



دانشکده برق و رباتیک
گروه کنترل

پایان نامه کارشناسی ارشد

کنترل فازی تطبیقی غیر مستقیم ربات
" پیوما ۵۶۰ " با روش گرادیان نزولی

منصوره قوشه

استاد راهنما :

دکتر محمد مهدی فاتح

شهریور ۱۳۸۹

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود
دانشکده برق و رباتیک
گروه کنترل

کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان نزولی

دانشجو : منصوره قوشه

استاد راهنما :

جناب آقای دکتر محمد مهدی فاتح

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۸۹

دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه کنترل

پایان نامه کارشناسی ارشد خانم منصوره قوشه

تحت عنوان:

کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان نزولی

در تاریخ ۱۳۹۱/۰۶/۸۹ توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه عالی مورد پذیرش قرار گرفت.

امضاء	استاد مشاور	امضاء	استاد راهنما
	نام و نام خانوادگی :		نام و نام خانوادگی :محمد مهدی فاتح

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی : امید رضا معروضی		نام و نام خانوادگی : سید علی سلیمانی
			نام و نام خانوادگی :حیدر طوسیان شاندیز

تقدیم به :

پدر و مادر عزیز و
همسر گرامیم که در طی دوران تحصیل مرا یاری نمودند .

تشکر و قدردانی

اینک که با یاری خداوند متعال ، این پایان نامه به اتمام رسیده است بر خود لازم می دانم از استاد محترم جناب آقای دکتر محمد مهدی فاتح که راهنمای من در تمام امور این پایان نامه بوده اند ، صمیمانه قدردانی نمایم . از پدر و مادر عزیزم که زحمات زیادی را نثارم نمودند بینهایت سپاسگذارم و همچنین از همسر مهربانم که همواره در طول دوران تحصیل مشوق من بوده اند و مرا در این امر یاری نموده اند ، تشکر و قدردانی می نمایم .

تعهد نامه

اینجانب منصوره قوشه دانشجوی دوره کارشناسی ارشد رشته مهندسی برق - کنترل دانشکده برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه " کنترل فازی تطبیقی غیر مستقیم ربات پیوما ۵۶۰ با روش گرادیان نزولی " تحت راهنمایی جناب آقای دکتر 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 پژوهش های 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 تاثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- 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 اخلاقی 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 افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد .

چکیده :

در این پایان نامه ، هدف کنترل یکی از بازوهای ماهر رباتیک ، ربات " پیوما ۵۶۰ " می باشد . این ربات دارای ۶ مفصل لولایی است که هر یک از ۶ مفصل ربات توسط یک محرک (موتور DC) به حرکت در آمده ، تا موقعیت نقطه انتهایی ربات به مکان مورد نظر ما برسد . این ربات کاربردهای زیادی در صنعت دارد .

از بین روش های کنترلی ، روش کنترل فازی تطبیقی غیر مستقیم را برای کنترل این ربات انتخاب نموده ایم ، که برای طراحی سیستم های فازی در این روش کنترلی ، پارامترهای این سیستم ها را با روش گرادیان نزولی تنظیم می نمائیم به منظور این که خطای ردگیری بین موقعیت های مطلوب با مفاصل ربات مینیمم گردد .

در انجام این پروژه ، قبل از طراحی کنترلر برای ربات پیوما ۵۶۰ ، این نوع کنترلر را برای یک موتور DC طراحی نموده ایم و پس از نتیجه بخش بودن آن ، از این کنترلر برای کنترل سیگنال ولتاژ هر یک از شش موتوری که مفاصل ربات را به حرکت در می آورد استفاده می نمائیم .

در واقع کنترل روی هر یک از مفاصل بصورت جداگانه انجام می گیرد که همان استراتژی کنترل مفصل مستقل است .

در این حالت کوپلینگ بین مفاصل ربات نادیده گرفته شده ، ولی چون سیستم کنترل ، تطبیقی است کوپلینگ بین مفاصل که همان اغتشاشات و عدم قطعیت ها در سیستم است در نظر گرفته می شود . لازم به ذکر است که ما هیچ رابطه ای از سیستم اصلی ربات " پیوما ۵۶۰ " نداریم و معادلات مربوط به ربات برای ما ناشناخته می باشد . برای شبیه سازی این سیستم کنترل از ربات پیوما ۵۶۰ موجود در جعبه ابزار رباتیک نرم افزار مطلب استفاده می کنیم .

در انتها این روش کنترل با یکی از روش های کنترل کلاسیک یعنی روش کنترل کلاسیک PD مقایسه شده و به این نتیجه خواهیم رسید که در روش کنترل فازی تطبیقی غیر مستقیم ، خطای حالت دائم کمتری داریم ولی نسبت به روش کنترل کلاسیک PD ، دیرتر سیستم کنترل تطبیق پیدا می کند .

کلمات کلیدی : کنترل فازی تطبیقی غیرمستقیم ، گرادیان نزولی ، موتور DC و ربات پیوما ۵۶۰

فهرست مطالب

بخش اول : کنترل فازی تطبیقی..... ۱.....

۱-۱ . کنترل فازی در مقایسه با کنترل کلاسیک..... ۲.....

۱-۲ . کنترل فازی تطبیقی..... ۵.....

۱-۲-۱ . کنترل فازی تطبیقی غیر مستقیم..... ۷.....

۱-۲-۲ . طراحی کنترل فازی تطبیقی غیر مستقیم..... ۸.....

۱-۲-۲-۱ . مشخصات مساله..... ۸.....

۱-۲-۲-۲ . طراحی کنترل کننده فازی..... ۹.....

بخش دوم : گرادیان نزولی..... ۱۴.....

۲-۱ . معرفی گرادیان نزولی..... ۱۵.....

۲-۲ . روش های گرادیان نزولی..... ۱۷.....

۲-۲-۱ . روش شدیدترین نزول..... ۱۷.....

۲-۲-۲ . روش گرادیان نزولی با ممنوم..... ۱۸.....

۲-۲-۳ . روش نیوتن..... ۱۹.....

۲-۲-۴ . روش لیونبرگ - مارکوارد..... ۱۹.....

۲-۳ . یافتن مینیمم تابع $f(x)$ با استفاده از الگوریتم گرادیان نزولی..... ۲۰.....

۲-۴ . طراحی سیستم های فازی با استفاده از روش آموزش گرادیان نزولی..... ۲۳.....

۲-۴-۱ . انتخاب ساختار سیستم فازی..... ۲۳.....

۲-۴-۲ . طراحی پارامترها بوسیله گرادیان نزولی ۲۴

۲-۴-۳ . طراحی سیستم های فازی با استفاده از روش آموزش گرادیان نزولی..... ۲۶

بخش سوم : سیستم کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان

نزولی..... ۲۸

۳-۱ . سیستم اصلی (ربات پیوما ۵۶۰)..... ۲۹

۳-۲ . سروموتور dc..... ۳۲

۳-۳ . کنترل مفصل مستقل..... ۳۷

۳-۳-۱ . طراحی سیستم فازی \hat{f} با روش گرادیان نزولی..... ۴۰

۳-۳-۲ . طراحی سیستم فازی \hat{g} با روش گرادیان نزولی ۴۴

۳-۳-۳ . مرحله ساخت قانون کنترل..... ۴۸

بخش چهارم : نتایج شبیه سازی..... ۵۲

۴-۱ . شبیه سازی کنترل فازی تطبیقی غیرمستقیم موتور dc با روش گرادیان نزولی..... ۵۳

۴-۱-۱ . پاسخ سیستم کنترل به ازای ورودی پله..... ۵۳

۴-۱-۲ . پاسخ سیستم کنترل به ازای ورودی \sin ۶۰

۴-۲ . شبیه سازی کنترل فازی تطبیقی غیرمستقیم ربات پیوما ۵۶۰ با روش گرادیان نزولی..... ۶۵

۴-۲-۱ . شبیه سازی سیستم کنترل به ازای ورودی های \sin ای ۶۵

۴-۲-۲ . شبیه سازی سیستم کنترل به ازای ورودی های پله ۷۲

بخش پنجم : مقایسه نتایج کنترل فازی تطبیقی غیر مستقیم به روش گرادیان نزولی با کنترل کلاسیک PD..... ۸۰

۵-۱ . مقایسه نتایج مربوط به سیستم کنترل موتور dc ۸۱

۵-۱-۱ . مقایسه نتایج به ازای ورودی های پله..... ۸۱

۵-۱-۲ . مقایسه نتایج به ازای ورودی های \sin ای..... ۸۲

۵-۲ . مقایسه نتایج مربوط به سیستم کنترل ربات پیوما ۵۶۰ ۸۳

۵-۲-۱ . مقایسه نتایج به ازای ورودی های پله..... ۸۳

۵-۲-۲ . مقایسه نتایج به ازای ورودی های \sin ای..... ۸۸

بخش ششم : نتیجه گیری..... ۹۵

پیوست..... ۹۸

پیوست ۱..... ۹۹

پیوست ۲..... ۱۰۱

پیوست ۳..... ۱۰۳

مراجع..... ۱۰۵

فهرست اشکال

- شکل ۱-۱ کنترل فازی در مقایسه با کنترل متعارف..... ۳
- شکل ۲-۱ ساختار اصلی سیستم های کنترل فازی تطبیقی..... ۶
- شکل ۳-۱ سیستم کنترل فازی تطبیقی غیر مستقیم..... ۱۳
- شکل ۳-۱ بازوی ماهر ربات " پیوما ۵۶۰ "..... ۳۰
- شکل ۲-۳ دیاگرام مفصلی ربات " پیوما ۵۶۰ "..... ۳۰
- شکل ۳-۳ سطح مقطع موتور DC مغناطیس دائم..... ۳۲
- شکل ۴-۳ دیاگرام مدار برای موتور DC کنترل شده با ولتاژ آرمیچر..... ۳۳
- شکل ۵-۳ مدل فشرده یک رابط با سیستم انتقال چرخ دنده - محرک..... ۳۵
- شکل ۶-۳ بلوک دیاگرام سیستم موتور dc..... ۳۶
- شکل ۷-۳ گروههای فازی مربوط به q و dq ۴۰
- شکل ۸-۳ گروههای فازی مربوط به خروجی سیستم فازی \hat{f} ۴۱
- شکل ۹-۳ بلوک دیاگرام مربوط به قوانین سیستم فازی \hat{f} ۴۲
- شکل ۱۰-۳ سطح خروجی به ازای تمام حالات در نظر گرفته شده برای ورودی های (q, dq) در بازه های موردنظر..... ۴۲
- شکل ۱۱-۳ بلوک دیاگرام سیستم فازی..... ۴۳
- شکل ۱۲-۳ گروههای فازی مربوط به ورودی q ۴۴
- شکل ۱۳-۳ گروههای فازی مربوط به خروجی سیستم فازی \hat{g} ۴۵
- شکل ۱۴-۳ بلوک دیاگرام مربوط به قوانین سیستم فازی \hat{g} ۴۶
- شکل ۱۵-۳ بلوک دیاگرام سیستم فازی \hat{g} ۴۶

- شکل ۳-۱۶ بلوک دیاگرام مربوط به کنترل یک مفصل از ربات پیوما ۵۶۰..... ۴۹
- شکل ۳-۱۷ بلوک دیاگرام داخل بلوک مربوط به کنترلر..... ۵۰
- شکل ۳-۱۸ بلوک دیاگرام کل سیستم کنترل ۶ مفصل ربات پیوما ۵۶۰..... ۵۱
- شکل ۴-۱ خطای ردگیری به ازای ورودی پله $q_d = 0.3u(t)$ ۵۳
- شکل ۴-۲ پارامترهای سیستم فازی g و f با روش گرادیان نزولی..... ۵۴
- شکل ۴-۳ خروجی سیستم فازی g و f ۵۵
- شکل ۴-۴ سیگنال کنترل ولتاژ..... ۵۶
- شکل ۴-۵ پاسخ سیستم کنترل q و ورودی مطلوب q_d ۵۶
- شکل ۴-۶ خطای ردگیری به ازای ورودی پله $q_d = 0.3u(t)$ ۵۷
- شکل ۴-۷ سیگنال کنترل ولتاژ..... ۵۷
- شکل ۴-۸ پارامترهای سیستم فازی f و g ۵۸
- شکل ۴-۹ خروجی سیستم های فازی f و g ۵۹
- شکل ۴-۱۰ پاسخ سیستم کنترل q و ورودی مطلوب q_d ۶۰
- شکل ۴-۱۱ خطای ردگیری به ازای ورودی $q_d = -0.2\cos(\pi*t)+0.2$ ۶۱
- شکل ۴-۱۲ پارامترهای سیستم فازی f ۶۱
- شکل ۴-۱۳ پارامترهای سیستم فازی g ۶۲
- شکل ۴-۱۴ خروجی سیستم های فازی f ۶۲
- شکل ۴-۱۵ خروجی سیستم های فازی g ۶۳
- شکل ۴-۱۶ سیگنال ولتاژ..... ۶۳
- شکل ۴-۱۷ پاسخ سیستم کنترل..... ۶۴
- شکل ۴-۱۸ پاسخ سیستم کنترل فازی تطبیقی غیر مستقیم به ازای مسیرهای مطلوب \sin ای..... ۶۵
- شکل ۴-۱۹ خطای ردگیری مربوط به کنترل هر ۶ مفصل ربات (با ۹ قانون برای سیستم فازی f)..... ۶۶

- شکل ۴-۲۰ سیگنال های ولتاژ اعمالی به موتورها..... ۶۶
- شکل ۴-۲۱ خروجی های f و g مربوط به ۶ کنترلر..... ۶۷
- شکل ۴-۲۲ گشتاورهای موجود روی ۶ مفصل ربات پیوما ۵۶۰..... ۶۸
- شکل ۴-۲۳ تغییر پارامترهای سیستم های فازی f در یک سیکل زمانی (۹ قانون برای هر سیستم تعریف شده است)..... ۶۸
- شکل ۴-۲۴ تغییر پارامترهای سیستم های فازی g ۷۰
- شکل ۴-۲۵ مسیر نقطه انتهایی ربات در فضای دکارتی..... ۷۱
- شکل ۴-۲۶ خطای ردگیری مربوط به کنترل هر ۶ مفصل ربات..... ۷۲
- شکل ۴-۲۷ سیگنال های کنترل هر ۶ مفصل ربات..... ۷۳
- شکل ۴-۲۸ خروجی های f مربوط به ۶ کنترلر..... ۷۴
- شکل ۴-۲۹ خروجی های g مربوط به ۶ کنترلر..... ۷۵
- شکل ۴-۳۰ گشتاورهای موجود روی ۶ مفصل ربات پیوما ۵۶۰..... ۷۶
- شکل ۴-۳۱ تغییر پارامترهای سیستم های فازی f در یک سیکل زمانی..... ۷۷
- شکل ۴-۳۲ تغییر پارامترهای سیستم های فازی g در یک سیکل زمانی..... ۷۸
- شکل ۴-۳۳ پاسخ های خروجی ۶ سیستم کنترل همان مفاصل ربات " پیوما ۵۶۰ "..... ۷۹
- شکل ۴-۳۴ مسیر نقطه انتهایی ربات در فضای دکارتی..... ۷۹
- شکل ۵-۱ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd..... ۸۱
- شکل ۵-۲ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd..... ۸۲
- شکل ۵-۳ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd مربوط به ۶ مفصل ربات پیوما..... ۸۶
- شکل ۵-۴ خروجی های سیستم های کنترل مفاصل ربات در هر دو حالت ۱. کنترل فازی تطبیقی غیر مستقیم ۲. کنترل کلاسیک PD..... ۸۷

شکل ۵-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۱ ربات پیوما ۵۶۰.....	۸۸
شکل ۶-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۲ ربات پیوما ۵۶۰.....	۸۹
شکل ۷-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۳ ربات پیوما ۵۶۰.....	۹۰
شکل ۸-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۴ ربات پیوما ۵۶۰.....	۹۱
شکل ۹-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۵ ربات پیوما ۵۶۰.....	۹۲
شکل ۱۰-۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd	مربوط به مفصل ۶ ربات پیوما ۵۶۰.....	۹۳
شکل ۱-۷ بلوک سیستم فازی \hat{f}	۹۹
شکل ۲-۷ بلوک سیستم فازی \hat{g}	۱۰۱
شکل ۳-۷ بلوک قانون کنترل	۱۰۳
جدول ۱-۳ پارامترهای دناویت - هارتنبرگ	۳۱

مطالب بخش اول : کنترل فازی تطبیقی

۱-۱. کنترل فازی در مقایسه با کنترل کلاسیک

۱-۲. کنترل فازی تطبیقی

۱-۲-۱. کنترل فازی تطبیقی غیر مستقیم

۱-۲-۲. طراحی کنترل کننده فازی تطبیقی غیر مستقیم

۱-۲-۲-۱. مشخصات مساله

۱-۲-۲-۲. طراحی کنترل کننده فازی

۱-۱. کنترل فازی در مقایسه با کنترل کلاسیک

کنترل فازی^۱ و کنترل کلاسیک^۲ دارای شباهت ها و تفاوت هایی می باشند . این دو در موارد زیر مشابه می باشند .

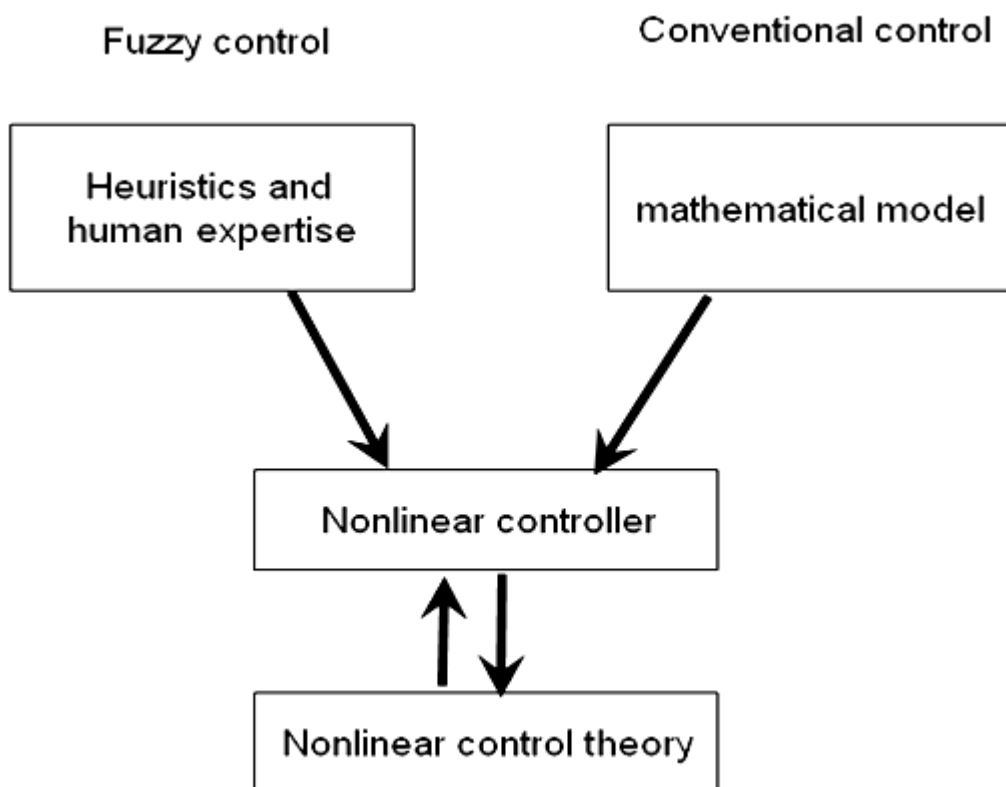
- این دو سعی در حل نوع یکسانی از مسائل را دارند ، که همان مسائل کنترل می باشند. بنابراین بایستی به جنبه های یکسانی که در هر مساله کنترل مشترک می باشد توجه داشته باشند . به عنوان مثال ، پایداری^۳ و عملکرد .
- ابزارهای ریاضی مورد استفاده جهت تحلیل سیستم های کنترل طراحی شده مشابه می باشند ، چرا که به مطالعه جنبه های یکسانی (پایداری ، همگرایی و غیره) برای یک سیستم می پردازند .

به هر حال ، یک تفاوت اساسی بین کنترل فازی و کنترل کلاسیک وجود خواهد داشت :

- کنترل کلاسیک جهت طراحی کنترل کننده با یک مدل ریاضی از فرآیند شروع می نماید و کنترل کننده ها برای مدل طراحی می گردند ، در حالیکه کنترل فازی با استفاده از تجربیات شخص خبره (بصورت قواعد اگر – آنگاه فازی)^۴ شروع می گردد و کنترل کننده ها بوسیله ترکیب این قواعد طراحی می گردند . بدین معنی که اطلاعات لازم جهت ساخت این دو نوع کنترل کننده فازی متفاوت است .

1. Fuzzy control
2. Classic control
3. Stability
4. If-Then Fuzzy rules

شکل ۱-۱ را ببینید. کنترل کننده های فازی پیشرفته می توانند با استفاده از هر دو مدل ریاضی^۱ و هیورستیک^۲ ساخته شوند .



شکل ۱-۱ کنترل فازی در مقایسه با کنترل متعارف

برای بسیاری از مسائل کنترل عملی مشاهده یک مدل ریاضی ساده و در عین حال دقیق مشکل می باشد ، اما می تواند آزمایشاتی توسط یک شخص ماهر و با تجربه فراهم شود ، که یک راه کار عملی و تجربی مفید را برای کنترل نمودن فرآیند ارائه نماید .
 کنترل فازی برای این دو نوع مسائل بیش از هر راه دیگری مفید می باشد .

-
1. Mathematical Model
 2. Heuristics

ما روش های طراحی برای کنترل کننده های فازی را به دو دسته طبقه بندی می کنیم ، رهیافت سعی و خطا و روش تئوریک .

در رهیافت سعی و خطا ، با استفاده از دانشی که بر پایه آزمایشات و تجربیات بدست آمده و به صورت شفاهی قابل بیان است ، همچنین با سوال هایی که از خبرگان مربوطه صورت می گیرد و پرسشنامه های دقیقی تنظیم می شود ، مجموعه ای از قواعد اگر - آنگاه فازی جمع آوری می گردد و آنگاه در ادامه کنترل کننده های فازی بر اساس این قواعد ساخته شده و در نهایت در سیستم مورد آزمایش قرار می گیرند . حال اگر به کارگیری کنترل کننده فازی طراحی شده در عمل مورد رضایت نباشد ، قواعد مجدداً به صورت مناسبی تغییر و تنظیم می گردند و یا دوباره از ابتدا ایجاد می شوند و اینکار آنقدر ادامه می یابد تا پس از چند چرخه سعی و خطا عملکرد نهایی رضایتبخش گردد . در روش تئوریک ، ساختار و پارامترهای کنترل کننده های فازی چنان طراحی می شوند که معیار عملکرد مشخصی (برای مثال - پایداری) تضمین گردد . البته بهتر است که در طراحی کنترل کننده های فازی برای سیستم های عملی ، هر دو روش را ترکیب نمائیم تا امکان دستیابی به بهترین کنترل کننده فازی فراهم آید . [2]

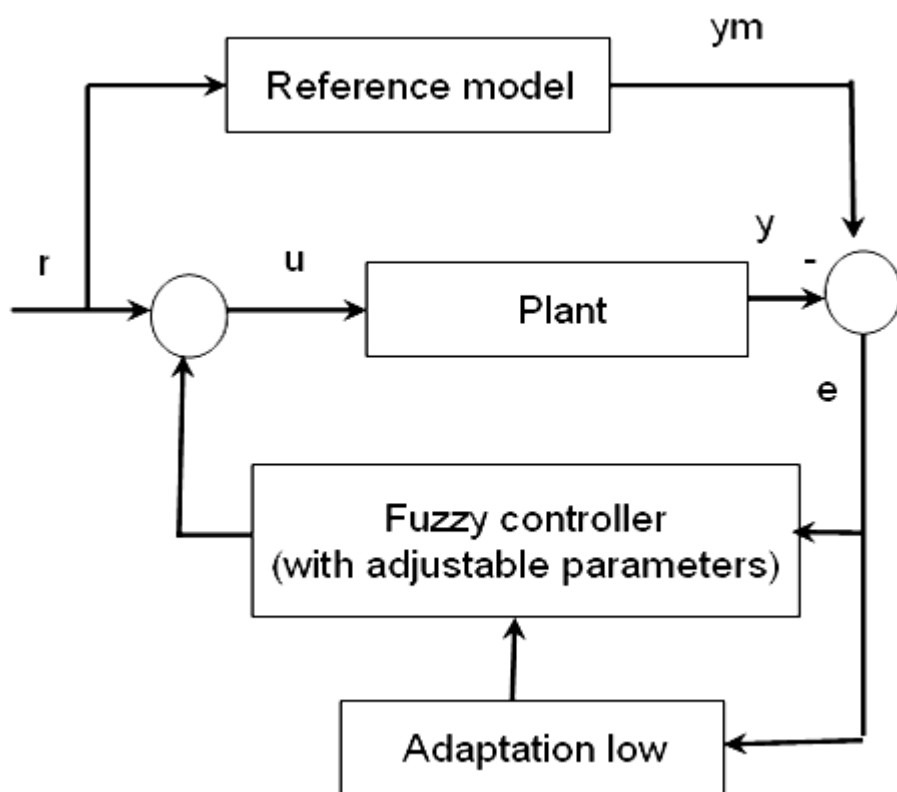
۲-۱. کنترل فازی تطبیقی

کنترل فازی می تواند به دو دسته کنترل فازی غیر تطبیقی و کنترل فازی تطبیقی^۱ دسته بندی شود. در کنترل فازی غیر تطبیقی، ساختار و پارامترهای کنترل کننده های فازی ثابت بوده و در طی انجام عملیات زمان حقیقی عوض نخواهد شد. در حالیکه در کنترل فازی تطبیقی، ساختار و پارامترهای کنترل کننده فازی برحسب شرایط در طی اعمال زمان حقیقی تغییر می یابد. کنترل فازی غیر تطبیقی ساده تر از کنترل فازی تطبیقی بوده اما به اطلاعات بیشتری از مدل فرآیند و قواعد هیورستیک نیاز دارد. از طرفی دیگر، کنترل فازی تطبیقی جهت به اجرا در آمدن پر هزینه بوده ولی در عین حال به اطلاعات کمتری نیاز خواهد داشت و در نتیجه ممکن است بهتر و مؤثرتر به اجرا در آید.

کنترل کننده های فازی فرض می شود در وضعیت هایی کار می کنند که در آن یک عدم قطعیت^۲ بزرگ با تغییرات نامعلوم در پارامترها و ساختارهای سیستم وجود دارد. عموماً، هدف اصلی کنترل تطبیقی عبارت از ثابت نگاه داشتن کارآیی یک سیستم در حضور این عدم قطعیت ها می باشد. بنابراین کنترل فازی پیشرفته باید تطبیقی باشد.

ساختار اصلی یک سیستم کنترل فازی تطبیقی در شکل ۱-۲ نشان داده شده است. مدل مرجع به منظور مشخص نمودن پاسخ ایده آل که سیستم کنترل فازی بایستی پیروی نماید، استفاده می گردد. فرض می شود که سیستم تحت کنترل دارای اجزایی ناشناخته باشد. کنترل کننده های فازی از سیستم های فازی ساخته می شوند که در آن پارامترهای سیستم قابل تنظیم می باشند. قاعده تعدیل^۳، پارامترهای سیستم فازی را به صورت بر خط چنان تنظیم می کند که خروجی سیستم تحت کنترل یعنی $y(t)$ ، خروجی مدل مرجع یعنی $y_m(t)$ را دنبال نماید.

-
1. Adaptive fuzzy control
 2. Uncertainty
 3. Adaptation Law



شکل ۲-۱ ساختار اصلی سیستم های کنترل فازی تطبیقی

از شکل ۲-۱ مشاهده می شود که تفاوت اصلی بین سیستم های کنترل فازی تطبیقی و سیستم های کنترل فازی غیر تطبیقی عبارتند از :

۱. کنترل کننده های فازی در سیستم کنترل فازی تطبیقی در طی عملیات زمان حقیقی^۱ تغییر می کنند ، در حالی که کنترل کننده فازی در سیستم کنترل فازی غیر تطبیقی قبل از عملیات زمان حقیقی مشخص و ثابت خواهند بود .
۲. یک جزء اضافی به نام قاعده تعدیل به سیستم کنترل فازی تطبیقی ارائه می گردد تا پارامترهای کنترل کننده فازی را تنظیم سازد .

مزیت های اصلی کنترل فازی تطبیقی نسبت به کنترل کننده فازی غیر تطبیقی عبارتند از :

۱. عملکرد و کار آیی بهتر معمولا قابل دستیابی می باشد ، چرا که کنترل کننده فازی تطبیقی می تواند خود را با توجه به تغییرات محیطی تنظیم نماید .
۲. دانش کمتری از سیستم تحت کنترل لازم می باشد ، چرا که قاعده تعدیل می تواند در جهت یادگیری دینامیک سیستم در طی عملیات زمان حقیقی کمک نماید .

معایب اصلی کنترل فازی تطبیقی نسبت به کنترل فازی غیر تطبیقی عبارتند از :

۱. تحلیل نتایج سیستم کنترل مشکل است زیرا سیستم کنترل نه تنها غیر خطی بلکه متغییر با زمان نیز می باشد .
۲. پیاده سازی آن گران و پرخرج است .

۱-۲-۱. کنترل فازی تطبیقی غیر مستقیم

کنترل کننده های تطبیقی را می توان به سه دسته طبقه بندی نمود :
غیرمستقیم ، مستقیم و ترکیب غیر مستقیم/ مستقیم .

کنترل فازی تطبیقی غیر مستقیم :

کنترل کننده فازی تعدادی سیستم فازی را شامل می شود که این سیستم های فازی (در ابتدا) از روی دانش سیستمی ساخته می شوند .

کنترل فازی تطبیقی مستقیم :

کنترل کننده فازی تنها یک سیستم فازی می باشد که (در ابتدا) از دانش کنترلی ساخته می شود

کنترل فازی تطبیقی مستقیم / غیر مستقیم :

کنترل کننده فازی یک متوسط وزن دار شده از کنترل کننده های فازی تطبیقی مستقیم و غیر مستقیم می باشد (بنابراین ، هر دو دانش سیستمی و دانش کنترلی مورد استفاده واقع می گردند)

۱-۲-۲. طراحی کنترل کننده فازی تطبیقی غیر مستقیم

۱-۲-۲-۱. مشخصات مساله :

فرض کنید که سیستم یک سیستم غیر خطی^۱ مرتبه n ام باشد که بوسیله معادلات دیفرانسیل زیر بیان می شود ، f و g توابع ناشناخته ، u و y به ترتیب ورودی - خروجی سیستم می باشند .

$$\dot{x}^{(n)} = f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u \quad (1-1)$$

$$y = x \quad (2-1)$$

اندازه گیری باشد . $X = (x_1, x_2, \dots, x_n)^T = (x, \dot{x}, \dots, x^{(n-1)})^T \in \mathbb{R}^n$ بردار حالت سیستم است که فرض می شود قابل

برای اینکه رابطه (۱-۱) کنترل پذیر باشد ، لازم است که $g(x) \neq 0$ باشد .

بدون از دست دادن عمومیت ، فرض می کنیم که $g(x) > 0$ باشد . در متون مربوط به کنترل غیر خطی این سیستم ها به فرم طبیعی بوده و دارای درجه نسبی برابر با n هستند .

هدف کنترل عبارت است از طراحی یک کنترل کننده بازخورد $u = u(X|\theta)$ بر پایه سیستم های فازی و یک قاعده تعدیل به منظور تنظیم بردار پارامتر θ ، بطوریکه خروجی سیستم یعنی y ، خروجی ایده آل یعنی y_m را دنبال نماید و اینکه مشتقات زمانی آن معلوم و کران دار باشد .

از آنجائیکه توابع $f(X)$ ، $g(X)$ در سیستم غیر خطی بوده و نامعلوم فرض می گردند ، ما با یک مساله کنترل غیر خطی تک ورودی - تک خروجی کاملاً عمومی سر و کار خواهیم داشت بنابراین در هدف کنترل ، بر این مورد پافشاری نخواهیم کرد که خروجی سیستم y حتماً بایستی به سوی خروجی y_m بطور مجانبی همگرا گردد ، بلکه تنها لازم داریم که y تا حد امکان نزدیک به y_m آن را دنبال نماید .

از آنجائیکه ما یک کنترل کننده فازی تطبیقی غیر مستقیم را در این بخش طراحی می کنیم ، مقداری دانش سیستمی موجود است . در حالت خاص ، فرض می کنیم که مجموعه ای از قواعد اگر - آنگاه فازی در دسترس می باشند که رفتار ورودی - خروجی $f(X)$ ، $g(X)$ را تشریح می کنند ، این قواعد به صورت زیر داده شده اند :

$$(3-1) \quad \text{اگر } x_1, x_2, \dots, x_n \text{ است و } F_n^f \text{ است ، آنگاه } f(x) \text{ است } C^r .$$

که $f(x)$ را تشریح می کند و

$$(4-1) \quad \text{اگر } x_1, x_2, \dots, x_n \text{ است و } G_n^s \text{ است ، آنگاه } g(x) \text{ است } D^s .$$

که $g(x)$ را شرح می دهد و در روابط بالا F_1^f, C^r, G_1^s, D^s مجموعه های فازی هستند و r و s به ترتیب تعداد قوانین فازی مربوط به سیستم های فازی f و g می باشند .

۲-۲-۱. طراحی کنترل کننده فازی :

اگر توابع غیر خطی $f(X), g(X)$ مشخص باشند ، آنگاه ما می توانیم کنترل u را چنان انتخاب نمائیم که قسمت غیر خطی را از بین ببرد و کنترل کننده را بر پایه تئوری کنترل خطی طراحی کنیم (برای مثال جایابی قطب) . در حالت خاص فرض کنید

$$K = (k_n, \dots, k_1) , e = (e, \dot{e}, \dots, e^{(n-1)})^T , e = y_m - y = y_m - x \quad -$$

باشند بطوریکه همه ریشه های چند جمله ای $s^n + k_1s^{n-1} + \dots + k_n$ در نیمه باز چپ صفحه مختلط قرار داشته باشند و قاعده کنترل را بدین صورت انتخاب کنید .

$$u^* = \frac{1}{g(X)} [-f(X) + y_m^{(n)} + K^T e] \quad (5-1)$$

با جایگذاری (3-1) در (1-1) ، سیستم حلقه بسته را بصورت زیر بدست می آوریم :

$$e^{(n)} + k_1 e^{(n-1)} + \dots + k_n e = 0 \quad (6-1)$$

بدلیل انتخاب K ، ما داریم $e(t) \rightarrow 0$ هرگاه $t \rightarrow \infty$ ، یعنی خروجی سیستم y به سوی خروجی حقیقی y_m به طور مجانبی همگرا می گردد .

از آنجائیکه $f(X)$ ، $g(X)$ نامعلوم می باشند ، لذا کنترل کننده ایده آل (5-1) نمی تواند مورد استفاده قرار گیرد . لیکن ما قواعد اگر - آنگاه فازی که رفتار ورودی - خروجی $f(X)$ ، $g(X)$ را شرح می دهند ، خواهیم داشت . بنابراین ، یک ایده معقول جایگذاری $f(X)$ ، $g(X)$ در (5-1) بوسیله سیستم های فازی $\hat{f}(X)$ ، $\hat{g}(X)$ می باشد ، که به ترتیب با استفاده از قواعد (3-1) ، (4-1) ساخته می شوند .

از آنجائیکه قواعد (3-1) ، (4-1) تنها اطلاعاتی تقریبی درباره $f(X)$ ، $g(X)$ فراهم می نماید ، لذا سیستم های فازی ساخته شده $\hat{f}(X)$ ، $\hat{g}(X)$ قادر نخواهد بود به اندازه کافی $f(X)$ ، $g(X)$ را تقریب بزند .

به منظور بهبود یافتن دقت y_m ، یک ایده آن است که برخی پارامترها را در $\hat{f}(X)$ ، $\hat{g}(X)$ آزاد بگذاریم تا در طی عملیات بر خط تغییر یابند بطوریکه دقت تقریب همراه با گذشت زمان بهبود یابد .

فرض کنید $\theta_f \in R^{Mf}$ ، $\theta_g \in R^{Mg}$ به ترتیب پارامترهای آزاد در $\hat{f}(X)$ ، $\hat{g}(X)$ باشند ، بنابراین ما عبارت $\hat{f}(X) = \hat{f}(X|\theta_f)$ ، $\hat{g}(X) = \hat{g}(X|\theta_g)$ را نمایش می دهیم . با جایگذاری $f(X)$ ، $g(X)$ در (5-1) به ترتیب بوسیله سیستم های فازی $\hat{f}(X|\theta_f)$ ، $\hat{g}(X|\theta_g)$ ، کنترل کننده فازی را به صورت زیر بدست می آوریم .

$$u = u_I = \frac{1}{\hat{g}(X|\theta_g)} [-\hat{f}(X|\theta_f) + y_m^{(n)} + K^T e] \quad (7-1)$$

این کنترل کننده فازی، کنترل کننده معادل قطعی^۱ نامیده می شود، زیرا اگر \hat{f} ، \hat{g} با f ، g متناظر خود باشند (که به این مفهوم خواهد بود که هیچگونه عدم قطعیتی در رابطه با f ، g وجود ندارد)، آنگاه کنترل کننده u_I همان کنترل کننده ایده آل u^* رابطه (5-1) خواهد شد.

