کنترل ولتاژ می باشد. بنابراین این فصل، به استراتژی کنترل گشتاور اختصاص داده شده است. در این فصل، ابتدا روش دینامیک وارون را مرور می کنیم و آن را روی ربات کروی اجرا می کنیم. در ادامه، روش کنترل غیر خطی مقاوم را بیان می کنیم و از آن برای کنترل ربات مورد نظر استفاده می کنیم. در شبیه سازی ها عملکرد استراتژی کنترل گشتاور را در برابر عدم قطعیت های پارامتری، دینامیک مدل نشده و اغتشاش خارجی بررسی خواهیم کرد. در فصل بعد استراتژی کنترل ولتاژ را معرفی می کنیم و نتایج این دو استراتژی را بررسی خواهیم کرد.

۴-۲- دینامیک وارون (گشتاور محاسباتی)

معادله دینامیکی ربات را به صورت زیر در نظر بگیرید:

$$D(q)\ddot{q} + h(q, \dot{q}) = \tau \quad (1-4)$$

قانون کنترل را با استفاده از خطی سازی فیدبکی به صورت زیر پیشنهاد می دهیم.

$$D(q)(\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q)) + h(q, \dot{q}) = \tau \quad (2-4)$$

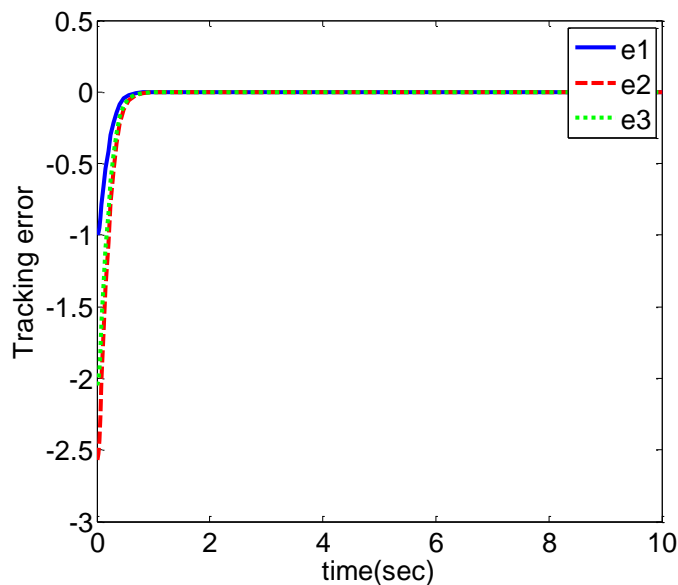
q_d مسیر مطلوب در فضای مفصلی می‌باشد. با انتخاب مقادیر مناسب برای ماتریس‌های قطری K_p و K_d می‌توان رفتار گذرای سیستم را کنترل نمود. با استفاده از روابط (۱-۴) و (۲-۴) دینامیک حلقه بسته به صورت زیر در می‌آید.

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (۳-۴)$$

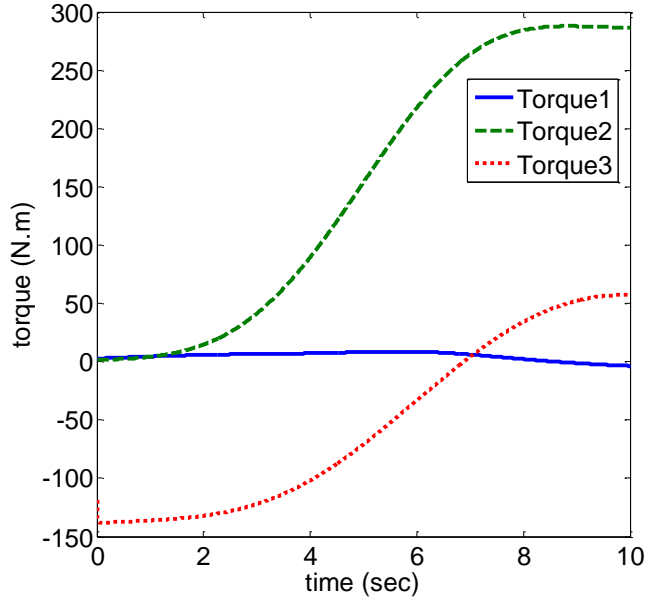
که در آن

$$e = q_d - q \quad (۴-۴)$$

با انتخاب $K_d = 20I$ و $K_p = 100I$ این روش را روی ربات کروی اجرا می‌کنیم. مسیر مطلوب برای هر سه مفصل $q_d = 1 - \cos\left(\frac{\pi}{10}t\right)$ می‌باشد. نتایج در شکل‌های زیر ارائه شده است.



شکل ۴-۱ خطای ردگیری روش خطی سازی فیدبکی بدون اعمال عدم قطعیت



شکل ۴-۲ سیگنال کنترل (گشتاور) روش خطی سازی فیدبکی بدون اعمال عدم قطعیت

۴-۳- کنترل غیر خطی مقاوم

فرض کنید معادله دینامیکی سیستم به صورت زیر باشد:

$$D(q)\ddot{q} + h(q, \dot{q}) = \tau \quad (۵-۴)$$

قانون کنترل زیر را در نظر بگیرید.

$$\hat{D}(q)u + \hat{h}(q, \dot{q}) = \tau \quad (۶-۴)$$

$\hat{D}(q)$ و $\hat{h}(q, \dot{q})$ تخمین هایی از $D(q)$ و $h(q, \dot{q})$ هستند (مدل نامی). برای ردگیری مسیر مطلوب q_d

در فضای مفصلی، u را به صورت زیر پیشنهاد می دهیم:

$$u = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) + u_r \quad (۷-۴)$$

u_r را برای غلبه بر عدم قطعیت ها، به قانون کنترل اضافه نموده‌ایم و در ادامه مقدار آن را بدست خواهیم آورد. با جایگذاری τ از رابطه (۶-۴) در (۵-۴)، معادله حلقه بسته

$$D(q)\ddot{q} + h(q, \dot{q}) = \widehat{D}(q)u + \widehat{h}(q, \dot{q}) \quad (۸-۴)$$

بدست می‌آید. در نتیجه خواهیم داشت:

$$\ddot{q} = D^{-1}(q)\widehat{D}(q)u + D^{-1}(q)(\widehat{h}(q, \dot{q}) - h(q, \dot{q})) \quad (۹-۴)$$

برای سادگی از شناسه های q و \dot{q} صرف نظر می‌کنیم. با اضافه و کم کردن u به سمت راست رابطه فوق خواهیم داشت:

$$\ddot{q} = u + (D^{-1}\widehat{D} - I)u + D^{-1}(\widehat{h} - h) \quad (۱۰-۴)$$

با جایگذاری u از (۷-۴) در (۱۰-۴) خواهیم داشت:

$$\ddot{q} = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_q - q) + u_r + (D^{-1}\widehat{D} - I)u + D^{-1}(\widehat{h} - h) \quad (۱۱-۴)$$

با تعاریف

$$e = q - q_q \quad (۱۲-۴)$$

$$\eta = (D^{-1}\widehat{D} - I)u + D^{-1}(\widehat{h} - h) \quad (۱۳-۴)$$

خواهیم داشت:

$$\ddot{e} + K_d\dot{e} + k_p e = u_r + \eta \quad (۱۴-۴)$$

اکنون با استفاده از روش مستقیم لیاپانوف u_r را بدست می‌آوریم. ابتدا معادلات حالت سیستم در فضای ردگیری را بدست می‌آوریم. با تعریف متغیرهای حالت به صورت

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (15-4)$$

معادلات حالت به صورت زیر در می آید.

$$\dot{x} = Ax + Bu_s \quad (16-4)$$

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (17-4)$$

$$u_s = u_r + \eta \quad (18-4)$$

تابع لیاپانوف را به صورت زیر پیشنهاد می دهیم

$$F(x) = x^T P x \quad (19-4)$$

که P یک ماتریس متقارن مثبت معین است. مشتق زمانی $F(x)$ برابر است با

$$\dot{F}(x) = x^T (A^T P + PA)x + 2x^T P B u_s = -x^T Q x + 2x^T P B u_s \quad (20-4)$$

اگر A هرویتز باشد، P از معادله ریکاتی زیر محاسبه می شود که در آن Q یک ماتریس متقارن مثبت معین است.

$$A^T P + PA + Q = 0 \quad (21-4)$$

با توجه به رابطه (۲۰-۴) برای اینکه سیستم پایدار مجانبی باشد، باید داشته باشیم:

$$x^T P B u_s = x^T P B (u_r + \eta) \leq 0 \quad (22-4)$$

فرض کنید $\|\lambda\| \leq \varphi$ باشد. در این صورت نشان خواهیم داد که اگر u_r را به صورت

$$u_r = \begin{cases} \frac{-(x^T P B)^T \varphi}{\|x^T P B\|} & \|x^T P B\| \neq 0 \\ 0 & \|x^T P B\| = 0 \end{cases} \quad (23-4)$$

انتخاب کنیم، رابطه (۲۲-۴) بر قرار خواهد شد و پایداری سیستم اثبات می شود.

اثبات: u_r از معادله (۲۳-۴) را در (۲۲-۴) جایگذاری می کنیم.

$$x^T PB \left(\frac{-(x^T PB)^T \varphi}{\|x^T PB\|} + \eta \right) = (-\|x^T PB\| \varphi + x^T PB \eta) \quad (24-4)$$

از طرفی می دانیم که:

$$x^T PB \eta \leq \|x^T PB\| \cdot \|\eta\| \leq \|x^T PB\| \varphi \quad (25-4)$$

بنابراین، خواهیم داشت:

$$x^T PB \eta - \|x^T PB\| \varphi \leq 0 \quad (26-4)$$

و شرط منفی بودن مشتق زمانی تابع لیاپانوف محقق می شود.

اکنون باید مقدار φ (باند عدم قطعیت) در رابطه (۲۳-۴) را بدست آوریم. با توجه به روابط (۱۳-۴) و

(۷-۴) داریم:

$$\eta = (D^{-1} \hat{D} - I) \{ \ddot{q}_d + K_d (\dot{q}_d - \dot{q}) + K_p (q_d - q) + u_r \} + D^{-1} (\hat{h} - h) \quad (27-4)$$

بنابراین η به صورت زیر محدود می شود:

$$\|\eta\| \leq \|(D^{-1} \hat{D} - I)\| \cdot \|\ddot{q}_d - K_d \dot{e} - K_p e + u_r\| + \|D^{-1}\| \cdot \|\hat{h} - h\| \quad (28-4)$$

به ازای تمام مقادیر q داریم [۱]:

$$D_1 \leq \|D^{-1}\| \leq D_2 \quad (29-4)$$

با انتخاب \hat{D} به صورت

$$\hat{D} = \frac{2I}{D_1 + D_2} \quad (30-4)$$

خواهیم داشت [۱]:

$$\|(D^{-1}\hat{D} - I)\| \leq \frac{D_1 - D_2}{D_1 + D_2} =: \alpha \leq 1 \quad (31-4)$$

بنابراین، نامساوی (۲۸-۴) را می‌توان به صورت زیر نوشت:

$$\|\eta\| \leq \alpha \cdot \|\ddot{q}_d - K_d \dot{e} - K_p e\| + \alpha \|u_r\| + \|D^{-1}\| \cdot \|\hat{h} - h\| \quad (32-4)$$

فرض کنید $\|u_r\| \leq \varphi$ در این صورت داریم [۱]:

$$\|\eta\| \leq \alpha \cdot \|\ddot{q}_d - K_d \dot{e} - K_p e\| + \alpha \varphi + D_2 \|\hat{h} - h\| := \varphi \quad (33-4)$$

در نتیجه باند عدم قطعیت به صورت زیر بدست می‌آید:

$$\varphi = \frac{\|\ddot{q}_d - K_d \dot{e} - K_p e\| + D_2 \|\hat{h} - h\|}{1 - \alpha} \quad (34-4)$$

نتایج شبیه سازی ها: روش کنترل غیر خطی مقاوم، با مقادیر پارامترهای زیر روی بازوی ماهر کروی پیاده سازی شد و نتایج آن در شکل های ۳-۴ تا ۴-۴ ارائه شده است.

$$K_p = 100I \quad K_d = 20I \quad D_2 = .5 \quad \alpha = .2 \quad (35-4)$$

در رابطه (۲۱-۴)، Q را ماتریس همانی در نظر می‌گیریم. ماتریس A نیز از رابطه (۱۷-۴) بدست می‌آید. با حل معادله (۲۱-۴) توسط دستور *lyap* در *MATLAB* ماتریس P به صورت زیر خواهد بود:

$$P = \begin{bmatrix} .1253 & 0 & 0 & -.5 & 0 & 0 \\ 0 & .1253 & 0 & 0 & -.5 & 0 \\ 0 & 0 & .1253 & 0 & 0 & -.5 \\ -.5 & 0 & 0 & 2.525 & 0 & 0 \\ 0 & -.5 & 0 & 0 & 2.525 & 0 \\ 0 & 0 & -.5 & 0 & 0 & 2.525 \end{bmatrix} \quad (36-4)$$

