

دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

پایان نامه کارشناسی ارشد برق - کنترل

عنوان:

## طراحی و ساخت پایدار ساز فازی سیستم قدرت

اساتید راهنما:

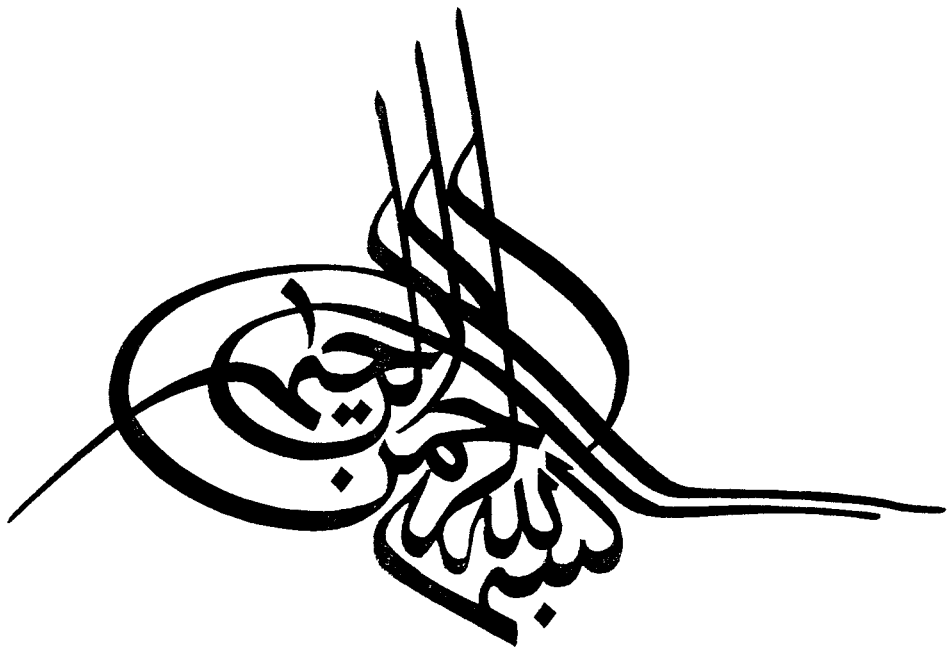
دکتر علی اکبر قره ویسی

دکتر ابراهیم حسینی

ارائه دهنده:

ابراهیم نور محمدی

شهریور ۱۳۸۳



تتمیم پیم

پدر بزرگوارم

و

مادر صبورم

.

.

و

همسر مهربانم

## چکیده

در این پایان نامه، ابتدا عملکرد کلی پایدارساز سیستم (PSS) قدرت به عنوان کنترل کننده‌ای که جهت افزودن میرایی به نوسانات روتور ژنراتور با اعمال یک سیگنال اضافی به کنترل تحریک، بکار می‌رود. تا با تولید یک جزء از گشتاور الکتریکی همفاز با تغییرات سرعت، میرایی لازم را برای این نوسانات فراهم کند، مورد بررسی قرار گرفته است.

سپس به بررسی و مدلسازی یک سیستم قدرت تک ماشینه که با خطوط انتقال به شین بی‌نهایت متصل شده پرداخته شده است. مدل بدست آمده حاکی از این است که سیستم قدرت یک سیستم بسیار غیرخطی است که عملکرد دینامیکی آن تحت تأثیر مجموعه وسیعی از تجهیزات تشکیل دهنده آن قرار دارد.

در ادامه با استفاده از یک PSS کلاسیک سعی در بدست آوردن یک شناخت کلی در مورد رفتار PSS در سیستم قدرت شده است که منجر به طراحی PSSهای فازی به شیوه‌های ممدانی و سوگنؤ گردیده است. و در نهایت با ساخت یک سخت افزار میکروکنترلی و انتقال پایدارساز به بیرون از کامپیوتر، نحوه ایجاد یک کنترل کننده فازی سخت‌افزاری بررسی شده و نتایج بدست آمده تحلیل گردیده است.