برای پیاده سازی کنترل کننده (7-1)، ما بایستی فرمولها و جزئیات $\hat{f}(X|\theta_f)$ ، $\hat{g}(X|\theta_g)$ را مشخص نمائیم. از آنجائیکه ممکن است تعداد قواعد در (3-1)، (4-1) کم باشد، بنابراین این تعداد قواعد جهت ایجاد $\hat{g}(X|\theta_g)$ ، $\hat{f}(X|\theta_f)$ عموماً کافی نخواهد بود و نمی توان آنها را تنها بر پایه L_f قاعده در (3-1) و L_g قاعده در (4-1) بنا نمود.

ما بایستی $\hat{f}(X|\theta_f)$ ، $\hat{g}(X|\theta_g)$ را بر پایه مجموعه های کاملی از قواعد که قواعد (3-1)، (4-1) به عنوان حالت های خاص شامل می شوند، بسازیم. در حالت خاص $\hat{g}(X|\theta_g)$ ، $\hat{f}(X|\theta_f)$ با استفاده از دو گام زیر ساخته می شود:

- گام اول: برای متغیرهای x_i ($i = 1, 2, \dots, n$)، p_i مجموعه فازی $A_i^{l_i}$ ($l_i = 1, 2, \dots, p_i$) تعریف کنید که F_i^r ($r = 1, 2, \dots, L_f$) در (3-1) را به عنوان حالت های خاص شامل شود و q_i مجموعه فازی $B_i^{l_i}$ ($l_i = 1, 2, \dots, q_i$) تعریف کنید، که G_i^s ($s = 1, 2, \dots, L_g$) در (4-1) را به عنوان حالت های خاص شامل شود.

- گام دوم: سیستم فازی $\hat{f}(X|\theta_f)$ را از $\prod_{i=1}^n p_i$ قاعده، (8-1) اگر x_1 ، $B_1^{l_1}$ است و ... و x_n ، $A_n^{l_n}$ است، آنگاه \hat{f} ، $E^{1 \dots l_n}$ است. بسازید که $(l_i = 1, 2, \dots, p_i)$ ، $i = 1, 2, \dots, n$ و $E^{1 \dots l_n}$ معادل با C^r می باشد. اگر قسمت اگر رابطه (8-1) با قسمت اگر رابطه (3-1) مطابقت داشته باشد و در غیر این صورت معادل با یک مجموعه فازی دلخواه خواهد بود.

به طور مشابه سیستم فازی $\hat{g}(X|\theta_g)$ را از $\prod_{i=1}^n q_i$ قاعده ،

(9-1) اگر $A_1^{l_1}, x_1$ است و ... و $B_n^{l_n}, x_n$ است ، آنگاه \hat{g} ، $H^{l_1 \dots l_n}$ است .

بسازید که $i = 1, 2, \dots, n$ ، $(l_i = 1, 2, \dots, q_i)$ و $H^{l_1 \dots l_n}$ معادل با D^s می باشد. اگر قسمت اگر رابطه (9-1) با قسمت اگر رابطه (4-1) مطابقت داشته باشد و در غیر این صورت معادل با یک مجموعه فازی دلخواه خواهد بود .

در حالت خاص ، با استفاده از موتور استنتاج¹ ضرب ، فازی ساز منفرد و غیر فازی ساز میانگین مراکز² ، بدست خواهیم آورد :

$$\hat{f}(X|\theta_f) = \frac{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \bar{y}_f^{l_1 \dots l_n} [\prod_{i=1}^n \mu_{A_i}^{l_i}(x_i)]}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} [\prod_{i=1}^n \mu_{A_i}^{l_i}(x_i)]} \quad (10-1)$$

$$\hat{g}(X|\theta_g) = \frac{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \bar{y}_g^{l_1 \dots l_n} [\prod_{i=1}^n \mu_{B_i}^{l_i}(x_i)]}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} [\prod_{i=1}^n \mu_{B_i}^{l_i}(x_i)]} \quad (11-1)$$

فرض کنید $\bar{y}_f^{l_1 \dots l_n}$ و $\bar{y}_g^{l_1 \dots l_n}$ پارامترهای آزاد باشند که به ترتیب در $\theta_f \in R^{\prod_{i=1}^n p_i}$ و $\theta_g \in R^{\prod_{i=1}^n q_i}$

جمع آوری شده باشند ، بنابراین ما روابط (10-1) و (11-1) را بصورت زیر بازنویسی می کنیم .

$$\hat{f}(X|\theta_f) = \theta_f^T \xi(x) \quad (12-1)$$

$$\hat{g}(X|\theta_g) = \theta_g^T \eta(x) \quad (13-1)$$

که $\xi(x)$ یک بردار $\prod_{i=1}^n p_i$ بعدی با عناصر $l_1 \dots l_n$ ام خودش می باشد :

$$\xi_{l_1 \dots l_n}(X) = \frac{\prod_{i=1}^n \mu_{A_i}^{l_i}(x_i)}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} [\prod_{i=1}^n \mu_{A_i}^{l_i}(x_i)]} \quad (14-1)$$

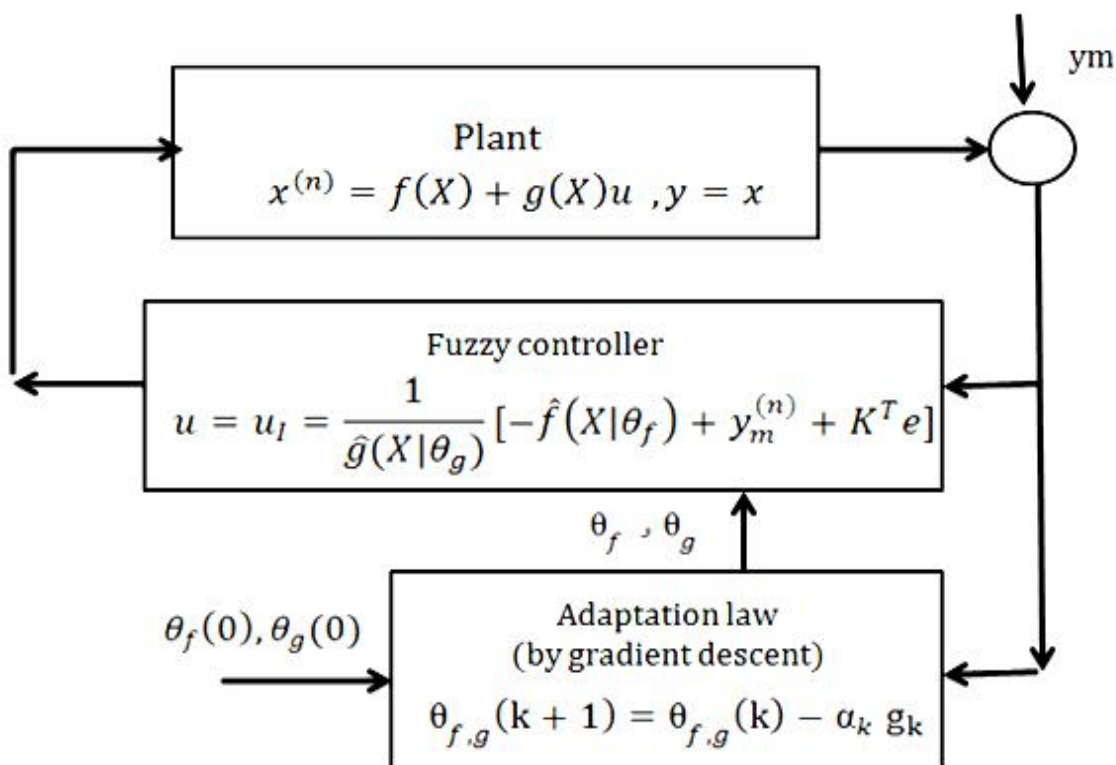
که $\eta(x)$ یک بردار $\prod_{i=1}^n q_i$ بعدی با عناصر $l_1 \dots l_n$ ام خودش می باشد :

$$\eta_{l_1 \dots l_n}(X) = \frac{\prod_{i=1}^n \mu_{B_i}^{l_i}(x_i)}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} [\prod_{i=1}^n \mu_{B_i}^{l_i}(x_i)]} \quad (15-1)$$

از گام دوم مشاهده می نمائیم که برخی پارامترها در θ_f, θ_g مطابق با قواعد (3-1) ، (4-1) انتخاب می گردند و سایر پارامترهای باقی مانده در θ_f, θ_g در طی عملیات بر خط تغییر می یابند ، این پارامترها به عنوان پارامترهای اولیه محسوب می شوند .

وظیفه بعدی ما طراحی یک قاعده تعدیل برای θ_f, θ_g می باشد ، بطوریکه خطای ردیابی e حداقل گردد . که ما قاعده تعدیل را به کمک گرادیان نزولی¹ طراحی می کنیم . بحث گرادیان نزولی در بخش بعدی آمده است .

سیستم کنترل فازی تطبیقی غیر مستقیم در شکل (۳-۱) مشاهده می شود . [2]



شکل ۳-۱ سیستم کنترل فازی تطبیقی غیر مستقیم

مطالب بخش دوم : گرادیان نزولی

۲-۱. معرفی گرادیان نزولی

۲-۲. روش های گرادیان نزولی

۲-۲-۱. روش شدیدترین نزول

۲-۲-۲. روش گرادیان نزولی باممنتوم

۲-۲-۳. روش نیوتن

۲-۲-۴. روش لیونبرگ - مارکوارد

۲-۳. یافتن مینیمم تابع $f(x)$ با استفاده از الگوریتم گرادیان نزولی

۲-۴. طراحی سیستم های فازی با استفاده از روش آموزش گرادیان نزولی

۲-۴-۱. انتخاب ساختار سیستم فازی

۲-۲-۲. طراحی پارامترها بوسیله گرادیان نزولی

۲-۴-۳. طراحی سیستم های فازی با استفاده از روش آموزش گرادیان نزولی

۱-۲. معرفی گرادیان نزولی

الگوریتم گرادیان نزولی پایه بسیاری از الگوریتم‌های بهینه‌سازی^۱ است و از آن در یادگیری شبکه‌های عصبی - فازی^۲ و حداقل‌سازی مقدار خطای شبکه برای تنظیم پارامترهای شبکه استفاده می‌شود. در این الگوریتم تنظیم پارامترهای سیستم فازی و یا تنظیم بایاس ها و وزن های شبکه عصبی در جهتی است که تابع هدف^۳ به شدت کاهش یابد .

دسته وسیعی از تحقیقات برای افزایش سرعت این الگوریتم در آموزش شبکه‌های عصبی چندلایه پیشرو انجام شده است؛ که از نتایج آن می‌توان به انواع روش‌های گرادیان مرتبه اول شامل استفاده از مومنتم^۴، نرخ آموزش وقتی، RPROP و انواع روش‌های گرادیان مرتبه دوم شامل روش‌های نیوتن، گرادیان توام و LM اشاره کرد.

در روش های بهینه سازی که معمولا سیستم مورد نظر غیر خطی است یک تابع هدف بصورت زیر داریم . هدف در روش های بهینه سازی ، مینیمم کردن تابع هدف که برحسب خطا تعریف می شود ، می باشد .

$$J(\mathbf{x}) = \frac{1}{2} \mathbf{e}^T \mathbf{e} = \frac{1}{2} \sum_{i=1}^N \mathbf{e}_i^2 \quad (1-2)$$

N تعداد داده های آموزشی در الگوریتم ، \mathbf{e} بردار خطای بین خروجی واقعی سیستم با خروجی مدلی که با الگوریتم گرادیان نزولی آموزش دیده ، می باشد .

-
1. Optimizing Algorithm
 2. Fuzzy – Neural networks
 3. Objective Function
 4. Momentum

بصورت زیر در یک جهت پارامترها یعنی \mathbf{x} را تغییر داده (k): مرحله k ام از جدید کردن پارامتر \mathbf{x} می باشد). طوری که به ازای این \mathbf{x} جدید، تابع هدف نسبت به حالت قبلی بهینه تر شود در

نتیجه \mathbf{x} به سمت نقطه هدف رفته به طوری که $\left. \frac{\partial J}{\partial \mathbf{x}} \right|_{\mathbf{x}(k+1)} = \mathbf{0}$ گردد.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (2-2)$$

α_k طول گام یا سرعت آموزش و \mathbf{d} جهت جستجو یا جهت جدید شدن پارامتر \mathbf{x} می باشد.

این جهت \mathbf{d} ، در روش گرادیان نزولی براساس مشتق یا گرادیان تعیین می گردد.

گرادیان تابع هدف نسبت به پارامتر \mathbf{x} بصورت زیر تعریف می گردد.

$$\mathbf{g} = \nabla_{\mathbf{x}} J = \frac{\partial J}{\partial \mathbf{x}} \quad (3-2)$$

در نتیجه از رابطه گرادیان (3-2) داریم:

$$\Delta J = \mathbf{g}^T \Delta \mathbf{x} \quad (4-2)$$

$$\Delta \mathbf{x} = \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{d}_k \quad (5-2)$$

چون هدف مینیمم کردت تابع J و $\left. \frac{\partial J}{\partial \mathbf{x}} \right|_{\mathbf{x}(k+1)} = \mathbf{0}$ نقطه مینیمم را میدهد در نتیجه J یک تابع

نزولی است که $J_{k+1} < J_k$ پس $\Delta J < 0$. از طرفی از رابطه (5-2, 4-2) داریم:

$$\Delta J = \alpha \mathbf{g}^T \mathbf{d} < 0 \quad -$$

α یک اسکالر مثبت و \mathbf{g} گرادیان تابع هدف نسبت به پارامتر یا همان مشتق است. با انتخاب

$\mathbf{d} = -\mathbf{Gg}$ خواهیم داشت:

$$\Delta J = -\alpha \mathbf{g}^T \mathbf{Gg} < 0 \quad (6-2)$$

برای درستی عبارت بالا چون $\alpha > 0$ باید $\mathbf{g}^T \mathbf{Gg} > 0$ پس G ماتریس مثبت معین باشد. [1]

۲-۲. روش های گرادیان نزولی

روش های مختلفی از گرادیان نزولی اعم از مرتبه اول و مرتبه دوم تاکنون ارائه شده است در زیر به معرفی مختصری از چند نمونه از این روش ها می پردازیم .

۲-۲-۱. روش شدیدترین نزول^۱

ساده ترین کاربرد از آموزش گرادیان نزولی ، جدید کردن پارامترهای سیستم در جهتی است که تابع هدف سریعاً کاهش یابد . یعنی در رابطه $\mathbf{d} = -\mathbf{G}\mathbf{g}$ در صورتیکه ماتریس \mathbf{G} را برابر با ماتریس همانی در نظر بگیریم در نتیجه $\mathbf{d} = -\mathbf{g}$ خواهد بود و رابطه (2-2) بصورت زیر بیان می شود .

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \quad (7-2)$$

تمام روش های گرادیان نزولی از این روش کلی (شدیدترین نزول) استفاده می کنند .

به طور کلی دو راه مختلف برای الگوریتم آموزش گرادیان نزولی وجود دارد :

۱- آموزش بصورت دسته ای^۲

۲- آموزش بصورت الگو به الگو^۳

۱- آموزش بصورت دسته ای : در مد دسته ای یا محلی ، ابتدا تمام داده های آموزشی به الگوریتم داده شده ، سپس آموزش انجام می شود . یعنی در این مد ، همه ورودی ها در شبکه به کار گرفته شده و براساس خطای مجموع داده های آموزشی بررسی می شود که چند مرحله نیاز به جدید کردن پارامترهای سیستم می باشد . (آموزش بصورت آفلاین^۴)

1 . Steepest Descent
2. Batch
3. Pattern to Pattern
4. Offline

۲- آموزش بصورت الگو به الگو :

در این مد بعد از ورود هر یک از داده ها ، آموزش شروع شده و پارامترها فقط مخصوص همان داده جدید می شوند . یعنی در این راه در هر مرحله یک الگوی خاص برای یک داده آموزشی وجود دارد که این الگو برای داده آموزشی دیگر استفاده نمی شود .
در این روش معیار تکرار آموزش برای هر داده آموزشی ، خطای مربوط به همان یک داده آموزشی است . در واقع این روش آموزش بصورت آنلاین^۱ می باشد .

۲-۲-۲ . روش گرادیان نزولی با ممنتوم

این الگوریتم آموزش اجازه می دهد یک شبکه فقط به گرادیان محلی پاسخ ندهد بلکه به جهت جدید در سطح خطا پاسخ دهد . فعالیت این الگوریتم آموزش شبیه به یک فیلتر پائین گذر می باشد . ممنتوم اجازه می دهد شبکه از نقاط کوچک در سطح خطا چشم پوشی کند . بدن ممنتوم ، یک شبکه می تواند یک فرورفتگی در یک مینیمم محلی کم عمق را بیابد در صورتیکه با ممنتوم به سمت نقطه مینیمم سرازیر می گردد .
این الگوریتم بصورت زیر بیان می گردد :

$$\Delta \mathbf{x}_{(k)} = \mu \Delta \mathbf{x}_{(k-1)} + (1 - \mu)(-\alpha_k \mathbf{g}_k) \quad (8-2)$$

μ ثابت ممنتوم مقداری بین صفر و یک را می گیرد . در یک شبکه که کاملاً غیر حساس به گرادیان محلی باشد ثابت ممنتوم برابر با مقدار یک می باشد که بنابراین در این حالت شبکه نمی تواند یاد بگیرد و در نتیجه آموزش به درستی انجام نمی شود .

$\mu = 0$ همان روش شدیدترین نزول خواهد بود . [1]

۲-۲-۳. روش نیوتن^۱

این روش ، جزء روش های گرادیان مرتبه دوم می باشد که برای بهینه سازی سریع مورد استفاده قرار می گیرد. الگوریتم آن بصورت زیر است :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \quad (9-2)$$

\mathbf{H} همان ماتریس هرمیشن^۲ (مشتق مرتبه دوم) است که بصورت زیر تعریف می شود :

$$\mathbf{H} = \frac{\partial^2 J}{\partial \mathbf{x} \partial \mathbf{x}^T} \quad (10-2)$$

روش نیوتن اغلب از روش های گرادیان ترکیبی سریعتر همگرا می شود .

محاسبه ماتریس هرمیشن برای شبکه های عصبی پیشرو ، پیچیده و گران است ، که در این حالت یک گروه از الگوریتم هایی که بر پایه روش نیوتن است اما نیاز به محاسبه مشتق دوم گرادیان تابع هدف (هرمیشن) ندارد ، استفاده می شود . [1]

۲-۲-۴. روش لیونبرگ - مارکوارد^۳

در روش نیوتن در صورتیکه ماتریس هرمیشن مربعی نباشد و یا دترمینال ماتریس صفر باشد نمی توان معکوس آن را برای جدید کردن پارامترهای شبکه استفاده کرد .

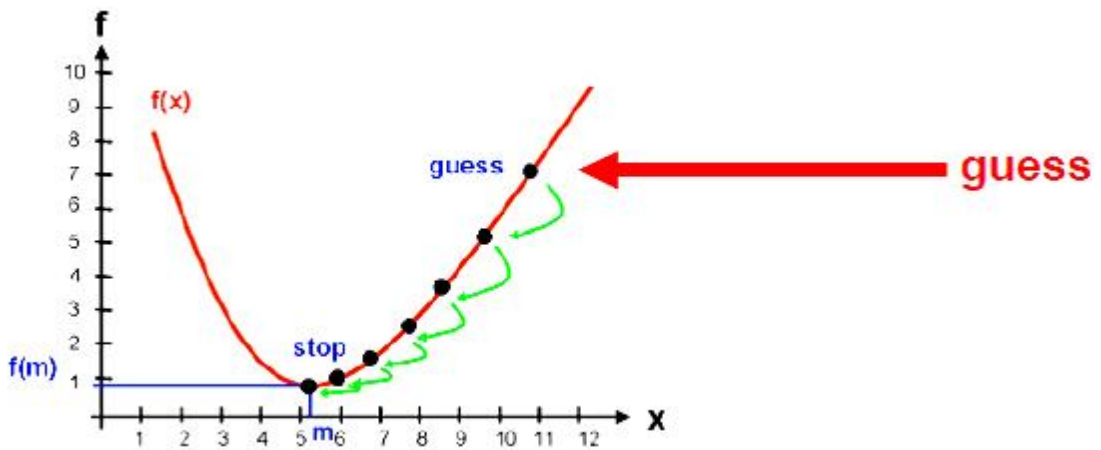
برای رفع این مشکل، روش لیونبرگ - مارکوارد الگوریتم زیر را پیشنهاد می دهد: [1]

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{H}_k + \lambda_k \mathbf{I})^{-1} \mathbf{g}_k \quad (11-2)$$

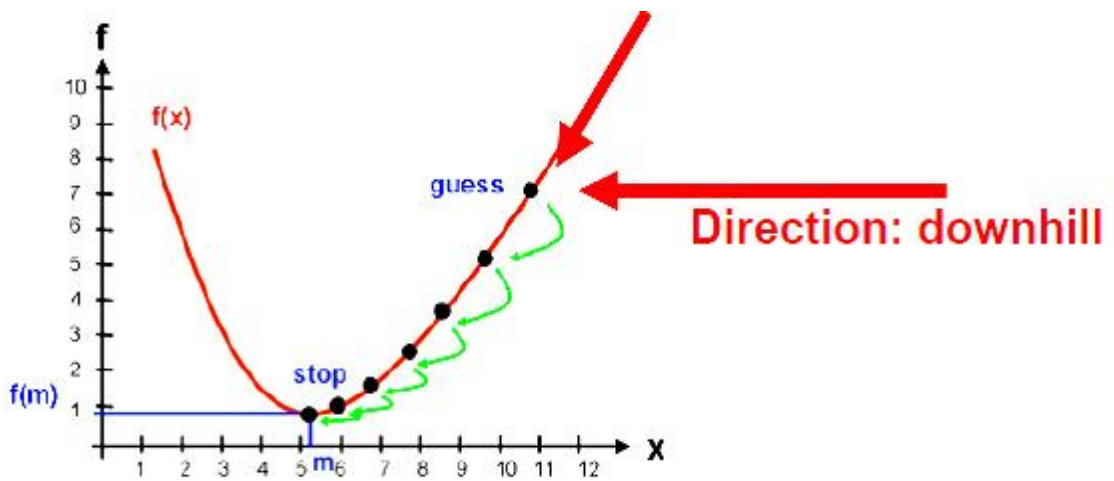
-
1. Nioton
 2. Hermittion
 3. Lewenberg - Marquard

۲-۳. یافتن مینیمم تابع $f(x)$ با استفاده از الگوریتم گرادینان نزولی

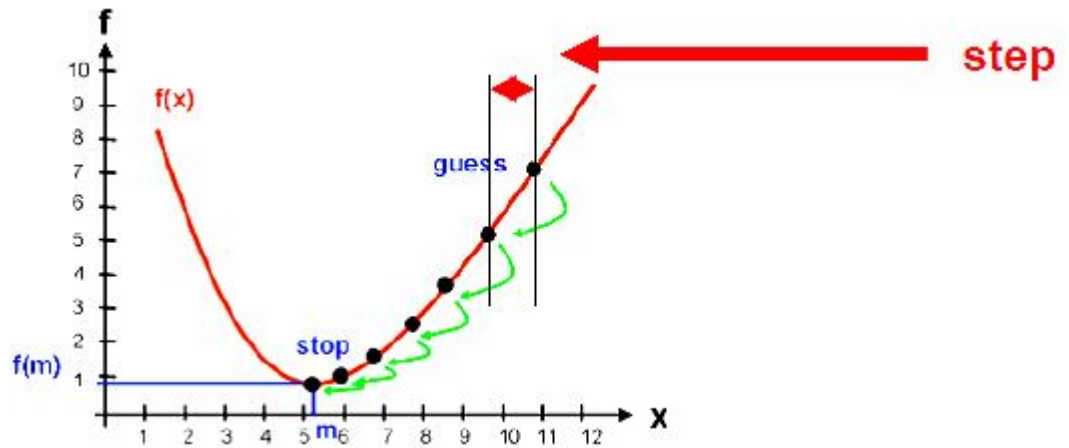
- در مرحله اول ، نقطه ی شروع برای حرکت را تعیین می کنیم . (نقطه ی شروع ، همان $guess$ می باشد) .



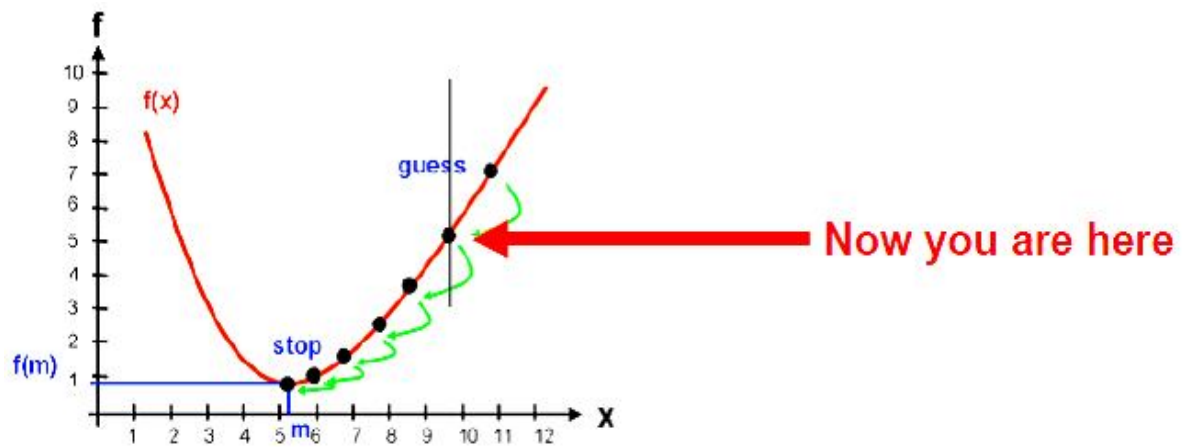
- در مرحله دوم ، جهت حرکت به سمت پایین (جهت نزول) را در نقطه ی شروع تعیین می کنیم .



- در مرحله سوم ، گام حرکت را تعیین میکنیم .



- در مرحله چهارم ، نقطه ی جدید (X) پیدا می شود .

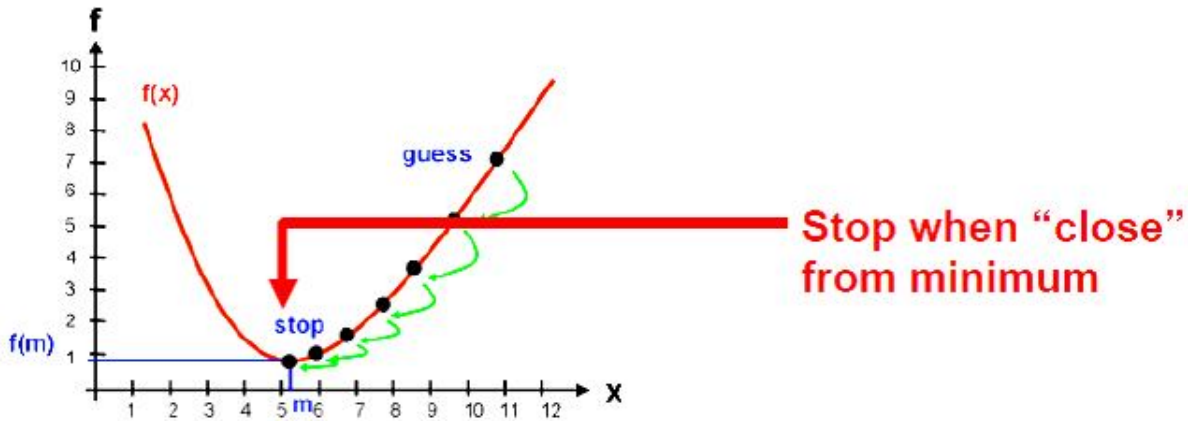


- مراحل ۲ و ۳ و ۴ برای نقطه ی جدید تکرار می شود .

تکرار این مراحل تا زمانی ادامه پیدا می کند که به نقطه هدف (نقطه ای که به ازای آن تابع

$f(x)$ مینیمم می گردد) برسیم .

بعد از رسیدن به نقطه ی مینیمم، این الگوریتم متوقف می شود .



در نتیجه می توان الگوریتم گرادیان نزولی برای یافتن مینیمم تابع $f(x)$ را بصورت زیر نوشت:

مرحله ۱) انتخاب مقدار اولیه x_0 برای پارامتر x_0 و انتخاب پارامتر همگرایی $\epsilon > 0$.

مرحله ۲) محاسبه گرادیان c_k تابع $f(x)$ در نقطه x_k . $(c_k = \Delta f(x_k))$

اگر $\|c_k\| < \epsilon$ ، پس تکرار آموزش را متوقف کن و $x^* = x_k$ قرار بده که x^* نقطه ی مینیمم می باشد

. در غیر این صورت باید به مرحله ی ۳ رفت .

مرحله ۳) محاسبه جهت حرکت در نقطه ی x_k .

$$d_k = -c_k \quad (12-2)$$

مرحله ۴) محاسبه اندازه ی گام حرکت α_k برای مینیمم سازی تابع $f(x)$.

مرحله ۵) متغییر x بصورت زیر تنظیم می شود :

$$x_{k+1} = x_k + \alpha_k d_k \quad (13-2)$$

و سپس به ازای مقدار جدید x ، به مرحله ۲ رفته و مراحل تکرار می شود.

-
1. Initial value
 2. Gradient
 3. Step size

۲-۴. طراحی سیستم های فازی با استفاده از روش گرادیان نزولی

۲-۴-۱. انتخاب ساختار سیستم فازی

در این قسمت ، روشی را برای طراحی سیستم های فازی معرفی می کنیم ، که در آن توابع تعلق براساس معیارهایی به شکلی بهینه انتخاب خواهند شد . در این روش ابتدا ساختار سیستم فازی مشخص می شود ، که در این ساختار تعدادی پارامتر متغییر وجود خواهد داشت که در مرحله بعد این پارامترها از روی داده های ورودی - خروجی معین می شوند .

در این صورت ابتدا ساختار سیستم فازی را مشخص می کنیم . در اینجا ما سیستم فازی را با موتور استنتاج ضرب ، فازی ساز منفرد ، غیر فازی ساز میانگین مراکز و توابع تعلق گوسین ، مطابق رابطه (14-2) طراحی می کنیم .

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l \left[\prod_{i=1}^n \exp \left(-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]}{\sum_{l=1}^M \left[\prod_{i=1}^n \exp \left(-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \right]} = \frac{\sum_{l=1}^M \bar{y}^l z^l}{\sum_{l=1}^M z^l} = \frac{a}{b} \quad (14-2)$$

که M ثابت \bar{y}^l و σ_i^l و \bar{x}_i^l و پارامترهای متغییری می باشند که به روش گرادیان نزولی تنظیم می گردند. اگر چه ساختار سیستم فازی مطابق رابطه (14-2) انتخاب شده ، ولی سیستم فازی هنوز طراحی نشده است ، چرا که پارامترهای \bar{y}^l و σ_i^l و \bar{x}_i^l مشخص نشده اند و با تعیین پارامترهای \bar{y}^l و σ_i^l است که سیستم فازی طراحی خواهد شد ، به عبارت دیگر طراحی سیستم فازی معادل تعیین پارامترهای \bar{y}^l و σ_i^l و \bar{x}_i^l می باشد .

برای تعیین بهینه این پارامترها نمایش سیستم فازی $f(x)$ به شکل یک شبکه پیش خورد ، مفید می باشد .

بطور مشخص ، نگاشتی از ورودی $x \in R^n$ به خروجی $f(x) \in V \subset R$ را می توان مطابق عملیات زیر پیاده سازی نمود :

ابتدا ورودی x از یک عملگر ضرب گوسین عبور کرده و به $z^l = \prod_{i=1}^n \exp \left(-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right)$ تبدیل می شود . سپس z^l از یک عملگر جمع کننده عبور کرده و عملگر جمع کننده وزنی مقادیر $a = \sum_{l=1}^M \bar{y}^l z^l$ ، $b = \sum_{l=1}^M z^l$ را نتیجه می دهد . سرانجام خروجی سیستم فازی بصورت $f(x) = \frac{a}{b}$ محاسبه می شود. [2]

۲-۴-۲. طراحی پارامترها بوسیله گرادیان نزولی

هدف ما طراحی سیستم فازی $f(x)$ مطابق رابطه (2-14) است به نحوی که خطای تطبیق $e^p = \frac{1}{2} [f(x_0^p) - y_0^p]^2$ مینیمم گردد . بدین معنی که هدف تعیین پارامترهای \bar{y}^l و σ_i^l و \bar{x}_i^l است ، به نحوی که e^p کمترین مقدار ممکن را داشته باشد . در ادامه بحث از e ، f و y به ترتیب برای نمایش e^p ، $f(x_0^p)$ ، y_0^p استفاده می کنیم .

ما از الگوریتم گرادیان نزولی برای تعیین این پارامترها استفاده می کنیم ، به عبارت دیگر برای تعیین \bar{y}^l از الگوریتم زیر استفاده می کنیم :

$$\bar{y}^l(q+1) = \bar{y}^l(q) - \alpha \left. \frac{\partial e}{\partial \bar{y}^l} \right|_q \quad (15-2)$$

که $q=0,1,2,\dots$ ، $l=1,2,\dots,M$ و α یک عدد ثابت بوده که طول گام را مشخص می کند . اگر هنگامی که q به سمت بی نهایت می رود $\bar{y}^l(q)$ همگرا شود ، آنگاه از روی (2.15) در نقطه همگرایی

داریم $\left. \frac{\partial e}{\partial \bar{y}^l} \right|_q = 0$ که این بدین معنی است که همگرایی \bar{y}^l در یک مینیمم محلی e می باشد .

لازم به ذکر است که f و در نتیجه e فقط از طریق a به \bar{y}^l بستگی دارد ، که $f(x) = \frac{a}{b}$ ،

$$z^l = \prod_{i=1}^n \exp \left(-\left(\frac{x_i - \bar{x}_i^l}{\sigma_i^l} \right)^2 \right) \text{ و } b = \sum_{l=1}^M z^l \text{ ، } a = \sum_{l=1}^M \bar{y}^l z^l$$

بنابراین با استفاده از قاعده زنجیره ای مشتق داریم :

$$\frac{\partial e}{\partial \bar{y}^l} = (f - y) \frac{\partial f}{\partial a} \frac{\partial a}{\partial \bar{y}^l} = (f - y) \frac{1}{b} z^l \quad (16-2)$$

با جایگذاری (16-2) در رابطه (15-2) ، الگوریتم آموزش را برای \bar{y}^l بدست می آوریم :

$$\bar{y}^l(q + 1) = \bar{y}^l(q) - \alpha \frac{(f-y)}{b} z^l \quad (17-2)$$

که $l=1,2,\dots,M$ ، $q=0,1,2,\dots$ می باشد .

برای تعیین \bar{x}_i^l داریم :

$$\bar{x}_i^l(q + 1) = \bar{x}_i^l(q) - \alpha \left. \frac{\partial e}{\partial \bar{x}_i^l} \right|_q \quad (18-2)$$

که $l=1,2,\dots,M$ ، $q=0,1,2,\dots$ و $i=1,2, \dots, n$. لازم به ذکر است که f و در نتیجه e فقط از

طریق z^l به \bar{x}_i^l بستگی دارد ، بنابراین با استفاده از قانون زنجیره ای خواهیم داشت :

$$\frac{\partial e}{\partial \bar{x}_i^l} = (f - y) \frac{\partial f}{\partial z} \frac{\partial z^l}{\partial \bar{x}_i^l} = (f - y) \frac{(\bar{y}^l - f)}{b} z^l \frac{2(x_{0i}^p - \bar{x}_i^l)}{\sigma_i^2} \quad (19-2)$$

با جایگذاری (19-2) در (18-2) ، الگوریتم آموزش را برای \bar{x}_i^l بدست می آوریم :

$$\bar{x}_i^l(q + 1) = \bar{x}_i^l(q) - \alpha (f - y) \frac{(\bar{y}^l - f)}{b} z^l \frac{2(x_{0i}^p - \bar{x}_i^l)}{\sigma_i^2} \quad (20-2)$$

که $l=1,2,\dots,M$ ، $q=0,1,2,\dots$ و $i=1,2, \dots, n$.

با استفاده از همین روال ، الگوریتم آموزش را برای σ_i^l بدست می آوریم :

$$\sigma_i^l(q + 1) = \sigma_i^l(q) - \alpha \left. \frac{\partial e}{\partial \sigma_i^l} \right|_q = \sigma_i^l(q) - \alpha (f - y) \frac{(\bar{y}^l - f)}{b} z^l \frac{2(x_{0i}^p - \bar{x}_i^l)^2}{\sigma_i^3} \quad (21-2)$$

که $l=1,2,\dots,M$ ، $q=0,1,2,\dots$ و $i=1,2, \dots, n$.

الگوریتم آموزش (17-2) و (20-2) و (21-2) یک روال پس انتشار خطا را انجام می دهند . برای

آموزش \bar{y}^l ، خطای نرمال شده $\frac{(f-y)}{b}$ به لایه \bar{y}^l برگشت داده می شود، سپس \bar{y}^l مطابق رابطه (17-2)

تصحیح می شود ، که در آن z^l ورودی \bar{y}^l می باشد .