مسیر مطلوب برای هر سه مفصل همان مسیر قبل می‌باشد. قانون کنترل از روابط (۴-۶)، (۴-۷) و (۴-۲۳) بدست می‌آید. φ در (۴-۲۳) از رابطه (۴-۳۴) بدست می‌آید. عدم قطعیت پارامتری را به صورت

$$\hat{D} = .8D \quad \hat{h} = .8h \quad (۴-۳۷)$$

اعمال می‌کنیم. در شبیه سازی مدل ربات به صورت

$$D(q)\ddot{q} + h(q, \dot{q}) + \tau_f + \tau_{dist} = \tau \quad (۴-۳۸)$$

در نظر گرفته شده است. τ_f دینامیک مدل نشده (گشتاور اصطکاک) است.

$$\tau_f = [20\dot{q}_1 + 8\text{sign}(\dot{q}_1) \quad 4\dot{q}_2 + .16\text{sign}(\dot{q}_2) \quad 20\dot{q}_3 + 8\text{sign}(\dot{q}_3)]^T \quad (۴-۳۹)$$

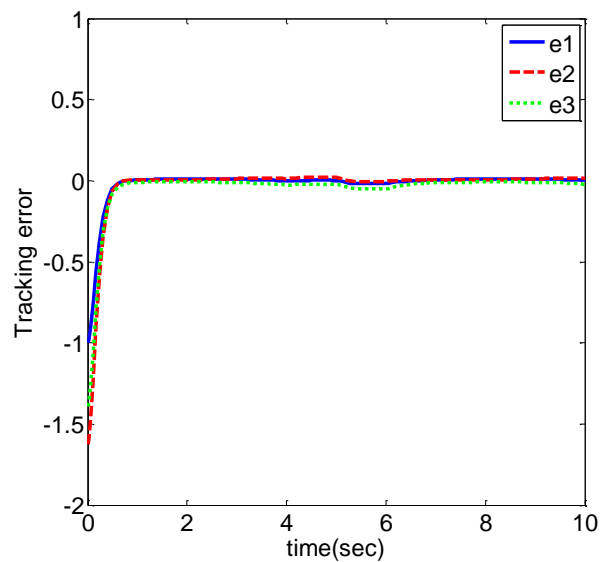
اغتشاش خارجی τ_{dist} نیز مجموع ۳ تابع پله می‌باشد که به صورت

$$\tau_{dist} = 25s(t - 4) - 50s(t - 5) + 25s(t - 6) \quad (۴-۴۰)$$

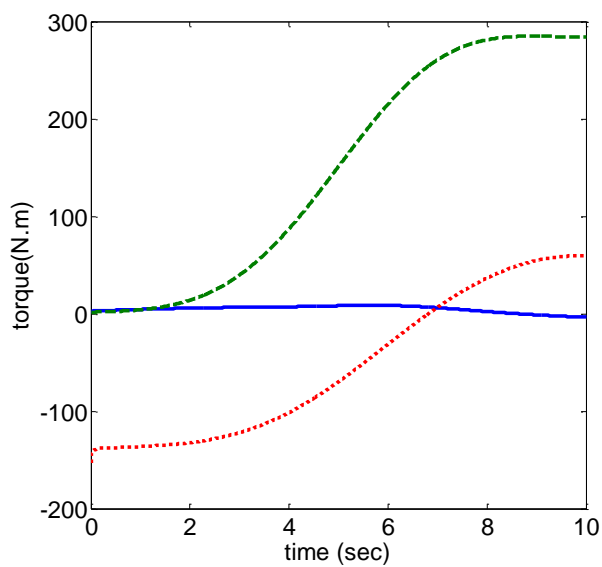
به سیستم اعمال شده است. $s(t)$ تابع پله واحد می‌باشد. دامنه گشتاور اعمالی در حالت ماندگار در روش دینامیک وارون (شکل ۴-۲) حداکثر ۱۲۰ نیوتون متر است و دامنه گشتاور اغتشاش تقریباً ۲۰ درصد آن انتخاب شده است. خطای ردگیری در شکل ۴-۳ نشان داده شده است. همان طور که ملاحظه می‌شود، روش کنترل مقاوم غیر خطی در جبران عدم قطعیت‌ها عملکرد قابل قبولی دارد. شکل ۴-۴ سیگنال کنترل را نمایش می‌دهد. پرش‌هایی در این سیگنال در لحظه‌های ۴، ۵ و ۶ ثانیه دیده می‌شود که بیانگر عملکرد صحیح کنترلی برای دفع اغتشاش خارجی است. به منظور راحت تر شدن مقایسه عملکرد روش‌ها، معیاری به صورت زیر در نظر می‌گیریم.

$$J = 0.1 \int_0^{10} (|e_1| + |e_2| + |e_3|) dt \quad (۴-۴۱)$$

و آن را برای هر روش محاسبه می‌کنیم. برای روش فوق مقدار J عدد ۰,۰۳۲۱۴ را نشان می‌دهد.



شکل ۳-۴ عملکرد روش کنترل مقاوم غیرخطی با اعمال عدم قطعیت



شکل ۴-۴ سیگنال کنترل (گشتاور) روش کنترل مقاوم غیر خطی

فصل پنجم

کنترل ولتاژ ربات‌های الکتریکی

❖ مقدمه

❖ معرفی استراتژی کنترل ولتاژ

❖ کنترل ولتاژ مقاوم

۵-۱- مقدمه

همان طور که در فصل اول اشاره شد، استراتژی کنترل گشتاور در پیاده سازی عملی با مشکلاتی اساسی مواجه است. در مقابل این استراتژی، استراتژی کنترل ولتاژ مطرح شده است که معایب کنترل گشتاور را ندارد. مهم ترین مزیت استراتژی کنترل ولتاژ این است که در طراحی قانون کنترل آن نیازی به مدل ربات نداریم. زیرا جریان موتور حاوی تمام تأثیرات غیرخطی بار (ربات) است. در نتیجه، با فیدبک گرفتن از جریان موتور دینامیک غیر خطی پیچیده ربات را در اختیار خواهیم داشت. در این فصل این استراتژی را معرفی می کنیم و دلایل برتری آن را نسبت به کنترل گشتاور تشریح خواهیم کرد. نتایج شبیه سازی تأییدی بر عملکرد صحیح این استراتژی است. سپس به کمک روش کنترل مقاوم غیرخطی که در فصل پیش معرفی شد، به طراحی کنترل کننده مقاوم مبتنی بر کنترل ولتاژ خواهیم پرداخت و پایداری کل سیستم (ربات، موتورها و کنترل کننده) را اثبات خواهیم کرد. در این فصل برای طراحی کنترل ولتاژ مقاوم ربات و موتورها را به عنوان یک سیستم یکپارچه در نظر می گیریم. اما در فصل ۶ طراحی کنترل ولتاژ مقاوم را بر مبنای کنترل موتور انجام می دهیم و نتایج این دو طراحی را مقایسه خواهیم نمود.

۵-۲- معرفی استراتژی کنترل ولتاژ

موتورهای DC در صنعت کاربردهای فراوانی دارند. دلیل آن نیز قابلیت کنترل پذیری بالای آنها می باشد. معادله حرکت موتور DC مغناطیس دائم به صورت زیر است:

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + r\tau_l = \tau \quad (۱-۵)$$

J_m ممان اینرسی موتور و B_m ضریب میرایی می‌باشند. r ضریب چرخ دنده است. θ_m موقعیت زاویه‌ای موتور، τ گشتاور تولید شده توسط موتور و τ_l گشتاور بار موتور می‌باشند که از معادله دینامیکی ربات به صورت زیر بدست می‌آید.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau_l \quad (2-5)$$

رابطه بین موقعیت موتور و متغیرهای ربات به صورت زیر است:

$$q = r\theta_m \quad (3-5)$$

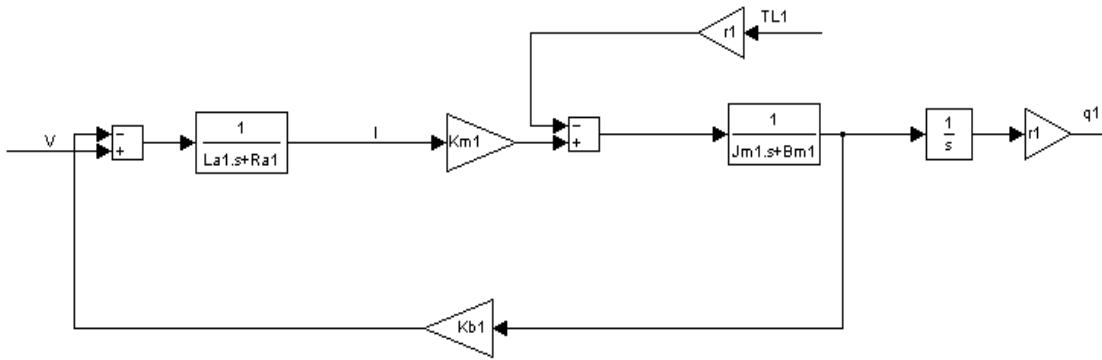
معادله kvl در مدار الکتریکی موتور را می‌توان به صورت زیر نوشت:

$$v = Ri + L \frac{di}{dt} + k_b \frac{d\theta_m}{dt} \quad (4-5)$$

که در آن R, L, k_b و R به ترتیب ثابت ضد محرکه القایی، اندوکتانس و مقاومت موتور و i و v جریان و ولتاژ ترمینال موتور می‌باشند. با توجه به رابطه (2-5) می‌توانیم (3-5) را بر حسب متغیر مفصل ربات به صورت زیر بنویسیم.

$$v = Ri + L \frac{di}{dt} + \frac{k_b}{r} \frac{dq}{dt} \quad (5-5)$$

با استفاده از روابط فوق می‌توان بلوک دیاگرام زیر را برای یک موتور و بار آن که یک بازوی ربات می‌باشد به صورت زیر رسم کرد.



شکل ۵-۱ بلوک دیاگرام یک موتور و بار آن

در شکل فوق $TL1$ درایه اول بردار گشتاور τ در معادله دینامیکی ربات (۳-۵۱) می‌باشد. همان طور که گفتیم در روش کنترل ولتاژ، کنترل ربات به کنترل موتور تبدیل می‌شود و هدف کنترل موقعیت زاویه‌ای موتور است و سیگنال کنترل، ولتاژ ترمینال موتور می‌باشد. بنابراین، معادله دینامیکی سیستم همان معادله kvl (۴-۵) می‌باشد. باید یک قانون کنترل ارائه دهیم تا خطای ردگیری موقعیت مفصل را به صفر برساند. قانون کنترل زیر را در نظر بگیرید:

$$v = Ri + L \frac{di}{dt} + \frac{k_p}{r} u \quad (۶-۵)$$

$$u = \dot{q}_d + k_p(q_d - q) \quad (۷-۵)$$

باید توجه داشت که در این قانون کنترل نگران دینامیک ربات نیستیم. چون با فیدبک گرفتن از جریان موتور تمام دینامیک های غیر خطی ربات در اختیار خواهند بود و با حذف شدن جریان توسط قانون کنترل فوق، معادله حلقه بسته مستقل از دینامیک غیر خطی پیچیده ربات خواهد شد. دلیل این ادعا با توجه به معادله زیر روشن تر می‌شود:

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + r\tau_l = \tau = K_m I \quad (۸-۵)$$

بنابراین جریان موتور حاوی تأثیرات دینامیکی ربات و موتور است [۳۸]. با استفاده از روابط (۵-۵) - (۵-۷) دینامیک حلقه بسته به صورت زیر در می‌آید:

$$\dot{q} = \dot{q}_d + k_p(q_d - q) \quad (۹-۵)$$

بنابراین می‌توان نتیجه گرفت که:

$$\dot{e} + k_p e = 0 \quad (۱۰-۵)$$

که در آن

$$e = q_d - q \quad (۱۱-۵)$$

بنابراین، اگر k_p مثبت باشد، با گذشت زمان خطا به سمت صفر میل می‌کند. همان طور که ملاحظه شد، قانون کنترل (۴-۶) خطای ردگیری را به صفر می‌رساند. مقایسه این قانون کنترل (روش کنترل ولتاژ) با قانون کنترل (۴-۲) (روش کنترل گشتاور) به وضوح سادگی روش ارائه شده را نشان می‌دهد. برای پیاده سازی قانون کنترل (۴-۲) نیاز به مدل پیچیده ربات یعنی ماتریس های M, C, G داریم. به دست آوردن این مدل کاری وقت گیر و دشوار می‌باشد. در حالی که برای پیاده سازی قانون کنترل روش کنترل ولتاژ، به پارامترهای ساده‌ای همچون مقاومت، اندوکتانس و ثابت ضد محرکه القایی نیاز داریم. بنابراین، حجم محاسبات کنترل کننده و فضای حافظه مورد نیاز کمتر می‌شود و سرعت پاسخ دهی سیستم بیشتر می‌شود. نکته مهم دیگر در تمایز این دو روش، فیدبک‌های مورد نیاز آنها می‌باشد. همان طور که روابط (۵-۶) و (۵-۷) نشان می‌دهند، پیاده سازی قانون کنترل روش کنترل ولتاژ به فیدبک های جریان، مشتق جریان و موقعیت زاویه‌ای موتور نیاز دارد. در حالی که پیاده سازی قانون کنترل روش کنترل گشتاور به فیدبک‌های موقعیت و سرعت مفاصل نیاز دارد. نکته بسیار مهم دیگر آن است که می‌توان در مدلی که برای موتور در نظر می‌گیریم از اندوکتانس آن صرف نظر کنیم یا اینکه جمله $L \frac{di}{dt}$ را به عنوان دینامیک