## فهرست مطالب

مقدمه	۱
<b>فصل اول - سیستم قدرت و پایدارسازهای سیستم قدرت</b>	<b>۴</b>
۱-۱ پایداری سیستم قدرت	۵
۲-۱ پایدارساز سیستم قدرت	۹
۳-۱ انواع پایدارسازهای سیستم قدرت	۱۱
۱-۳-۱ ورودی سرعت	۱۲
۲-۳-۱ ورودی توان	۱۲
۳-۳-۱ ورودی فرکانس	۱۳
۴-۳-۱ پایدارسازهای دیجیتالی	۱۴
۴-۱ ملاحظات بر اهداف اجرایی PSS	۱۴
<b>فصل دوم - مدلسازی سیستم قدرت</b>	<b>۱۷</b>
۱-۲ مدلسازی سیستم قدرت	۱۹
۲-۲ معادلات ماشین سنکرون	۲۰
۳-۲ نمایش سیستم تحریک	۲۵
۴-۲ نمایش PSS کلاسیک	۲۶
۵-۲ نمایش PSS فازی	۲۸
۶-۲ نمایش PSS فازی سخت افزاری	۲۹
<b>فصل سوم - منطق فازی و کنترل فازی</b>	<b>۳۰</b>
۱-۳ منطق فازی	۳۲
۲-۳ سیستم‌های فازی	۳۴

۳۹	.....کنترل فازی	۳-۳
۴۰	.....کنترل فازی در مقایسه با کنترل کلاسیک	۴-۳
۴۱	.....نحوه ایجاد کنترل کننده‌های فازی در <b>MATLAB</b>	۵-۳
۴۲	.....طراحی PSS فازی	۶-۳
۴۳	.....۱-۶-۳ طراحی PSS ممدانی	۳-۳-۶
۴۵	.....۲-۶-۳ طراحی PSS سوگنؤ	۳-۳-۶
۴۶	<b>فصل چهارم – پیاده‌سازی سخت‌افزاری پایدار ساز سیستم قدرت .....</b>	
۴۸	.....۱-۴ ساختارهای سخت‌افزاری برای سیستم‌های منطق فازی	۴-۱
۴۸	.....۲-۴ انواع پردازشگرهای اختصاصی برای منطق فازی	۴-۲
۴۹	.....۱-۲-۴ نسل اول - پردازشگرهای فازی آنالوگ	۴-۲-۱
۵۰	.....۲-۲-۴ نسل دوم - پردازشگرهای فازی دیجیتال	۴-۲-۲
۵۲	.....۳-۲-۴ نسل سوم - پردازشگرهای فازی مجتمع	۴-۲-۳
۵۳	.....۳-۴ میکروکنترلرها و میکروپروسورهای استاندارد	۴-۳
۵۵	.....۴-۴ ساخت PSS فازی میکروکنترلری	۴-۴
۵۵	.....۱-۴-۴ ایجاد ارتباط با سخت افزار توسط <b>Simulink</b>	۴-۴-۱
۵۹	.....۲-۴-۴ شرح سخت افزار	۴-۴-۲
۶۸	<b>فصل پنجم – مقایسه و تحلیل نتایج .....</b>	
۶۹	.....۱-۵ توصیف سیستم قدرت استفاده شده:	۵-۱
۷۰	.....۱-۱-۵ پارامترهای ژنراتور	۵-۱-۱
۷۰	.....۲-۱-۵ پارامترهای سیستم تحریک	۵-۱-۲
۷۱	.....۲-۵ شبیه‌سازی‌های انجام شده	۵-۲