برای آموزش \bar{x}_i^1 و σ_i^1 ، خطای نرمال شده $z^1 \frac{(\bar{y}^1 - f)}{b}$ به لایه ای که خروجی آن z^1 است ، برگشت داده می شود . آنگاه \bar{x}_i^1 و σ_i^1 مطابق رابطه (20-2) ، (21-2) تصحیح می شوند و متغیرهای باقیمانده x_{0i}^p ، \bar{x}_i^1 و σ_i^1 را می توان به صورت محلی بدست آورد . با توجه به مطالب گفته شده ، این الگوریتم را الگوریتم آموزش پس انتشار خطا نیز می نامند . [2,3,4,5]

۳-۴-۲ . طراحی سیستم های فازی با استفاده از روش آموزش گرادیان نزولی

- **گام اول : تعیین ساختار و پارامترهای اولیه .** سیستم فازی را به شکل رابطه (2-14) انتخاب کرده و مقدار M را تعیین کنید . انتخاب M بزرگتر ، پارامترهای بیشتری را بوجود آورده و حجم محاسبات را بیشتر می کند ولی در عین حال تقریب با دقت بهتری را نتیجه می دهد . پارامترهای اولیه $\bar{y}^1(0)$ و $\sigma_i^1(0)$ و $\bar{x}_i^1(0)$ را مشخص کنید . این پارامترهای اولیه را می توان مطابق قواعد زبانی از روی دانش انسان خبره تعیین کرد و یا به نحوی انتخاب کرد که توابع تعلق متناظر به شکل یکنواخت فضای ورودی - خروجی را در بر بگیرند .
- **گام دوم : اعمال ورودی و محاسبه خروجی سیستم فازی .** برای یک زوج ورودی - خروجی داده شده $(x_0^p; y_0^p)$ ، $p=1,2,\dots$ و در مرحله q آموزش ، $q=0,1,2,\dots$ ، x_0^p را به لایه ورودی سیستم فازی اعمال کرده و a ، b و z^1 و در نتیجه $f(x) = \frac{a}{b}$ را محاسبه کنید .
- **گام سوم : تصحیح پارامترها .** با استفاده از الگوریتم آموزش مطابق روابط (2-17) ، (2-20) (21-2) ، پارامترهای تصحیح شده را محاسبه کنید .

- **گام چهارم:** روال فوق را بارفتن به گام ۲ با $q=q+1$ تکرار کنید تا آنجاکه $|f - y_0^p|$ از یک مقدار مشخص ε کوچکتر شده و یا از q به یک عدد از قبل مشخص شده برسد .

- **گام پنجم:** روال فوق را با رفتن به گام ۲ با $p=p+1$ تکرار کنید . بدین معنی که پارامترها را با استفاده از زوج ورودی - خروجی بعدی $(x_0^{p+1}; y_0^{p+1})$ تصحیح کنید .

- **گام ششم:** در صورت لزوم و امکان ، $p=1$ قرار داده و مراحل ۲ تا ۵ را تا رسیدن به یک سیستم فازی رضایت بخش تکرار کنید . برای کنترل بلادرنگ و شناسایی سیستم های دینامیک این مرحله امکان پذیر نیست چرا که داده های ورودی - خروجی یکی پس از دیگری دریافت می شوند . برای مسائل شناسایی الگو ، از آنجا که داده های ورودی - خروجی بطور همزمان وجود دارند ، این مرحله معمولا امکان پذیر می باشند .

بدلیل اینکه الگوریتم آموزش مطابق روابط (2-17) و (2-20) و (2-21) ، یک الگوریتم گرادیان نزولی است ، انتخاب پارامترهای اولیه برای نتیجه بخش بودن الگوریتم بسیار مهم است .

اگر پارامترهای اولیه نزدیک به مقادیر بهینه انتخاب شوند ، الگوریتم شانس زیادی دارد که پارامترها به مقدار بهینه همگرا شوند ، در غیر این صورت الگوریتم ممکن است به یک مقدار غیر بهینه همگرا شده و یا حتی واگرا شوند . مزیت استفاده از سیستم فازی این است که پارامترهای \bar{y}_i^1 و σ_i^1 و \bar{x}_i^1 مفاهیم فیزیکی مشخص داشته و روش هایی برای انتخاب مقادیر اولیه خوب برای آن ها وجود دارد .

فرض کنید پارامترهای \bar{y}_i^1 مرکز مجموعه های فازی در بخش آنگاه قواعد فازی و پارامترهای σ_i^1 و \bar{x}_i^1 مراکز و پهنای توابع تعلق گوسین در بخش اگر قواعد باشند . بنابراین با داشتن یک سیستم فازی طراحی شده ، ما می توانیم قواعد اگر - آنگاه که آن سیستم را تشکیل می دهند را بدست آوریم . این قواعد بازیابی شده از روی سیستم ، می تواند برای تشریح سیستم فازی طراحی شده به ما کمک کند

[2,3,4,5] .

مطالب بخش سوم : سیستم کنترل فازی تطبیقی غیر مستقیم ربات
" پیوما ۵۶۰ " با روش گرادیان نزولی

۳-۱. سیستم اصلی (ربات پیوما ۵۶۰)

۳-۲. سرو موتور dc

۳-۳. کنترل مفصل مستقل

۳-۳-۱. طراحی سیستم فازی \hat{f} با روش گرادیان نزولی

۳-۳-۲. طراحی سیستم فازی \hat{g} با روش گرادیان نزولی

۳-۳-۳. مرحله ساخت قانون کنترل

سیستم کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان نزولی

سیستم کنترل مورد استفاده شامل قسمت های زیر می باشد :

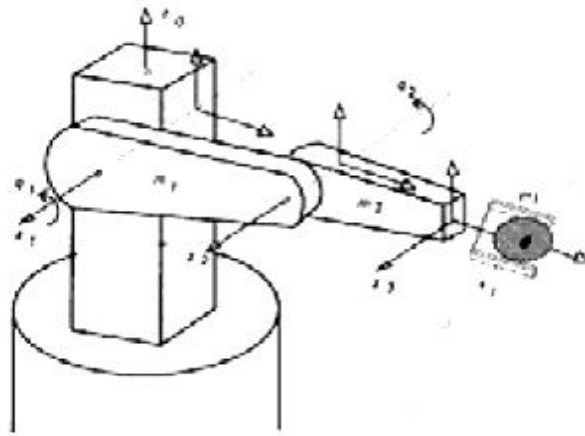
- ۱- سیستم اصلی (ربات پیوما ۵۶۰) ^۱
- ۲- ۶ سروموتور ^۲ برای به حرکت در آوردن ۶ مفصل ربات ^۳
- ۳- ۶ کنترلر برای کنترل ۶ سیگنال ولتاژ هر یک از موتور ها و در نتیجه کنترل ۶ مفصل ربات .
طراحی هر یک از کنترلرها براساس کنترل فازی تطبیقی غیرمستقیم صورت می گیرد که قانون تعدیل در هر یک از آن ها ، همان الگوریتم آموزش گرادیان نزولی است .

۱-۳ . سیستم اصلی (ربات پیوما ۵۶۰)

ربات پیوما ۵۶۰ یک بازوی ماهر از ربات های صنعتی است که برای جابجایی اجسام و قرار دادن آن ها در یک موقعیت مشخص در فضای کار استفاده می گردد .

این بازوی ماهر رباتیک ^۴ دارای ۶ مفصل لولایی ^۵ (۶ درجه آزادی) می باشد که ۳ مفصل اول مربوط به بازو و ۳ مفصل آخر مربوط به مچ می باشد . مجری نهایی از بازوی ربات می تواند به یک نقطه در فضای کار در جهت خاصی برسد . [7]

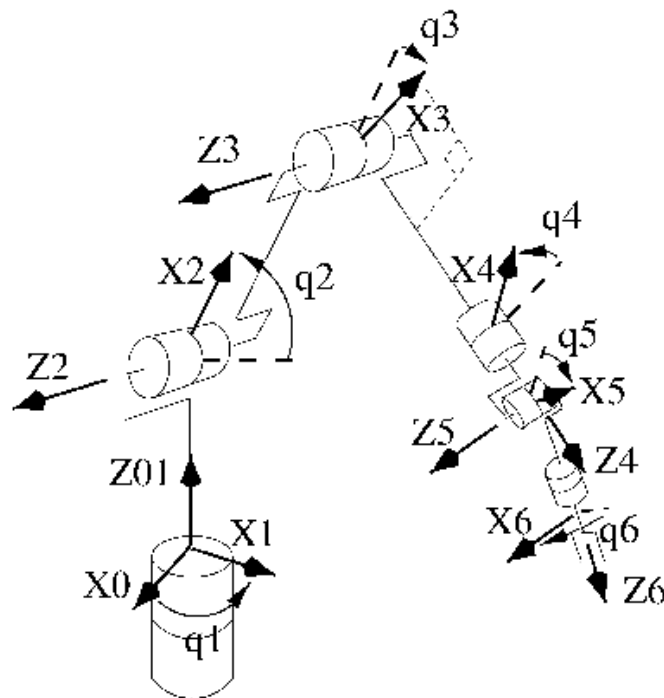
1. Puma 560
2. Servo motor (DC motor)
3. Robot Joints
4. Robotic Mainpulator
5. Revolute joint



1 PUMA robot

شکل ۱-۳ بازوی ماهر ربات " پیوما ۵۶۰ "

دیاگرام مفصلی ربات پیوما ۵۶۰ بصورت زیر می باشد :



شکل ۲-۳ دیاگرام مفصلی ربات " پیوما ۵۶۰ "

جدول پارامترهای دناویت - هارتنبرگ^۱ مربوط به ربات پیوما ۵۶۰ بصورت زیر است: [7,9]

جدول ۱-۳ جدول پارامترهای دناویت - هارتنبرگ

joint	qi	ai	$\alpha_i(\text{degree})$	Di
1	q1	0	90	0
2	q2	0.4318	0	0
3	q3	0.0203	-90	0.15005
4	q4	0	90	0.4318
5	q5	0	-90	0
6	q6	0	0	0

در این پروژه می خواهیم ۶ مفصل از ربات را کنترل نمائیم بطوریکه هر یک از مفاصل ربات ، مسیر مطلوب را طی نمایند . با توجه به آن که دینامیک ربات غیر خطی است و کوپلینگ قوی بین مفاصل وجود دارد در نتیجه خیلی اوقات در معرض نامعینی های ساختاری و غیرساختاری می باشد ، به همین خاطر در این پروژه ، از کنترل فازی تطبیقی غیر مستقیم برای کنترل بازوی ماهر رباتیک به منظور جبران سازی عدم قطعیت ها و نامعینی ها استفاده می گردد . لازم به ذکر است که تطبیق در این سیستم کنترل توسط الگوریتم گرادیان نزولی انجام می گیرد .

برای کنترل ، با توجه به آن که ربات دارای ۶ درجه آزادی است کنترل مفاصل ربات را بطور مستقل انجام می دهیم (استراتژی کنترل مفصل مستقل) و کنترل ، روی سیگنال ولتاژ هر موتور از مفصل ربات انجام می شود و هیچ کنترلی روی گشتاور^۲ مفاصل نداریم .

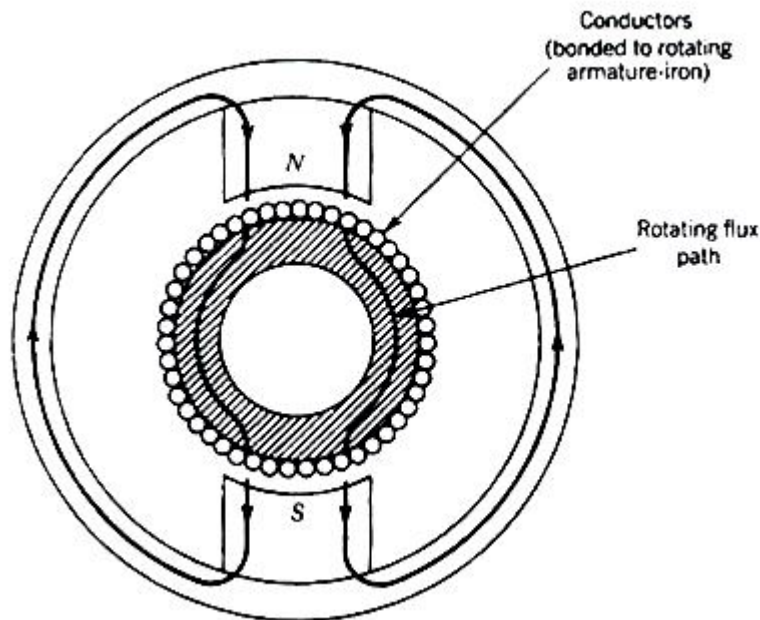
از ۶ موتور برای به حرکت در آوردن ۶ مفصل ربات استفاده می شود که هر یک سرو موتور dc می باشند .

1. Denavit - Hartenberg
2. torque

۳-۲. سرو موتور dc

یکی از محرکه ها برای مفاصل ربات ، موتور dc مغناطیس دائم است . موتور dc اصولا بر این اساس کار می کند که یک هادی حامل جریان در میدان مغناطیسی ، نیروی $F = \varphi \times i$ را حس می نماید که در آن φ شار مغناطیسی و i جریان هادی است . موتور از یک استاتور^۱ ثابت و یک رتور^۲ متحرک که در داخل استاتور می چرخد مطابق شکل ۳-۳ تشکیل می شود .

اگر استاتور یک شار مغناطیسی شعاعی تولید نماید و جریان رتور (آرمیچر هم نامیده می شود) i باشد ، آنگاه گشتاوری روی موتور خواهد بود که باعث چرخیدن آن می شود .



شکل ۳-۳ سطح مقطع موتور DC مغناطیس دائم

اندازه این گشتاور عبارت است از :

$$\tau_m = k_1 \varphi i_a \quad (1-3)$$

که در آن τ_m گشتاور موتور (N-m) ، φ میدان مغناطیسی (و بر) ، i_a جریان آرمیچر (آمپر) و k یک ثابت فیزیکی است .

-
1. Stator
 2. Rotor

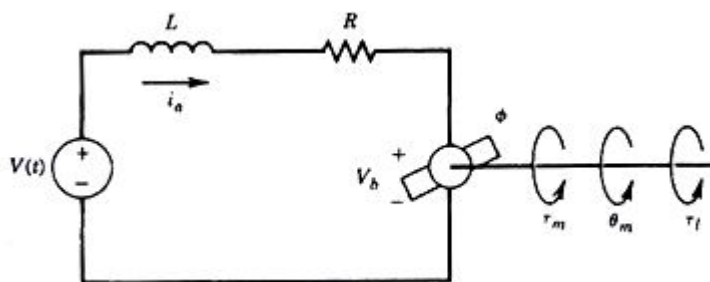
بعلاوه هرگاه یک هادی در یک میدان مغناطیسی^۱ حرکت کند، یک ولتاژ v_b بین ترمینال های آن تولید می شود که متناسب با سرعت هادی در میدان است. این ولتاژ نیروی ضد محرکه الکتریکی نامیده می شود و تمایل به مخالفت با جریان هادی دارد. بنابراین علاوه بر گشتاور τ_m در (1-3) رابطه نیروی ضد محرکه را داریم.

$$v_b = k_2 \phi \omega_m \quad (2-3)$$

که در آن v_b ولتاژ ضد محرکه بر حسب ولت، ω_m سرعت زاویه ای رتور بر حسب (rad/sec) و k_2 ثابت تناسب است.

موتورهای dc به طریقی دسته بندی می شوند که در آن میدان مغناطیسی تولید می شود و آرمیچر طراحی می گردد. بحث ما اینجا فقط روی موتورهای مغناطیس دائم است که استاتور آنها مغناطیس دائم می باشند. در این حالت می توانیم شار ϕ را ثابت فرض کنیم. گشتاور روی رتور با کنترل جریان آرمیچر i_a انجام می شود.

به دیاگرام شماتیک شکل ۳-۴ توجه نمائید که در آن $v(t) =$ ولتاژ آرمیچر و $L =$ اندوکتانس آرمیچر و R مقاومت آرمیچر و $v_b =$ ولتاژ ضد محرکه و $i_a =$ جریان آرمیچر و $\theta_m =$ موقعیت رتور (رادیان) و $\tau_m =$ گشتاور تولیدی و $\tau_l =$ گشتاور بار و ϕ شار مغناطیسی می باشد.



شکل ۳-۴ دیاگرام مدار برای موتور DC کنترل شده با ولتاژ آرمیچر

پس معادله دیفرانسیل جریان آرمیچر به صورت زیر است :

$$L \frac{di_a}{dt} + Ri_a = v - v_b \quad (3-3)$$

از آنجا که شار ϕ ثابت است ، گشتاور تولیدی موتور برابر است با

$$\tau_m = k_1 \phi i_a = k_m i_a \quad (4-3)$$

که در آن k_m ثابت گشتاور برحسب N-m/amp است . از (2-3) داریم :

$$v_b = k_2 \phi \omega_m = k_b \omega_m = k_b \frac{d\theta_m}{dt} \quad (5-3)$$

که در آن k_b ثابت ضد محرکه القایی است .

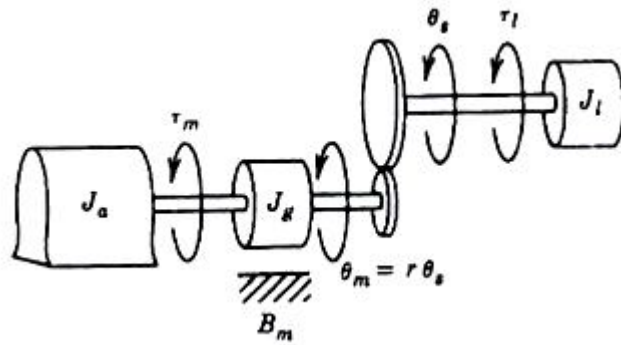
ثابت گشتاور موتور dc را می توانیم با استفاده از دسته منحنی های گشتاور - سرعت برای مقادیر مختلف ولتاژ اعمال شده تعیین نمائیم . وقتی موتور در حالت ترمز است ، گشتاور رتور - قفل در ولتاژ اسمی با τ_o نشان داده می شود . با استفاده از معادله (3-3) برای $v_b = 0$ و $\frac{di_a}{dt} = 0$ داریم :

$$v_r = Ri_a = \frac{R\tau_o}{k_m} \quad (6-3)$$

بنابراین ثابت گشتاور عبارت است از :

$$k_m = \frac{R\tau_o}{v_r} \quad (7-3)$$

برای باقیمانده بحث به شکل ۳-۵ رجوع نمائید که در آن یک موتور dc با اتصال به سیستم انتقال چرخ دنده ای با نسبت $r : 1$ و یک رابط از بازوی ماهر قرار دارد . نسبت دنده r معمولا دارای مقداری بین 0.02 تا 0.002 است که نسبت کاهنده ۵۰ به ۱ تا ۵۰۰ به ۱ را نشان می دهد . با مراجعه به شکل ۳-۳ ، $j_m = j_a + j_g$ است که جمع اینرسی های محرکه و چرخ دنده می باشد .



شکل ۳-۵ مدل فشرده یک رابط با سیستم انتقال چرخ دنده - محرک

پس معادله حرکت این سیستم بصورت زیر می باشد :

$$J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} = \tau_m - r\tau_l = k_m i_a - r\tau_l \quad (8-3)$$

در حوزه لاپلاس سه معادله (3-3)، (8-3)، (5-3) می توانند ترکیب شوند و به صورت زیر نوشته شوند :

$$(Ls + R)I_a(s) = V(s) - k_b s\theta_m(s) \quad (9-3)$$

$$(J_m s^2 + B_m s)\theta_m(s) = k_m I_a(s) - r\tau_l(s) \quad (10-3)$$

بلوک دیاگرام سیستم بالا در شکل ۳-۶ نشان داده شده است . تابع تبدیل از $V(s)$ به $\theta_m(s)$ بصورت

زیر است : (با $\tau_l = 0$)

$$\frac{\theta_m(s)}{V(s)} = \frac{k_m}{s[(Ls+R)(J_m s+B_m)+k_m k_b]} \quad (11-3)$$

تابع تبدیل از گشتاور بار τ_l به $\theta_m(s)$ بصورت زیر است : (با $V = 0$)

$$\frac{\theta_m(s)}{\tau_l(s)} = \frac{-r(Ls+R)}{s[(Ls+R)(J_m s+B_m)+k_m k_b]} \quad (12-3)$$

مکرراً فرض می شود که " ثابت زمانی الکتریکی " $\frac{L}{R}$ " خیلی کوچکتر از " ثابت زمانی مکانیکی " $\frac{J_m}{B_m}$ است. این یک فرض قابل قبول برای همه سیستم های الکترومکانیکی است و آن را به مدل مرتبه پایین تر از دینامیک محرکه رهنمون می سازد .

در نتیجه خواهیم داشت : تابع تبدیل از $V(s)$ به $\theta_m(s)$ بصورت زیر است : (با $\tau_1 = 0$)

$$\frac{\theta_m(s)}{V(s)} = \frac{k_m/R}{s[(J_ms+B_m)+k_mk_b/R]} \quad (11-3)$$

تابع تبدیل از گشتاور بار τ_1 به $\theta_m(s)$ بصورت زیر است : (با $V = 0$)

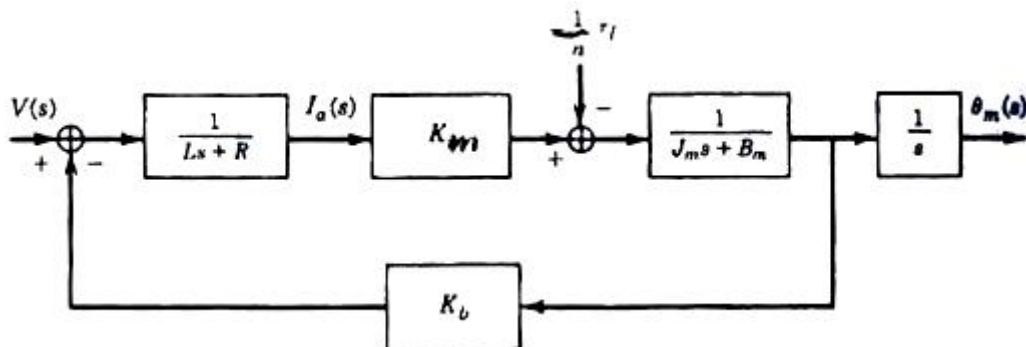
$$\frac{\theta_m(s)}{\tau_1(s)} = \frac{-r}{s[(J_ms+B_m)+k_mk_b/R]} \quad (12-3)$$

$$\theta = r\theta_m \quad (13-3)$$

در حوزه زمان معادلات (11-3) و (12-3) و (13-3) با جمع آثار توسط معادله دیفرانسیل مرتبه دوم عرضه می شوند . [8]

$$J_m\ddot{\theta}(t) + \left(B_m + \frac{k_mk_b}{R}\right)\dot{\theta}(t) = \left(\frac{k_m r}{R}\right)v(t) - r^2\tau_1(s) \quad (14-3)$$

بلوک دیاگرام سیستم موتور dc بصورت زیر می باشد :



شکل ۳-۶ بلوک دیاگرام سیستم موتور dc

۳-۳. کنترل مفصل مستقل

می دانیم سیستم غیر خطی شامل (موتور dc و گشتاور ناشی از یک مفصل از ربات پیوما ۵۶۰) به فرم زیر بوده که در این فرم، توابع f و g ناشناخته هستند.

$$\begin{aligned} \dot{x}^{(n)} &= f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u & - \\ y &= x & - \end{aligned}$$

که با توجه به آنچه در مطالب بخش ۱ بیان شد، برای کنترل چنین سیستمی از قانون کنترل زیر استفاده می گردد:

$$u^* = \frac{1}{g(X)} \left[-f(X) + y_m^{(n)} + K^T e \right]$$

با توجه به آنکه f و g توابع نامعلوم می باشند، در نتیجه برای اعمال قانون کنترل ناچاریم f و g را توسط توابع فازی \hat{f}, \hat{g} که به روش گرادیان نزولی طراحی می گردند، تقریب بزنیم و سپس قانون کنترل u را ساخته و بدین ترتیب سیستم مورد نظر که شامل یک موتور dc و یک مفصل از ربات پیوما ۵۶۰ است را کنترل نمائیم.

با ساختن توابع \hat{f}, \hat{g} (سیستم های فازی که براساس گرادیان نزولی طراحی شده)، قانون کنترل بصورت رابطه زیر خواهد بود:

$$u = \frac{1}{\hat{g}(X|\theta_g)} \left[-\hat{f}(X|\theta_f) + y_m^{(n)} + K^T e \right] \quad (7-1)$$

که همان طور که در بخش ۱ بیان شد با جایگذاری این قانون کنترل (7-1) در (1-1) به معادله دینامیکی زیر خواهیم رسید :

$$e^{(n)} + k_1 e^{(n-1)} + \dots + k_n e = [f - \hat{f}] + u[g - \hat{g}] \quad (15-3)$$

اگر بتوان \hat{g}, \hat{f} را طوری ساخت که $\hat{g} \rightarrow g, \hat{f} \rightarrow f$ میل کند در نتیجه به معادله دینامیکی سیستم حلقه بسته (6-1) خواهیم رسید و با انتخاب مناسب مقادیر k ، قطب های سیستم حلقه بسته درای مقادیری با قسمت حقیقی منفی خواهند بود، در نتیجه سیستم پایدار می شود. پس گام نخست برای طراحی کنترلر انتخاب مکان مناسب برای قطب های سیستم حلقه بسته می باشد تا شرط لازم برای پایداری را فراهم نماید.

سیستم مورد نظر یک موتور dc که باعث به حرکت در آوردن یکی از مفاصل ربات پیوما ۵۶۰ می شود، می باشد. معادله مربوط به موتور همان طور که در قسمت ۲ همین فصل بیان شد به صورت زیر می باشد :

$$J_m \ddot{\theta}(t) + \left(B_m + \frac{k_m k_b}{R} \right) \dot{\theta}(t) = \left(\frac{k_m r}{R} \right) v(t) - r^2 \tau_1(s) \quad (14-3)$$

و همچنین معادله دینامیکی مربوط به یک مفصل ربات که همان گشتاور بار را فراهم می نماید به صورت زیر است :

$$M(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) = \tau_1 \quad (16-3)$$

از روابط (14-3) و (16-3) می توان به معادله دیفرانسیل (18-3) زیر رسید.

$$M1(\theta) = (J_m + r^2 M(\theta)) \quad (17-3)$$

$$\ddot{\theta}(t) = \left(\frac{k_m r}{R} \right) (M1(\theta))^{-1} v(t) - [r^2 H(\theta, \dot{\theta}) + \left(B_m + \frac{k_m k_b}{R} \right) \dot{\theta}(t)] (M1(\theta))^{-1} \quad (18-3)$$

در نتیجه معادله دیفرانسیل سیستم مورد نظر به فرم رابطه (1-1) و (2-1) در فصل ۱ در آمده که در آن :

$$\dot{x}^{(n)} = f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u \quad (1-1)$$

$$y = x \quad (2-1)$$

$$f(\theta, \dot{\theta}) = -[r^2 H(\theta, \dot{\theta}) + (B_m + \frac{k_m k_b}{R}) \dot{\theta}(t)] (M1(\theta))^{-1} \quad (19-3)$$

$$g(\theta) = \left(\frac{k_m r}{R}\right) (M1(\theta))^{-1} \quad (20-3)$$

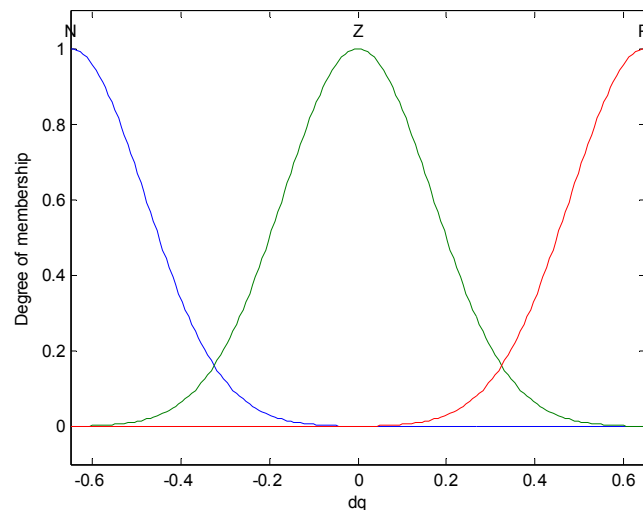
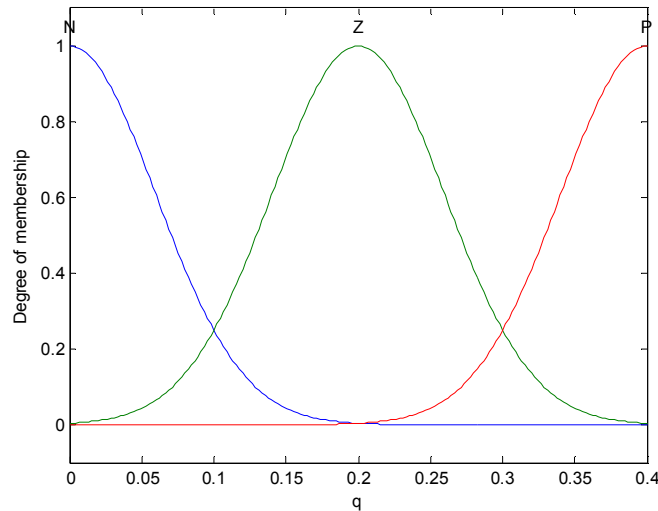
$$y = x = \theta \quad -$$

با توجه به آنکه $M(\theta)$ و $H(\theta, \dot{\theta})$ توابع ناشناخته می باشند ، در نتیجه $g(\theta)$ و $f(\theta, \dot{\theta})$ نیز نامعلوم

هستند ، پس از تقریب آنها یعنی \hat{f} و \hat{g} برای قانون کنترل استفاده می کنیم . [8]

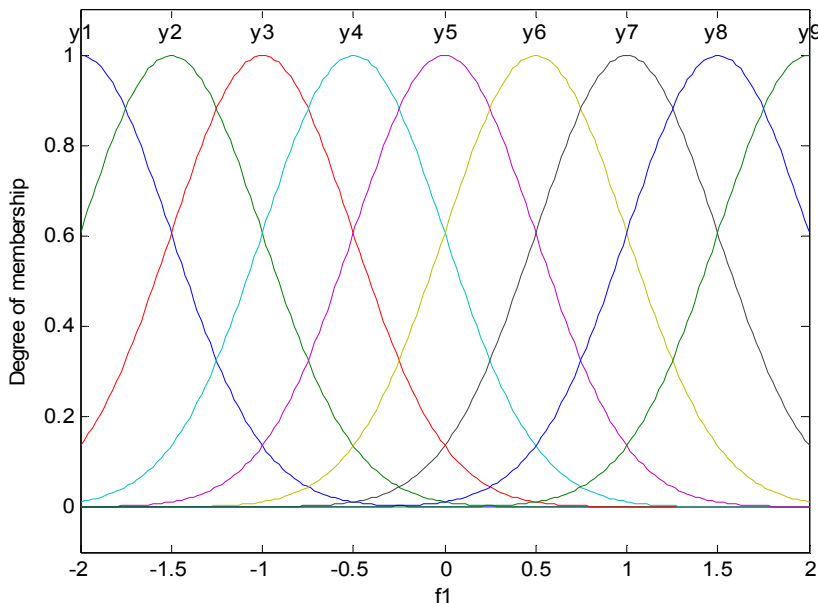
۳-۳-۱. طراحی سیستم فازی \hat{f} با روش گرادیان نزولی

برای طراحی سیستم فازی لازم است ابتدا ورودی های این سیستم تعیین گردد . ورودی های سیستم \hat{f} , dq , q همان مشتق زاویه مفصل از ربات و خود زاویه مفصل ربات می باشد . سپس برای هر یک از این دو ورودی سیستم ، ۳ گروه فازی P و Z و N بصورت توابع عضویت گوسین در نظر می گیریم . توابع عضویت^۱ مربوط به q و dq در شکل زیر قابل مشاهده می باشند که پهنای هر یک از توابع تعلق مربوط به q را 0.06 و مربوط به dq را 0.17 در نظر گرفته ایم .



شکل ۳-۷ گروههای فازی مربوط به q و dq

لازم به ذکر است که در حین استفاده از این سیستم فازی در کنترلر مراکز و پهنای توابع تعلق را ثابت نگه داشته و آنها را تغییر نمی دهیم و فقط مراکز خروجی ها در هر قانون تغییر می کنند ، پس مراکز خروجی ها در هر قانون بعنوان پارامترهای سیستم فازی هستند که قابل تنظیم می باشند . مقادیر اولیه برای مراکز خروجی ها در هر قانون در بازه ای که شامل اعداد کوچک می باشد انتخاب می گردد . ما در اینجا خروجی ها در هر قانون را در بازه ی $[-2,2]$ در نظر گرفتیم و چون ۹ قانون داریم در نتیجه ۹ گروه فازی برای خروجی خواهیم داشت .



شکل ۳-۸ گروههای فازی مربوط به خروجی سیستم فازی \hat{f}

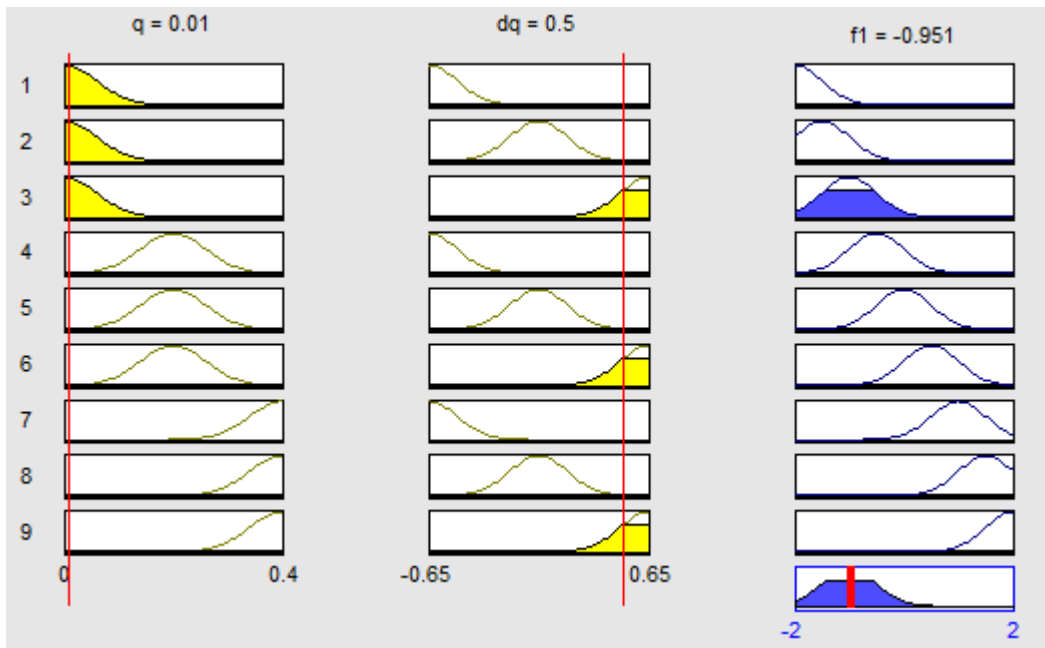
۹ قانون فازی نیز بصورت زیر است که در هر قانون خروجی ها بصورت دلخواه انتخاب شده اند و ما هیچ اطلاعاتی راجع به آنها نداریم :

1. If (q is N) and (dq is N) then (f1 is y1)
2. If (q is N) and (dq is Z) then (f1 is y2)
3. If (q is N) and (dq is P) then (f1 is y3)
4. If (q is Z) and (dq is N) then (f1 is y4)
5. If (q is Z) and (dq is Z) then (f1 is y5)
6. If (q is Z) and (dq is P) then (f1 is y6)
7. If (q is P) and (dq is N) then (f1 is y7)
8. If (q is P) and (dq is Z) then (f1 is y8)
9. If (q is P) and (dq is P) then (f1 is y9)

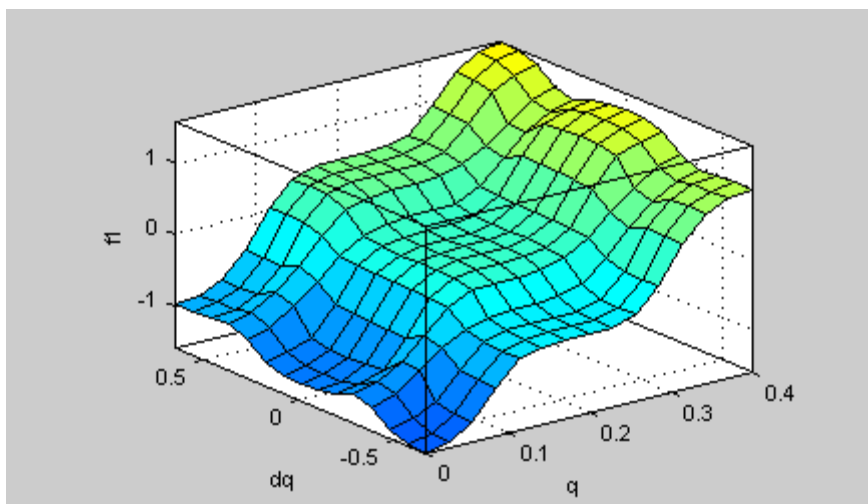
با استفاده از این قوانین مثلاً به ازای ورودی $(q,dq)=(0.01,0.5)$ خروجی سیستم فازی -0.951-

خواهد شد ، اما این در حالی است که پارامترهای سیستم فازی تغییر نکند .

بلوک دیاگرام مربوط به این حالت در زیر آمده است .

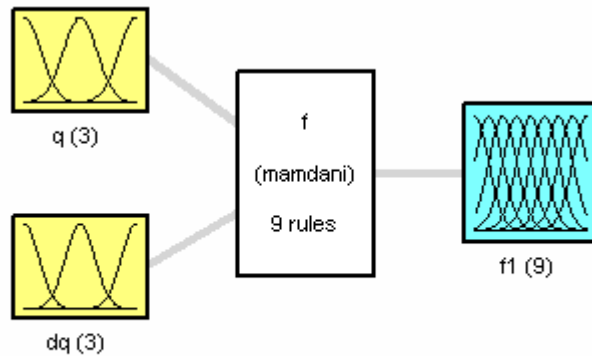


شکل ۳-۹ بلوک دیاگرام مربوط به قوانین سیستم فازی \hat{f}



شکل ۳-۱۰ سطح خروجی به ازای تمام حالات در نظر گرفته شده برای ورودی های (q,dq) در بازه های موردنظر

و همچنین بلوک دیاگرام سیستم فازی مورد نظر بصورت زیر می باشد .