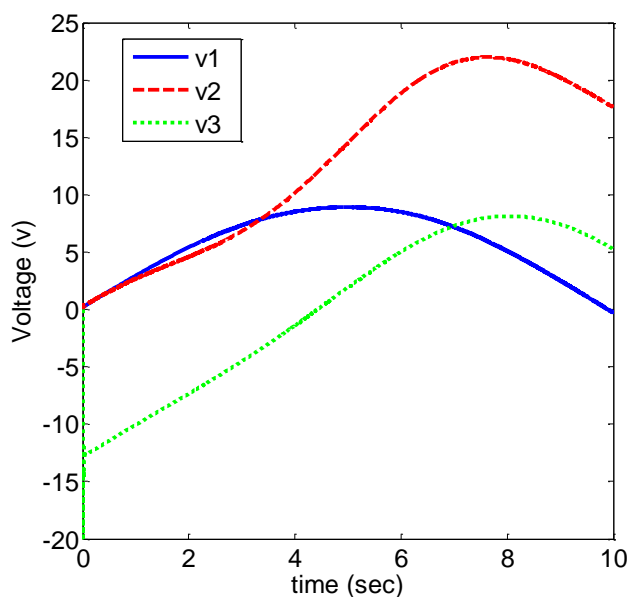
مدل نشده در نظر بگیریم که در این صورت نیاز به فیدبک مشتق جریان نداریم. در فصل بعد از این تکنیک استفاده خواهیم کرد.

نتایج شبیه سازی ها: پارامترهای مربوط به ربات در فصل ۳ ارائه شده است. جدول ۱-۵ پارامترهای موتورها را نشان می‌دهد. مسیر مطلوب همان مسیر فصل ۴ می‌باشد. مقدار k_p را عدد ۱۰ انتخاب می‌کنیم. شکل ۲-۵ سیگنال ولتاژ موتورها را نشان می‌دهد.

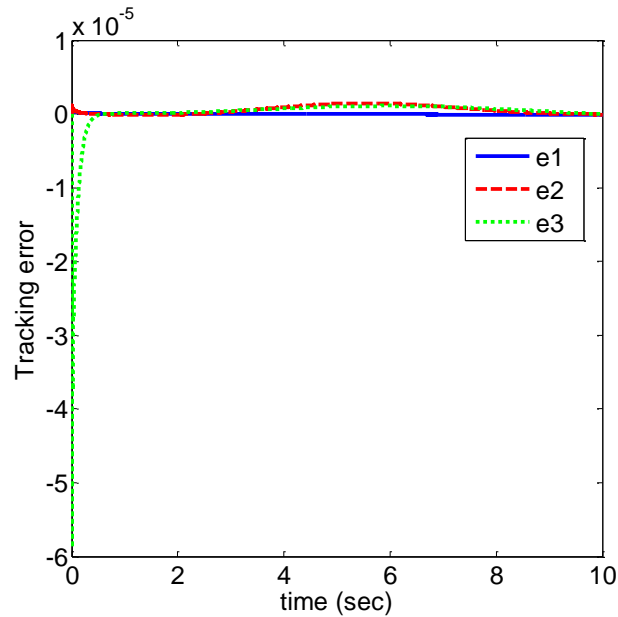
موتور	R	L	k_m	J_m	B_m	r	u_{\max}
۳ و ۲ و ۱	1.26	0.001	0.36	2×10^{-4}	0.001	0.01	40

جدول ۱-۵ پارامترهای موتور

همان طور که ملاحظه می‌شود سیگنالی نرم است، نوسانات شدید ندارد و در محدوده کاری موتورها قرار دارد. خطاهای ردگیری در حالتی که عدم قطعیت نداریم در شکل ۳-۵ رسم شده‌اند.



شکل ۲-۵ سیگنال ولتاژ موتورها در روش کنترل ولتاژ بدون عدم قطعیت



شکل ۳-۵ خطای ردگیری روش کنترل ولتاژ بدون عدم قطعیت

۲-۵- کنترل ولتاژ مقاوم

اکنون با استفاده از روش کنترل مقاوم غیرخطی که در فصل قبل معرفی شد به طراحی کنترل مقاوم با استفاده از استراتژی کنترل ولتاژ می‌پردازیم. مدل ربات را به صورت زیر در نظر بگیرید:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (۱۲-۵)$$

که پارامترهای آن در فصل ۳ معرفی شده اند. معادلات موتورها نیز به صورت زیر می باشند:

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m + r\tau = K_m I \quad (۱۳-۵)$$

$$V = RI + L \frac{dI}{dt} + K_b \dot{\theta}_m \quad (۱۴-۵)$$

روابط فوق ماتریسی هستند. I و V بردار جریان و ولتاژ موتورها هستند. r ، J_m و B_m ماتریس‌های قطری ضریب چرخ دنده، ممان اینرسی و ضریب میرایی موتورها می‌باشند. K_m ماتریس قطری ثوابت

گشتاور است که با K_b (ماتریس قطری ثوابت ضد محرکه القایی) برابر است. R و L ماتریس‌های قطری مقاومت و اندوکتانس موتورها می‌باشد. با استفاده از (۳-۵) و روابط فوق، پس از اندکی عملیات ریاضی خواهیم داشت [۷۵]:

$$RK_m^{-1}(J_m r^{-1} + rD(q))\ddot{q} + (RK_m^{-1}B_m r^{-1} + RK_m^{-1}rC(q, \dot{q}) + K_b r^{-1})\dot{q} + RK_m^{-1}rG(q) = \quad (۱۵-۵)$$

رابطه فوق را می‌توان به صورت ساده تر زیر نوشت

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + W(q) = V \quad (۱۶-۵)$$

$$M(q) = RK_m^{-1}(J_m r^{-1} + rD(q)) \quad (۱۷-۵)$$

$$N(q, \dot{q}) = RK_m^{-1}B_m r^{-1} + RK_m^{-1}rC(q, \dot{q}) + K_b r^{-1} \quad (۱۸-۵)$$

$$W(q) = RK_m^{-1}rG(q) \quad (۱۹-۵)$$

برای کامل کردن مدل سیستم و در نظر گرفتن عدم قطعیت‌هایی از قبیل دینامیک مدل نشده (مانند اصطکاک و ...) و اغتشاش خارجی، مدل زیر پیشنهاد می‌دهیم

$$M(q)\ddot{q} + H(q, \dot{q}) = V \quad (۲۰-۵)$$

$$H(q, \dot{q}) = N(q, \dot{q})\dot{q} + W(q) + d \quad (۲۱-۵)$$

d بیانگر دینامیک مدل نشده و اغتشاش خارجی می‌باشد. می‌توان رابطه (۲۰-۵) را با توجه به مدل نامی به صورت زیر نوشت:

$$\hat{M}(q)\ddot{q} + \hat{H}(q, \dot{q}) + V_d = V \quad (۲۲-۵)$$

$\hat{M}(q)$ و $\hat{H}(q, \dot{q})$ مدل نامی سیستم را نشان می‌دهند و ساختاری همانند $M(q)$ و $H(q, \dot{q})$

دارند، ولی مقادیر پارامترهای آنها متفاوت می‌باشد. عبارت متغیر با زمان V_d شامل عدم قطعیت‌ها اعم از عدم قطعیت پارامتری، دینامیک مدل نشده و اغتشاش خارجی است. با توجه به (۲۰-۵) و (۲۱-۵) می‌توان نتیجه گرفت که V_d کران بالا دارد. چون می‌توان نوشت

$$(M(q) - \hat{M}(q))\ddot{q} + (H(q, \dot{q}) - \hat{H}(q, \dot{q})) = V_d \quad (23-5)$$

بنابراین V_d کران بالایی به صورت زیر دارد

$$\|V_d\| \leq \|M(q) - \hat{M}(q)\| \cdot \|\ddot{q}\| + \|H(q, \dot{q}) - \hat{H}(q, \dot{q})\| \quad (24-5)$$

$\|\cdot\|$ نشان دهنده نرم ۲ یک بردار یا ماتریس می‌باشد. اکنون قانون کنترل زیر را در نظر بگیرید

$$\hat{M}(q)u + \hat{H}(q, \dot{q}) = V \quad (25-5)$$

با جایگذاری V از رابطه (۲۲-۵) در (۲۵-۵)، معادله حلقه بسته به صورت زیر بدست می‌آید

$$\ddot{q} + \hat{M}^{-1}(q)V_d = u \quad (26-5)$$

برای ردگیری مسیر مطلوب، u را به صورت زیر پیشنهاد می‌دهیم

$$u = \ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) + u_r \quad (27-5)$$

u_r را برای غلبه بر عدم قطعیت‌ها، به قانون کنترل اضافه نموده‌ایم و در ادامه مقدار آن را بدست خواهیم

آورد. با جایگذاری u از رابطه (۲۷-۵) در (۲۶-۵)، معادله حلقه بسته به صورت زیر بدست می‌آید

$$\ddot{q}_d - \ddot{q} + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) = \lambda - u_r \quad (28-5)$$

که در آن

$$\lambda = \hat{M}^{-1}(q)V_d \quad (29-5)$$

با توجه به رابطه (۵-۱۷)، $M(q)$ به صورت زیر محدود می‌شود

$$\|M(q)\| < \|RK_m^{-1}J_m r^{-1}\| + \|RK_m^{-1}r\| \cdot \|D(q)\| \quad (۵-۳۰)$$

ماتریس اینرسی $D(q)$ به ازای تمام q کران بالا دارد [۱]. بنابراین، با توجه به رابطه اخیر می‌توان گفت

که $M(q)$ نیز کران بالا دارد. چون ساختار $\hat{M}(q)$ مانند $M(q)$ می‌باشد، می‌توان نتیجه گرفت که $\hat{M}(q)$

نیز کران بالا دارد. بنابر این $\hat{M}^{-1}(q)$ کران بالایی مانند C دارد. چون طبق (۵-۲۴) V_d کراندار است، λ

کراندار است. اکنون با استفاده از روش مستقیم لیاپانوف u_r را بدست می‌آوریم. ابتدا معادلات حالت

سیستم در فضای ردگیری را بدست می‌آوریم. با استفاده از (۵-۲۸) و تعریف $e = q_d - q$ خواهیم داشت

$$\ddot{e} + K_d \dot{e} + K_p e = \lambda - u_r \quad (۵-۳۱)$$

با تعریف متغیرهای حالت به صورت

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (۵-۳۲)$$

معادلات حالت به صورت زیر در می‌آید:

$$\dot{x} = Ax + Bu_s \quad (۵-۳۳)$$

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (۵-۳۴)$$

$$u_s = \lambda - u_r \quad (۵-۳۵)$$

تابع لیاپانوف را به صورت زیر پیشنهاد می‌دهیم

$$F(x) = x^T P x \quad (۵-۳۶)$$

که P یک ماتریس متقارن مثبت معین است. مشتق زمانی $F(x)$ برابر است با

$$\dot{F}(x) = x^T(A^T P + PA)x + 2x^T P B u_s = -x^T Q x + 2x^T P B u_s \quad (37-5)$$

اگر A هرویتز باشد، P از معادله ریکاتی زیر محاسبه می‌شود که در آن Q یک ماتریس متقارن مثبت معین است

$$A^T P + PA + Q = 0 \quad (38-5)$$

با توجه به رابطه (37-5) برای اینکه سیستم پایدار مجانبی باشد، باید داشته باشیم

$$x^T P B u_s = x^T P B (\lambda - u_r) \leq 0 \quad (39-5)$$

فرض کنید $\|\lambda\| \leq \varphi$ باشد. در این صورت نشان خواهیم داد که اگر u_r را به صورت

$$u_r = \begin{cases} \frac{(x^T P B)^T \varphi}{\|x^T P B\|} & \|x^T P B\| \neq 0 \\ 0 & \|x^T P B\| = 0 \end{cases} \quad (40-5)$$

انتخاب کنیم، رابطه (39-5) برقرار خواهد شد و پایداری سیستم اثبات می‌شود.

اثبات: u_r از معادله (40-5) را در (39-5) جایگذاری می‌کنیم.

$$x^T P B \left(\lambda - \frac{(x^T P B)^T \varphi}{\|x^T P B\|} \right) = (x^T P B \lambda - \|x^T P B\| \varphi) \leq 0 \quad (41-5)$$

از (41-5) می‌توان نتیجه گرفت که

$$x^T P B \lambda \leq \|x^T P B\| \varphi \quad (42-5)$$

اگر نامساوی فوق برقرار باشد پایداری سیستم تضمین شده است. با توجه به نامساوی کوشی-شوارتز می‌توان نوشت

$$x^T P B \lambda \leq \|x^T P B\| \cdot \|\lambda\| \quad (43-5)$$

بنابراین حداکثر مقدار $x^T PB \lambda$ عبارتست از $\|\lambda\| \|x^T PB\|$. اگر این مقدار از $\|x^T PB\| \varphi$ کمتر باشد، نامساوی (۴۲-۵) برقرار می‌شود و پایداری سیستم اثبات می‌شود. بنابراین باید داشته باشیم

$$\|x^T PB\| \cdot \|\lambda\| \leq \|x^T PB\| \varphi \quad (۴۴-۵)$$

نامساوی فوق نیز با توجه به فرض $\|\lambda\| \leq \varphi$ همواره برقرار است. اکنون باید از صحت این فرض اطمینان حاصل کنیم و مقدار φ را بدست آوریم. چون برای تعیین u_r در (۴۰-۵) مقدار φ باید معلوم باشد. از (۲۳-۵) و (۲۹-۵) نتیجه می‌شود که

$$\lambda = \widehat{M}^{-1}(q)(M(q) - \widehat{M}(q))\ddot{q} + \widehat{M}^{-1}(q)(H(q, \dot{q}) - \widehat{H}(q, \dot{q})) \quad (۴۵-۵)$$

بنابراین λ به صورت زیر محدود می‌شود

$$\|\lambda\| \leq \|\widehat{M}^{-1}(q)\| \|(M(q) - \widehat{M}(q))\| \|\ddot{q}\| + \|\widehat{M}^{-1}(q)\| \|H(q, \dot{q}) - \widehat{H}(q, \dot{q})\| \quad (۴۶-۵)$$

اگر فرض کنیم

$$\|(M(q) - \widehat{M}(q))\| \leq a, \|H(q, \dot{q}) - \widehat{H}(q, \dot{q})\| \leq b, \|\widehat{M}^{-1}(q)\| \leq c \quad (۴۷-۵)$$

آنگاه باند عدم قطعیت به صورت زیر در می‌آید

$$\|\lambda\| \leq ca \|\ddot{q}\| + cb = \varphi \quad (۴۸-۵)$$

در رابطه فوق $\|\ddot{q}\|$ باید اندازه گیری شود و a, b, c معلوم هستند. اگر $\|\ddot{q}\|$ قابل اندازه گیری نباشد می‌توانیم آن را محاسبه کنیم. با استفاده از (۲۸-۵) و (۴۵-۵) داریم:

$$\lambda = \widehat{M}^{-1}(M - \widehat{M})(\ddot{q}_d + K_d \dot{e} + K_p e + u_r) + \widehat{M}^{-1}(H - \widehat{H}) \quad (۴۹-۵)$$

بنابراین λ به صورت زیر محدود می‌شود:

$$\|\lambda\| \leq ca\|(\ddot{q}_d + K_d\dot{e} + K_p e)\| + ca\|u_r\| + cb \quad (50-5)$$

تذکر: برای کاهش لرزش سیگنال کنترل می‌توانیم u_r در (5-40) را به صورت زیر اصلاح کنیم [75].

$$u_r = \begin{cases} \frac{(x^{TPB})^T \varphi}{\|x^{TPB}\|} & \|x^{TPB}\| \geq \varepsilon \\ \frac{(x^{TPB})^T \varphi}{\varepsilon} & \|x^{TPB}\| < \varepsilon \end{cases} \quad (51-5)$$

ε مقدار کوچکی است.

نتایج شبیه سازی ها: در شبیه سازی از قانون کنترل (5-25) با u_r که در رابطه (5-51) داده شده است

به ازای $\varepsilon = 0.1$ استفاده شده است. مقدار φ در رابطه (5-51) از (5-48) و (5-47) بدست می‌آید. $\|M\|$

در شکل 5-4 رسم شده است. همان طور که مشاهده می‌شود حداکثر مقدار $\|M\|$ ، 2,5 است. بنابراین

برای a در (5-47) حداکثر مقدار 0,5 بدست می‌آید. برای b در (5-47) مقدار 10 بدست می‌آید، زیرا

حداکثر مقدار $\|H\|$ عدد 50 می‌باشد. نمودار $\|H\|$ در شکل 5-5 نشان داده شده است. برای C حداکثر

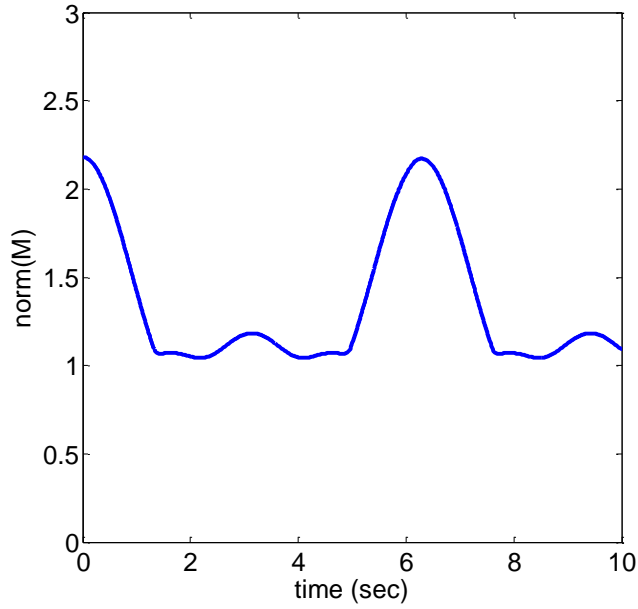
مقدار 1,2 بدست می‌آید. مقادیر k_p و k_d همان مقادیر فصل قبل می‌باشند. مدل ریاضی بازوی ماهر

کروی در فصل 3 معرفی شده است. پارامترهای موتورها نیز همان پارامترهای جدول 5-1 می‌باشند. گشتاور

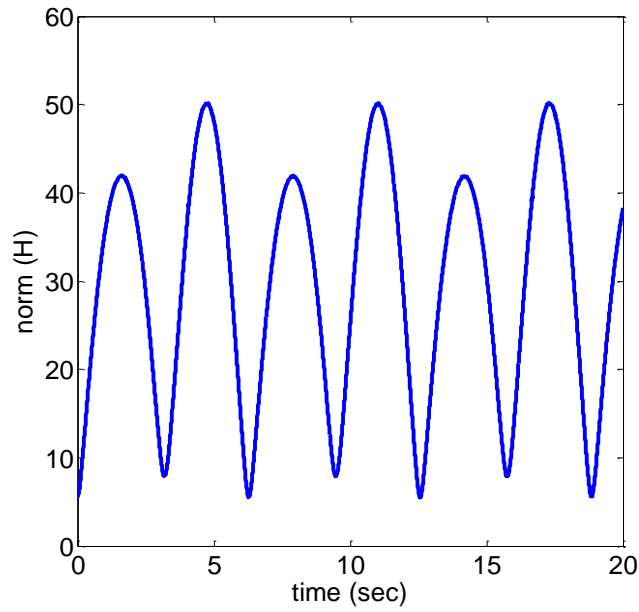
اصطکاک نیز گشتاور رابطه (4-40) می‌باشد. در شبیه سازی مدل سیستم را به صورت

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + W(q) + RK_m^{-1}r\tau_f + V_{dist} = V \quad (52-5)$$

در نظر می‌گیریم که در آن V_{dist} بردار اغتشاش خارجی است و τ_f گشتاور اغتشاش است.



شکل ۴-۵ نمودار $\|M\|$



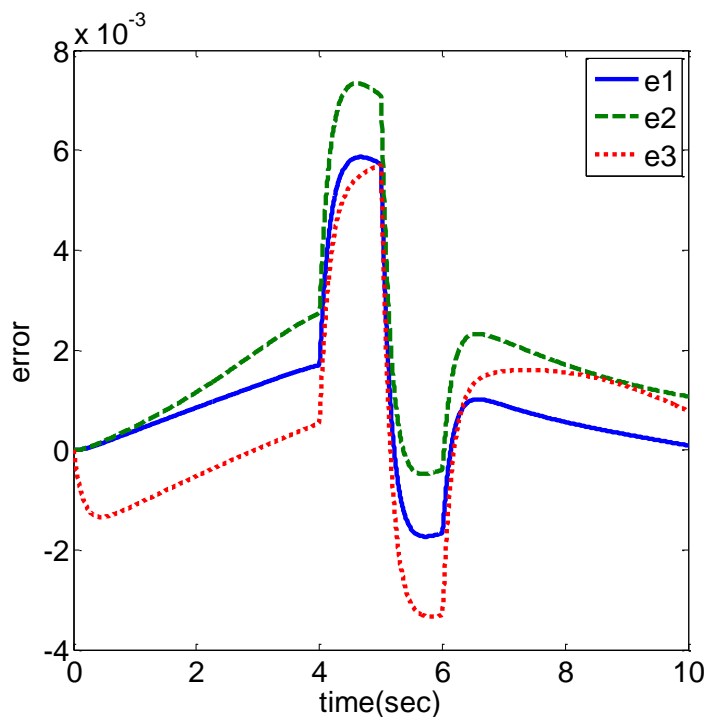
شکل ۵-۵ نمودار $\|H\|$

مانند فصل قبل، دامنه اغتشاش خارجی را ۲۰ درصد دامنه سیگنال کنترل در شرایطی که عدم قطعیت نداریم، انتخاب می‌کنیم. همانطور که در شکل ۲-۵ مشاهده می‌شود دامنه سیگنال ولتاژ بدون اعمال عدم

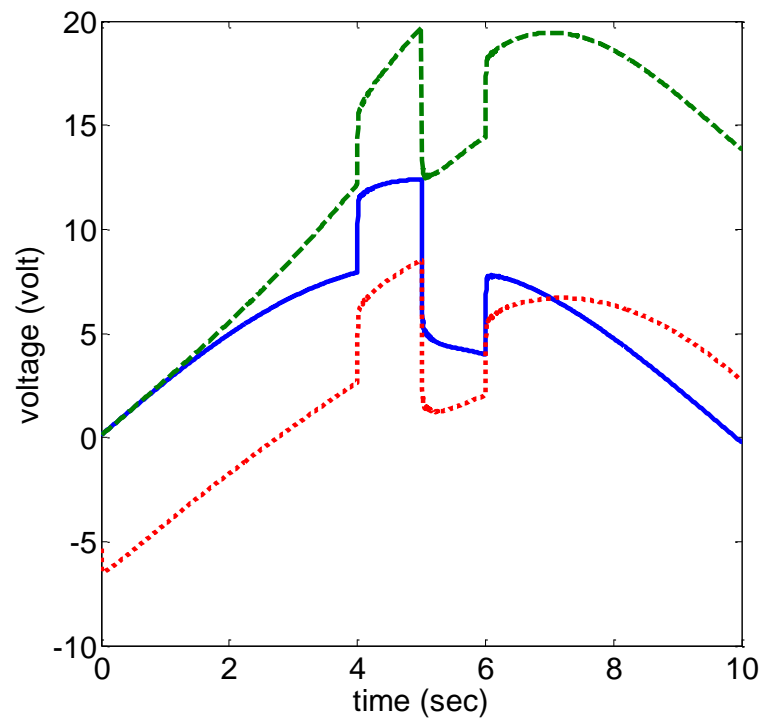
قطعیت، حداکثر ۲۵ ولت است. بنابراین، اغتشاش خارجی هر مفصل را به صورت زیر به سیستم اعمال می‌کنیم:

$$v_{dist} = 5s(t - 4) - 10s(t - 5) + 5s(t - 6) \quad (5-53)$$

عدم قطعیت پارامتری را به صورت فصل قبل در نظر می‌گیریم. خطای ردگیری در شکل ۵-۶ نشان داده شده است. معیار عملکرد J که در رابطه (۴-۴۱) تعریف شده است، برای شکل ۵-۴ مقدار $0,006892$ را نشان می‌دهد در حالی که با استراتژی کنترل گشتاور که در فصل ۴ معرفی شد معیار عملکرد J مقدار $0,03214$ بدست می‌آید. سیگنال کنترل نیز در شکل ۵-۷ نشان داده شده است.