System f: 2 inputs. 1 outputs. 9 rules
 شکل ۱۱-۳ بلوک دیاگرام سیستم فازی

پارامترهای سیستم فازی یعنی همان خروجی ها در هر قانون با الگوریتم گرادیان نزولی قابل تنظیم هستند طوری که خطای ردگیری مینیمم گردد ، یعنی در طراحی این سیستم ، قانون تعدیل ، گرادیان نزولی است و الگوریتم براساس مینیمم سازی تابع هزینه $J_k = \frac{1}{2} [e(k)]^2$ می باشد .

می دانیم رابطه تنظیم پارامترها با گرادیان نزولی بصورت زیر است :

$$y_{k+1} = y_k - \alpha_k g_k \quad (21-3)$$

که g_k همان گرادیان تابع هدف نسبت به پارامترهای سیستم فازی یعنی (y_k) همان خروجی در هر قانون است ، که توسط رابطه زیر محاسبه می گردد .

$$g_k = \frac{\partial J_k}{\partial q} \frac{\partial q}{\partial u} \frac{\partial u}{\partial \hat{f}} \frac{\partial \hat{f}}{\partial y_k} \quad (22-3)$$

می دانیم :

$$\frac{\partial J_k}{\partial q} = -[q_m(k) - q(k)] \quad (23-3)$$

برای محاسبه $\frac{\partial q}{\partial u}$ ، چون خود سیستم اصلی را نمی‌شناسیم، در نتیجه از رابطه زیر برای محاسبه استفاده می‌کنیم:

$$\frac{\partial q}{\partial u} = \frac{\frac{dq}{dt}}{\frac{du}{dt}} = \frac{\dot{q}}{\dot{u}} \quad (24-3)$$

که با توجه به قانون کنترل (7-1) داریم:

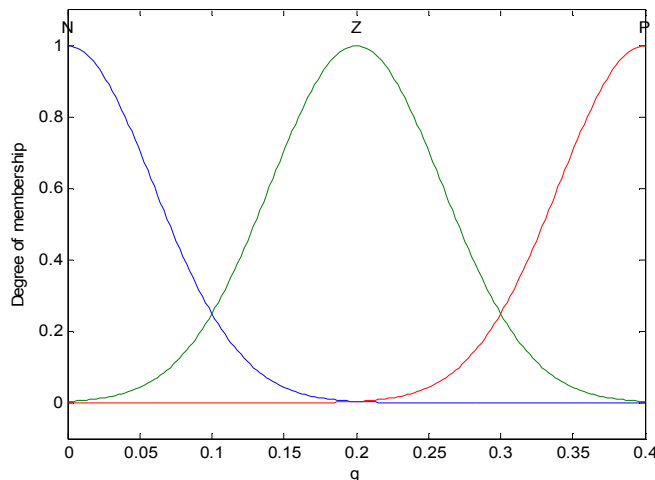
$$\frac{\partial u}{\partial \hat{f}} = \frac{-1}{\hat{g}} \quad (25-3)$$

برای محاسبه $\frac{\partial \hat{f}}{\partial y_k}$ ، ابتدا سیستم فازی \hat{f} را با استفاده از تشکیل قوانین فازی ساخته و سپس آن را محاسبه می‌کنیم. [2, 3]

برنامه مربوط به سیستم فازی \hat{f} با تنظیم پارامترها در نرم افزار مطلب، در پیوست ۱ آمده است.

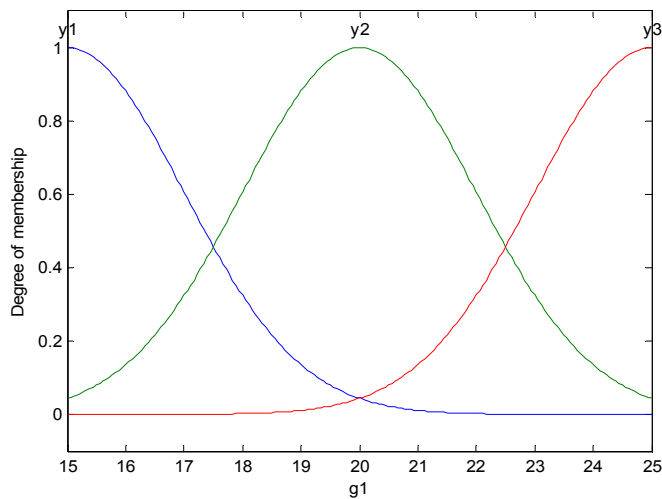
۳-۳-۲. طراحی سیستم فازی \hat{g} با روش گرادیان نزولی

برای طراحی سیستم فازی لازم است ابتدا ورودی‌های این سیستم تعیین گردد. ورودی سیستم \hat{g} ، همان زاویه مفصل ربات می‌باشد. ۳ گروه فازی P و Z و N بصورت توابع عضویت گوسین برای این ورودی در نظر گرفته می‌شود. توابع عضویت مربوط به q در شکل زیر قابل مشاهده می‌باشند که پهنای هر یک از توابع تعلق مربوط به q را 0.06 در نظر گرفته ایم.



شکل ۳-۱۲ گروه‌های فازی مربوط به ورودی q

لازم به ذکر است که در حین استفاده از این سیستم فازی در کنترلر مراکز و پهنای توابع تعلق را ثابت نگه داشته و آنها را تغییر نمی دهیم و فقط مراکز خروجی ها در هر قانون تغییر می کنند ، پس مراکز خروجی ها در هر قانون بعنوان پارامترهای سیستم فازی هستند که قابل تنظیم می باشند .مقادیر اولیه برای مراکز خروجی ها در هر قانون در بازه ای [15, 25] انتخاب می گردد زیرا مقدار g برای موتور به تنهایی بدون اعمال گشتاور بار ناشی از مفصل 20 می باشد . در نتیجه ۳ قانون داریم و ۳ گروه فازی برای خروجی خواهیم داشت .

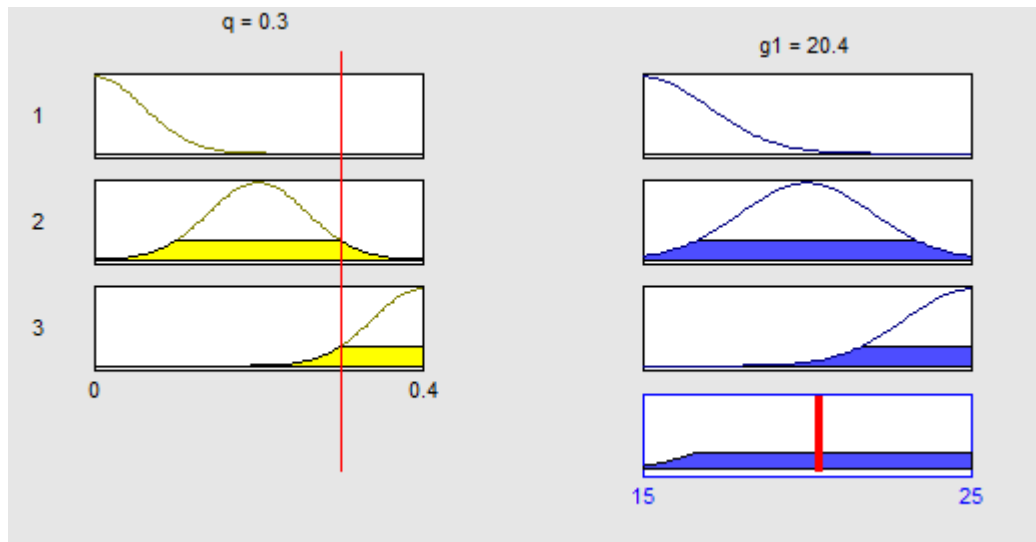


شکل ۳-۱۳ گروههای فازی مربوط به خروجی سیستم فازی \hat{g}

۳ قانون فازی نیز بصورت زیر است که در هر قانون خروجی ها بصورت دلخواه انتخاب شده اند و ما هیچ اطلاعاتی راجع به آنها نداریم :

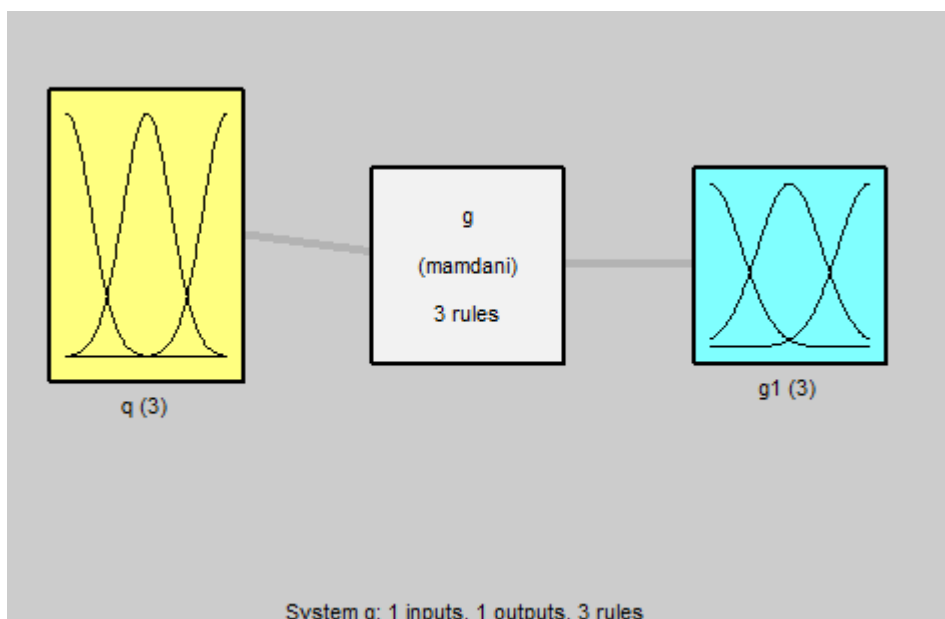
1. If (q is N) then (g1 is y1)
2. If (q is Z) then (g1 is y2)
3. If (q is P) then (g1 is y3)

با استفاده از این قوانین مثلا به ازای ورودی $q=0.3$ خروجی سیستم فازی 20.4 خواهد شد ، اما این در حالی است که پارامترهای سیستم فازی تغییر نکند. بلوک دیاگرام مربوط به این حالت در زیر آمده است .



شکل ۳-۱۴ بلوک دیاگرام مربوط به قوانین سیستم فازی \hat{g}

و همچنین بلوک دیاگرام سیستم فازی مورد نظر بصورت زیر می باشد .



شکل ۳-۱۵ بلوک دیاگرام سیستم فازی \hat{g}

پارامترهای سیستم فازی یعنی همان خروجی ها در هر قانون با الگوریتم گرادیان نزولی قابل تنظیم هستند طوری که خطای ردگیری مینیمم گردد ، یعنی در طراحی این سیستم ، قانون تعدیل ، گرادیان نزولی است و الگوریتم براساس مینیمم سازی تابع هزینه $J_k = \frac{1}{2} [e(k)]^2$ می باشد .
می دانیم رابطه تنظیم پارامترها با گرادیان نزولی بصورت زیر است :

$$y_{k+1} = y_k - \alpha_k g_k \quad (21-3)$$

که g_k همان گرادیان تابع هدف نسبت به پارامترهای سیستم فازی یعنی (y_k) همان خروجی در هر قانون است ، که توسط رابطه زیر محاسبه می گردد .

$$g_k = \frac{\partial J_k}{\partial q} \frac{\partial q}{\partial u} \frac{\partial u}{\partial \hat{g}} \frac{\partial \hat{g}}{\partial y_k} \quad (22-3)$$

می دانیم :

$$\frac{\partial J_k}{\partial q} = -[q_m(k) - q(k)] \quad (23-3)$$

برای محاسبه $\frac{\partial q}{\partial u}$ ، چون خود سیستم اصلی را نمی شناسیم ، در نتیجه از رابطه زیر برای محاسبه استفاده می کنیم :

$$\frac{\partial q}{\partial u} = \frac{\frac{dq}{dt}}{\frac{du}{dt}} = \frac{\dot{q}}{\dot{u}} \quad (24-3)$$

که با توجه به قانون کنترل (7-1) داریم :

$$\frac{\partial u}{\partial \hat{g}} = \frac{-u}{\hat{g}} \quad (25-3)$$

برای محاسبه $\frac{\partial \hat{g}}{\partial y_k}$ ، ابتدا سیستم فازی \hat{g} را با استفاده از تشکیل قوانین فازی ساخته و سپس آن را محاسبه می کنیم . [2,3]

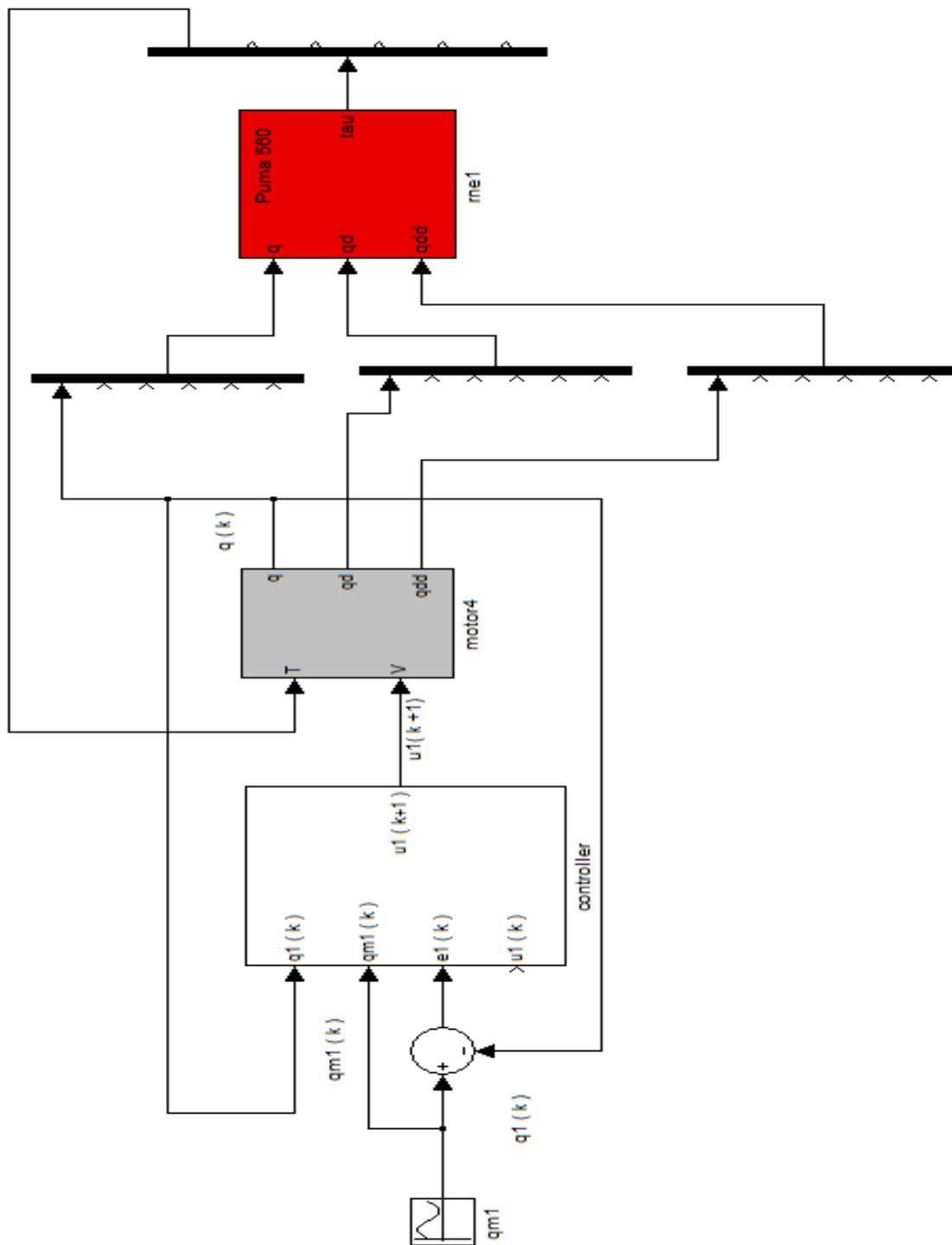
برنامه مربوط به سیستم فازی \hat{g} با تنظیم پارامترها در نرم افزار مطلب ، در پیوست ۲ آمده است .

۳-۳-۳. مرحله ساخت قانون کنترل

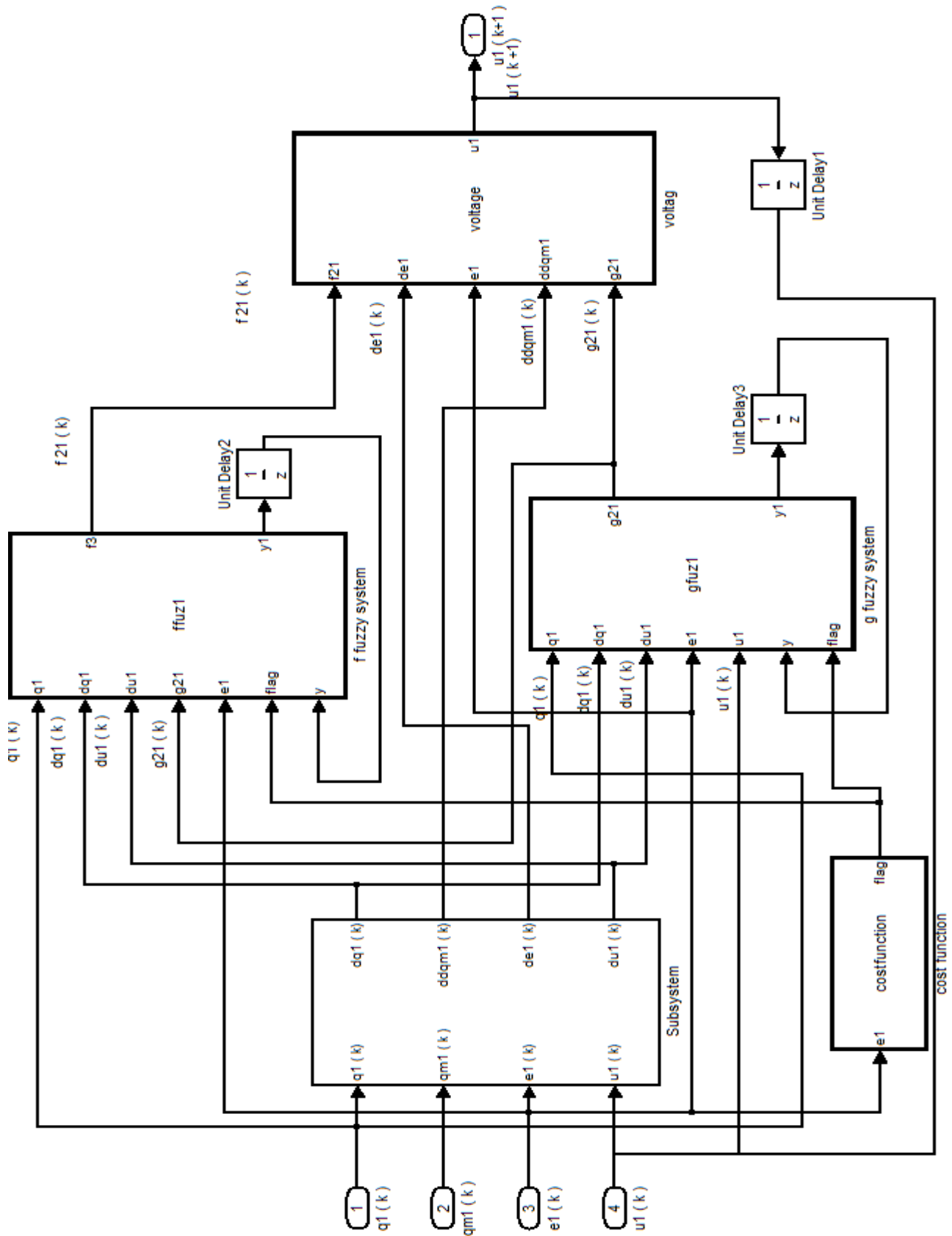
در مرحله قبل همان طور که مشاهده شد دو سیستم فازی \hat{g} , \hat{f} را طراحی کردیم. حال به کمک این دو سیستم می خواهیم قانون کنترل در رابطه (7-1) را بسازیم. در پیوست ۳، برنامه ساخت این قانون کنترل آمده است.

با طراحی کنترلر برای سیستم موردنظر، یعنی (یک موتور dc با گشتاور بار ناشی از یک مفصل از ربات پیوما ۵۶۰)، می توان از ۶ سیستم طراحی شده بالا برای کنترل زوایای هر ۶ مفصل از ربات استفاده نمود.

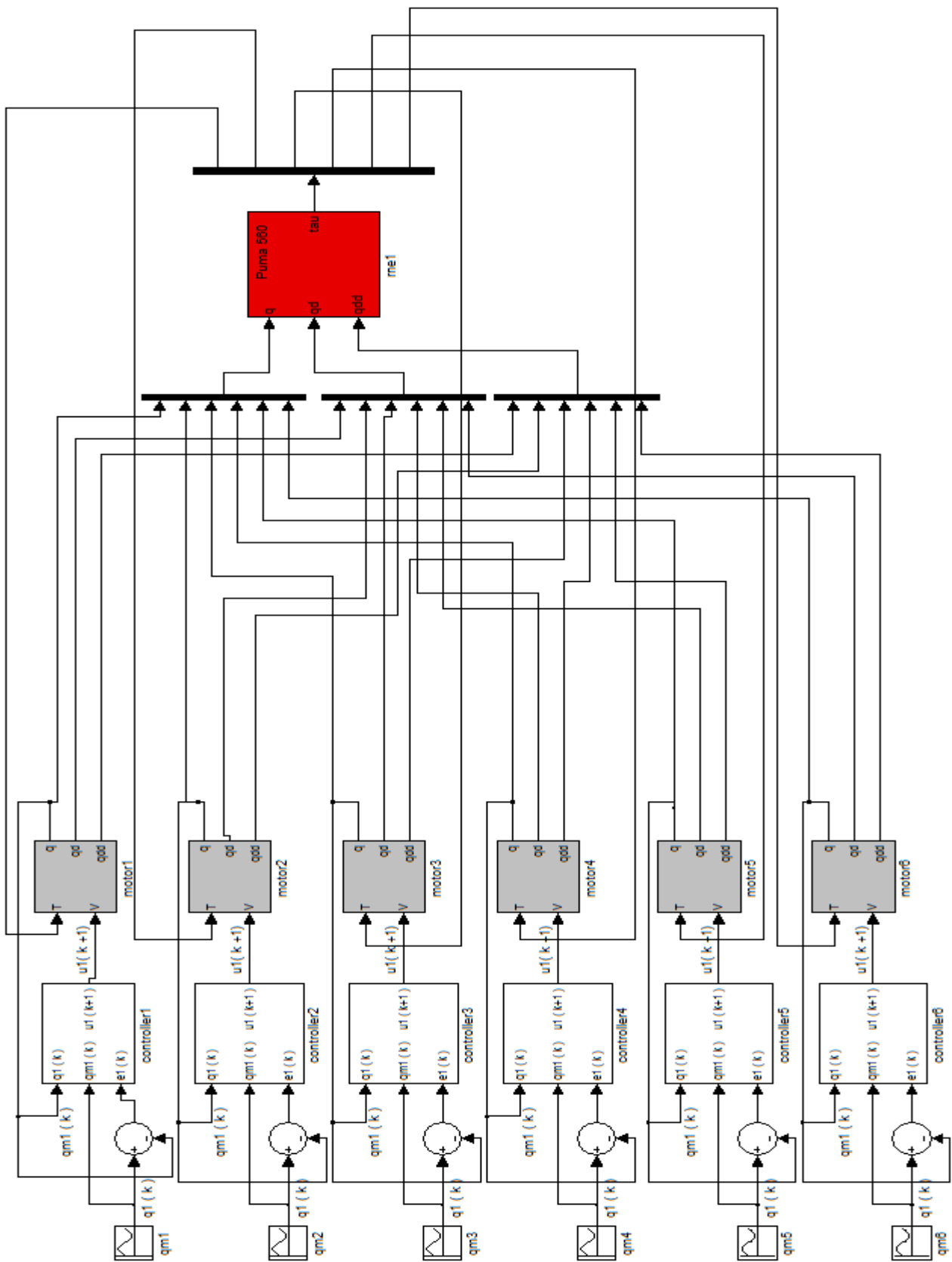
بلوک دیاگرام مربوط به کنترل یک مفصل از ربات پیوما ۵۶۰ در شکل ۳-۱۶ آمده است. اجزای سیستم کنترل یک مفصل از ربات پیوما ۵۶۰، شامل کنترلر، موتور dc به عنوان محرک و ربات پیوما ۵۶۰ می باشد.



شکل ۳-۱۶ بلوک دیاگرام مربوط به کنترل یک مفصل از ربات پیوما ۵۶۰



شکل ۳-۱۷ بلوک دیاگرام داخل بلوک مربوط به کنترلر



شکل ۳-۱۸ بلوک دیاگرام کل سیستم کنترل ۶ مفصل ربات پیوما ۵۶۰

مطالب بخش چهارم : نتایج شبیه سازی

۴-۱ . نتایج شبیه سازی سیستم کنترل فازی تطبیقی غیر مستقیم موتور dc با روش گرادیان نزولی

۴-۱-۱ . پاسخ سیستم کنترل به ازای ورودی پله

۴-۱-۲ . پاسخ سیستم کنترل به ازای ورودی \sin

۴-۲ . نتایج شبیه سازی سیستم کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان نزولی

۴-۲-۱ . پاسخ سیستم کنترل به ازای ورودی \sin

۴-۲-۲ . پاسخ سیستم کنترل به ازای ورودی پله

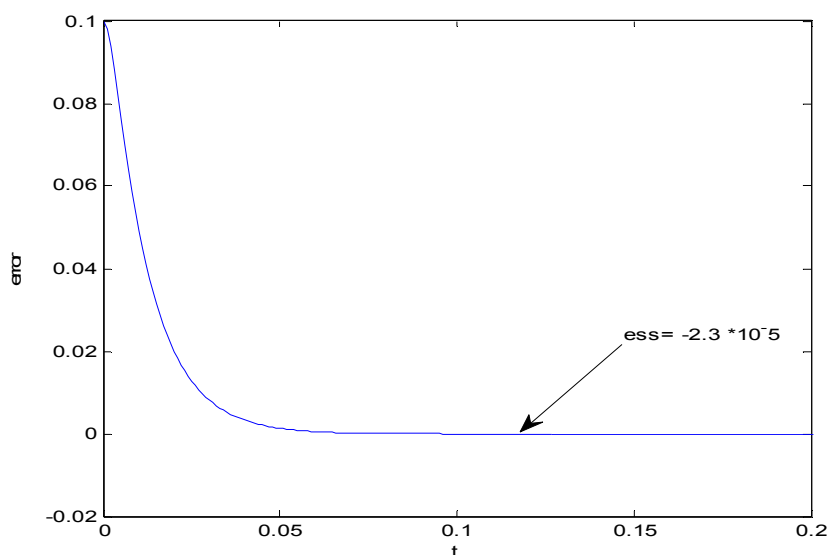
۴-۱. نتایج شبیه سازی کنترل فازی تطبیقی غیر مستقیم موتور dc با روش گرادیان نزولی

همان طور که می دانیم، ربات " پیوما ۵۶۰ " دارای ۶ مفصل لولایی است ، که برای به حرکت در آوردن هر یک از مفاصل ربات ، نیازمند یک محرک هستیم ، که محرک مورد استفاده در این پروژه ، موتور dc می باشد . ما در قسمت اول پروژه ، ابتدا به کنترل فازی تطبیقی غیرمستقیم یک موتور dc با گشتاور متوسط $T=30\sin(q)$ با روش گرادیان نزولی می پردازیم و سپس همین کنترل را روی هر یک از ۶ موتور dc ربات اعمال می نماییم .

نتایج شبیه سازی کنترل فازی تطبیقی غیر مستقیم موتور dc را در دو حالت زیر بررسی می نماییم . (لازم به ذکر است که در هر دو حالت تعداد قوانین سیستم فازی f برابر ۴ و تعداد قوانین سیستم فازی g را برابر با ۲ در نظر گرفته ایم)

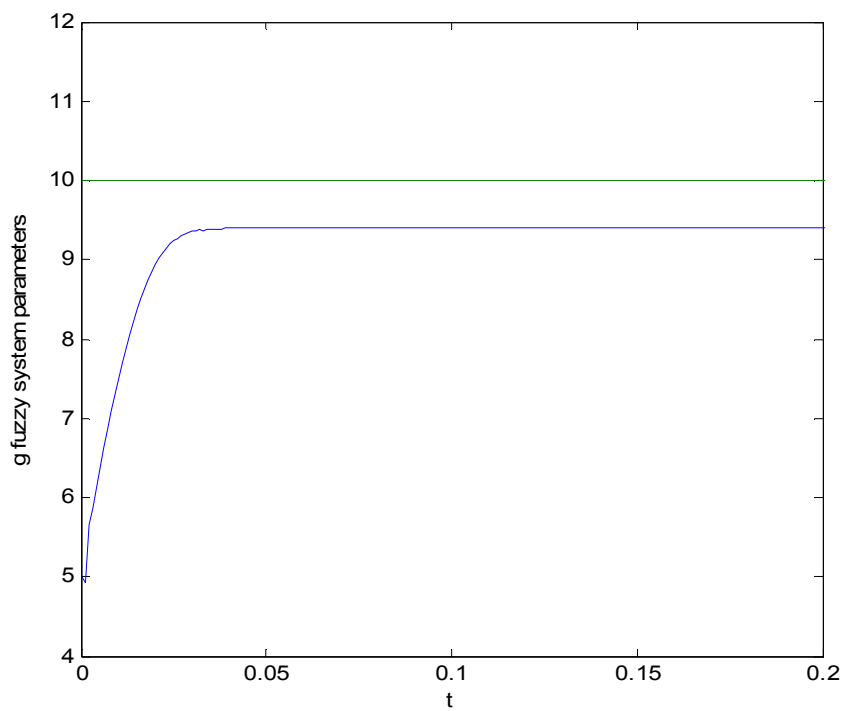
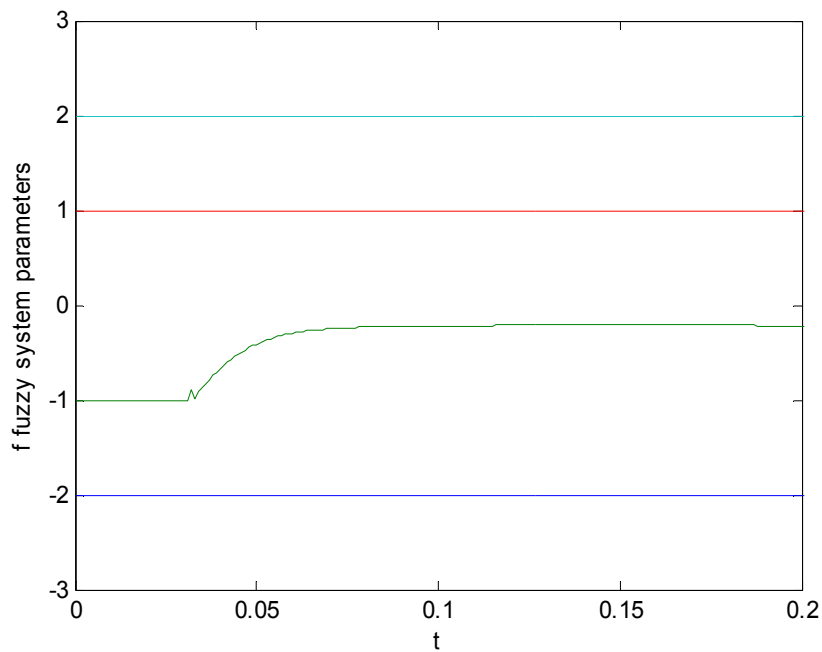
۴-۱-۱. پاسخ سیستم کنترل به ازای ورودی پله

قطب های سیستم حلقه بسته را در موقعیت های $s_1 = -50 + 85i$ ، $s_2 = -50 - 85i$ قرار می دهیم .

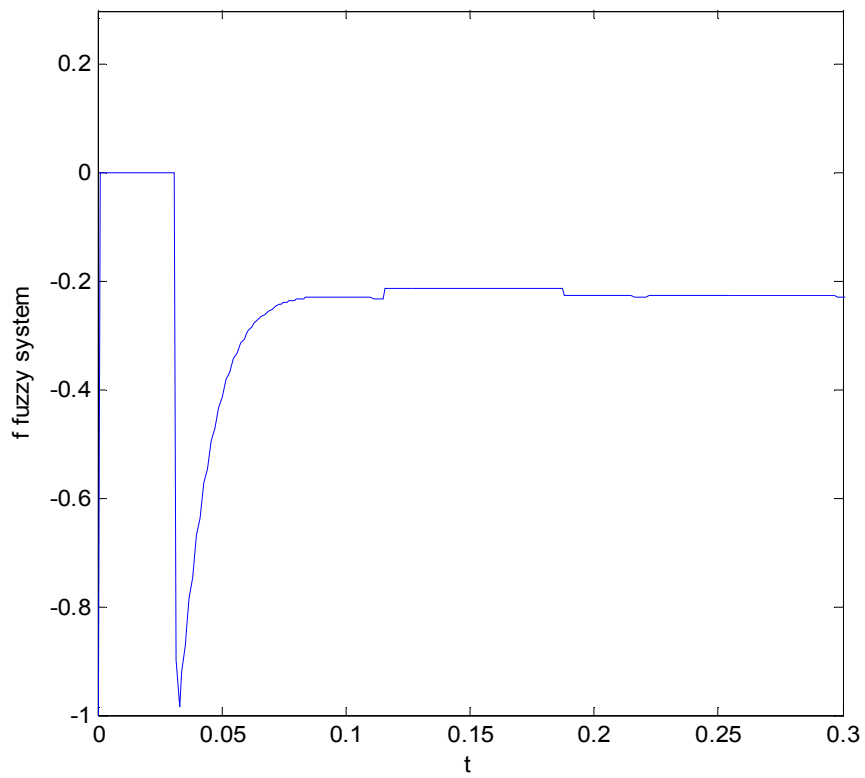
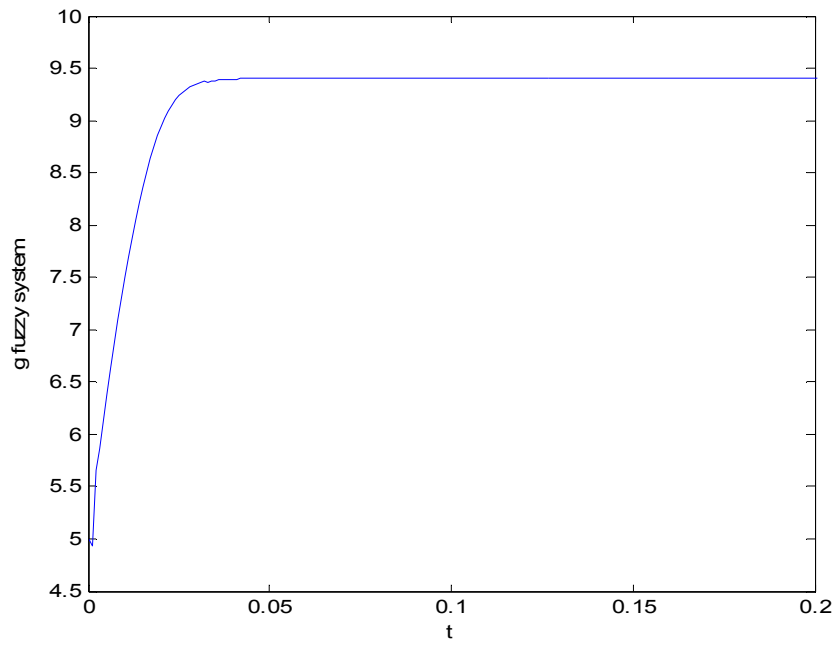


شکل ۴-۱ خطای ردگیری به ازای ورودی پله $q_d = 0.3u(t)$

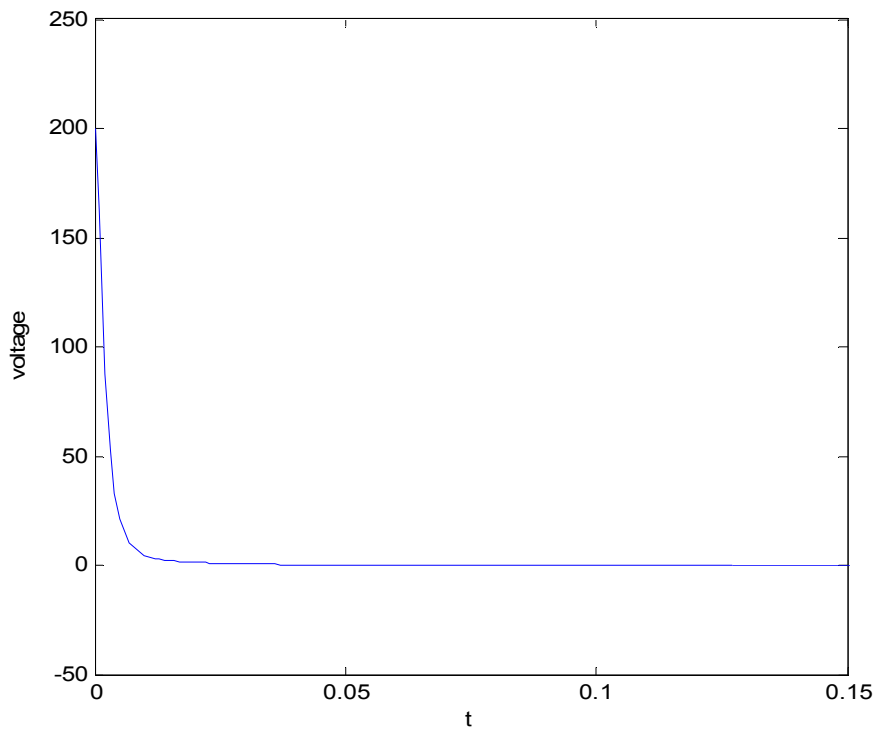
پارامترهای سیستم فازی f و g با روش گرادیان نزولی به صورت زیر تنظیم می شوند ، طوری که خطای ردگیری را در زمان 0.1 ثانیه به -2.3×10^{-5} خواهد رسید .