شکل ۵-۶ خطای ردگیری روش کنترل ولتاژ مقاوم با اعمال عدم قطعیت



شکل ۵-۷ سیگنال کنترل (ولتاژ) روش کنترل ولتاژ مقاوم با اعمال عدم قطعیت

فصل ششم

کنترل بهینه غیر خطی

❖ مقدمه

❖ کنترل بهینه گشتاور

❖ کنترل بهینه ولتاژ

۶-۱ مقدمه

در این فصل با استفاده از الگوریتم بهینه سازی پرندگان، عملکرد سیستم کنترل را در ردگیری مسیر مطلوب بهبود می‌بخشیم و مقدار بهینه‌ای برای پارامترهای قانون کنترل بدست می‌آوریم به طوری معیار عملکرد J در (۴-۴۱) کمترین مقدار را داشته باشد. همان طور که در طراحی سیستم کنترل در فصل های چهارم و پنجم دیدیم، قانون کنترل پارامترهایی دارد که مقدار برخی از آنها بستگی به نظر طراح دارد (مانند k_p و k_d). برخی دیگر از پارامترهای قانون کنترل، پارامترهای مدل سیستم هستند که ممکن است اطلاعات دقیقی از آنها در اختیار نباشد. رابطه مقدار این پارامترها و خطای ردگیری پیچیده است و نمی‌توان به طور مشخص گفت که افزایش یا کاهش این پارامترها چه تأثیری بر خطای ردگیری می‌گذارد. به همین دلیل، تعیین مقدار این پارامترها را به عهده الگوریتم بهینه سازی پرندگان می‌گذاریم.

می‌توانیم با استفاده از الگوریتم بهینه سازی پرندگان ابتدا پارامترهای سیستم را شناسایی کنیم. سپس بر مبنای مدل شناسایی شده کنترل کننده را طراحی کنیم. اما همان طور که اشاره شد، برخی از پارامترهای قانون کنترل به نظر طراح بستگی دارد. برای انتخاب بهینه این پارامترها به یک الگوریتم بهینه‌سازی نیاز داریم. در این پایان نامه این دو مرحله را باهم ادغام می‌کنیم و مقدار بهینه تمام پارامترهای مورد نیاز (پارامترهای سیستم و پارامترهای کنترل کننده) را بدست می‌آوریم.

همان طور که در فصل دوم اشاره شد، روشهای بهینه‌سازی مبتنی بر مشتق مکانیزم خوبی برای فرار از جواب های بهینه محلی ندارند. علاوه بر این، یکی از پارامترهای تأثیر گذار در عملکرد سیستم ε می باشد. در روشهای بهینه‌سازی مبتنی بر مشتق، برای تنظیم پارامترها به مشتق تابع هزینه نسبت به پارامترها نیاز داریم. مشتق گیری از تابع هزینه نسبت به ε با توجه به ناپیوسته بودن ضابطه قانون کنترل معادله تازه سازی این پارامتر را پیچیده می‌کند. به همین دلیل در این پایان نامه به یک روش بدون

مشتق نیاز داریم. الگوریتم ژنتیک و الگوریتم بهینه‌سازی پرندگان مهمترین روشهای بهینه‌سازی بدون مشتق می‌باشند. با توجه به مقایسه‌ای که در فصل ۲ بین این دو الگوریتم انجام شد، مشخص می‌شود که الگوریتم بهینه‌سازی پرندگان روش ساده‌تری می‌باشد و پیاده‌سازی آن راحت‌تر است.

در این فصل، کنترل بهینه را بر مبنای استراتژی‌های کنترل گشتاور و کنترل ولتاژ انجام خواهیم داد و نتایج را با هم مقایسه خواهیم کرد. علاوه بر این، در این فصل کنترل ولتاژ مقاوم را بر مبنای کنترل موتور می‌دهیم. در فصل ۵ موتورهای وربات را یک سیستم بزرگ در نظر گرفتیم و کنترل ولتاژ مقاوم را برای این سیستم پیچیده انجام دادیم. نتایج شبیه‌سازی این فصل نشان می‌دهد اگر کنترل ولتاژ مقاوم را بر مبنای کنترل موتور انجام دهیم خطای ردگیری کاهش چشمگیری می‌یابد.

۲-۶- کنترل بهینه گشتاور

دینامیک سیستم را به صورت زیر در نظر بگیرید:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) = \tau \quad (1-6)$$

که $\tau_f(\dot{q})$ گشتاور اصطکاک می‌باشد. می‌توان نوشت [۱]:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q}) \cdot P \quad (2-6)$$

که در آن P بردار پارامترهای ربات می‌باشد. $Y(q, \dot{q}, \ddot{q})$ در پیوست معرفی شده‌اند. با قانون کنترل مقاوم غیر خطی

$$\hat{D}(q)(\ddot{q}_a + K_a(\dot{q}_a - \dot{q}) + K_p(q_a - q)) + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) + \tau_r = \tau \quad (3-6)$$

سیستم حلقه بسته به صورت زیر در می‌آید:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) = \widehat{D}(q) \left(\ddot{q}_d + K_d(\dot{q}_d - \dot{q}) + K_p(q_d - q) \right) + \widehat{C}(q, \dot{q})\dot{q} + \widehat{G}(q) + \tau_r \quad (۴-۶)$$

با توجه به (۲-۶) می‌توان نوشت:

$$\widehat{D}(q)\ddot{q} + \widehat{C}(q, \dot{q})\dot{q} + \widehat{G}(q) = Y(q, \dot{q}, \ddot{q}).\widehat{P} \quad (۵-۶)$$

برای ساده تر شدن روابط از شناسه های توابع صرف نظر می‌کنیم. بنابراین می‌توان نوشت:

$$(D - \widehat{D})\ddot{q} + (C - \widehat{C})\dot{q} + (G - \widehat{G}) + \tau_f = \widehat{D}(\ddot{q}_d + K_d\dot{e} + K_p e) + \tau_r \quad (۶-۶)$$

با توجه به (۲-۶) و (۵-۶) می‌توان نوشت:

$$\widehat{D}(\ddot{q}_d + K_d\dot{e} + K_p e) + \tau_r = Y(q, \dot{q}, \ddot{q})(P - \widehat{P}) \quad (۷-۶)$$

با توجه به آنچه در روابط (۱۶-۴) تا (۲۴-۴) فصل ۴ گفتیم τ_r به صورت زیر بدست می‌آید:

$$\tau_r = \widehat{D}(q)u_r \quad (۸-۶)$$

$$v = \begin{cases} \frac{(x^T P B)^T \varphi}{\|x^T P B\|} & \|x^T P B\| > \varepsilon \\ \frac{(x^T P B)^T \varphi}{\varepsilon} & \|x^T P B\| \leq \varepsilon \end{cases} \quad (۹-۶)$$

$$\varphi = \|\widehat{D}^{-1}\| \cdot \|Y\| \cdot \|P - \widehat{P}\| + \|\widehat{D}^{-1}\| \|\tau_f(\dot{q})\| \quad (۱۰-۶)$$

با توجه به روابط (۳-۶)، (۵-۶) و (۸-۶) قانون کنترل به صورت:

$$Y(q, \dot{q}, e, \dot{e}, \ddot{q}_d).\widehat{P}_1 + \tau_r = \tau \quad (۱۱-۶)$$

در می‌آید که در آن \hat{P}_1 شامل بردار \hat{P} و ضرایب k_p و k_d است که می‌خواهیم توسط الگوریتم بهینه سازی پرنده‌گان مقدار آنها را بدست آوریم. تعدادی دیگر از پارامترهای قانون کنترل مربوط به τ_r می‌شوند که عبارتند از: ε و φ . سمت راست (۶-۱۰) کران بالای عدم قطعیت را نشان می‌دهد که یک کمیت اسکالر متغیر با زمان می‌باشد. با توجه به در اختیار بودن \hat{D}^{-1} ، $\|\hat{D}^{-1}\|$ قابل محاسبه است. Y در پیوست ۱ ارائه شده است و $\|Y\|$ قابل محاسبه می‌باشد. برای $\|P - \hat{P}\|$ نیز حداکثر مقدار ۴۰ را قرار می‌دهیم. بدست آوردن این مقدار مشابه بدست آوردن $\|(M(q) - \hat{M}(q))\|$ در فصل ۵ می‌باشد. با توجه به پیوست ۱ و توضیحات فوق، بردار پارامترهایی که باید الگوریتم بهینه سازی پرنده‌گان مقادیر آنها را تعیین کند به صورت زیر در می‌آید:

$$\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12}]^T \quad (۱۲-۶)$$

$$\theta_1 = I_{221} + I_{332} + I_{333} \quad (۱۳-۶)$$

$$\theta_2 = I_{112} + I_{113} - I_{332} - I_{333} + m_2 l_{c2}^2 + m_3 l_{c3}^2 \quad (۱۴-۶)$$

$$\theta_3 = 2(I_{132} + I_{133}) \quad (۱۵-۶)$$

$$\theta_4 = m_3 \quad (۱۶-۶)$$

$$\theta_5 = 2m_3 l_{c3} \quad (۱۷-۶)$$

$$\theta_6 = I_{122} + I_{123} \quad (۱۸-۶)$$

$$\theta_7 = I_{232} + I_{233} \quad (۱۹-۶)$$

$$\theta_8 = m_2 l_{c2}^2 + m_3 l_{c3}^2 + I_{222} + I_{223} \quad (۲۰-۶)$$

$$\theta_9 = g(m_2 l_{c2} + m_3 l_{c3}) \quad (۲۱-۶)$$

$$\theta_{10} = k_p \quad (22-6)$$

$$\theta_{11} = k_d \quad (23-6)$$

$$\theta_{12} = \varepsilon \quad (24-6)$$

به ازای برخی از مقادیر پارامترها، سیگنال کنترل نوسانات شدید دارد، اما سیستم پایدار است و خطای ردگیری مناسب است. در اجرای الگوریتم بهینه سازی پرندگان به این جواب‌ها هزینه‌ای بزرگ اختصاص می‌دهیم تا به عنوان بهترین جواب انتخاب نشوند. تعداد ۲۰ پرنده به صورت بردار θ در (۱۲-۶) در نظر می‌گیریم. برای پارامترهای ربات بازه‌هایی به طول ۲۰ درصد مقدار واقعی هر پارامتر حول آن پارامتر در نظر می‌گیریم. الگوریتم بهینه سازی پرندگان در این بازه‌ها جواب بهینه را جستجو می‌کند. اگر تغییراتی که الگوریتم به پارامترها اعمال می‌کند موجب شود که برخی از پارامترها خارج از بازه مجاز قرار گیرند آن پارامتر را به مرزهای بازه مربوط محدود می‌کنیم. به عنوان مثال، مقدار واقعی θ_1 برابر است با ۱۹,۹۱. بنابراین بازه مجاز برای تغییرات θ_1 بازه [۲۳,۸۹ ۱۵,۹۲] می‌باشد. اگر θ_1 از ۲۳,۸۹ بیشتر شود آن را به این مقدار محدود می‌کنیم. همین کار را برای مرز پایین انجام می‌دهیم. برای سایر پارامترها بازه‌ها را به صورت زیر در نظر می‌گیریم:

$$\theta_{10} \in [1 \ 300] \quad (25-6)$$

$$\theta_{11} \in [1 \ 150] \quad (26-6)$$

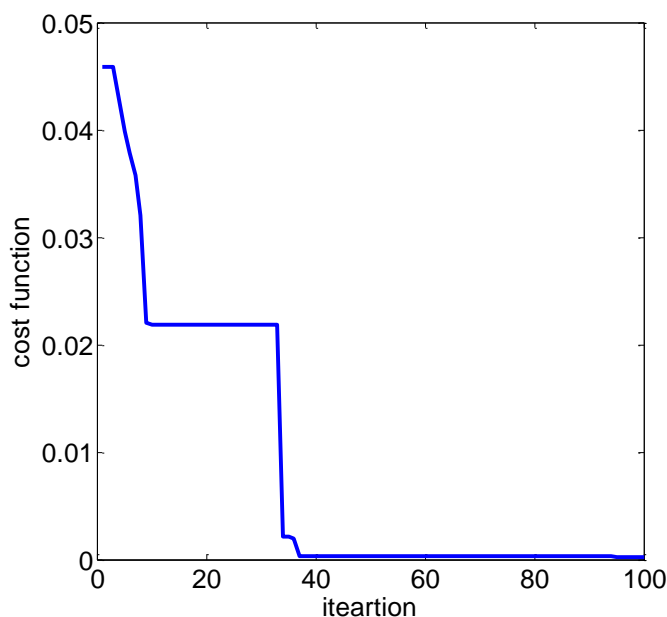
$$\theta_{12} \in [0.001 \ 1] \quad (27-6)$$

در مدل واقعی سیستم گشتاور اصطکاک وجود دارد ولی در مدلی که ما برای سیستم در نظر گرفته‌ایم از آن صرف نظر کردیم. بنابراین نمی‌توان انتظار داشت که پارامترهای ربات به سمت مقادیر واقعی خود همگرا شوند. جدول ۱-۶ مقادیر واقعی پارامترهای ربات را نشان می‌دهد.

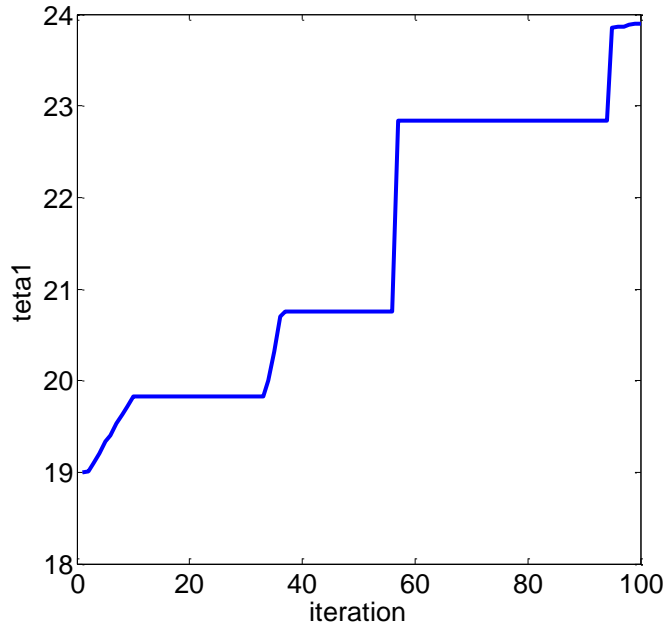
پارامتر	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
مقدار واقعی	19.91	-2.12	-0.02	14.32	11.45	-2.29	0.75	16.87	39

جدول ۱-۶ مقادیر واقعی پارامترهای ربات

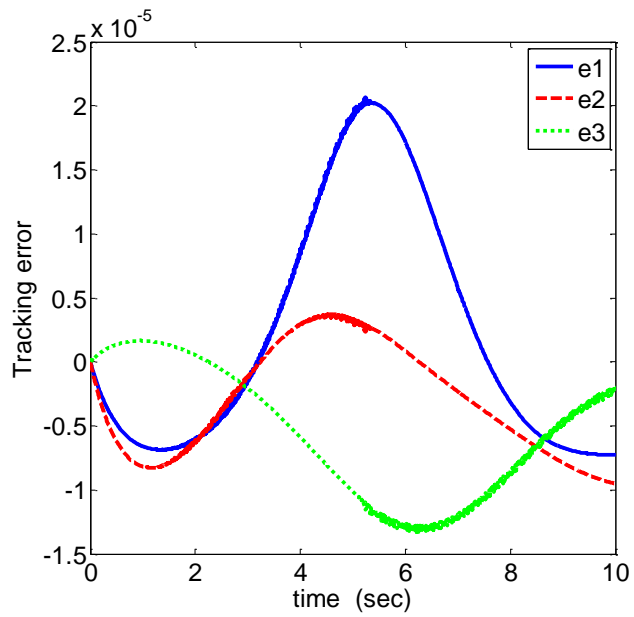
تابع هزینه‌ای که می‌خواهیم مینیمم کنیم همان تابع عملکرد رابطه (۴-۴۱) می‌باشد. الگوریتم بهینه‌سازی پرندگان را ۲۰ مرتبه اجرا می‌کنیم. کمترین (بهترین) مقدار بدست آمده برای تابع هزینه، $0,0007004417$ و بیشترین (بدترین) مقدار $0,0002881$ می‌باشد. میانگین جواب‌ها $0,0002342$ است. هر اجرای الگوریتم شامل ۱۰۰ گام می‌باشد. پارامترهای مربوط به بهترین جواب در جدول ۲-۶ نشان داده شده است. شکل ۱-۶ چگونگی کاهش تابع هزینه مربوط به بهترین جواب را نشان می‌دهد. تغییرات پارامتر θ_1 بهینه در ۱۰۰ گام شبیه‌سازی در شکل ۲-۶ نشان داده شده است. شکل ۳-۶ خطای ردگیری و شکل ۴-۶ سیگنال کنترل مربوط به بهترین جواب را نشان می‌دهد.



شکل ۱-۶ چگونگی کاهش تابع هزینه مربوط به بهترین



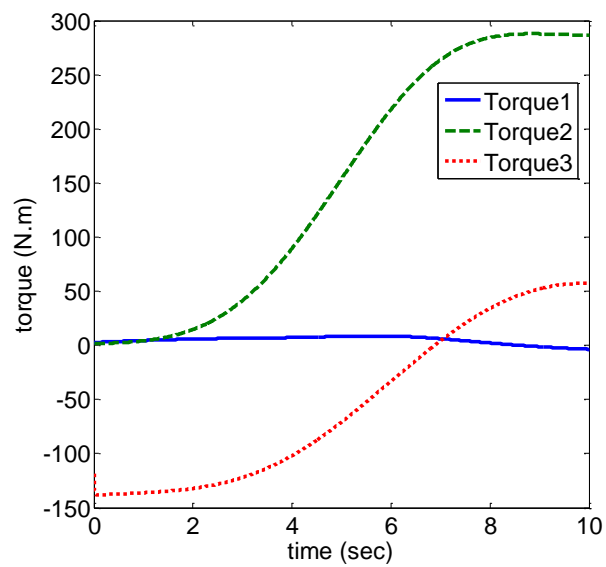
شکل ۶-۲ تغییرات پارامتر θ_1 بهینه در ۱۰۰ گام شبیه سازی



شکل ۶-۳ خطای ردگیری جواب بهینه کنترل گشتاور

پارامتر	مقدار بهینه
θ_1	23.892
θ_2	-2.5451
θ_3	-0.0024
θ_4	13.77
θ_5	-13.7472
θ_6	-2.748
θ_7	0.7672
θ_8	19.40
θ_9	46.8466
θ_{10}	294.64
θ_{11}	1.54
θ_{12}	1

جدول ۲-۶ پارامترهای بهینه در کنترل گشتاور



شکل ۴-۶ سیگنال کنترل جواب بهینه در کنترل گشتاور

۳-۶- کنترل بهینه ولتاژ

دینامیک سیستم (معادله ولتاژ موتورها) را به صورت زیر در نظر بگیرید:

$$V = RI_a + L\dot{I}_a + K_b r^{-1} \dot{q} + \varphi \quad (28-6)$$

که در آن R, L, K_b به ترتیب ماتریس های ثوابت ضد محرکه القایی، اندوکتانس ها، مقاومت موتورها و I_a و V بردار جریان و ولتاژ ترمینال موتورها می باشند. قانون کنترل را به صورت زیر پیشنهاد می دهیم:

$$V = \widehat{R}I_a + \widehat{k}_b \widehat{r}^{-1} (\dot{q}_d + k_p(q_d - q)) + V_r \quad (29-6)$$

q_d مسیر مطلوب می باشد. همان طور که ملاحظه می شود در این قانون کنترل از $L \frac{di}{dt}$ صرف نظر شده است. به عبارت دیگر آن را به عنوان دینامیک مدل نشده در نظر گرفته ایم. v_r را برای غلبه بر عدم قطعیت ها به قانون کنترل اضافه نموده ایم. معادله حلقه بسته با توجه به (۲۸-۶) و (۲۹-۶) به صورت زیر بدست می آید:

$$RI_a + L\dot{I}_a + K_b r^{-1} \dot{q} + \varphi = \widehat{R}I_a + \widehat{k}_b \widehat{r}^{-1} (\dot{q}_d + k_p(q_d - q)) + V_r \quad (30-6)$$

به عبارت دیگر می توان چنین نوشت:

$$\dot{e} + k_p e = \widehat{r} \widehat{k}_b^{-1} [(R - \widehat{R})I_a + L\dot{I}_a + (k_b r^{-1} - \widehat{k}_b \widehat{r}^{-1}) \dot{q} - V_r] \quad (31-6)$$

با تعاریف

$$\eta = \widehat{r} \widehat{k}_b^{-1} [(R - \widehat{R})I_a + L\dot{I}_a + (k_b r^{-1} - \widehat{k}_b \widehat{r}^{-1}) \dot{q}] \quad (32-6)$$

$$\varphi = -\widehat{r} \widehat{k}_b^{-1} V_r \quad (33-6)$$

خواهیم داشت:

$$\dot{e} + k_p e = \eta + \varphi \quad (34-6)$$

متغیر حالت را

$$x = e \quad (35-6)$$

تعریف می کنیم. با توجه به آنچه در روابط (۱۷-۴) تا (۲۴-۴) فصل ۴ گفتیم φ برابر خواهد شد با:

$$\varphi = \begin{cases} \frac{-(x^T k_p)^T \lambda}{\|x^T k_p\|} & \|x^T k_p\| > \varepsilon \\ \frac{-(x^T k_p)^T \lambda}{\varepsilon} & \|x^T k_p\| \leq \varepsilon \end{cases} \quad (36-6)$$

در رابطه فوق λ باند عدم قطعیت می باشد ($|\eta| < \lambda$) که با توجه به (۳۲-۶) به صورت زیر بدست می آید:

$$\lambda = \hat{r} \hat{k}_b^{-1} (\|R - \hat{R}\| \|I_a\| + \|L\| \|I_a\| + \|k_b r^{-1} - \hat{k}_b \hat{r}^{-1}\| \|\dot{q}\|) \quad (37-6)$$

با توجه به (۲۹-۶) و (۳۶-۶) نتیجه می گیریم که پارامترهایی که باید توسط الگوریتم بهینه سازی پرندگان

مقدار آنها تعیین شوند عبارتند از: ε , k_p , \hat{R} , \hat{k}_b و \hat{r} . هر سه موتور را یکسان فرض می کنیم. بنابراین

$$k_p = k_p I_{3 \times 3} \quad (38-6)$$

\hat{r} و \hat{k}_b , \hat{R} نیز همین فرم را دارند. بردار پارامترهایی که باید الگوریتم بهینه سازی پرندگان مقادیر آنها را

تعیین کند به صورت زیر در می آید:

$$\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^T \quad (39-6)$$

$$\theta_1 = \hat{R} \quad (40-6)$$

$$\theta_2 = \hat{k}_b \quad (41-6)$$

$$\theta_3 = \hat{r} \quad (42-6)$$

$$\theta_4 = k_p \quad (43-6)$$

$$\theta_5 = \varepsilon \quad (44-6)$$

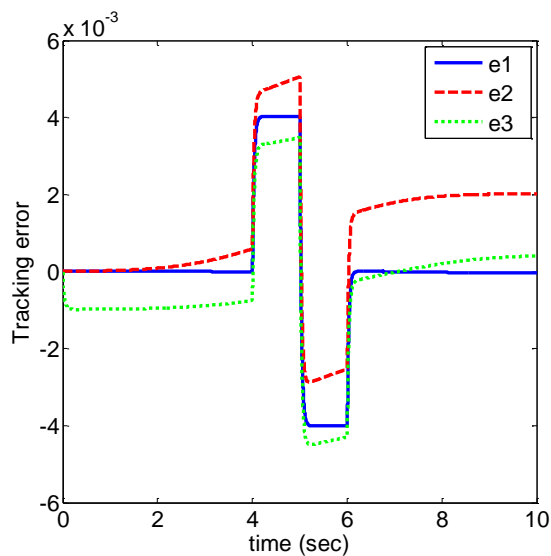
برای پارامترهای موتور بازه هایی به طول ۲۰ درصد مقدار واقعی هر پارامتر حول آن پارامتر در نظر

می‌گیریم. برای سایر پارامترها بازه هایی به صورت زیر در نظر می‌گیریم:

$$\theta_4 \in [1 \ 200] \quad (45-6)$$

$$\theta_5 \in [0.001 \ 1] \quad (46-6)$$

شکل ۵-۶ خطای ردگیری یکی از جوابهای تصادفی اولین تکرار الگوریتم بهینه سازی پرندگان را نشان می‌دهد. معیار عملکرد J برای این جواب ۰,۰۰۳۵ می‌باشد که نسبت به شکل ۴-۵ مقدار بسیار کمتری است. بنابراین، اگر استراتژی کنترل ولتاژ را بر مبنای کنترل موتور انجام دهیم، خطای ردگیری بسیار کمتر از حالتی است که موتور و ربات را با هم در نظر بگیریم.