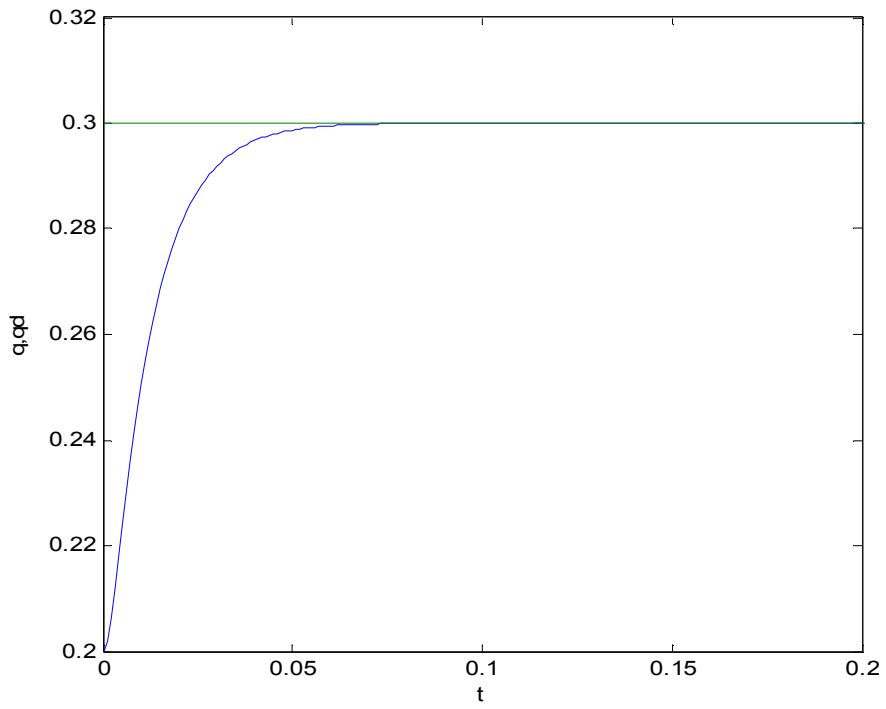
شکل ۲-۴ پارامترهای سیستم فازی f و g با روش گرادیان نزولی



شکل ۳-۴ خروجی سیستم فازی g و f

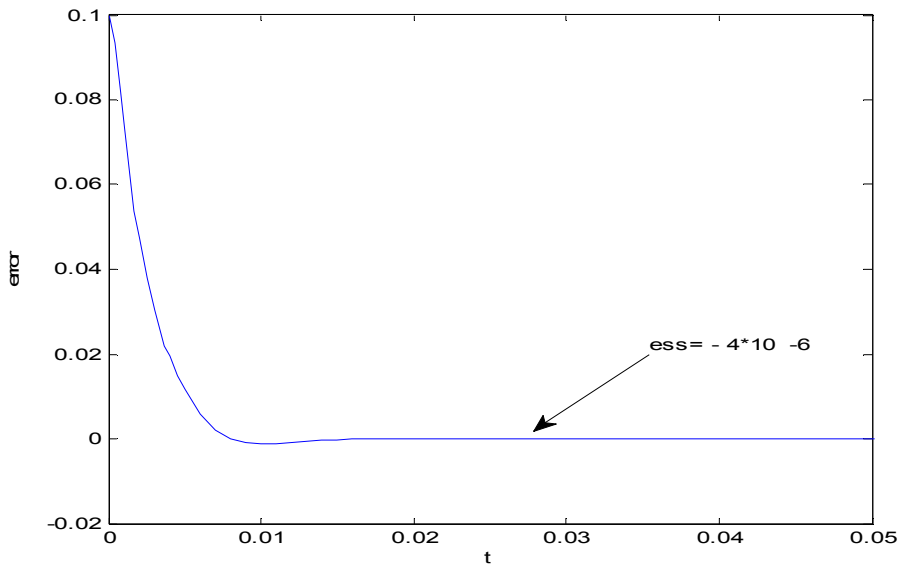


شکل ۴-۴ سیگنال کنترل ولتاژ

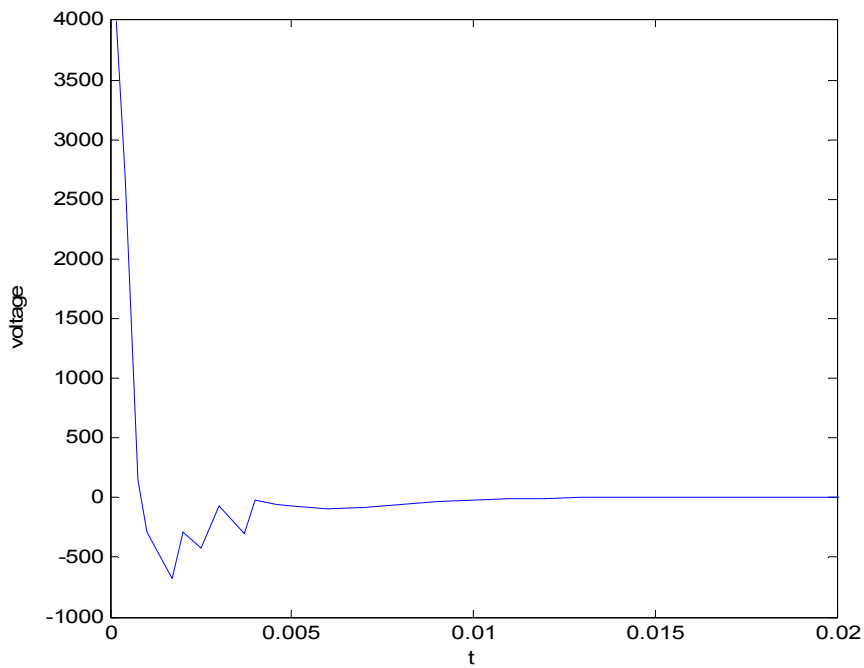


شکل ۴-۵ پاسخ سیستم کنترل q و ورودی مطلوب q_d

قطب های سیستم حلقه بسته را در موقعیت های $s_1 = -300 + 400i$, $s_2 = -300 - 400i$ قرار می دهیم .

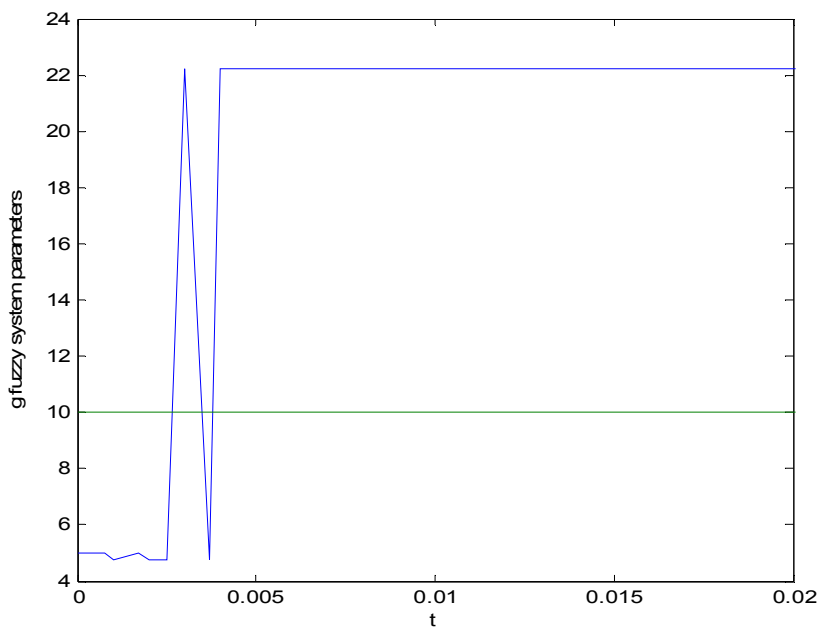
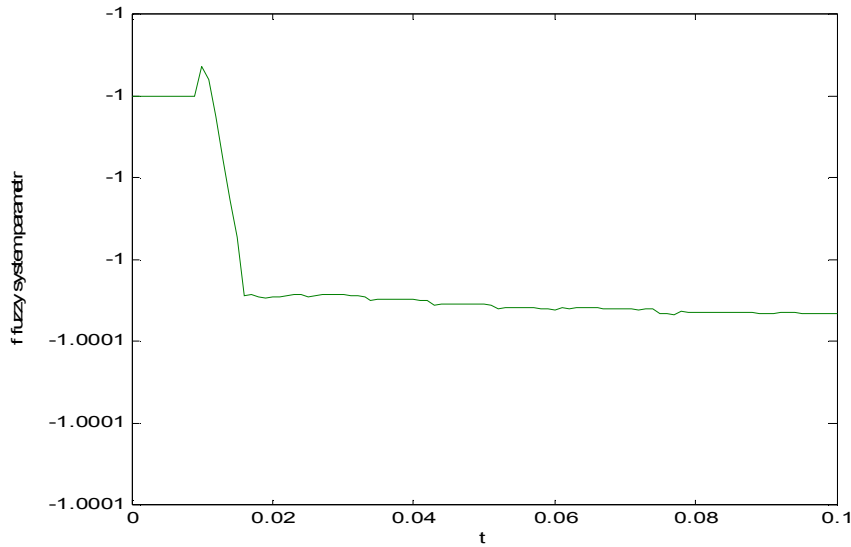


شکل ۴-۶ خطای ردگیری به ازای ورودی پله $q_d = 0.3u(t)$

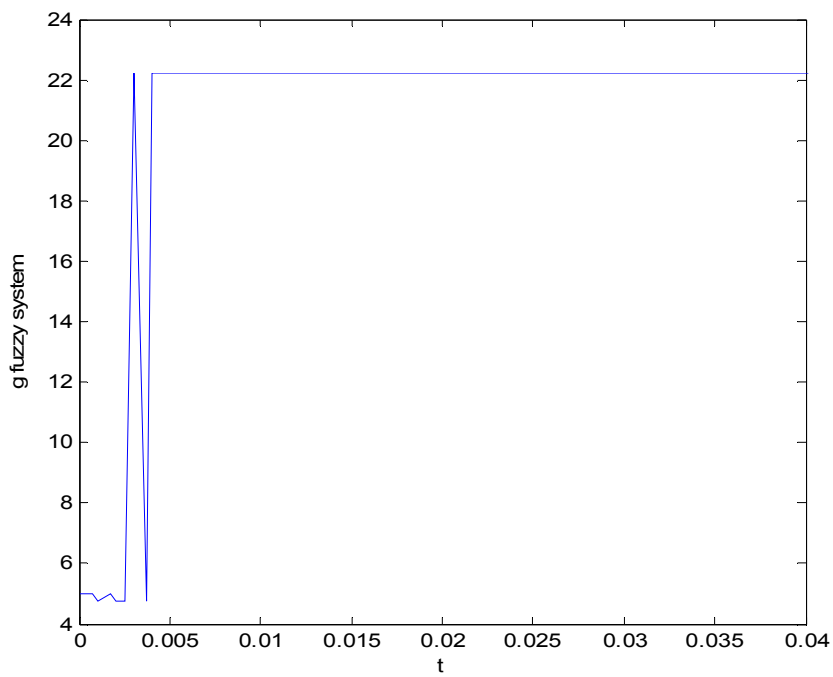
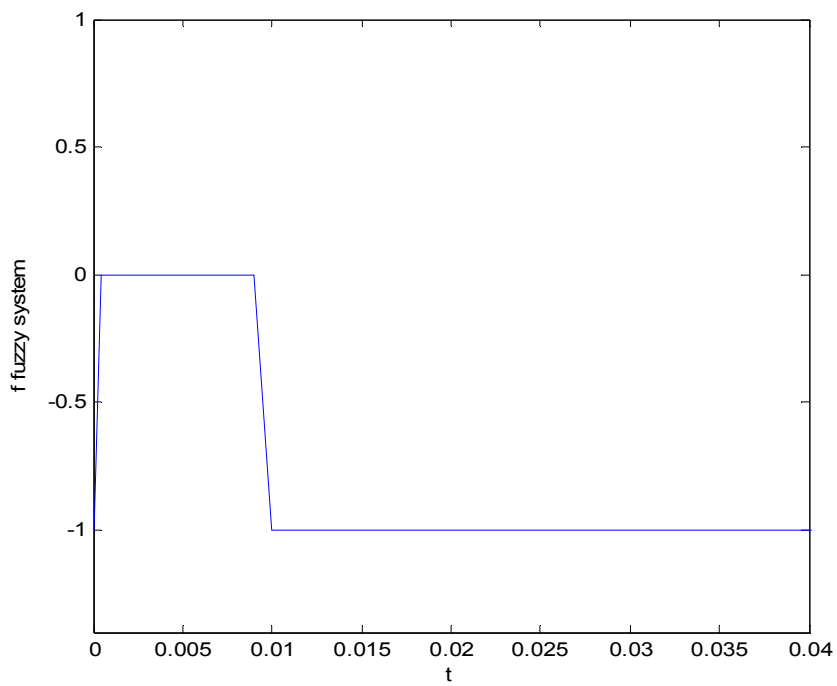


شکل ۴-۷ سیگنال کنترل ولتاژ

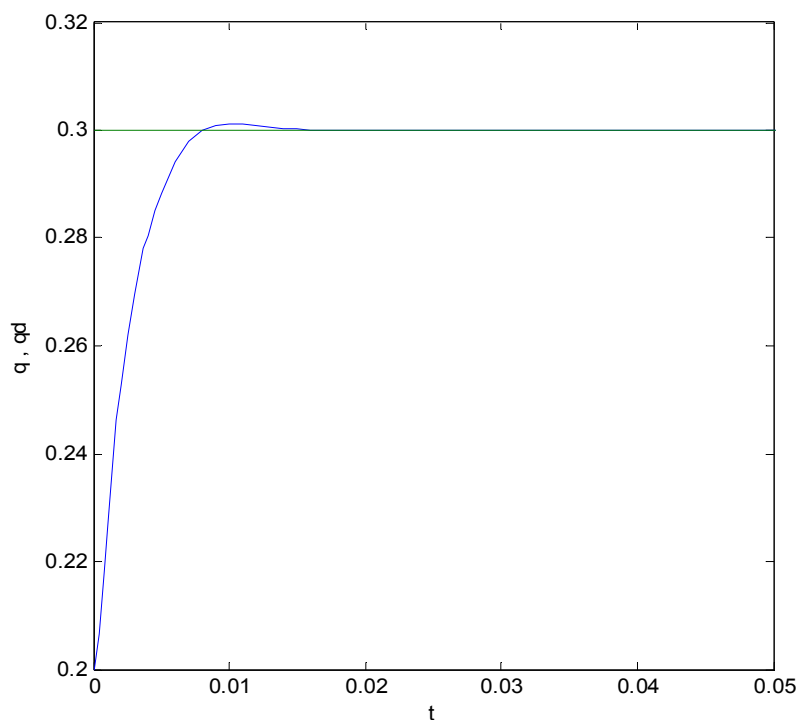
در این حالت به علت فاصله زیاد قطب ها از مبدا ، تطبیق سریعتر صورت گرفته ، در نتیجه پارامترهای سیستم فازی f که با روش گرادیان نزولی به صورت زیر تنظیم می شوند ، تغییرات چندانی ندارند ، طوری که خطای ردگیری را در زمان 0.1 ثانیه به $4 * 10^{-6}$ خواهد رسید . در سیستم فازی f ، فقط پارامتر 1- به صورت زیر تغییر می کند .



شکل ۴-۸ پارامترهای سیستم فازی f و g



شکل ۴-۹ خروجی سیستم های فازی f و g

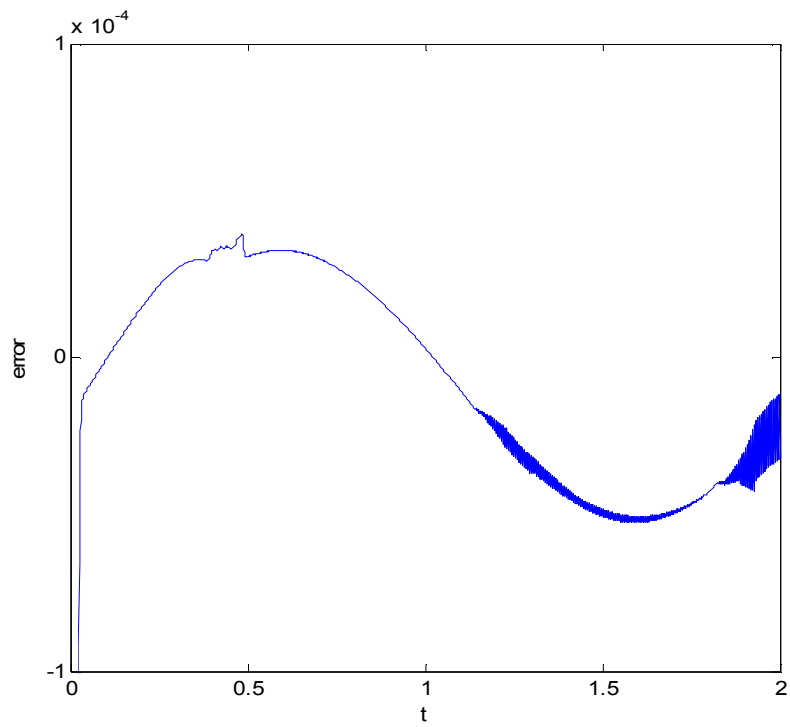


شکل ۴-۱۰ پاسخ سیستم کنترل q و ورودی مطلوب qd

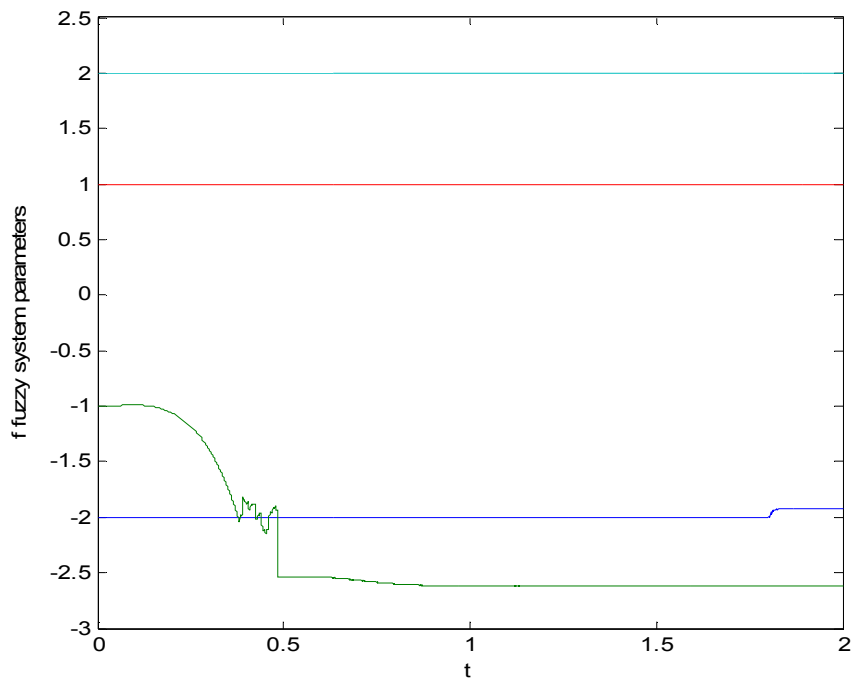
از مقایسه نتایج در دو حالت ، خواهیم فهمید هر چه قطب های سیستم حلقه بسته را دورتر از مبدا قرار دهیم ، خطای ردگیری ناشی از محل قطب ها سریعتر کاهش یافته ، در نتیجه سریعتر تطبیق انجام گرفته ، و در نهایت خطای حالت دائم ، کوچکتر از حالتی است که قطب ها نزدیکند .

۴-۱-۲ . پاسخ سیستم کنترل به ازای ورودی \sin

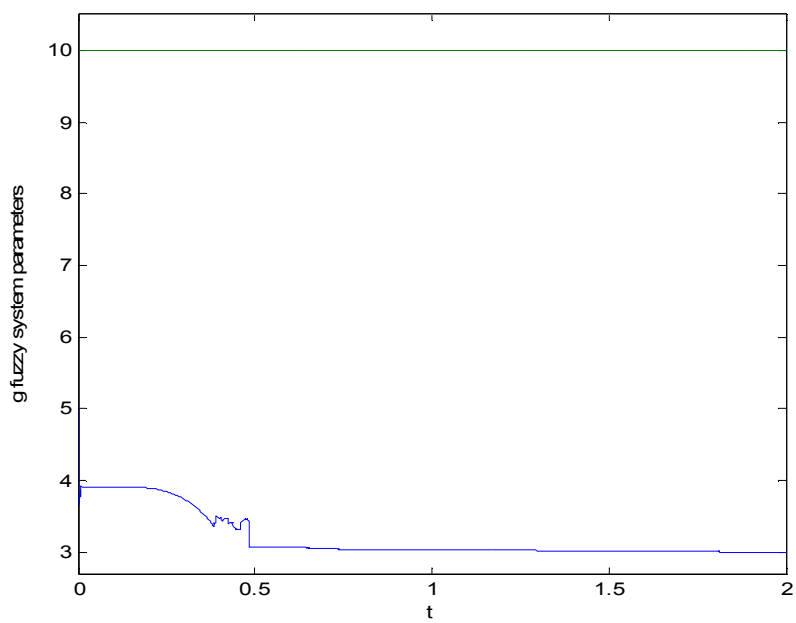
نتایج شبیه سازی کنترل فازی تطبیقی غیر مستقیم موتور dc را در حالتی که قطب های سیستم حلقه بسته را در موقعیت های $s_1 = -300 + 400i$, $s_2 = -300 - 400i$ قرار می دهیم در زیر بررسی می نماییم . (لازم به ذکر است که در هر دو حالت تعداد قوانین سیستم فازی f برابر ۴ و تعداد قوانین سیستم فازی g را برابر با ۲ در نظر گرفته ایم)



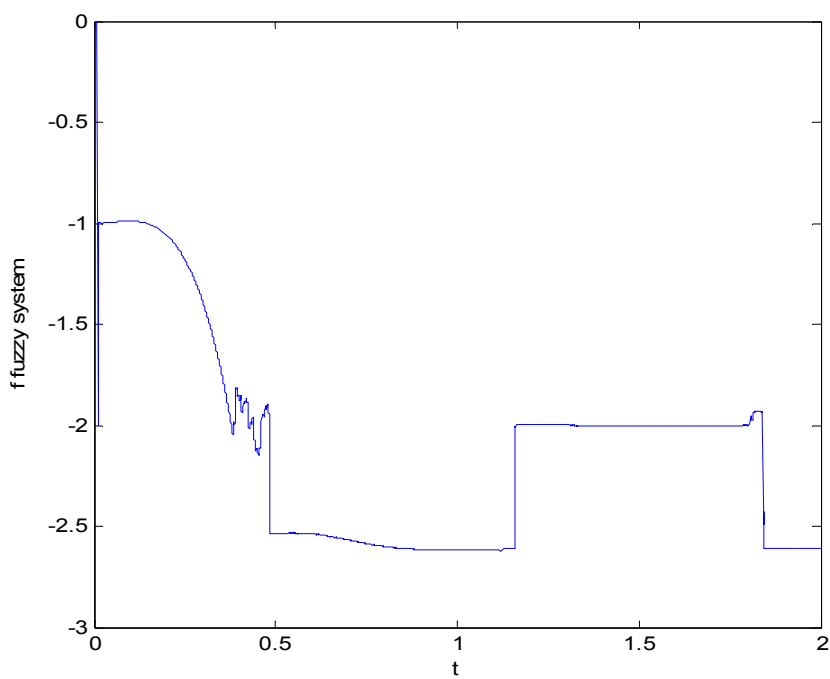
شکل ۴-۱۱. خطای ردگیری به ازای ورودی $qd = -0.2\cos(\pi*t)+0.2$



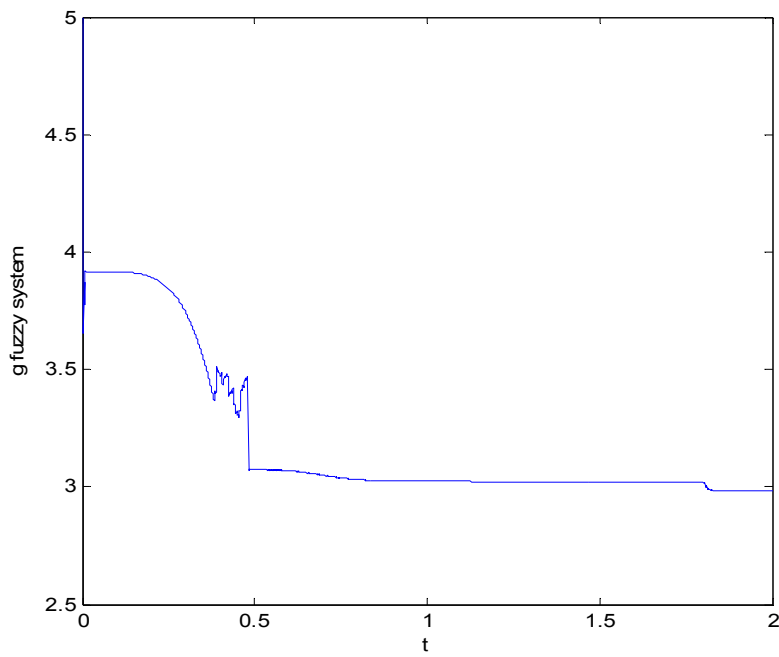
شکل ۴-۱۲. پارامترهای سیستم فازی f



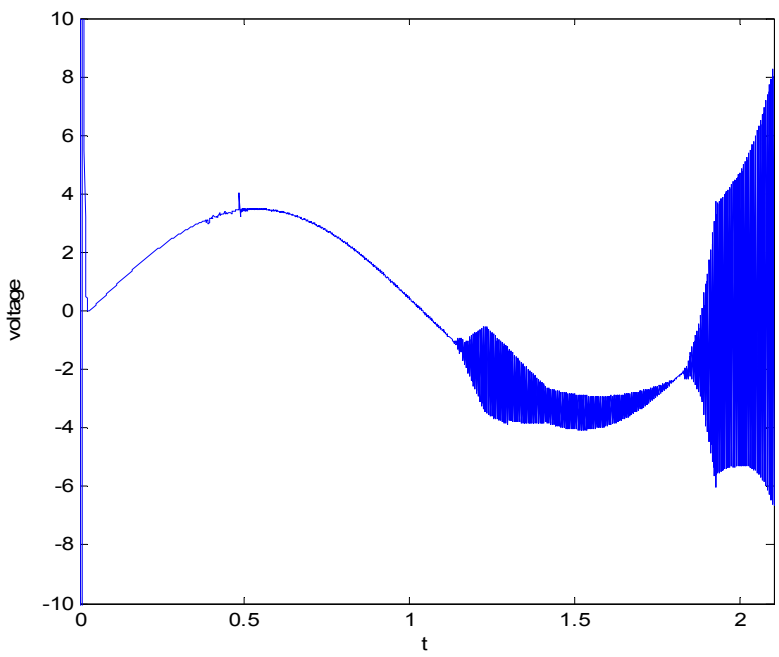
شکل ۴-۱۳. پارامترهای سیستم فازی g



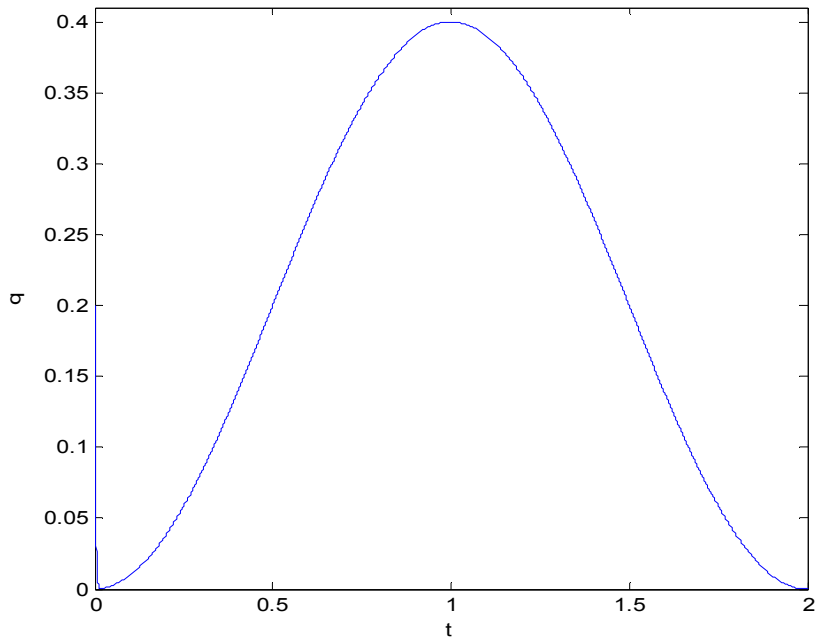
شکل ۴-۱۴. خروجی سیستم های فازی f



شکل ۴-۱۵ . خروجی سیستم های فازی g



شکل ۴-۱۶ . سیگنال ولتاژ



شکل ۴-۱۷. پاسخ سیستم کنترل

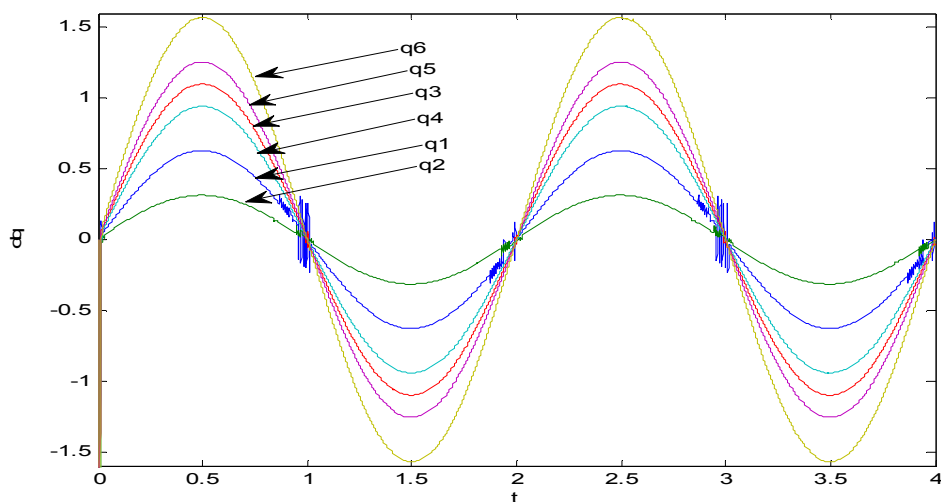
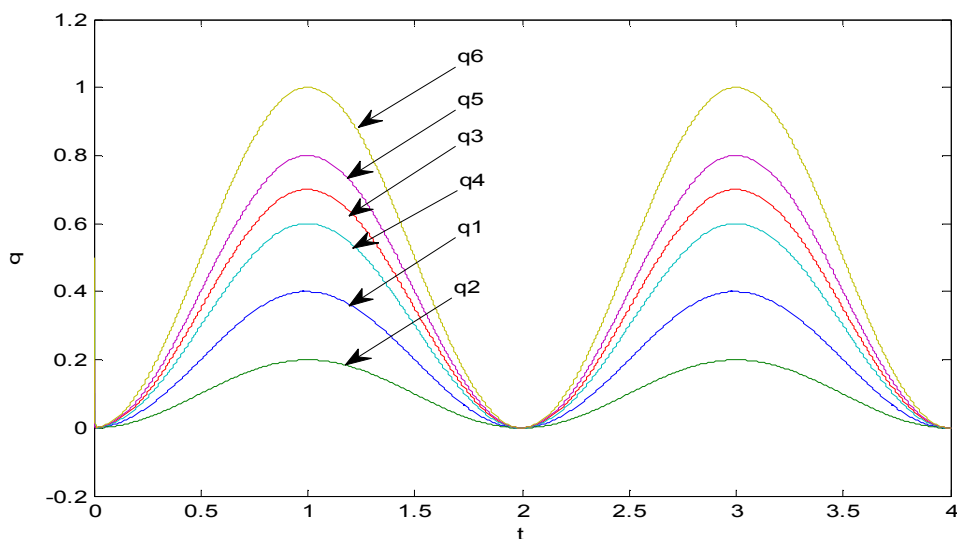
۴-۲. نتایج شبیه سازی سیستم کنترل فازی تطبیقی غیر مستقیم ربات " پیوما ۵۶۰ " با روش گرادیان نزولی

۴-۲-۱. مفاصل ربات مسیر های \sin ای را طی می کنند .