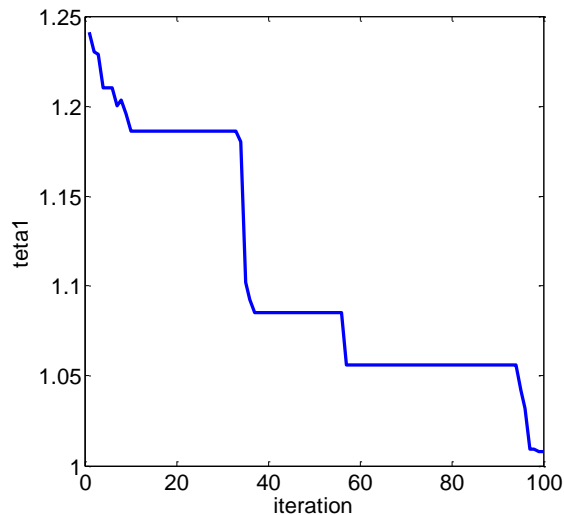


شکل ۵-۶ خطای ردگیری یکی از جوابهای تصادفی اولین تکرار الگوریتم بهینه سازی پرندگان

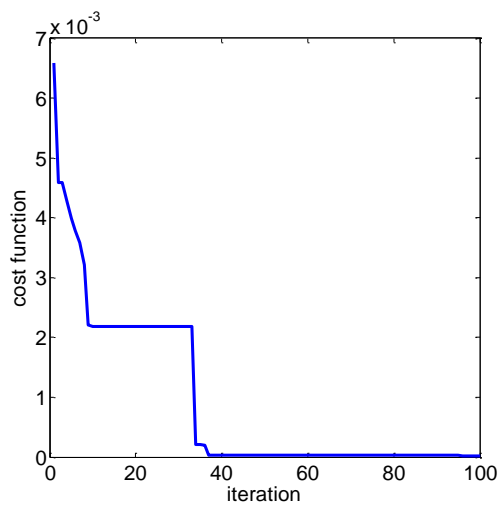
الگوریتم بهینه سازی پرندگان را ۲۰ مرتبه اجرا می کنیم. کمترین (بهترین) مقدار بدست آمده برای تابع هزینه، $0,00002454$ و بیشترین (بدترین) مقدار $0,00007275$ می باشد. میانگین جوابها $0,0004523$ است. هر اجرای الگوریتم شامل ۱۰۰ گام می باشد. پارامترهای واقعی موتور همان پارامترهای فصل قبل است. پارامترهای مربوط به بهترین جواب در جدول ۳-۶ نشان داده شده است. شکل ۶-۶ چگونگی کاهش تابع هزینه مربوط به بهترین جواب را نشان می دهد. تغییرات پارامتر θ_1 بهینه در ۱۰۰ گام شبیه سازی در شکل ۷-۶ نشان داده شده است.

موتور	R	k_m	r	k_p	ε
۳و۲و۱	1.008	0.432	0.012	160	0.001

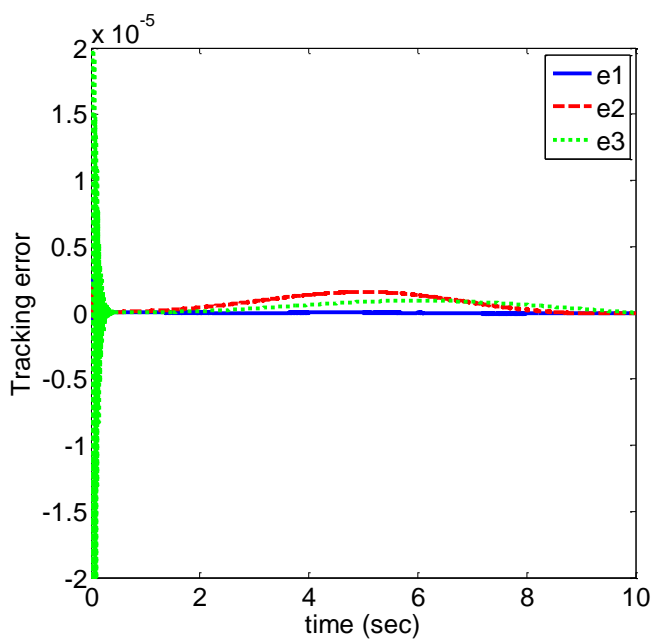
جدول ۳-۶ پارامترهای بهینه در کنترل ولتاژ



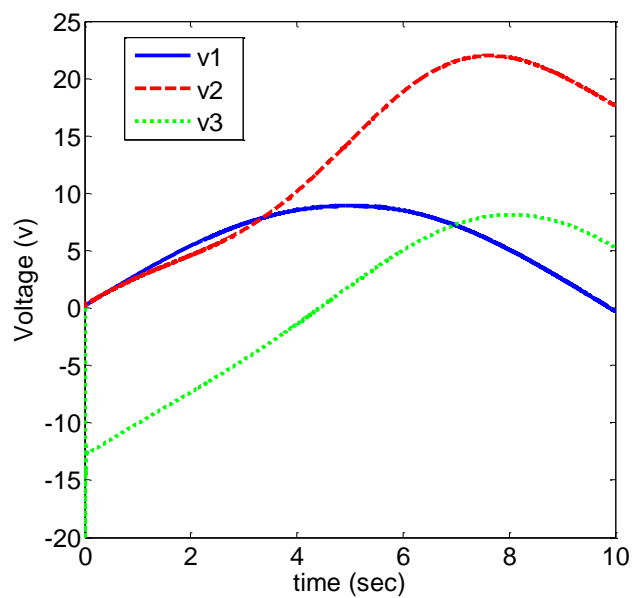
شکل ۶-۶ تغییرات پارامتر θ_1 بهینه در کنترل ولتاژ



شکل ۶-۷ کاهش تابع هزینه جواب بهینه در کنترل ولتاژ



شکل ۶-۸ خطای ردگیری جواب بهینه در کنترل ولتاژ



شکل ۶-۹ سیگنال کنترل جواب بهینه در کنترل ولتاژ

فصل هفتم

نتیجه گیری

در این پایان نامه پس از مرور مزایا و معایب انواع روش‌های کنترل مقاوم ربات‌ها، روش کنترل غیرخطی مقاوم برای کنترل ربات مورد مطالعه در فضای مفصلی انتخاب شد. سپس دو استراتژی کنترل ولتاژ و کنترل گشتاور را مقایسه نمودیم و با توجه به برتری‌های استراتژی کنترل ولتاژ نسبت به کنترل گشتاور، استراتژی کنترل ولتاژ را برگزیدیم. در فصل دوم الگوریتم بهینه سازی پرندگان معرفی و به نمونه‌هایی از کاربردهای آن در شناسایی و کنترل سیستم‌ها اشاره شد. در فصل سوم چگونگی مدلسازی سینماتیکی و دینامیکی بازوهای ماهر مکانیکی را مرور کردیم و مدل ریاضی ربات کروی را معرفی نمودیم. روش کنترل مقاوم غیرخطی مبتنی بر استراتژی کنترل گشتاور را به تفصیل در فصل چهارم بررسی کردیم و با استفاده از آن به کنترل مقاوم بازوی ماهر کروی پرداختیم. در فصل پنجم چگونگی کنترل ربات‌ها با استفاده از استراتژی کنترل ولتاژ را مطرح کردیم. سپس با استفاده از این استراتژی و روش کنترل مقاوم غیرخطی، کنترل مقاوم مبتنی بر کنترل ولتاژ را بررسی نمودیم. در فصل هفتم به کمک الگوریتم بهینه سازی پرندگان به تنظیم پارامترهای قوانین کنترل گشتاور و کنترل ولتاژ پرداختیم. نتایج شبیه سازی‌ها بیانگر تأثیر چشمگیر الگوریتم بهینه سازی پرندگان در کاهش خطای ردگیری می‌باشد. همچنین مقایسه نتایج کنترل ولتاژ و گشتاور نشان می‌دهد که استراتژی کنترل ولتاژ در مقایسه با استراتژی کنترل گشتاور عملکرد بهتری دارد و در حضور عدم قطعیت‌های یکسان، کنترل ولتاژ منجر به خطای ردگیری کمتری می‌شود. علاوه بر این، کنترل ولتاژ مبتنی بر کنترل موتور، نسبت به کنترل ولتاژ با در نظر گرفتن موتورهای ربات به عنوان یک سیستم یکپارچه، منجر به خطای ردگیری بسیار کمتری می‌شود.

پیوست‌ها

پیوست ۱

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = D(q)\ddot{q} + h(q, \dot{q}) = \tau$$

با توجه به مدل ربات در فصل ۳ و [۱] می‌توان نوشت:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Y(q, \dot{q}, \ddot{q})P$$

که در آن P بردار پارامترهای ربات می‌باشد که در ربات کروی به صورت زیر است:

$$P = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9]^T$$

$$p_1 = I_{221} + I_{332} +$$

$$p_2 = I_{112} + I_{113} -$$

$$p_3 = 2(I_{132} + I_{133})$$

$$p_4 = m_3$$

$$p_5 = 2m_3l_{c3}$$

$$p_6 = I_{122} + I_{123}$$

$$p_7 = I_{232} + I_{233}$$

$$p_8 = m_2l_{c2}^2 + m_3l$$

$$p_9 = g(m_2 l_{c2} + m_3 l_{c3})$$

و $Y(q, \dot{q}, \ddot{q})$ ماتریس زیر است:

$$Y(q, \dot{q}, \ddot{q}) = [Y_1, Y_2, Y_3]^T$$

$$Y_1 = [\ddot{q}_1, \ddot{q}_1 \sin$$

$$\ddot{q}_1 q_3^2 \sin$$

$$\ddot{q}_1 q_3 \sin^2$$

$$\ddot{q}_2 \sin$$

$$\ddot{q}_2 q_3$$

$$\ddot{q}_2 q_3 + \dot{q}_2 \dot{q}_3 -$$

$$Y_3 = [0, 0, 0, \ddot{q}_3 - q_3 \dot{q}_1^2 - q_3 \dot{q}_2^2 - g \cos(q_2) + \dot{q}_1^2 q_3 \cos^2(q_2),$$

- [1] Spong M., W., Vidyasagar M., (1989), “*Robot dynamic and control*”, Wiley, New York.
- [2] Akhrif O., Okou F.A., Dessaint L.A., Champagne R., (1999), “Application of a multivariable feedback linearization scheme for rotor angle stability and voltage regulation of power systems”, *IEEE Transaction on power systems*, Vol. 14, No. 2, pp. 620-628.
- [3] Ortega R., Spong M., W.,(1988), “Adaptive motion control of rigid robots: a tutorial”, Proceedings of the 27th conference on decision and control, pp. 1575-1584, Austin, Texas.
- [4] Spong M., W., (1992), “ On the robust control of robot manipulators”, *IEEE Transaction on automatic control*, Vol. 37, No. 11, pp. 1782-1786.
- [5] Spong M., W., Vidyasagar M., (1985), “Robust nonlinear control of robot manipulators”, Proceedings of the 24th conference on decision and control, pp. 1767-1772, Ft. Lauderdale, FL.
- [6] Spong M., W., Vidyasagar M., (1987), “Robust linear compensator design for nonlinear robotic control”, *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, pp. 345-351.
- [7] Fateh M.M., Soltanpour M.R., (2009), “Robust task-space control of robot manipulators under imperfect transportation of control space”, *International Journal of Innovative Computing, Information and Control*, Vol. 5, No. 11(A), pp. 3949-3960.
- [8] Golea N., (2002), “Indirect fuzzy adaptive model-following control for robot manipulators”, Proceedings of the 2002 IEEE international conference on control applications, pp. 198-202, Galasgow, Scotland, U.K.
- [9] Wei L., Yang L., Wang H., (2006), “Indirect fuzzy adaptive control for trajectory tracking of uncertain robots”, *Electric Machines and control*, pp. 393-397.
- [10] Wang L.X., (1994), “*Adaptive fuzzy systems and control*”, Prentice Hall.

- [11] Hong-rui W., Zeng-wei C., Li-xin W., Xue-ying T., Xiu-ling L., (2007), "Direct adaptive fuzzy control for robots in cartesian space", Proceedings of Sixth International Conference on Machine Learning Cybernetics, pp. 482-486, Hong Kong.
- [12] Er M. J., Chin S.H., (2000) "Hybrid adaptive fuzzy controllers of robot manipulators with bounds estimation", *IEEE Transaction on industrial electronics*, Vol. 47, No. 5, pp. 1151-1160.
- [13] Neo S.S., Er M.J., (1995), "Adaptive fuzzy control of robot manipulators".
- [14] Chin S.H., Er M. J., (1998), "Hybrid adaptive fuzzy controllers of robot manipulators", Proceedings of the 1998 IEEE/RSJ International conference on intelligent robots and systems, pp. 1132-1137, Victoria, B.C., Canada.
- [15] Yoo B.K., Woon C. H., (2000), "Adaptive control of robot manipulators using fuzzy compensator", *IEEE Transaction on fuzzy systems*, Vol. 8, No. 2, pp.186-199.
- [16] Kim E., (2004), "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic", *IEEE Transaction on fuzzy systems*, Vol. 12, No. 3, pp. 368-378.
- [17] Kumbla K.K., Jamshidi M., (1994), "Control of robotic manipulator using fuzzy logic", Proceedings of the Third IEEE conference on fuzzy systems, pp. 518-523.
- [18] Lim C.M., Hiyama T.,(1991)," Application of fuzzy logic control to a manipulator", *IEEE Transaction on Robot. Autom.*, Vol. 1, No. 5, pp. 688-691.
- [19] Ham C., Qu Z., Johnson R., (2000), " Robust fuzzy control for robot manipulators", *IEE Proc., control theory appl.* Vol. 147, No. 2, pp. 212-216.
- [20] Hwang J.P., Kim E., (2006), "Robust tracking control of an electrically driven robot: Adaptive fuzzy approach", *IEEE Transaction on Fuzzy Systems*, Vol. 14, No. 2, pp. 232-247.
- [21] Tsai C.H., Wang C.H., Lin W.S.,(2000), "Robust fuzzy model-following control of robot manipulators", *IEEE Transaction on Fuzzy Systems*, Vol. 8, No. 4, pp. 462-469.
- [22] Jin Y., (1998), "Decentralized adaptive fuzzy control of robot manipulators", *IEEE Transaction on syst. Man cybern.*, Vol. 28, No. 1, pp. 47-57.
- [23] Kim V.T.,(2002), "Independent joint adaptive fuzzy control of robot manipulators", The 5th Biannual world automation congress, Vol. 14, pp. 645-652.