پاسخ سیستم کنترل فازی تطبیقی غیر مستقیم به ازای مسیرهای مطلوب \sin ای برای مفاصل ربات

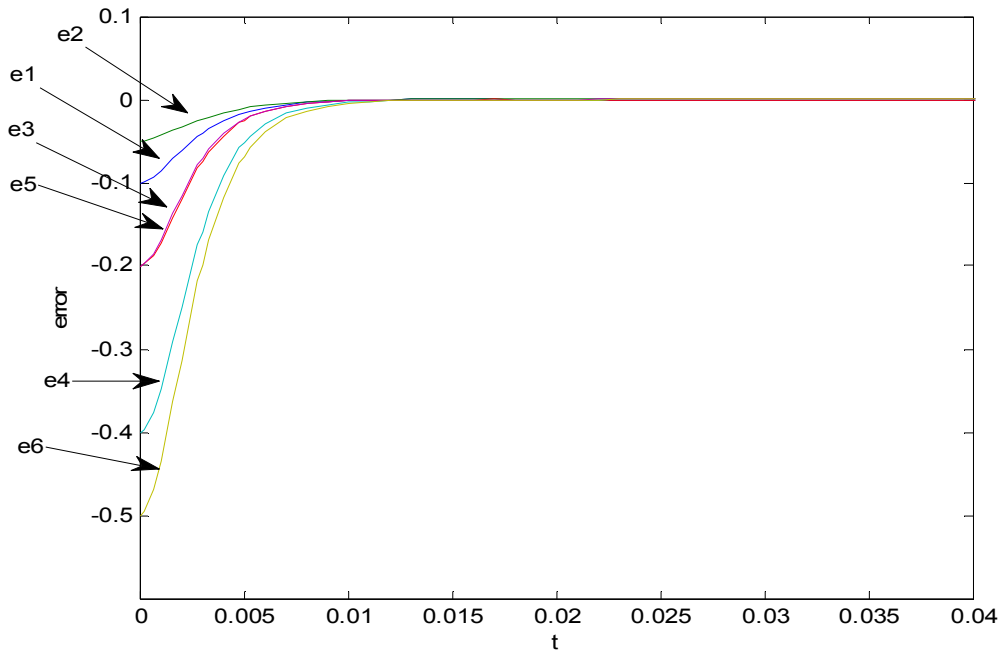
در صورتیکه قطب های سیستم حلقه بسته برای هر ۶ سیستم کنترل مفصل مستقل در

$s_1 = -300 + 400i$, $s_2 = -300 - 400i$ واقع شده باشد ، بصورت زیر می باشد :

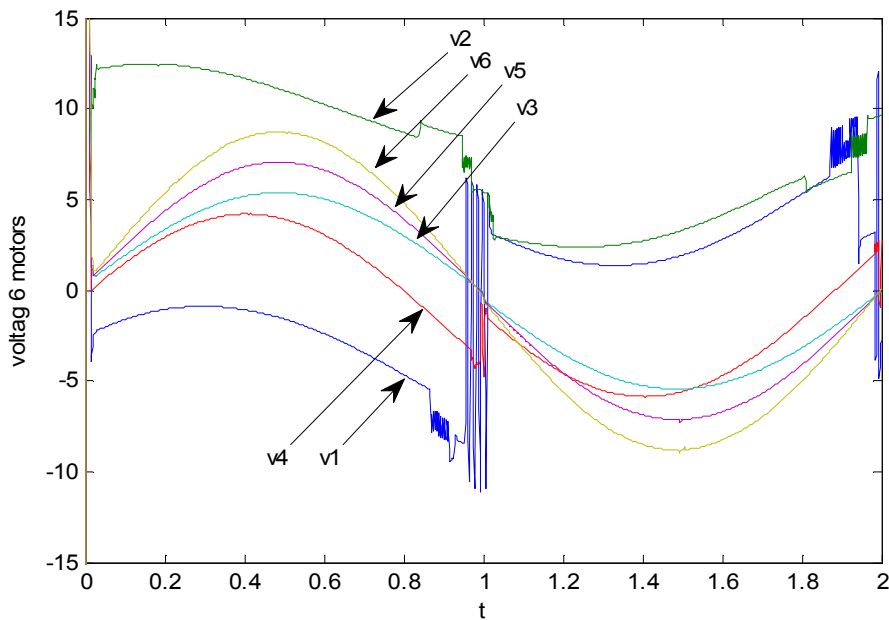


شکل ۴-۱۸. پاسخ سیستم کنترل فازی تطبیقی غیر مستقیم به ازای مسیرهای مطلوب \sin ای

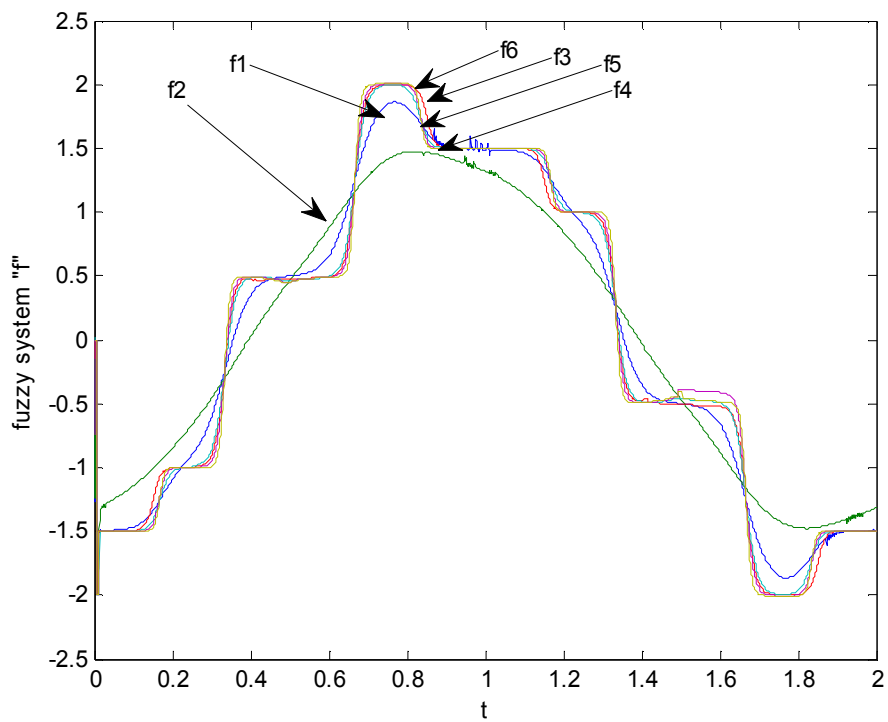
برای مفاصل ربات



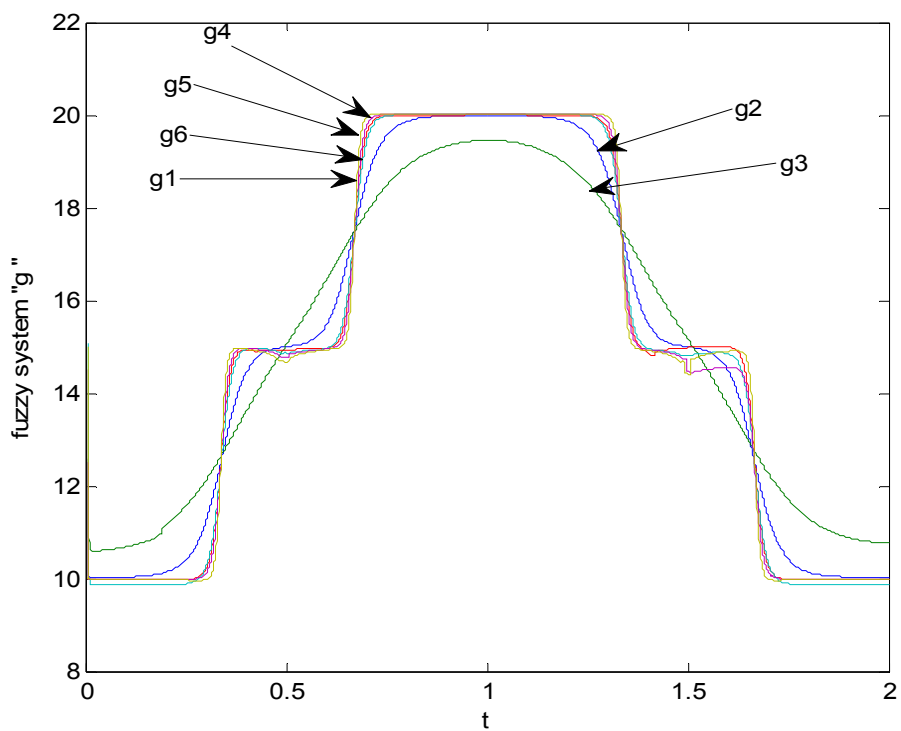
شکل ۴-۱۹. خطای ردگیری مربوط به کنترل هر ۶ مفصل ربات (با ۹ قانون برای سیستم فازی \hat{f})



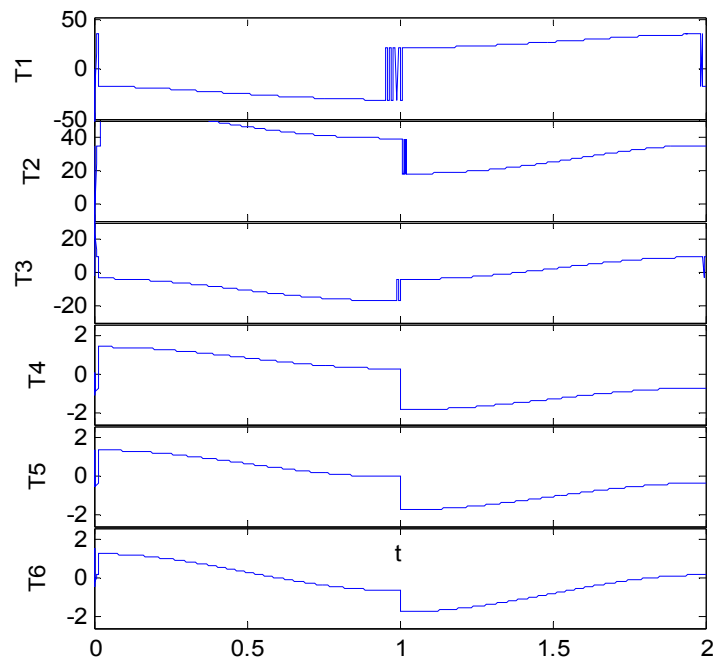
شکل ۴-۲۰. سیگنال های ولتاژ که به موتورها اعمال می شوند تا مفاصل را به اندازه ی موردنظر جابجا کنند



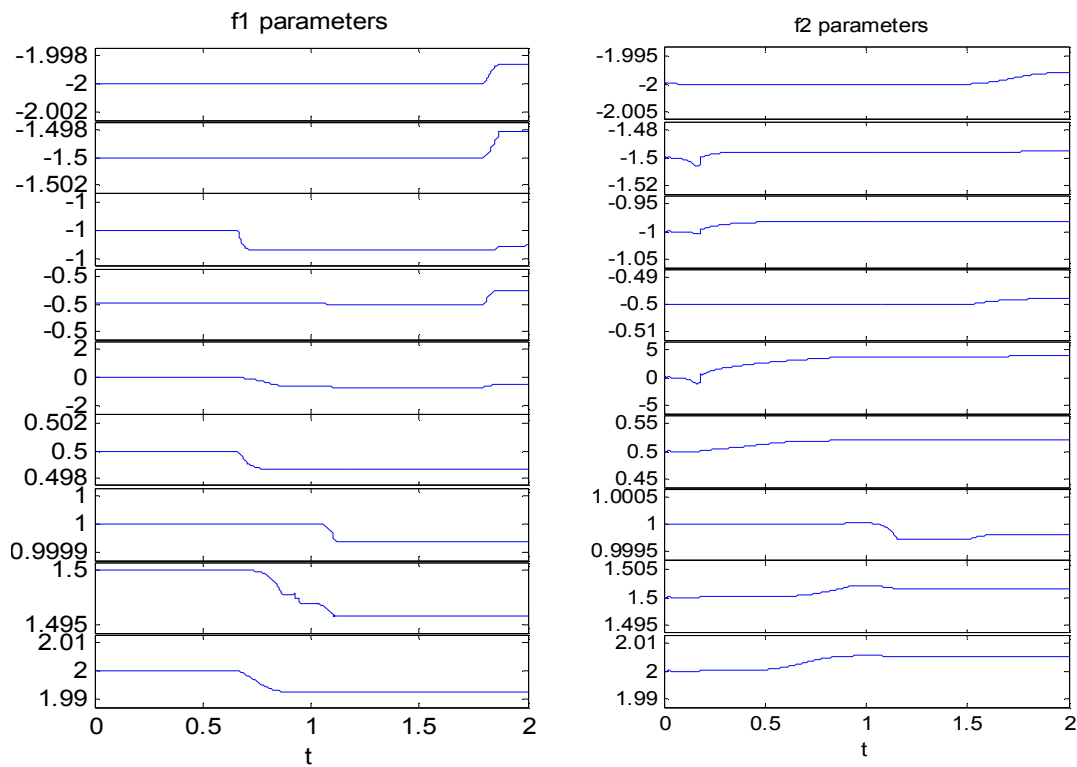
شکل ۴-۲۱. خروجی های f مربوط به ۶ کنترلر



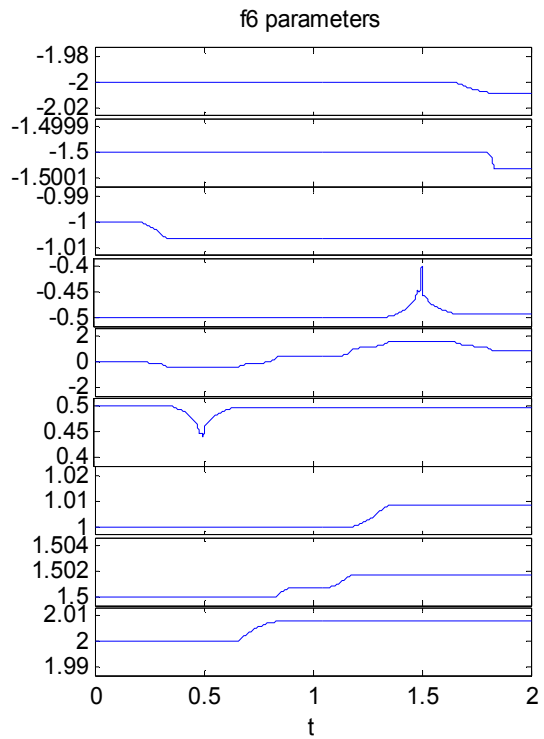
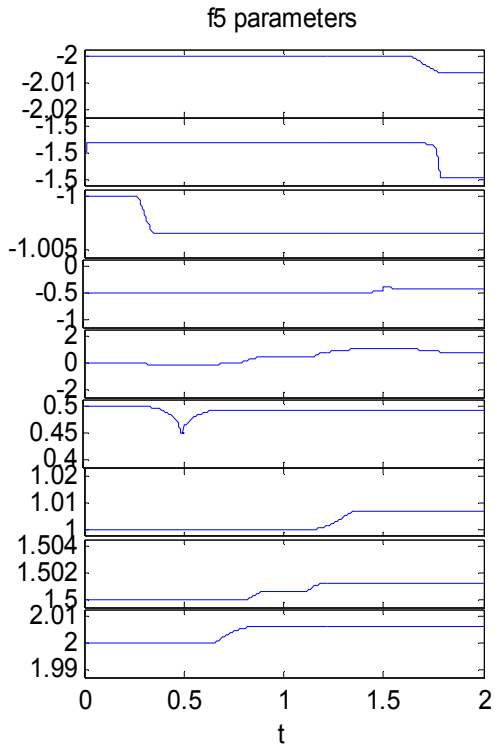
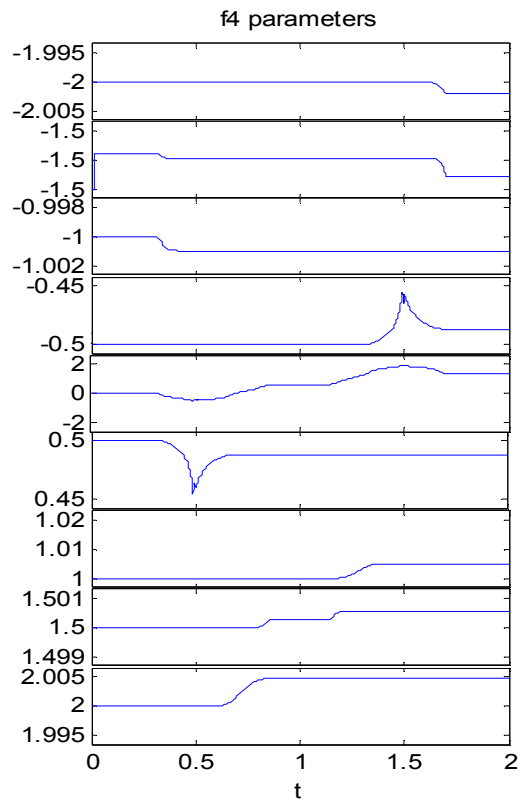
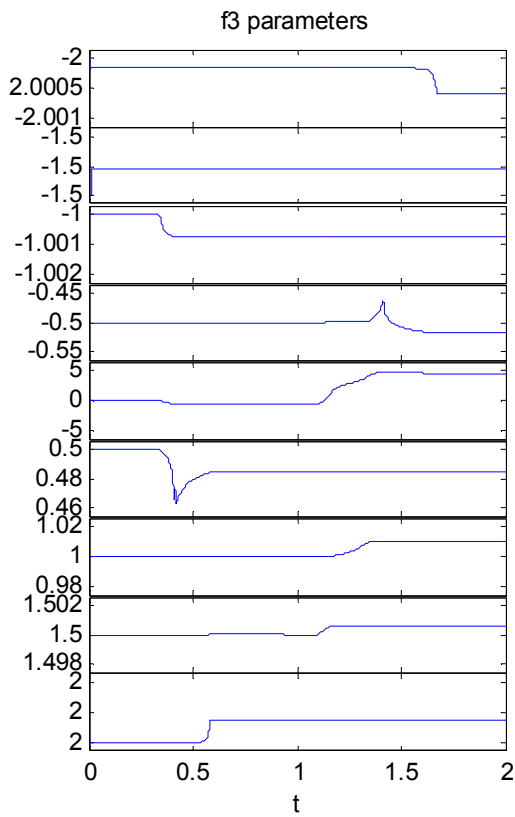
شکل ۴-۲۱. خروجی های g مربوط به ۶ کنترلر



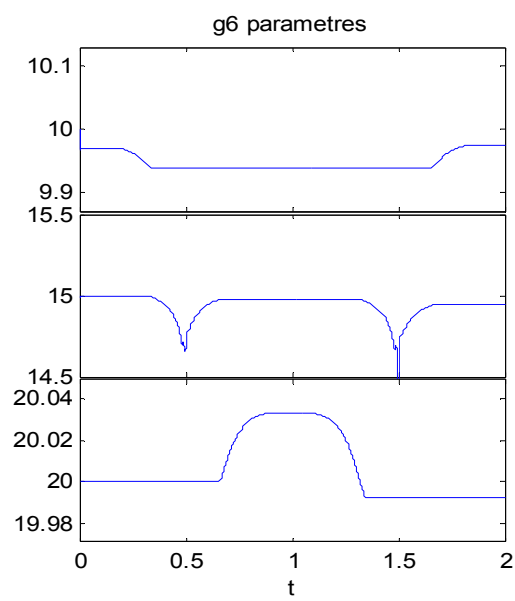
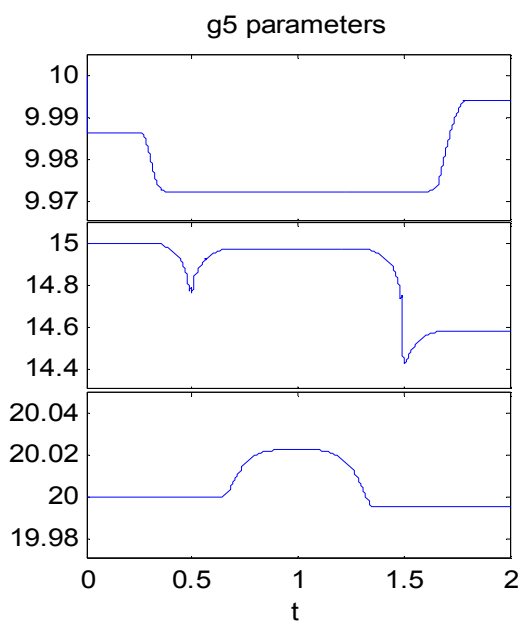
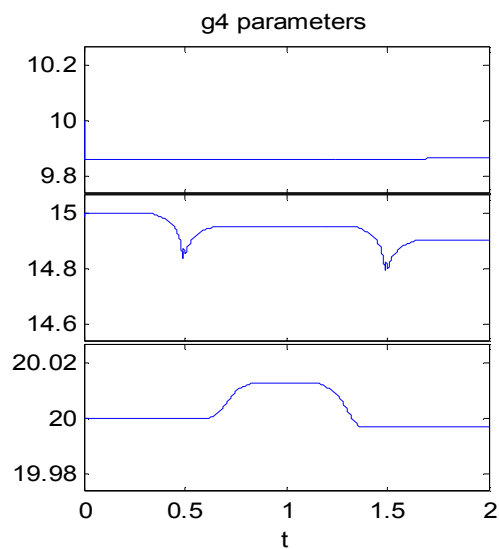
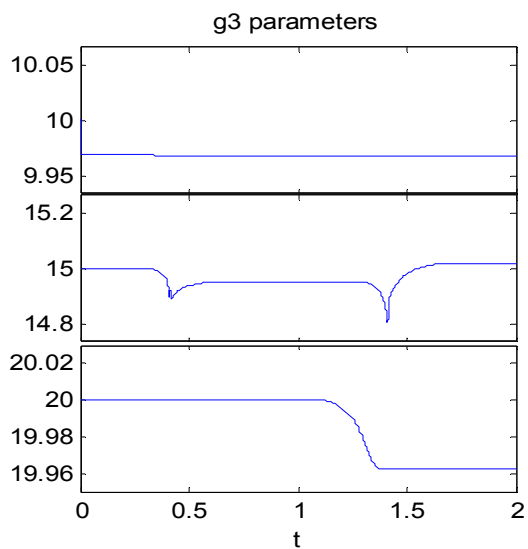
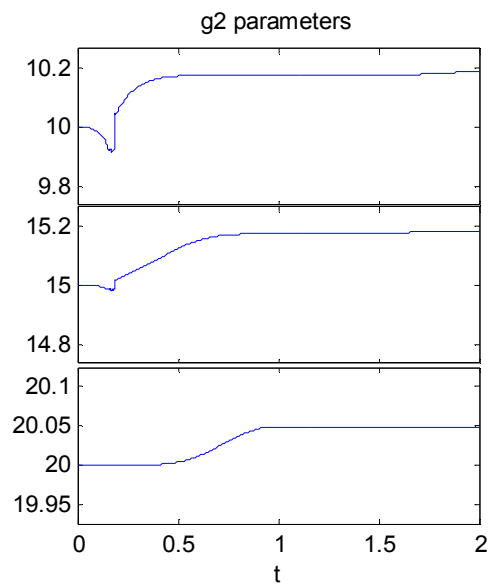
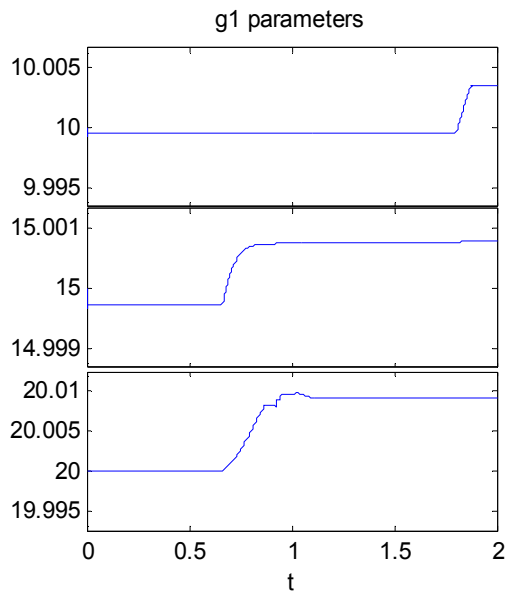
شکل ۴-۲۲. گشتاورهای موجود روی ۶ مفصل ربات پیوما ۵۶۰



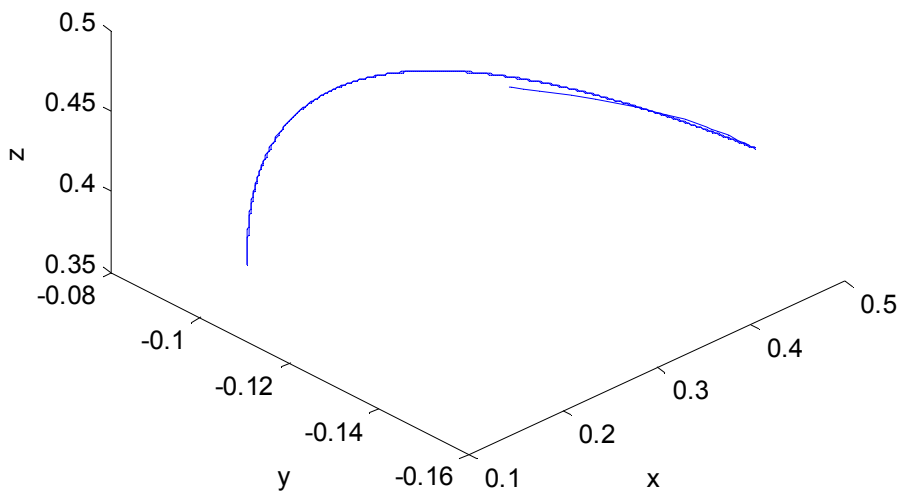
شکل ۴-۲۳. تغییر پارامترهای سیستم های فازی \tilde{f} در یک سیکل زمانی (۹ قانون برای هر سیستم تعریف شده است)



شکل ۴-۲۳. تغییر پارامترهای سیستم های فازی f در یک سیکل زمانی (۹ قانون برای هر سیستم تعریف شده است)



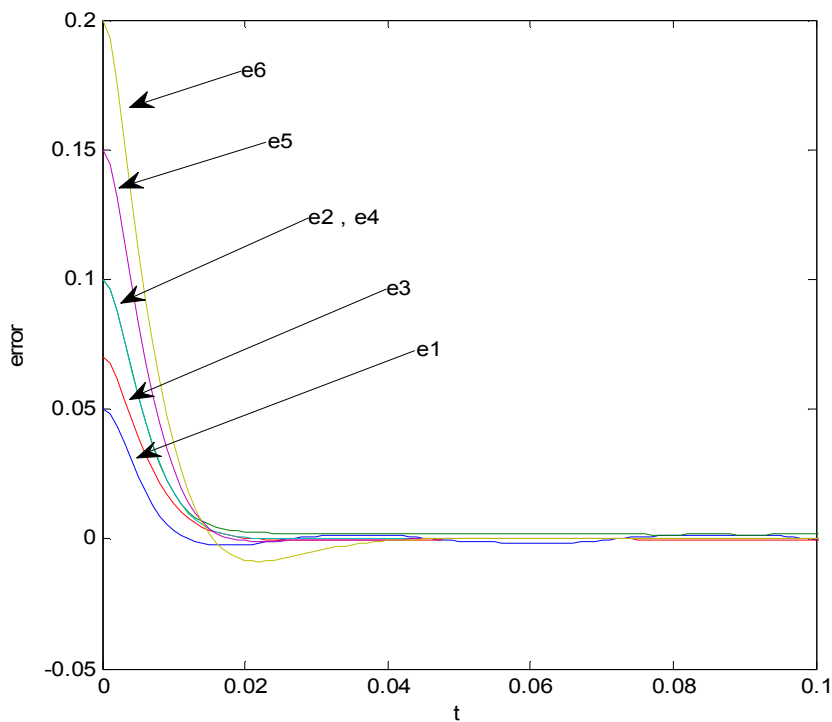
شکل ۴-۲۴. تغییر پارامترهای سیستم های فازی g (۳ قانون برای هر سیستم تعریف شده است)



شکل ۴-۲۵. مسیر نقطه انتهایی ربات در فضای دکارتی

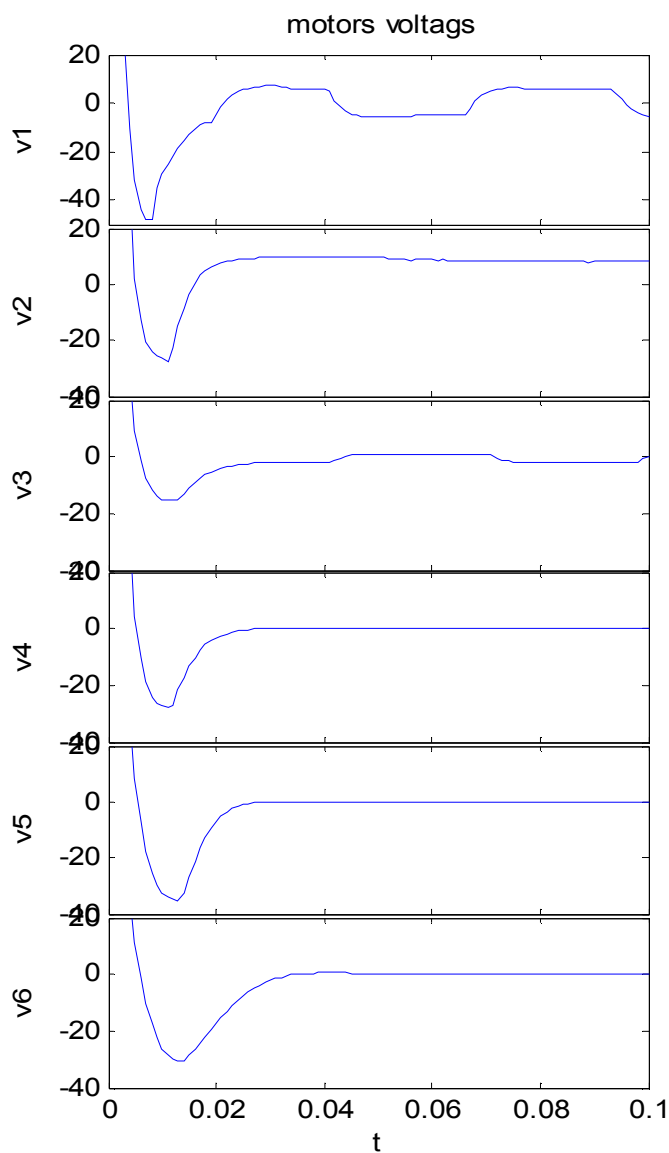
۴-۲-۲. شبیه سازی سیستم کنترل به ازای ورودی های پله

نتایج شبیه سازی کنترل فازی تطبیقی غیر مستقیم ربات پیوما ۵۶۰ را در حالی که قطب های سیستم حلقه بسته مربوط به هر کنترل مفصل مستقل را در موقعیت های $s_1 = -50 + 130i$ و $s_2 = -50 - 130i$ قرار می دهیم در زیر بررسی می نماییم. (لازم به ذکر است که در تمام ۶ سیستم کنترل مفصل مستقل تعداد قوانین سیستم فازی f برابر ۴ و تعداد قوانین سیستم فازی g را برابر با ۲ در نظر گرفته ایم).

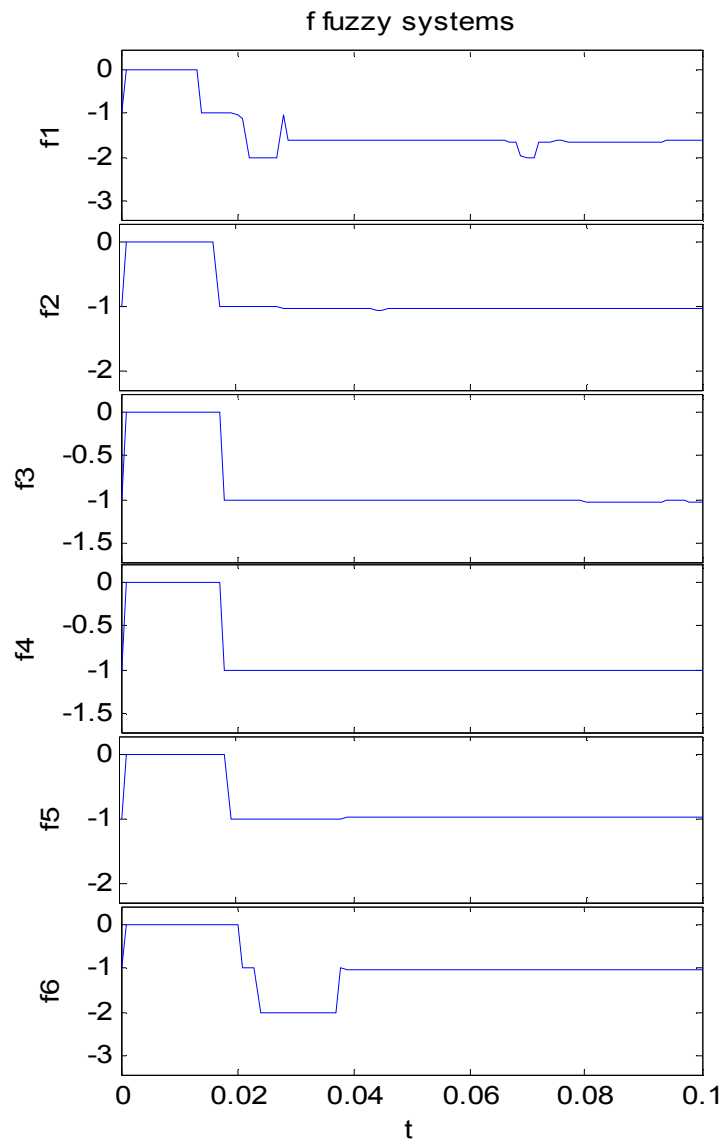


شکل ۴-۲۶. خطای ردگیری مربوط به کنترل هر ۶ مفصل ربات

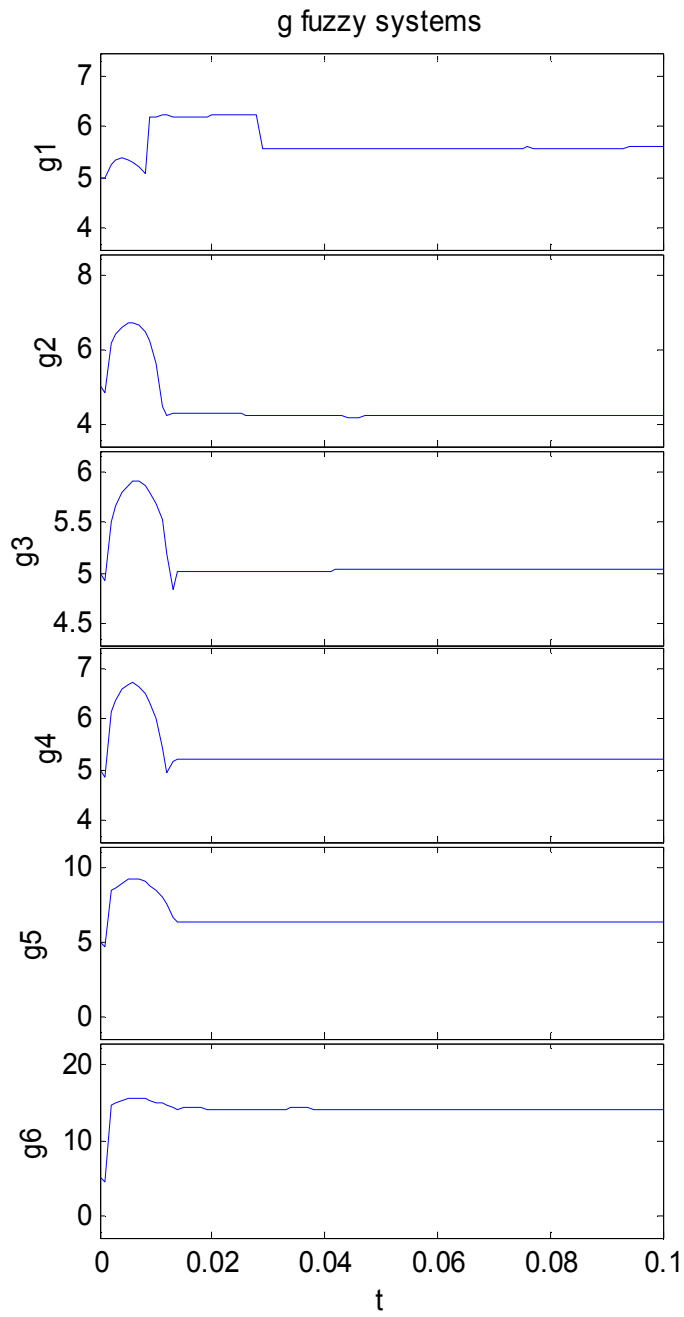
سیگنال های کنترل ایجاد شده توسط کنترل کننده های طراحی شده ، همان ولتاژهای ورودی موتورها هستند ، که به موتورها اعمال می شوند تا مفاصل را به اندازه ی موردنظر جابجا کنند . که در شبیه سازی انجام شده نتایج ناشی از سیستم کنترل بصورت زیر می باشد .



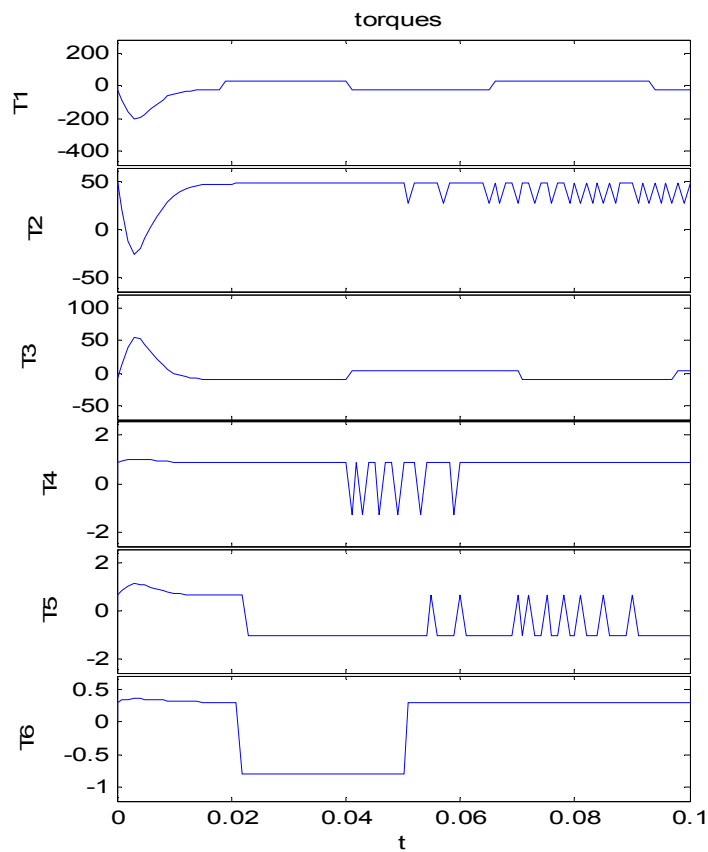
شکل ۴-۲۷ . سیگنال های کنترل هر ۶ مفصل ربات



شکل ۴-۲۸. خروجی های f مربوط به ۶ کنترلر

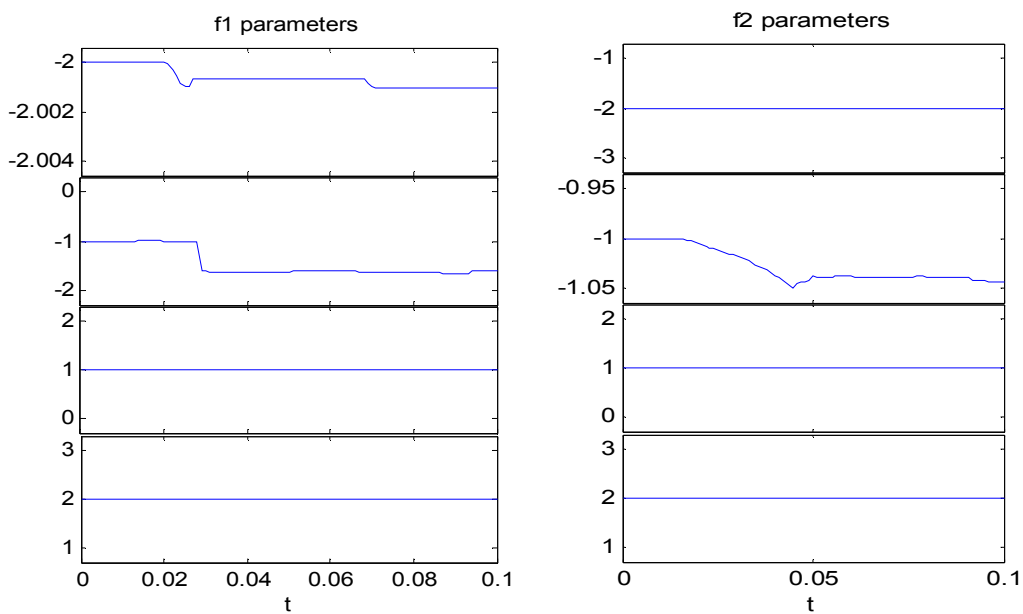


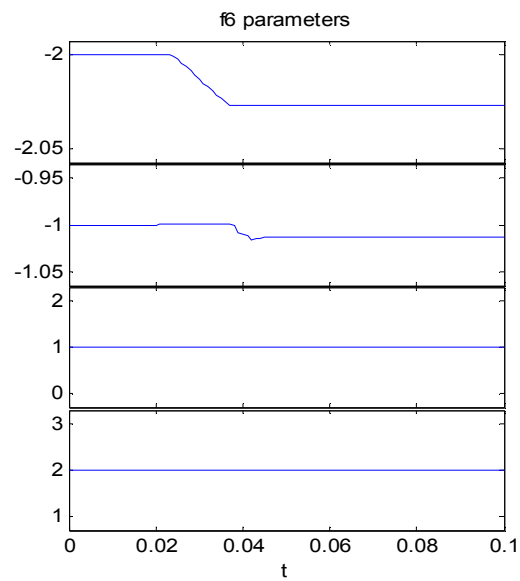
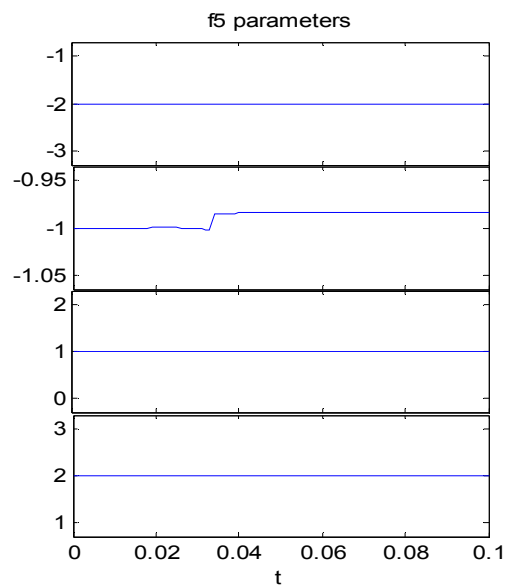
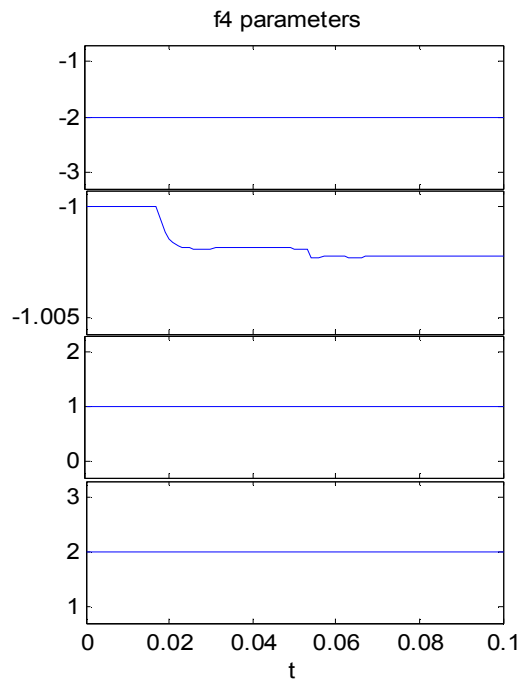
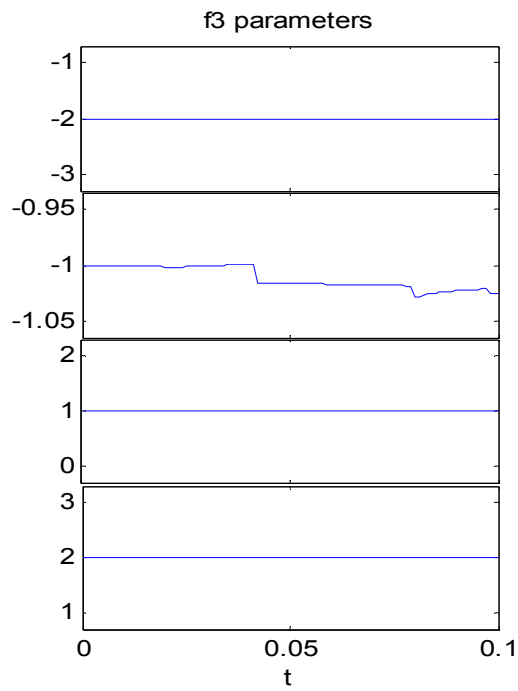
شکل ۴-۲۹. خروجی های g های مربوط به ۶ کنترلر



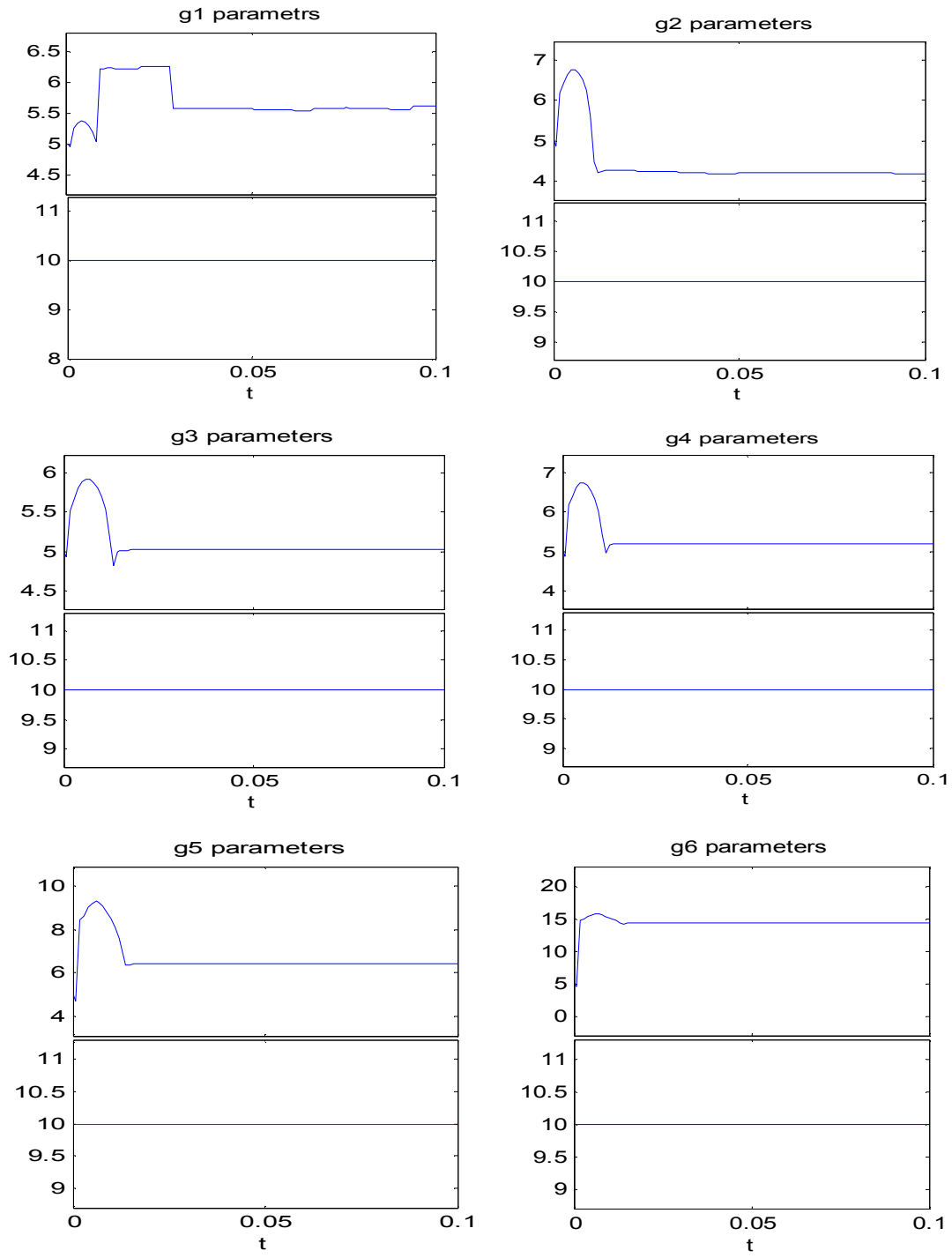
شکل ۴-۳۰. گشتاورهای موجود روی ۶ مفصل ربات پیوما ۵۶۰

پارامترهای سیستم های فازی f در یک سیکل زمانی بصورت زیر تغییر می کنند. (۴ قانون برای هر سیستم تعریف شده است)

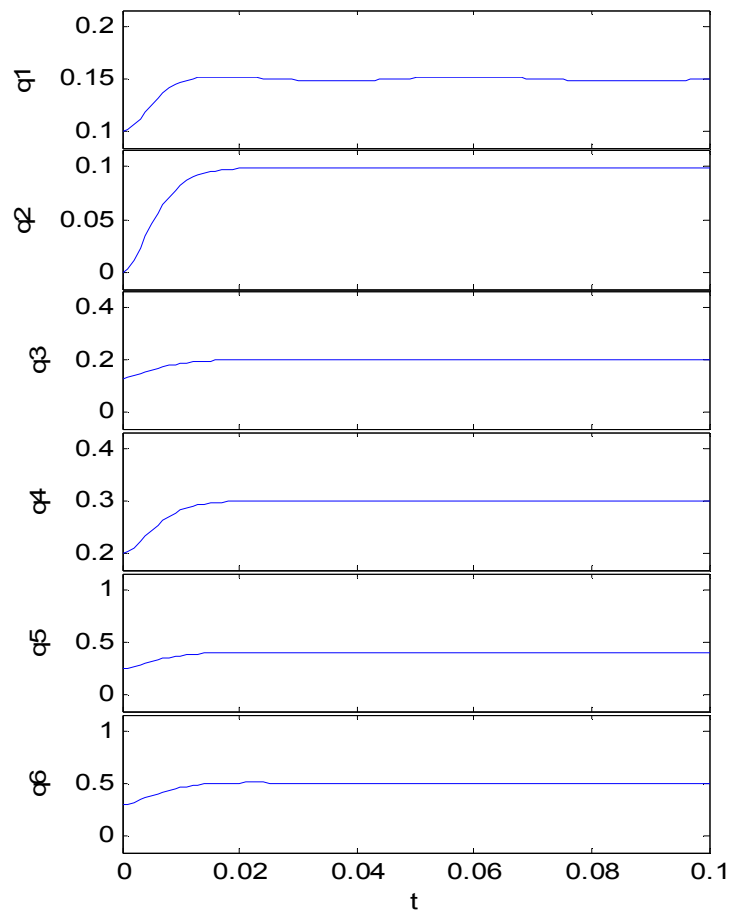




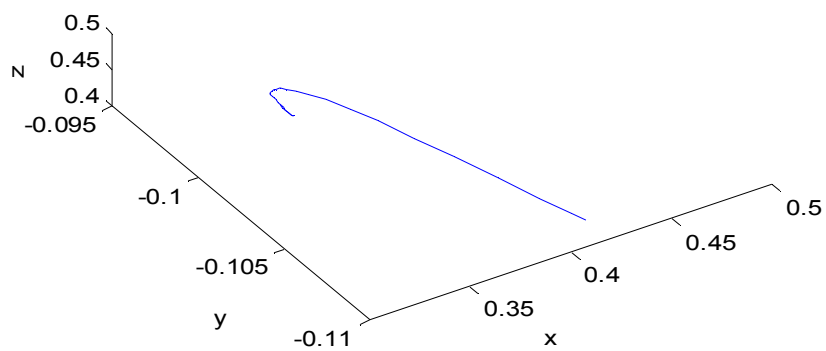
شکل ۴-۳۱. تغییر پارامترهای سیستم های فازی f در یک سیکل زمانی
(۴ قانون برای هر سیستم تعریف شده است)



شکل ۴-۳۲. تغییر پارامترهای سیستم های فازی g در یک سیکل زمانی
(۲ قانون برای هر سیستم تعریف شده است)



شکل ۴-۳۳. پاسخ های خروجی ۶ سیستم کنترل همان مفاصل ربات " پیوما ۵۶۰ "



شکل ۴-۳۴. مسیر نقطه انتهایی ربات در فضای دکارتی

مطالب بخش پنجم : مقایسه نتایج کنترل فازی تطبیقی غیر مستقیم به
روش گرادیان نزولی با کنترل کلاسیک **pd**

۵-۱ . مقایسه نتایج مربوط به سیستم کنترل موتور **dc**

۵-۱-۱ . مقایسه نتایج به ازای ورودی پله

۵-۱-۲ . مقایسه نتایج به ازای ورودی **sin**

۵-۲ . مقایسه نتایج مربوط به سیستم کنترل ربات پیوما ۵۶۰

۵-۲-۱ . مقایسه نتایج به ازای ورودی پله

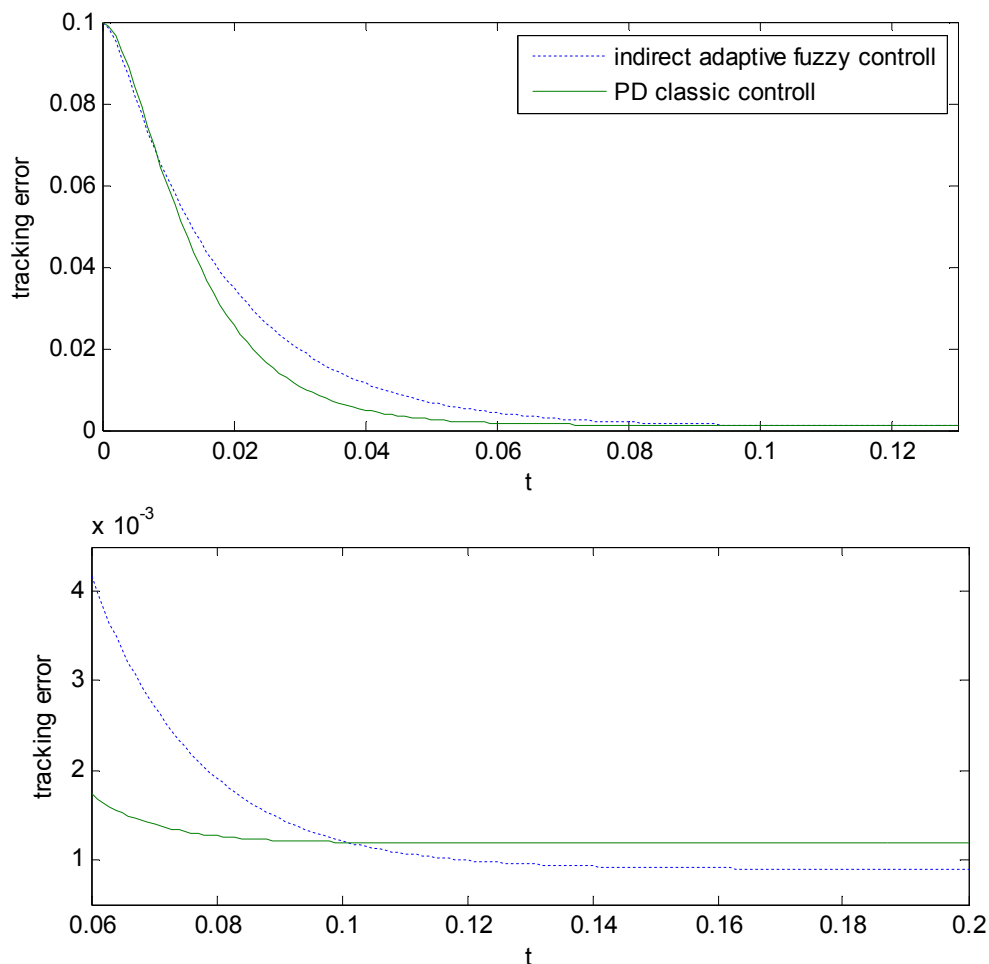
۵-۲-۲ . مقایسه نتایج به ازای ورودی **sin**

۵-۱. مقایسه نتایج مربوط به سیستم کنترل موتور dc

۵-۱-۱. مقایسه نتایج به ازای ورودی پله

در این قسمت می خواهیم نتایج کنترل فازی تطبیقی غیر مستقیم به روش گرادیان نزولی با کنترل کلاسیک pd به ازای ورودی پله مقایسه نماییم .

لازم به ذکر است که قطب های سیستم کنترل حلقه بسته در هر دو حالت (کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک PD) در $s_1 = -200, s_2 = -100$ دارد .

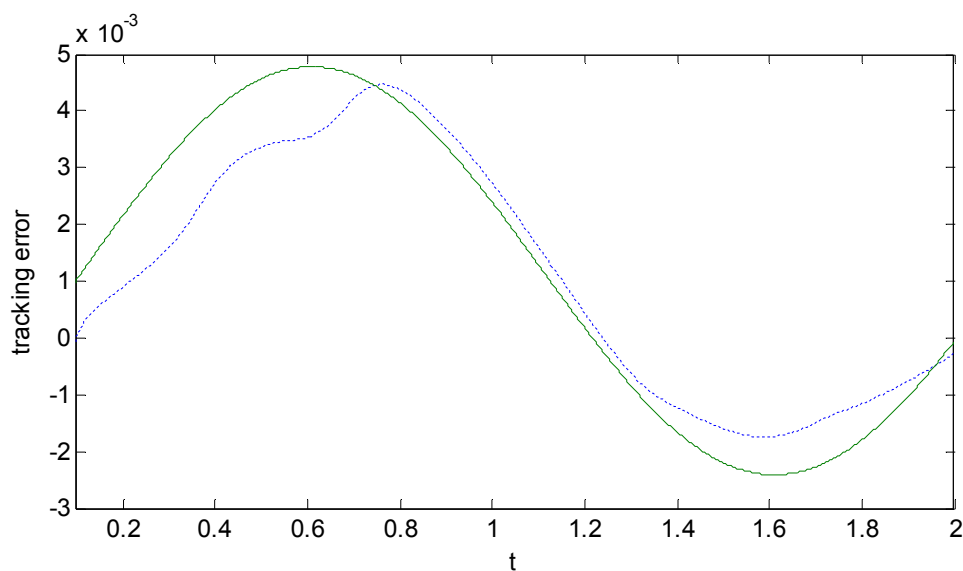
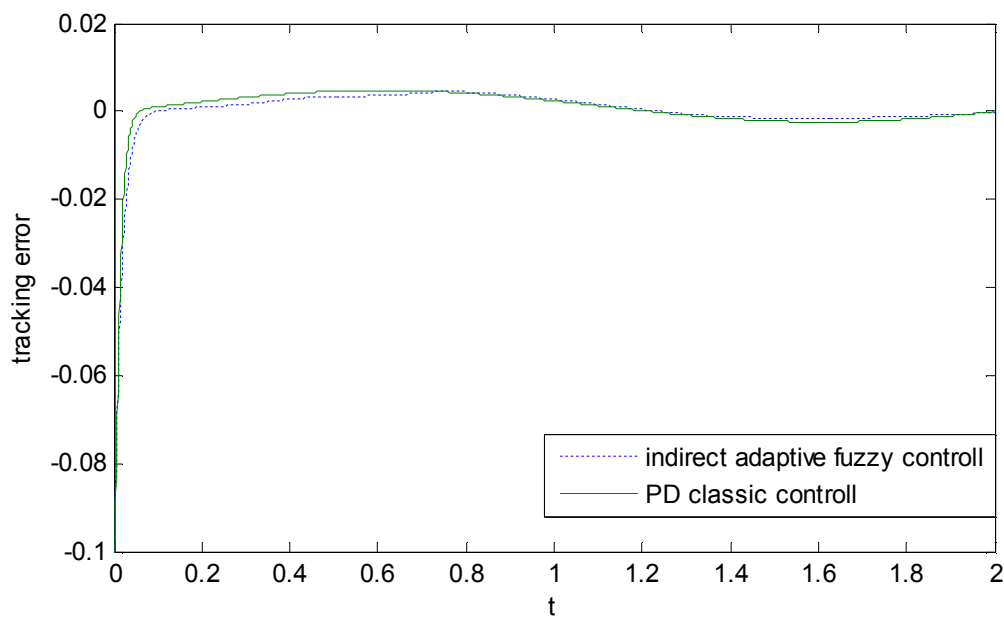


شکل ۵.۱. خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd

با مقایسه نمودارها ، خواهیم دید که خطای ردگیری در کنترل کلاسیک سریعتر خودش را به مقدار خطای دائم می رساند (در زمان 0.05s) ، در حالی که در کنترل فازی تطبیقی ، خطای ردگیری

دیرتر خودش را به مقدار دائم می رساند (در زمان $0.07s$) ولی خطای حالت دائم آن کوچکتر و در حد 0.0009 می باشد در حالیکه خطای حالت دائم در کنترل کلاسیک در حد 0.0013 می باشد .

۵-۱-۲ . مقایسه نتایج به ازای ورودی \sin ای



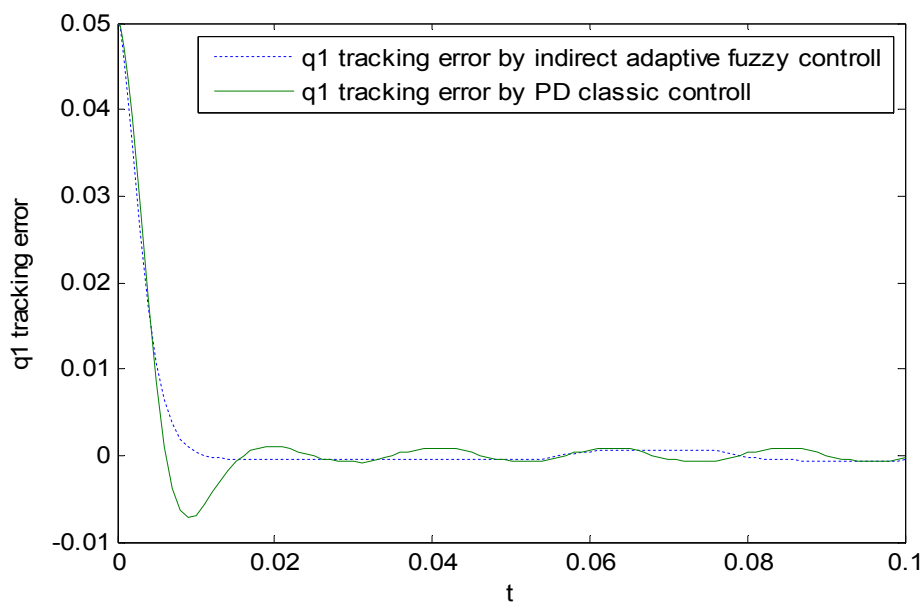
شکل ۵.۲ . خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd

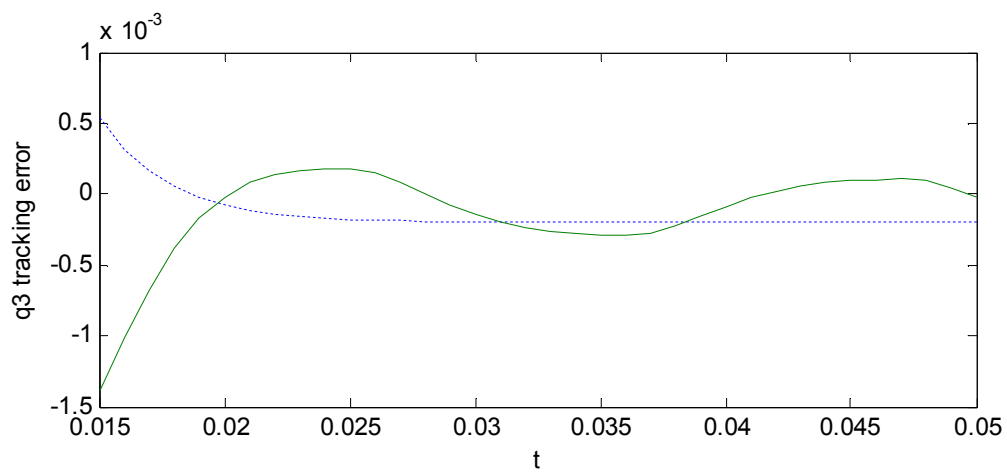
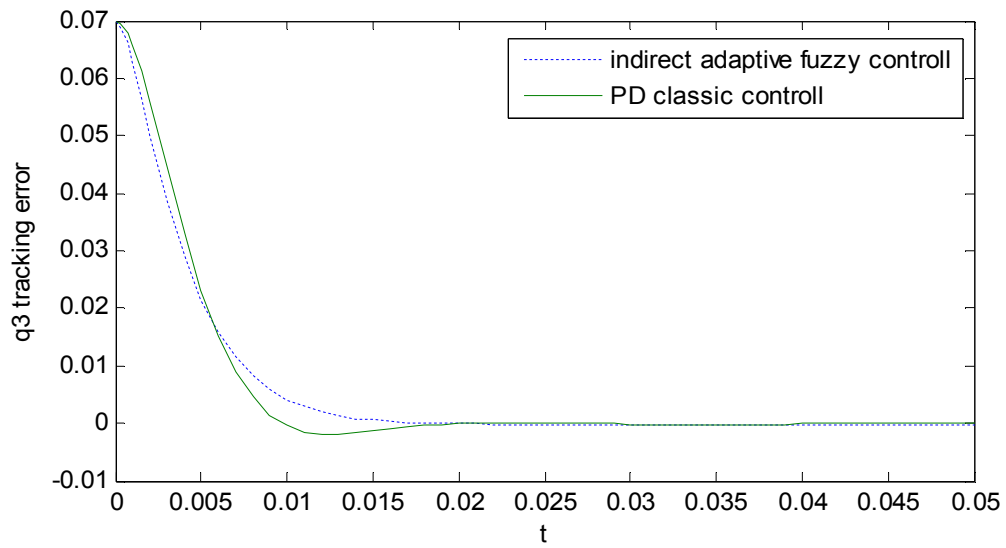
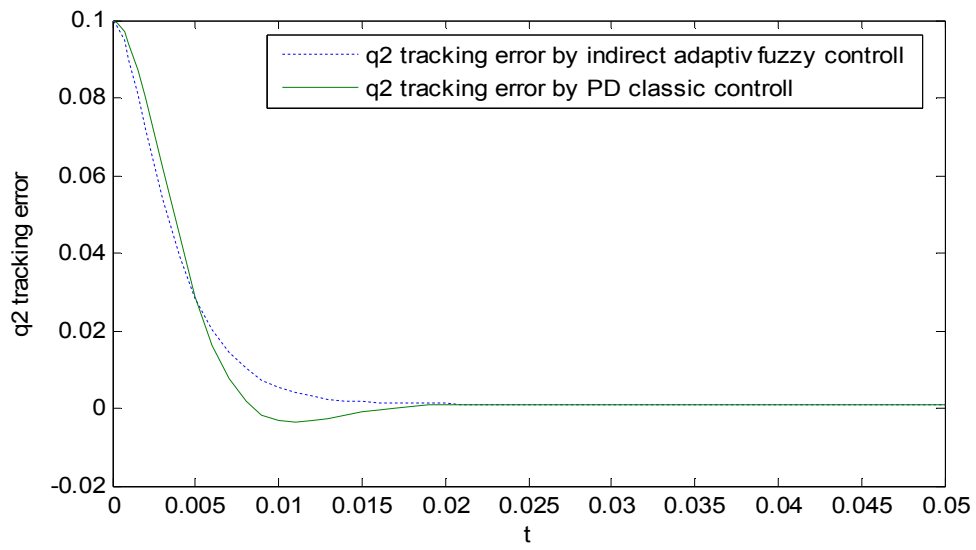
با مقایسه نمودارها ، خواهیم دید که ماکزیمم خطای ردگیری در حالت دائم در کنترل کلاسیک در حد 0.005 در حالیکه ماکزیمم خطای ردگیری در کنترل فازی تطبیقی در حد 0.0045 می باشد .

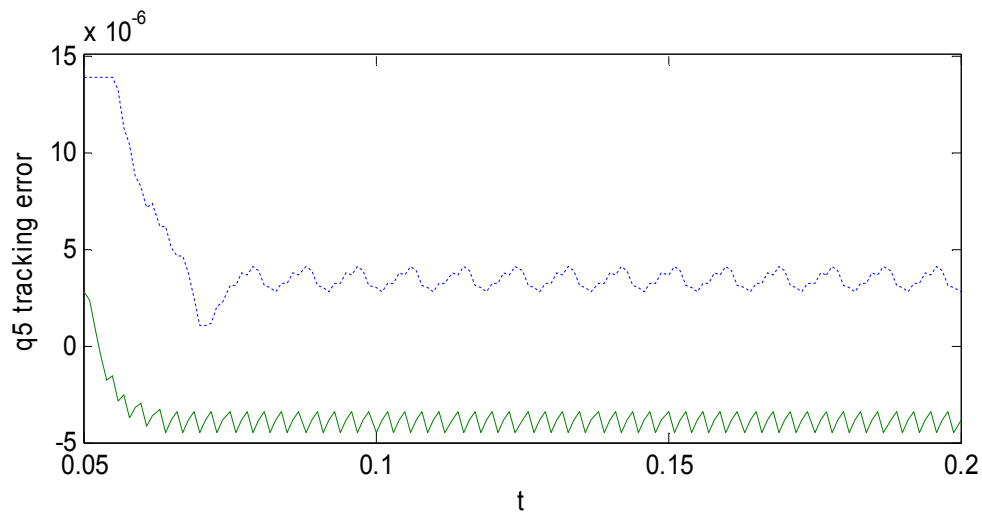
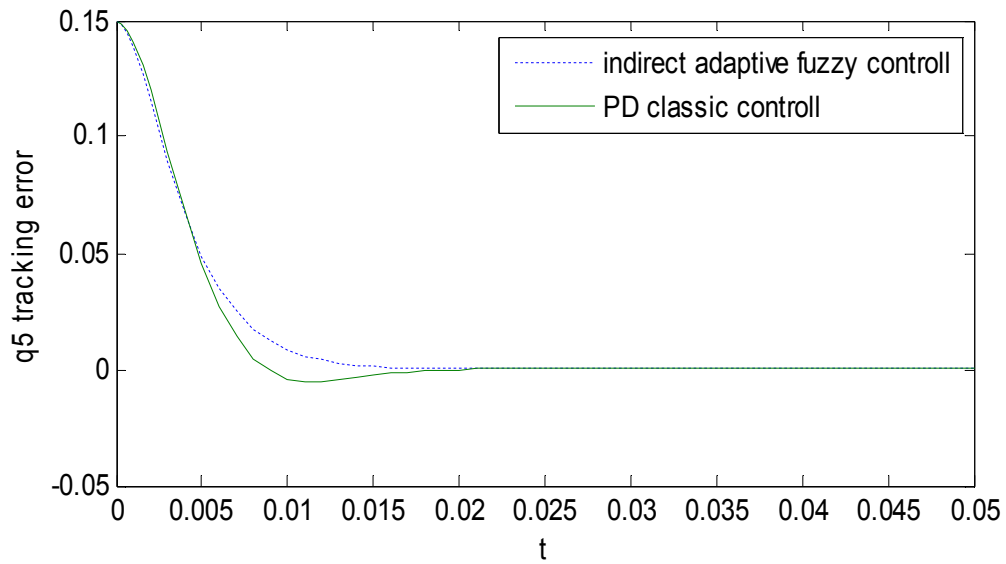
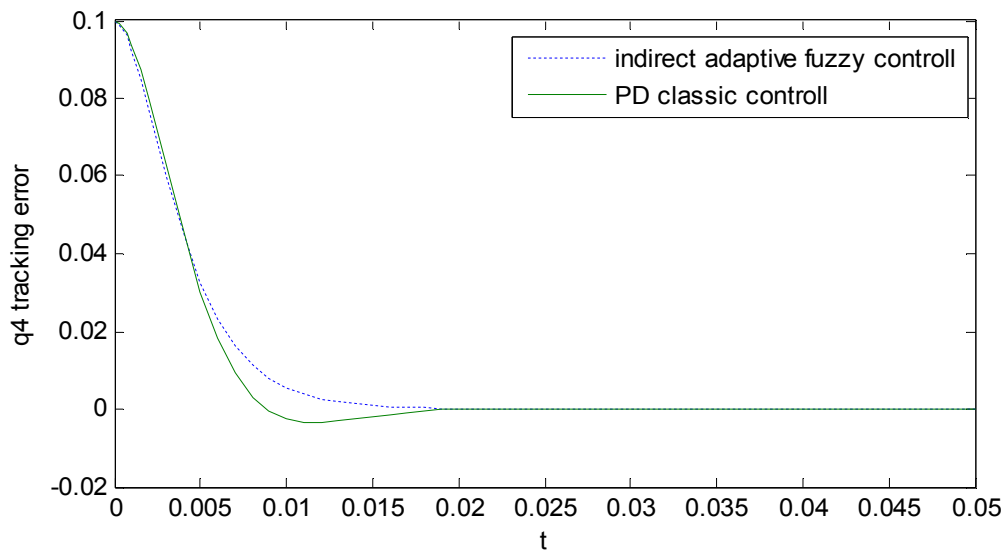
۵-۲. مقایسه نتایج مربوط به سیستم کنترل ربات پیوما ۵۶۰

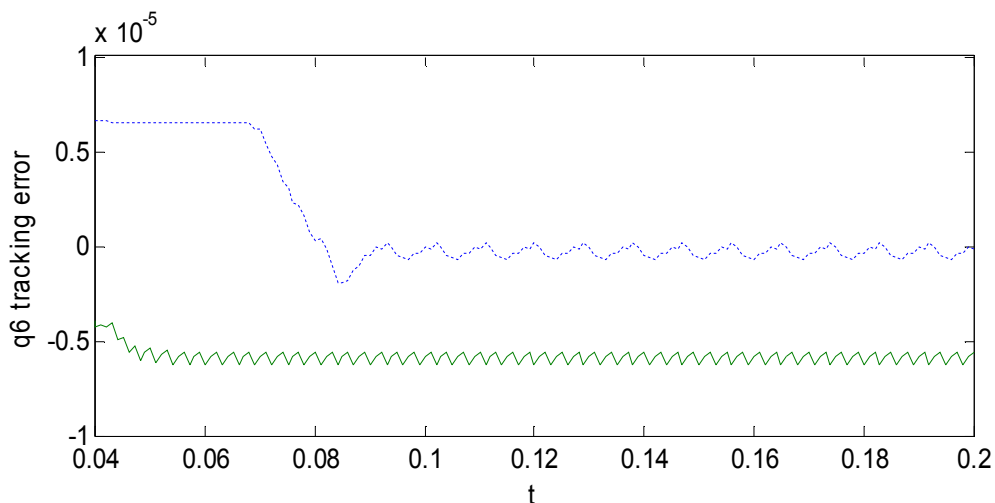
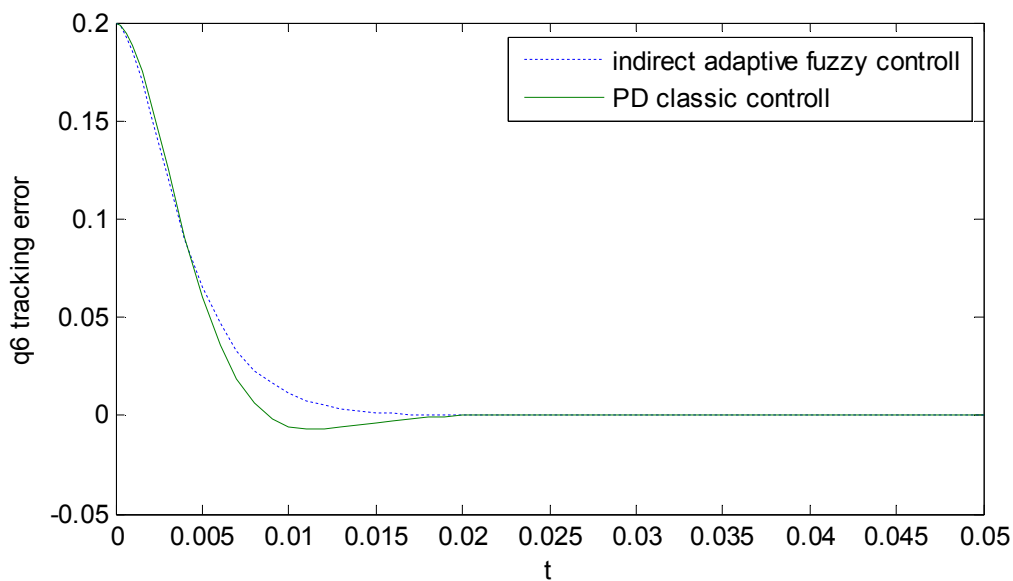
۵-۲-۱. مقایسه نتایج به ازای ورودی های پله

در این قسمت می خواهیم نتایج کنترل فازی تطبیقی غیر مستقیم به روش گرادیان نزولی با کنترل کلاسیک pd به ازای ورودی های پله مقایسه نماییم . لازم به ذکر است که قطب های هر ۶ سیستم کنترل حلقه بسته در هر دو حالت (کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک PD) در $s_1=-255+273.815i$, $s_2=-255-273.815i$ قرار دارد . خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به ۶ مفصل ربات پیوما ۵۶۰ بصورت زیر می باشد.