- [24] Purwar S., Kar I.N., Jha A.N., (2005), “ Adaptive control of robot manipulators using fuzzy logic systems under actuator constraints”, *Fuzzy Sets and Systems*, Vol.152, No. 3, pp. 651-664.
- [25] Fateh M.M., (2010), “Fuzzy task-space control of a welding robot”, *International Journal of Robotics and Automation*, Vol. 25, No. 4, pp. 372-378.
- [26] Fateh M. M., Shahrabi Frahani S., Khatamianfar A.,(2010), “Task space control of a welding robot using a fuzzy coordinator”, *International Journal of Control, Automation, and Systems*, Vol. 8, No. 3, pp. 574–582.
- [27] Jihong Q., Yaping D., Jinkun L., Hongyan W., (2007), “Robust adaptive fuzzy output tracking control of uncertain robot system using backstepping design” , Proceedings of the 27th Chinese control conference, pp. 303-308, Zhangjiajie, Hunan, China.
- [28] Sun F.C., Sun Z.Q., Feng G., (1996), “Design of adaptive fuzzy sliding mode controller for robot manipulators”, Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, Vol. 1, pp. 62-67.
- [29] Zhang D., Song S., (2008), “Research on the adaptive fuzzy sliding mode control of 2-DOF robot”, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 236-240.
- [30] Zhu D., Mei T., Luo M., Zhu D., (2009), “Adaptive sliding mode control for robots based on fuzzy support vector machines”, Proceedings of the 2009 IEEE international conference on mechatronics and automation, Changchun, China.
- [31] Hsu F.Y., Fu L.C., (1995), “Nonlinear control of robot manipulators using adaptive fuzzy sliding mode control”, Proceedings of the 1995 IEEE/RSJ international conference on intelligent robots and systems, Vol. 1, pp. 156-161.
- [32] T.K.J. Koo (1995), “Model reference adaptive fuzzy control of robot manipulator”
- [33] Giordano V., Topalov A.V., Kaynak O., Turchiano B., (2004), "Sliding-mode approach for on-line neural identification of robotic manipulators", 5th Asian Control Conference., Vol.3, pp. 2060- 2065.
- [34] Sadati N., Ghadami R., Bagherpour M., (2005), "An adaptive neural network sliding controller for robotic manipulators" IEEE International Conference on Industrial Technology, pp.1246-1251.

- [35] Xuemei R., Rad A.B., Lewis F.L. , (2007), "Neural Network-Based Compensation Control of Robot Manipulators with Unknown Dynamics," American Control Conference, pp.13-18.
- [36] Wai R.J., (2003), "Tracking control based on neural network strategy for robot manipulators", *Neurocomputing*, Vol. 51, pp. 425-445.
- [37]Tsuji T., Tanaka Y., (2005), "On-line learning of robot arm impedance using neural networks", *Robotics and Autonomous Systems*, Vol. 52, pp.257-271.
- [38] Fateh M.M., (2008), "On the voltage based control of electrical manipulators", *International Journal of Control, Automation and System*, Vol. 6, No.5, pp. 702–712.
- [39] Fateh M. M.,(2010), "Robust control of electrical manipulators by joint acceleration", *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 12, pp. 5501-5510.
- [40] Fateh M. M., (2009), "Robust control of electrical manipulators by reducing the effects of uncertainties", *World Applied Sciences Journal*, Vol. 7, pp.161–167.
- [41] Fateh M.M., (2010), "Robust fuzzy control of electrical manipulators", *Journal of Intelligent and Robotic Systems*, Vol. 60, No. (3), pp. 415-434.
- [42]Soltanpour M. R., Fateh M. M., (2009), "Adaptive robust tracking control of robot manipulators in the task-space under uncertainties", *Australian Journal of Basic and Applied Sciences*, Vol. 3, No. 1, pp. 308–322.
- [43] Soltanpour M. R., Fateh M.M., (2009), "Sliding mode robust control of robot manipulator in the task space by support of feedback linearization and backstepping control", *World Applied Sciences Journal*, Vol. 6, No. 1, pp. 70–76.
- [44] Shahrabi Farahani S., Fateh M. M. and Khatamianfar A., (2009), "Fuzzy position control of a SCARA welding robot in task space", 17th Iranian Conference on Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
- [45] Cheah C.C., Hirano M., Kawamura S., Arimoto S., (2003), "Approximate jacobian control for robot with uncertain kinematics and dynamics", *IEEE Transaction on Robotics and Automation*, Vol. 19, No. 4, pp. 692-702.

- [46] Fateh M.M., (2010), “Robust voltage control of electrical manipulators in task-space”, *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 6, pp.2691-2700.
- [47] Spong M.W., Khorasani K., Kokotovic P.V.,(1987), “An integral manifold approach to the feedback control of flexible joint robots”, *IEEE J. Robot. Autom.* RA-3, 291–300.
- [48] Wang D., (1995), “A simple iterative learning controller for manipulators with flexible joints”, *Automatica*, Vol. 31, No. 9, pp.1341–1344.
- [49] Zeman V., Patel R.V., Khorasani K., (1997), “Control of a flexible-joint robot using neural networks”, *IEEE Trans. Control Syst. Technol.* Vol. 5, No. 4, pp.453–462.
- [50] Kugi, A., Ott, C., Albu-Schaffer, A., Hirzinger, G., (2008), “On the passivity-based impedance control of flexible joint robots”, *IEEE Trans. Robot. Autom.* Vol. 24, No.2, pp. 416–429.
- [51] Talole, E., Kolhe, P., Phadke, B., (2009), “Extended state observer based control of flexible joint system with experimental validation”, *IEEE Trans. Ind. Electron*, doi:10.1109/TIE.2009.2029528.
- [52] Tomei P.,(1991), “A simple PD controller for robots with elastic joints,” *IEEE Transaction on Automatic Control*, Vol. 36, No.10, pp. 1208–1213.
- [53] Luca A.D., Isidori A., Nicolo F., (1985), “Control of robot arm with elastic joints via nonlinear dynamic feedback”, Proceedings of the 24th conference on decision and control, pp. 1671-1679, Ft. Lauderdale, FL.
- [54] Spong M.W., (1987), “Modeling and control of elastic joint robots”, *ASME J. Dyn. Syst. Meas. Control*, Vol. 109, pp. 310–319.
- [55] Wilson G.A., (1994), “Robust tracking of elastic joint manipulators using sliding mode control”, *Trans. Inst. Meas. Control* Vol. 16, No. 2, pp. 99–107.
- [56] Spong M.W., (1985), “Adaptive control of flexible joint manipulators: comments on two papers”, *Automatica*, Vol. 31, No. 4, pp. 585–590.
- [57] Chang L.L., Chuan C.C.,(1995), “Rigid model-based fuzzy control of flexible-joint manipulators”, *Journal of Intelligent and Robotic Systems*, Vol. 13, No. 2, pp.107–126.
- [58] Fateh M.M., (2011), “Robust control of flexible-joint robots using voltage control strategy”, *Nonlinear Dynamics*, DOI 10.1007/s11071-011-0086-3.

- [59] Astrom K. J., Wittenmark B., (1995), “Adaptive control”, Massachusetts, Addison-Wesley.
- [60] Alfi A., Fateh M.M., (2011), “Intelligent identification and control using improved fuzzy particle swarm optimization”, *Expert Systems with Applications*, Vol. 38,pp. 12312–12317.
- [61] H. Modares, A. Alfi, M. M. Fateh,(2011), “Parameter Identification of Chaotic Systems Through an Improved Particle Swarm Optimization”, *Expert systems with application*, vol. 37, pp 3714-3720.
- [62] A. M. Abdelbar, S. Abdelshahid, D. C. Wunsch, “Fuzzy PSO: A Generalization of Particle Swarm Optimization”, *Proceeding of International Joint Conference on Neural Networks*, Montreal, Canada, July 31- August 4, 2005, pp 1086-1091.
- [63] M. M. Fateh and M. Moradi Zirkohi, (2011), “Adaptive impedance control of a hydraulic suspension system using particle swarm optimization”, *Vehicle System Dynamics*, Published Online: 4 April 2011. DOI: 10.1080/00423114.2011.564289.
- [64] A. Alfi and M. M. Fateh, (2011), “Identification of nonlinear systems using modified particle swarm optimization: A hydraulic suspension system”, *Vehicle System Dynamics*, Vol. 49, No. 6, pp. 871-887.
- [65] M. Moradi zirkohi, M. M. Fateh, A. Akbarzadeh,(2010), “Design of radial basis function network using adaptive particle swarm optimization”, *Journal of Software Engineering and Applications*, Vol. 3, pp. 704–708.
- [66] A. Alfi and M. M. Fateh, (2010), “Parameter identification based on a modified PSO applied to suspension system”, *Journal of Software Engineering and Applications*, Vol. 3, pp. 221–229.
- [67] Y. D. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R. G. Harley, (2008), “Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems”, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, Vol. 12, No. 2, pp. 171-195.
- [68] C. C. A. Rajan, M. R. Mohan, (2004), “An Evolutionary Programming-Based Tabu Search Method For Solving The Unit Commitment Problem”, *IEEE TRANSACTIONS ON POWER SYSTEMS*, Vol. 19, No. 1, pp. 577-585.

- [69] K. S. Swarup, S. Yamashiro, (2002), "Unit Commitment Solution Methodology Using Genetic Algorithm", *IEEE TRANSACTIONS ON POWER SYSTEMS*, Vol. 17, No. 1, pp. 87-91.
- [70] J.S.R.Jang, C.T. Sun, (1997), "*Neuro-Fuzzy and Soft Computing*", Prentice Hall.
- [71] J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE Int. Conf. Neural Networks*, Vol. 4, pp. 1942–1948, 1995.
- [72] Y. Meiyong and W. Xiaodong, (2007), "PSO-based Parameter Estimation of Nonlinear Systems", *Proceedings of the 26th Chinese Control Conference*, pp 533-536.
- [73] K. Rajeswari and P. Lakshmi, (2010), "PSO Optimized Fuzzy Logic Controller for Active Suspension System", *International Conference on Advances in Recent Technologies in Communication and Computing*, pp 278-283.

[۷۴] علیزاده، الف، ۱۳۹۰، پایان نامه ارشد، مدلسازی و کنترل غیرخطی مقاوم ردگیری ربات کروی،

دانشگاه مهندسی برق و رباتیک، دانشگاه صنعتی شاهرود.

- [75] Fateh, M.M., (2010), "Proper Uncertainty Bound Parameter to Robust Control of Electrical Manipulators Using Nominal Model", *Nonlinear Dynamics*, Vol. 61, pp. 655-666.

Abstract This thesis is concerned with the robust nonlinear control of electrically driven robot manipulators and presents a novel method based on voltage control strategy to compensate the tracking error caused by uncertainties including parametric uncertainty, external disturbances and unmodeled dynamics. The voltage control strategy is superior than the conventional torque control strategy since it is free of manipulator dynamics. Consequently, the computational load of the controller is less in comparison with the torque control strategy. The most important novelty of this thesis is using the particle swarm optimization algorithm (PSO) to minimize the tracking error in the joint-space. The stability of the system is guaranteed through the Lyapunov direct method and the PSO is employed to enhance the control system performance. The case study is a spherical robot manipulator driven by permanent magnet dc motors. Simulation results illustrate the capability of the PSO algorithm in reducing the tracking error. Moreover, the comparison of the voltage control strategy and the torque control strategy reveals the superiority of the voltage control strategy.

Keywords: robot manipulator, torque control strategy, voltage control strategy, nonlinear robust control, particle swarm optimization.



Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

**Optimal Nonlinear Control of Spherical Robot Manipulator
Using Particle Swarm Optimization**

Saeed Khorashadizadeh

**Supervisor(s):
Dr. Mohammad Mehdi Fateh**

Summer 2011