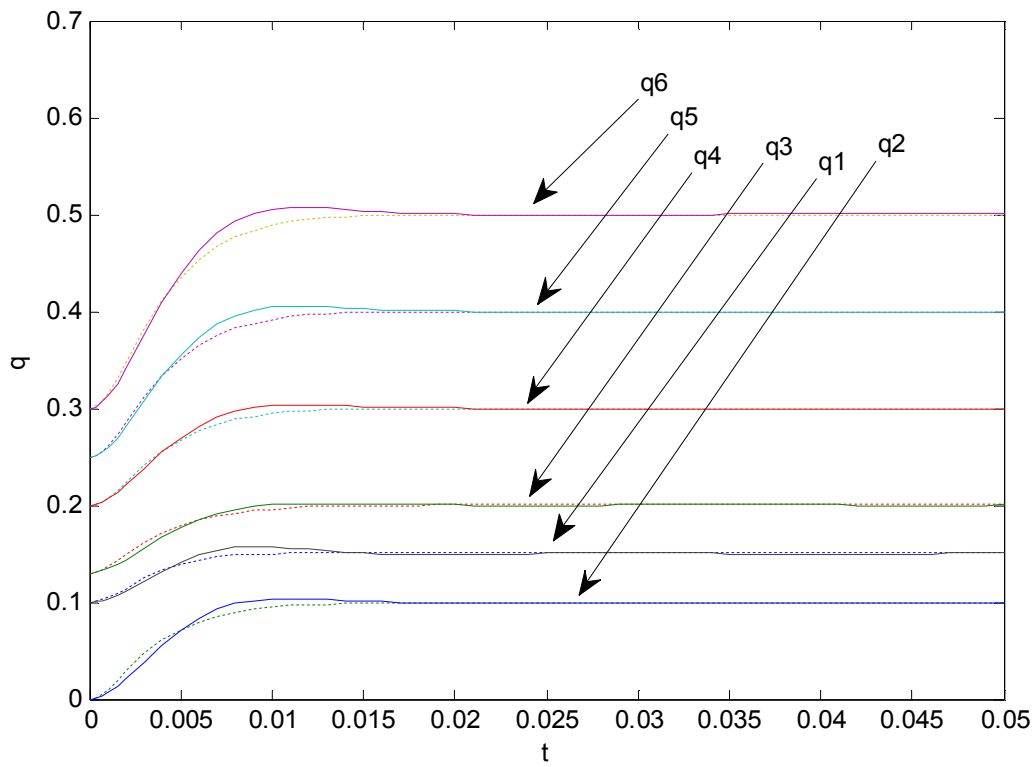






شکل ۵.۳. خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک Pd مربوط به ۶ مفصل ربات پیوما

با مقایسه نتایج می بینیم که در کنترل کلاسیک هر ۶ مفصل ربات ، خطای ردگیری دیرتر به مقدار نهایی خود میل می کند (در زمان 0.02) ، در حالی که در کنترل فازی تطبیقی این طور نیست (تقریباً در زمان 0.015 خطا به سمت مقدار نهایی خود میل می کند) و خطای حالت دائم مربوط به مفاصل q1 و q3 و q6 در کنترل فازی تطبیقی نسبت به کنترل کلاسیک کوچکتر است . و خطای حالت دائم مربوط به مفاصل q2 و q5 و q4 تقریباً در دو حالت برابر است .

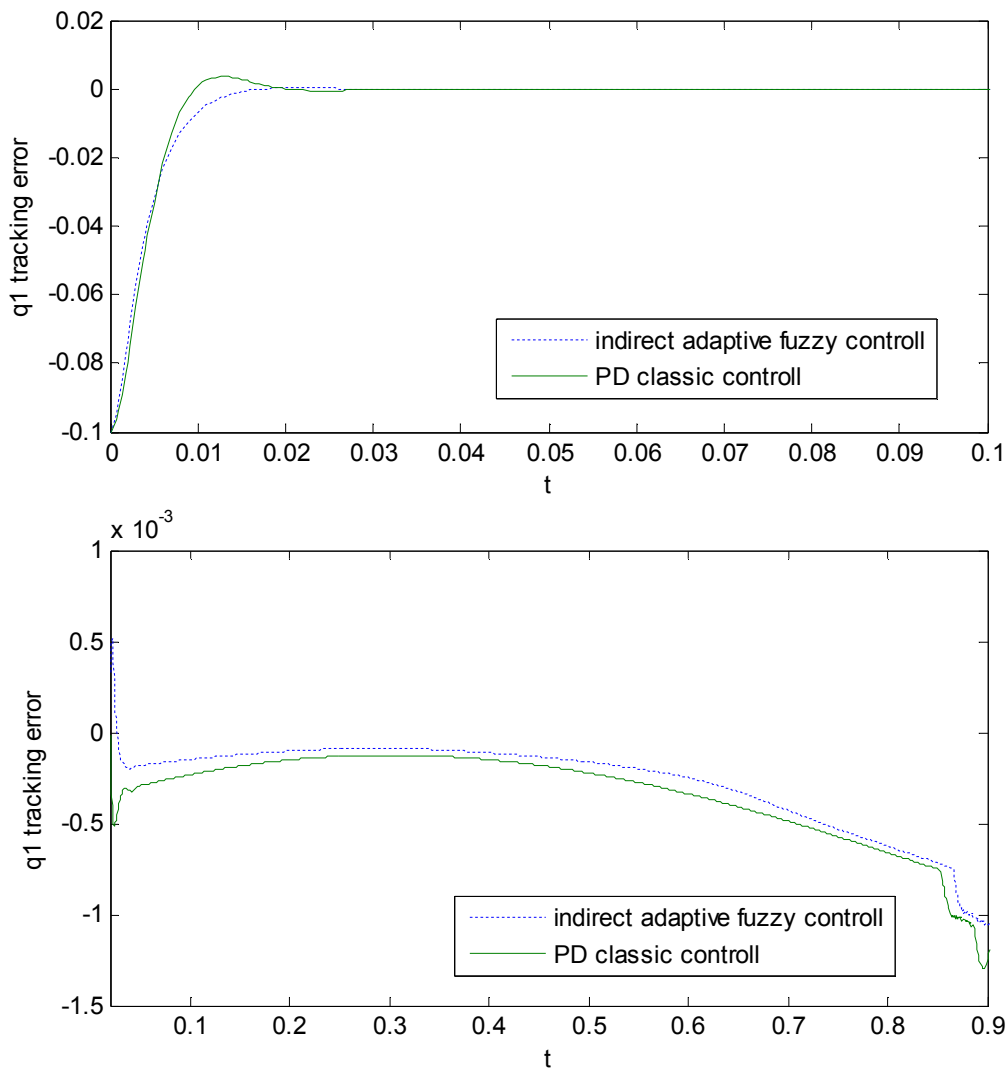


شکل ۵.۴. خروجی های سیستم های کنترل مفاصل ربات در هر دو حالت

۱. کنترل فازی تطبیقی غیر مستقیم (نمودارهای خط چین) ۲. کنترل کلاسیک PD (نمودارهای خط پر)

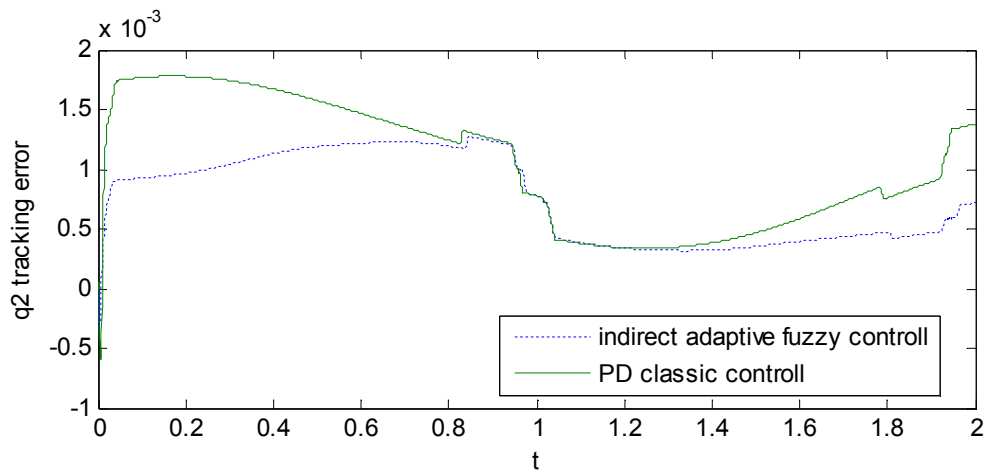
۵-۲-۲. مقایسه نتایج به ازای ورودی های \sin ای

در این قسمت می خواهیم نتایج کنترل فازی تطبیقی غیر مستقیم به روش گرادیان نزولی با کنترل کلاسیک pd به ازای ورودی های پله مقایسه نماییم . لازم به ذکر است که قطب های هر ۶ سیستم کنترل حلقه بسته در هر دو حالت (کنترل فازی تطبیقی غیر مستقیم و کنترل کلاسیک PD) در $s_1 = -255 + 273.815i$ ، $s_2 = -255 - 273.815i$ قرار دارد .



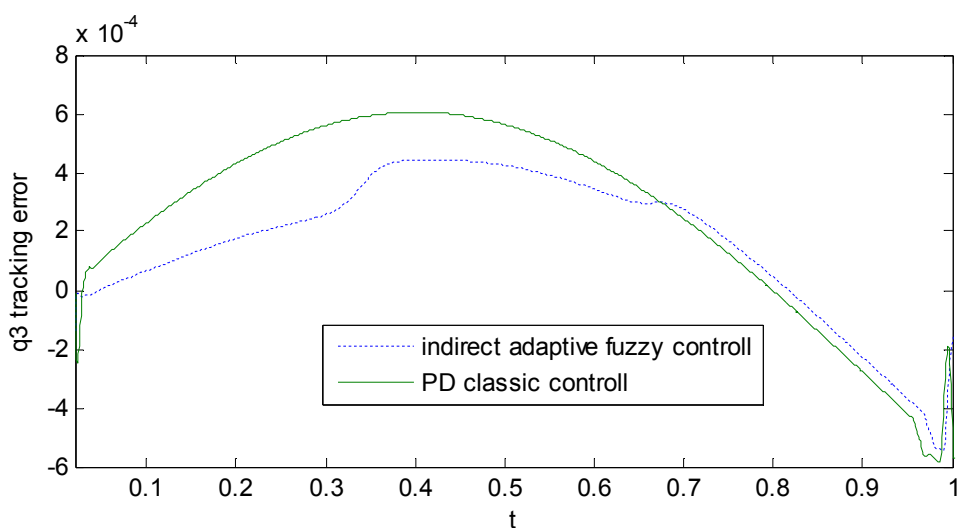
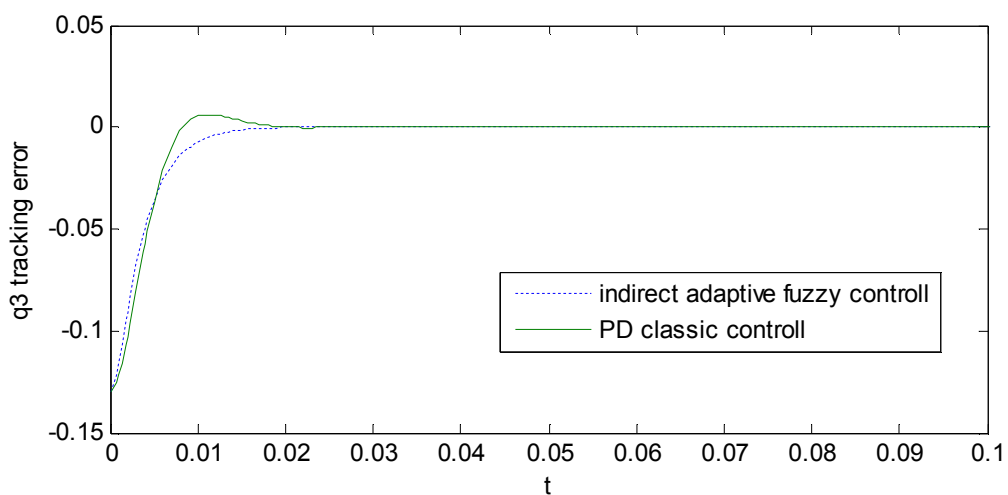
شکل ۵.۵ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۱ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۵ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۱ در کنترل فازی تطبیقی -0.0002 ، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک -0.0003 می باشد . در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد .



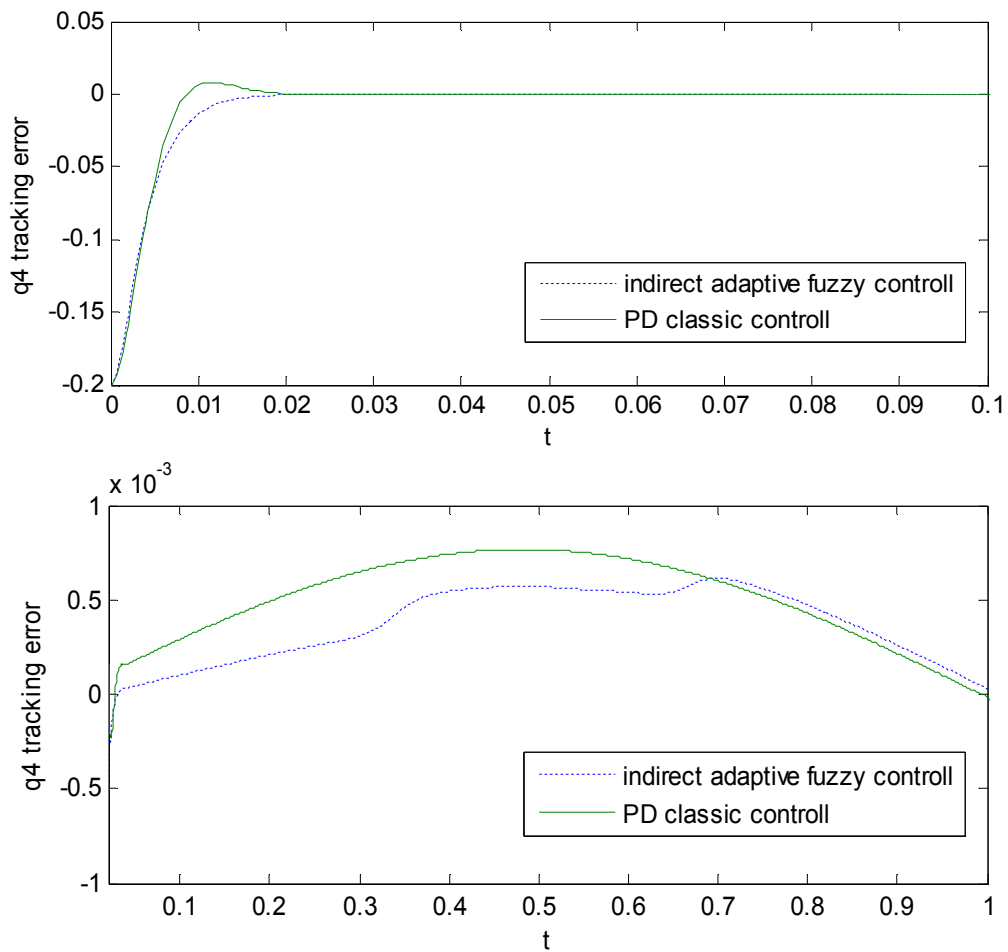
شکل ۵.۶ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۲ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۶ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۲ در کنترل فازی تطبیقی 0.0013 ، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک 0.0017 می باشد . در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد .



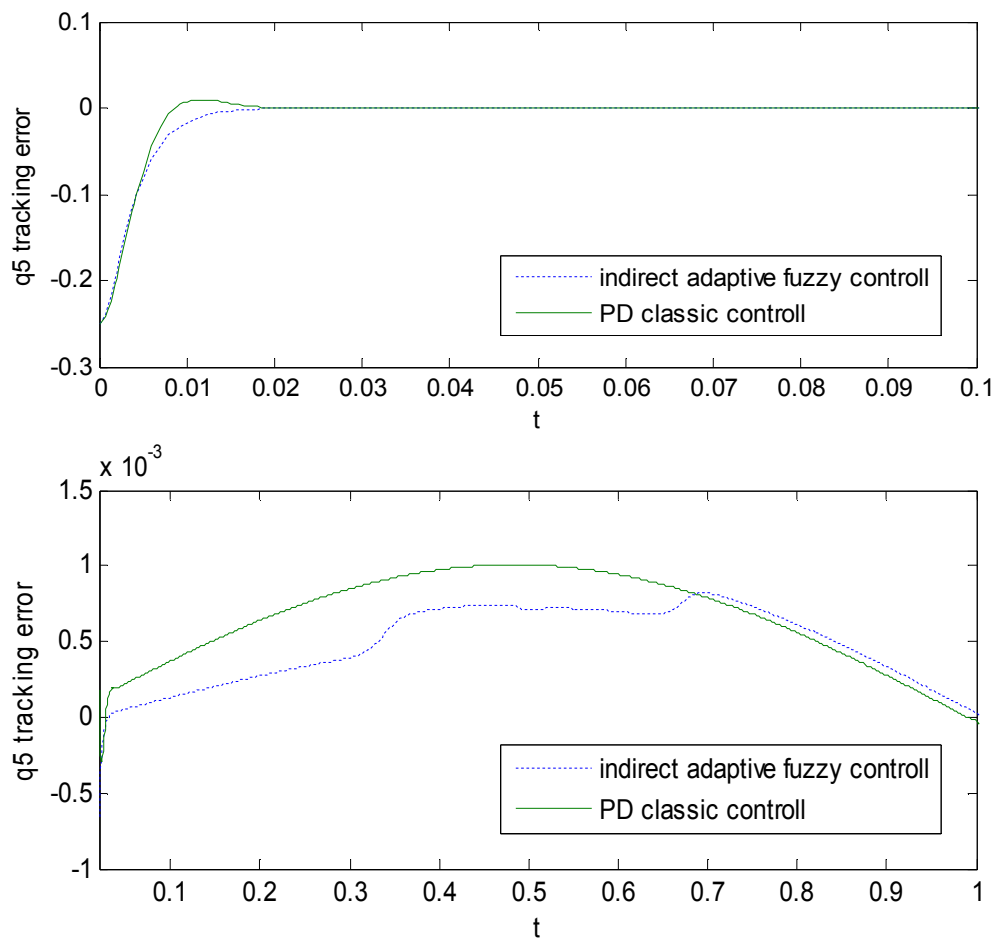
شکل ۵.۷ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۳ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۷ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۳ در کنترل فازی تطبیقی 0.00045، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک 0.0006 می باشد. در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد.



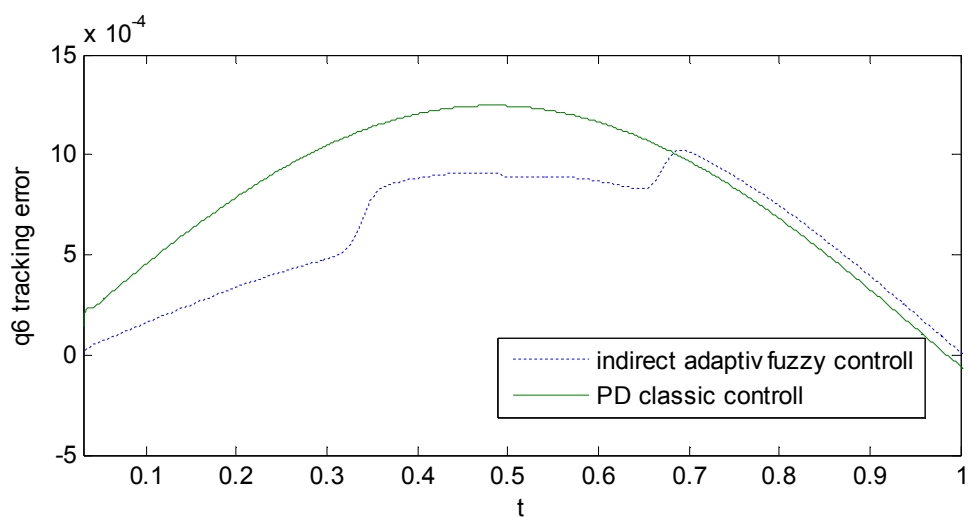
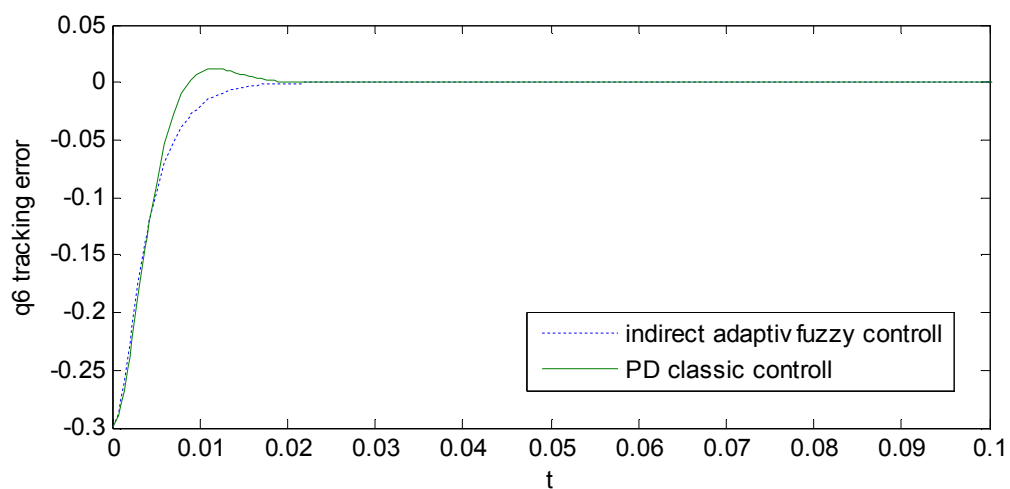
شکل ۵.۸ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۴ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۸ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۴ در کنترل فازی تطبیقی 0.0006 ، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک 0.0008 می باشد . در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد .



شکل ۵.۹ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۵ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۹ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۵ در کنترل فازی تطبیقی 0.0008 ، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک 0.001 می باشد . در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد .



شکل ۵.۱۰ خطای ردگیری سیستم کنترل فازی تطبیقی غیر مستقیم و سیستم کنترل کلاسیک pd مربوط به مفصل ۶ ربات پیوما ۵۶۰

با مقایسه دو نمودار در شکل ۵.۱۰ دیده می شود که ماکزیمم خطای ردگیری مربوط به مفصل ۶ در کنترل فازی تطبیقی 0.001، در حالیکه ماکزیمم خطای ردگیری در کنترل کلاسیک 0.0012 می باشد. در نتیجه کنترل فازی تطبیقی نتیجه بهتری می دهد.

مطالب بخش ششم:

نتیجه گیری

- با توجه به آن که در کنترل فازی یک سیستم پیچیده ، دارای عدم قطعیت و متغییر با زمان ، برای کاهش خطای ردگیری ، نیاز به جابجایی توابع تعلق ورودی و خروجی کنترلر فازی یعنی تنظیم پارامترهای کنترلر داریم ، الگوریتم آموزش گرادیان نزولی را برای بهینه سازی پارامترها پیشنهاد می دهیم .
- برای آموزش سیستم کنترل با الگوریتم گرادیان نزولی، انتخاب پارامترهای اولیه برای همگرایی الگوریتم ، بسیار حائز اهمیت می باشد . اگر پارامترهای اولیه انتخابی نزدیک به مقادیر بهینه باشند ، الگوریتم آموزش در پاسخ بهینه سراسری^۱ همگرا می شود . در غیر این صورت ممکن است در پاسخ بهینه محلی^۲ همگرا شود وحتی واگرا شود .
- انتخاب تعداد قوانین بیشتر (M بزرگتر) ، پارامترهای بیشتری را بوجود آورده و حجم محاسبات را بیشتر می کند ولی در عین حال تقریب با دقت بهتری را نتیجه می دهد .
- در بهینه سازی سیستم های فازی براساس روش های گرادیان ، این روش ها یکسری مزایا و معایبی دارند . از جمله :
 ۱. همگرایی سریعتری نسبت به روشهای گرادیان آزاد دارند .
 ۲. مشکل همگرایی به بهینه های محلی و گیر کردن در تله های بهینه محلی وجود دارد. برای رفع این مشکل، بایستی مقادیر اولیه پارامترهای سیستم فازی را آنقدر تغییر داد و یا از روش ممنوم برای تنظیم^۳ کردن تغییرات پارامترها استفاده کرد تا در دام مینیمم های محلی گیر نیافتیم.
 ۳. با توجه به نیاز به شرایط مشتق پذیری محدودیتهایی در تعریف توابع هدف ، نوع موتور استنتاج شکل توابع تعلق وجود دارد .

1. Global
2. Local
3.Update

۴. مزیت کنترل تطبیقی نسبت به کنترل غیر تطبیقی این است که، در کنترل تطبیقی تغییر مقادیر ضرایب یا دیگر پارامترها در قانون کنترل مطابق با برخی الگوریتم های بلادرنگ می باشد. در این نوع کنترل، کنترل کننده می تواند یک مجموعه از پارامترها را در حین دوره ی بهره برداری یاد بگیرد. این ایده مخصوصا برای بازوهای ماهر که کارهای تکراری انجام می دهند بسیار مفید است. بدون تطبیق، خطاهای ردیابی نیز تکراری هستند. با تطبیق، کارایی ردیابی می تواند از طریق تکرار پی در پی بهبود یابد.
۵. اگر سرعت آموزش خیلی بزرگ باشد، آن الگوریتم ناپایدار می شود. اگر سرعت آموزش مقدار کوچکی انتخاب شود، الگوریتم آموزش زمان طولانی را برای همگرا شدن پاسخ به مقدار نهایی، صرف می کند. بنابراین لازم است سرعت آموزش مقدار مناسبی انتخاب گردد.
۶. تنظیم پارامترهای سیستم فازی و تغییرات زیاد آن ها به محل قطب ها نیز وابسته است، هر چه قطب ها نزدیکتر باشند، پارامترها بیشتر (بزرگتر) تغییر می کنند و رنج تغییرات زیاد است و همچنین هر چه قطب ها دورتر باشند، چون سریعتر پاسخ خروجی به ورودی همگرا شده، تغییرات پارامترها خیلی کمتر است.
۷. هر چه تعداد قوانین (تعداد گروههای فازی ورودی در یک سیستم فازی) بیشتر باشد، تطبیق سریعتر انجام گرفته و پارامترها تغییرات زیادی ندارند، یعنی رنج تغییرات پارامترها خیلی کم است.
۸. در زمان های بالا در سیستم کنترل، با استفاده از گرادیان نزولی، چون dJ به سمت صفر میل میکند، در نتیجه پارامترها تقریبا ثابت شده و تغییر نمی کنند (حتی اگر ورودی سیستم کنترل مثلا \sin ای باشد).

عیب روش کنترل فازی تطبیقی غیر مستقیم با روش گرادیان نزولی

۹. از جمله معایب روش گرادیان نزولی این است که :

برای داشتن خطای ردگیری حالت دائم کوچکتر (در حد 0.0001) باید قطب های سیستم کنترل حلقه بسته فازی تطبیقی غیر مستقیم را خیلی دور از مبدا قرار داد (قسمت حقیقی قطب ها در حد -300) ، در غیر این صورت خطای ردگیری در حالت دائم زیاد (در حد 0.01) خواهد شد .

پیوست ها

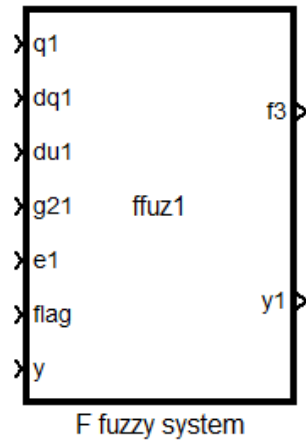
پیوست ۱. برنامه مربوط به سیستم فازی \hat{f} با تنظیم پارامترها در نرم افزار مطلب

پیوست ۲. برنامه مربوط به سیستم فازی \hat{g} با تنظیم پارامترها در نرم افزار مطلب

پیوست ۳. برنامه ساخت قانون کنترل در نرم افزار مطلب

پیوست ۱.

برنامه مربوط به سیستم فازی \hat{f} با تنظیم پارامترها در نرم افزار مطلب ، بصورت زیر است که داخل بلوک Embedded matlab functions در simulink مطلب نوشته می شود .



شکل ۷.۱. بلوک سیستم فازی \hat{f}

```
function [f3,y1]=ffuz1(q1 , dq1 , du1 , g21 , e1 , flag , y)
```

گام اول : تعیین مراکز و پهنای گروه های فازی مربوط به ورودی که همواره برای تمام دیتاهای ورودی ثابت باقی می مانند .

```
gama=100;
a=[0;0.2;0.4;-0.65;0;0.65];
b=[0.06;0.06;0.06;0.17;0.17;0.17];
```

گام دوم : تشکیل گروه های فازی مربوط به دو ورودی (توابع عضویت به صورت گوسین هستند)

```
A11=exp(-(q1-a(1))^2/b(1));
A12=exp(-(q1-a(2))^2/b(2));
A13=exp(-(q1-a(3))^2/b(3));
A21=exp(-(dq1-a(4))^2/b(4));
A22=exp(-(dq1-a(5))^2/b(5));
A23=exp(-(dq1-a(6))^2/b(6));
```

گام سوم : تشکیل جدول قوانین فازی

```
h1=A11*A21;  
h2=A11*A22;  
h3=A11*A23;  
h4=A12*A21;  
h5=A12*A22;  
h6=A12*A23;  
h7=A13*A21;  
h8=A13*A22;  
h9=A13*A23;
```

گام چهارم : تشکیل خروجی سیستم فازی

```
s=h1*y(1)+h2*y(2)+h3*y(3)+h4*y(4)+h5*y(5)+h6*y(6)+h7*y(7)+h8*y(8)+h(9)*y(9);  
n=h1+h2+h3+h4+ h5+h6+h7+h8+h9;
```

```
if n==0  
    f3=0;  
else  
    f3=s/n;  
end
```

گام پنجم : بررسی تابع هزینه و در صورتیکه این معیار هنوز مینیمم نباشد در این مرحله وارد شده و

پارامترهای سیستم فازی که همان خروجی های قوانین فازی می باشند با الگوریتم گرادیان نزولی

قابل تنظیم هستند .

```
if flag==1  
  
    if (du1*g21*n)==0  
        g1=0;  
    else  
        g1=(e1)*(dq1)/[du1*g21*n];  
    end  
  
    y11=y(1)-gama*h1*g1;  
    y12=y(2)-gama*h2*g1;  
    y13=y(3)-gama*h3*g1;  
    y14=y(4)-gama*h4*g1;  
    y15=y(5)-gama*h5*g1;  
    y16=y(6)-gama*h6*g1;  
    y17=y(7)-gama*h7*g1;  
    y18=y(8)-gama*h8*g1;  
    y1=[y11;y12;y13;y14; y15;y16;y17;y18;y19];  
y=y1;
```

```

s=h1*y(1)+h2*y(2)+h3*y(3)+h4*y(4)+h5*y(5)+h6*y(6)+h7*y(7) +h8*y(8)+h(9)*y(9);

if n==0
    f3=0;
else
    f3=s/n
end

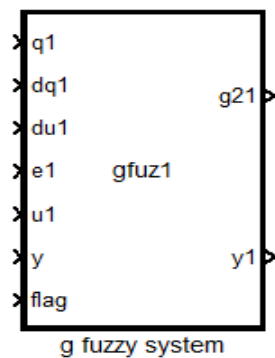
end

y1=y;
end

```

پیوست ۲ .

برنامه مربوط به سیستم فازی \hat{g} با تنظیم پارامترها در نرم افزار مطلب ، بصورت زیر است که داخل بلوک Embedded matlab functions در simulink مطلب نوشته می شود .



شکل ۷.۲. بلوک سیستم فازی \hat{g}

```
function [g21,y1]=gfuz1(q1,dq1,du1,e1,u1,y,flag)
```

گام اول : تعیین مراکز و پهنای گروه های فازی مربوط به ورودی که همواره برای تمام دیتاهای ورودی ثابت باقی می ماند .

```

a=[0;0.1;0.2];
b=[0.06;0.06;0.06];
gama=100;

```

گام دوم : تشکیل گروه های فازی مربوط به دو ورودی (توابع عضویت به صورت گوسین هستند)

```
A1=exp(-(q1-a(1))^2/b(1));  
A2=exp(-(q1-a(2))^2/b(2));  
A3=exp(-(q1-a(3))^2/b(3));
```

گام سوم : تشکیل جدول قوانین فازی

```
h1=A1;  
h2=A2;  
h3=A3;
```

گام چهارم : تشکیل خروجی سیستم فازی

```
s=h1*y(1)+h2*y(2)+h3*y(3);  
n=h1+h2+h3;
```

```
if n==0  
    g21=0;  
else  
    g21=s/n;  
end
```

گام پنجم : بررسی تابع هزینه و در صورتیکه این معیار هنوز مینیمم نباشد در این مرحله وارد شده و

پارامترهای سیستم فازی که همان خروجی های قوانین فازی می باشند با الگوریتم گرادیان نزولی

قابل تنظیم هستند .

```
if flag==1  
    if (g21*du1*n)==0  
        g1=0;  
    else  
        g1=(e1)*dq1*[u1]/(du1*g21*n);  
    end
```

```

y11=y(1)-gama*(g1*h1);
y12=y(2)-gama*(g1*h2);
y13=y(3)-gama*(g1*h3);
y1=[y11;y12;y13];

y=y1;

s=h1*y(1)+h2*y(2)+h3*y(3);

if n==0
    g21=0;
else
    g21=s/n;
end

end

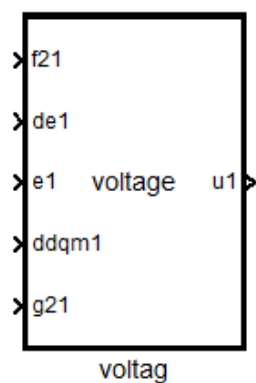
y1=y;

end

```

پیوست ۳ .

برای ساخت این قانون کنترل از بلوک Embedded matlab function در محیط simulink مطلب استفاده کردیم .



شکل ۷.۳. بلوک قانون کنترل

و قانون کنترل بصورت زیر در آن آمده است .

```
function u1=voltage(f21,de1,e1,ddqm1,g21)

if g21==0
    u1=0;
else
    u1=(-f21+ddqm1+600*de1+250000*e1)/g21;
end
end
```

- [1] . J. S. R. Jang , C. T. Sun, and E. Mizutani, " Neuro–Fuzzy and Soft Computing ". Englewood Cliffs, NJ : Prentice-Hall, 1997.
- [2] . L-X. Wang , "A Course in Fuzzy Systems And Control" , Prentice - Hall PTR , Upper Saddle River, New Jersey, 1997.
- [3] . J . Xiong and T . Li and H . Xiao and R . Deng , " Study on Adaptive Fuzzy control system Based on Gradient Descent learning Algorithm " , pp. 1009-1020, 2007.
- [4] . Y. shi and M. Mizumoto and N.Yubazaki and M. otani , " A Learning Algorithm for tuning Fuzzy Rules Based on the gradient Descent Method " , pp. 55-61, 1996 .
- [5]. Y.P. Huuange and C.C. Huang and C.H. Huang , " Optimizing the fuzzy Adaptive Learning by the Gradient Descent Approach " , pp. 701-705, 1994.
- [6]. C.-C.Wong , B.-C.Huang and J.-Y.Chen , " Rule regulation of indirect adaptive fuzzy controller " , IEE Proc – control theory Appl., vol. 145 , No .6 , November 1998 , pp.513-518, 1998.
- [7]. K.- Ch.Chiou and S.-J.Huang , " An Adaptive fuzzy controller for Robot manipulator " , pp. 151-177 , 29 July 2004 .
- [8] . Mark W. Spong and M.Vidyasagar , " Robot Dynamics and Control " , 1983.

- [9]. A. Lesewed , J . Kurek , " Calculation of Robot Parameters Based on Neural Nets " , Fifth International Workshop on Robot Motion and Control , pp.117-122 , 2005.
- [10]. N. Golea , " Indirect Fuzzy Adaptive Model – following control for Robot Manipulators " , Proc . IEEE 2002 Int. Conf . on control , Applications , pp.198- 202 , 2002.
- [11]. M. S. Hitam , and K. F. Gill , " Indirect Adaptive control of a Robot using Fuzzy Neural Network " , IEEE Trans . pp. 303-307 , 2000 .
- [12].V. M. Hung and U. J. Na , " Adaptive Neural fuzzy control for Robot Manipulator Frictions and Disturbance compensator " , IEEE 2008 Int. Conf. on control , Automation and systems, pp. 2569-2574 , 2008.
- [13]. S. H. Kim, Y. H. Kim, K. B. Sim, and H. T. Jeon, " On Developing an Adaptive Neural – Fuzzy control system " , Proc . IEEE 1993 Int. Conf. On Intelligent Robots and systems " , pp. 950-957 , July 26-30 , 1993.
- [14]. H. Nomura, I. Hayashi, and N. wakami, " A learning method of fuzzy inference rules by descent method , " Proc. Fuzz-IEEE, pp.203-210, 1992.
- [15]. Guely and P.Siarry, " Gradient descent method for optimizing various fuzzy rule bases " , Proc. Fuzz-IEEE , pp. 1241-1246, 1993.
- [16]. Y.-P. Hung , C.-C. Hung ,and , C.-C. Chang , " A systematic approach to adjust the fuzzy set parameters" , Proc. Fisrt Asian Fuzzy Systems Symp. , Singapore , Nov .1993.

Abstract

In this thesis, the goal of arms control expert Robotics, Robot "Puma 560" is. This robot has six joints that hinge joint of each robot 6 by a stimulus (motor DC) to the move to position the robot end points to reach our desired location. This robot has many applications in industry.

Between methods of control, adaptive fuzzy control method for indirect control of the robot we have chosen, for the design of fuzzy systems in this control method, the parameters of these systems with a gradient descending method we adjust to the tracking error between the desired position with the robot joints is minimum.

In doing this project, before the controller design for robot Puma 560, this type of controller for a DC motor design we have followed its result, the controller voltage signal to control each of the six motor to the robot joints motion we use brings.

The control on each joint is done separately, which is independent of the detailed control strategy.

In this case the coupling between the robot joints have been ignored, but because the control system, Adaptive coupling between joints is that the unrest and uncertainty in the system are considered.

It should be noted that our original system, no relationship of robot "Puma 560" and not related to the equations of the robot is unknown to us.

Simulation system for robot control in the toolbox Puma 560 Robotic content use software.

Finally, this control method with a classical control methods that classical PD control method and to compare this result we will find that the method of indirect adaptive fuzzy control, steady state error, but have less control compared to classic PD, later control system should be matched.

Keywords: indirect adaptive fuzzy control, gradient descending, DC motor and robot Puma 560



Shahrood University of Technology

Faculty of Electrical, Electronic & Robotic Engineering

**Indirect Adaptive Fuzzy Control of a Puma 560 Robot
using Gradient Descent training**

Mansoureh Ghoosheh

**Supervisor :
Dr. M. M. Fateh**

September 2010