۸۱	..... فصل ششم - نتیجه‌گیری و پیشنهادات
۸۲	..... ۱-۶ نتیجه‌گیری
۸۴	..... ۲-۶ پیشنهادات
۸۵	..... مراجع و ماخذ
۸۹	..... ضمائم
۹۰	..... ضمیمه الف - Mfile مربوط به ارتباط سخت افزاری
۹۴	..... ضمیمه ب - کد اسمبلی مربوط به سخت افزار





## مقدمه

با پایه‌ریزی و بسط منطق فازی توسط محققین و مهندسين مختلف که در این زمینه تلاش می‌کردند، دری تازه به روی حل مشکلات صنعتی گشوده شد و سیستم‌های فازی و کنترل کننده‌های فازی به‌عنوان یکی از گزینه‌هایی که از سیستم پردازش ذهن انسان برای کنترل فرایندهای مختلف استفاده می‌کرد، وارد عرصه صنعت شدند، و هم اکنون سیستم‌های بسیاری در زندگی روزمره می‌توان یافت که کنترل کننده‌های فازی نقشی اساسی را در آنها بازی می‌کنند.

کنترل کننده‌های فازی توانسته‌اند با ورود به عرصه کنترل صنعتی گامی جدید را در هوشمند کردن سیستم‌های صنعتی بردارند، به عنوان مثال پایدارسازهای فازی ارائه شده در مقالات مختلف عملکرد بهتری را ارائه داده است، در این پایان‌نامه نیز پس از بررسی پایدارسازهای سیستم قدرت که به طراحی پایدارسازهای فازی انجامیده است، سعی در ساخت سخت‌افزاری شده‌است که بتواند به عنوان پایدارساز فازی سخت‌افزاری با کامپیوتر ارتباط برقرار کرده و مدل سیستم‌قدرت شبیه‌سازی شده در داخل کامپیوتر را کنترل نماید.

با این مقدمه به معرفی اجمالی فصولی که در این پایان‌نامه آمده‌است می‌پردازیم.

- در فصل اول ضمن مروری بر پایداری سیستم قدرت نقش پایدارسازهای سیستم قدرت (PSS) را در این سیستم بررسی شده است.
- در فصل دوم به بررسی معادلات سیستم قدرت تک ماشین و مدلسازی اجزاء مختلف آن پرداخته شده است.
- در فصل سوم ضمن مروری بر منطق فازی، سیستم‌های فازی و کنترل‌کننده‌های فازی به نحوه طراحی پایدارسازهای فازی در Matlab پرداخته شده و در ادامه پایدارسازهای فازی طراحی شده با مطلب تشریح شده است.
- در فصل چهارم پردازشگرهای فازی که برای ایجاد پردازش فازی بکار می‌روند بررسی شده‌اند و نحوه ساخت سخت‌افزار و ایجاد ارتباط با کامپیوتر و همچنین نحوه انجام محاسبات فازی و روشی که درپیش گرفته شده، به‌طور مفصل تشریح شده است.
- در فصل پنجم به مقایسه و تحلیل نتایج بدست آمده با پایدارسازهای مختلف پرداخته شده است.
- در فصل ششم به نتیجه‌گیری و ارائه پیشنهاداتی جهت گسترش و بهبود کارایی این پروژه پرداخته شده است.

## فصل اول

**سیستم قدرت**

**و**

**پایدارسازهای سیستم قدرت**

## سیستم قدرت و پایدارسازهای سیستم قدرت

وظیفه سیستم قدرت این است که انرژی را از یکی از صورتهای طبیعی موجود به صورت انرژی الکتریکی درآورد و آنرا به نقاط مصرف منتقل نماید. از محاسن صورت الکتریکی انرژی، این است که می‌تواند براحتی منتقل شود و با درجه بالایی از قابلیت و اعتماد، نسبتاً به سادگی کنترل شود. از این رو یک سیستم قدرت با طراحی و بهره‌برداری صحیح باید نیازهای اساسی ذیل را برآورده سازد:

- ۱- سیستم باید بتواند تقاضای بار حقیقی و راکتیو مرتباً در حال تغییر را تأمین نماید.
- ۲- سیستم باید انرژی را با کمترین هزینه در تولید، انتقال و توزیع فراهم کند.
- ۳- کیفیت توان عرضه‌شده باید با توجه به عوامل زیر دارای حداقل استانداردهای لازم باشد:

الف- تثبیت فرکانس؛

ب - تثبیت زاویه؛

ج - تثبیت ولتاژ؛

د - سطح قابلیت اعتماد؛

به منظور تأمین نیازهای فوق، سطوح مختلفی از کنترل شامل مجموعه پیچیده‌ای از تجهیزات به کار گرفته می‌شود. اهداف این کنترل‌کننده‌ها بستگی به شرایط کاری و بهره‌برداری از سیستم قدرت دارد. در حالت عادی هدف این است که در عین اینکه ولتاژ و فرکانس نزدیک به مقادیر اسمی باشند، سیستم در مواجهه با خطاهای محتمل که در سیستم به وقوع می‌پیوندد به صورت پایداری عمل کرده و در سریعترین زمان ممکن به اغتشاشهای مختلف پاسخ دهد، و با ارائه چنین عملکردی حداکثر بازده سیستم را تضمین کند.

### ۱-۱ پایداری سیستم قدرت

پایداری سیستم قدرت موضوع پیچیده‌ای است که در طول سالها مورد توجه و بحث مهندسان سیستم قدرت بوده‌است. مروری بر تاریخچه موضوع به درک بهتر مسائل امروزی پایداری کمک می‌کند.

پایداری سیستم قدرت ابتدا در سال ۱۹۲۰ میلادی به عنوان یک مسأله مهم در سیستم قدرت مطرح گردید [۱]. نتایج اولین آزمایشها بر روی سیستمهای با مقیاس کوچک در سال ۱۹۲۴ ارائه شد. اولین آزمایشهای میدان<sup>۱</sup>، مربوط به مسائل پایداری یک سیستم قدرت واقعی در سال ۱۹۲۵ میلادی انجام پذیرفت. به تدریج که سیستمهای قدرت رشد کرد پیچیدگی مسائل پایداری، افزایش یافت. گامی مهم در بهبود روشهای محاسباتی، پایداری، با ساخت تحلیلگر شبکه<sup>۲</sup> در سال ۱۹۳۰ میلادی برداشته شد. آنچه از دیدگاه نظری در دهه ۱۹۲۰ و ۱۹۳۰ میلادی انجام پذیرفت، پایه لازم برای درک مسأله پایداری سیستم قدرت از طرف صنعت را بنیان گذاشت. در اوایل دهه ۱۹۵۰ میلادی، کامپیوترهای آنالوگ برای بررسی مسائلی که نیازمند مدلسازی مشروح تر ماشین سنکرون، سیستم تحریک و گاورنر سرعت بودند مورد استفاده واقع شدند. از دهه ۱۹۶۰ میلادی بیشتر تلاش و توجه صنعت به پایداری گذرای سیستم معطوف بوده‌است. سیستمهای قدرت از دیدگاه معیارهای مربوط به پایداری گذرا طراحی و بهره‌برداری می‌شوند.

---

1. Field Tests  
1. Network Analyzer

همچنان که ذکر شد پایداری سیستم قدرت در سال ۱۹۲۰ شناسایی و گزارش شد. از آن پس مهندسی که در این مورد تحقیق می‌کردند، برای راحتی تحلیل و درک طبیعت مسئله پایداری و همچنین برای پیدا کردن روشهای مناسب حل مشکل ناپایداری، موضوع پایداری را به دو گروه طبقه‌بندی کردند:

**الف- پایداری برای اغتشاش‌های کوچک (پایداری سیگنال کوچک)**، توانایی سیستم را برای حفظ حالت سنکرونیزه در اثر اغتشاشهای کوچک نشان می‌دهد. این اغتشاشها به علت تغییرات کوچک بار و تولید دائماً اتفاق می‌افتد. این گونه اغتشاشها را می‌توان به اندازه کافی کوچک فرض کرد تا اجازه خطی کردن معادلات سیستم را برای بررسی پایداری داشته باشیم. در سیستم‌های قدرت امروزی، پایداری اغتشاش کوچک، عمدتاً به علت کمبود میرایی نوسانها اتفاق می‌افتد. که این مدها می‌توانند شامل مدهای محلی<sup>۱</sup>، مدهای بین‌ناحیه‌ای<sup>۲</sup>، مدهای کنترلی<sup>۳</sup> و مدهای پیچشی<sup>۴</sup> باشند.

**ب- پایداری برای اغتشاش‌های بزرگ (پایداری گذرا)**، توانایی سیستم را به منظور حفظ حالت سنکرونیزه در اثر بروز یک اغتشاش شدید گذرا را نشان می‌دهد. در این حالت پایداری هم به نقطه کار اولیه سیستم و هم به شدت اغتشاش بستگی دارد. معمولاً در این حالت سیستم دستخوش تغییر می‌شود به گونه‌ای که نقطه کار حالت ماندگار سیستم بعد از اغتشاش با نقطه کار قبل از اغتشاش متفاوت است.

پایداری یا ناپایداری سیستم قدرت برای اغتشاشهای بزرگ به وضعیت سیستم در چند ثانیه بعد از اغتشاش برمی‌گردد. در حالی که سیستم قدرت تقریباً همیشه کار طبیعی خود را انجام می‌دهد و بطور دائم در معرض اغتشاشهای کوچک می‌باشد. لذا سیستم قدرت وقتی بطور رضایت بخش کار خواهد کرد که علاوه بر پایداری گذرا دارای پایداری سیگنال کوچک نیز باشد.

- 
1. Local Modes
  2. Interarea Modes
  3. Control Modes
  4. Torsional Modes

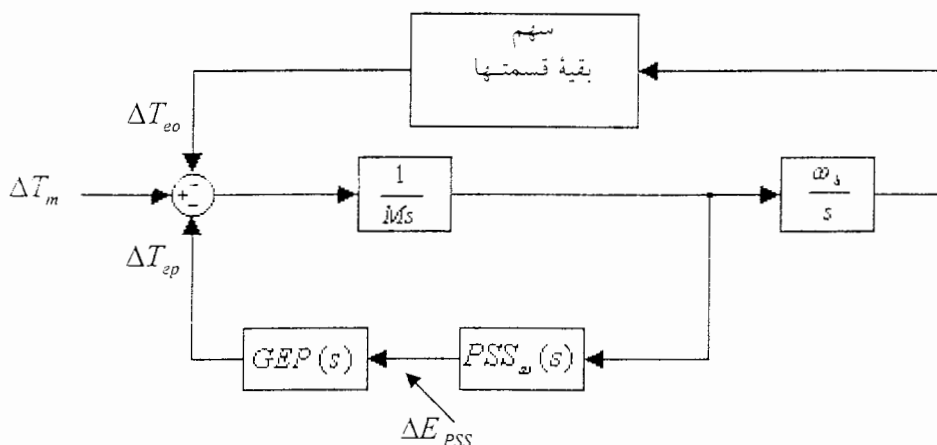
در ابتدا پایداری سیگنال کوچک را از پایداری گذرا نتیجه می‌گرفتند. این برداشت تا زمانی که ابعاد سیستم قدرت کوچک بود صحت داشت، ولی امروزه سیستم قدرت یکی از سیستم‌های بزرگ یا خیلی بزرگ بحساب می‌آید و ممکن است سیستم دارای پایداری گذرا باشد ولی دارای پایداری سیگنال کوچک نباشد. لذا نه تنها لازم است روشهایی برای بررسی پایداری سیگنال کوچک وجود داشته باشد بلکه باید اطمینان حاصل شود که سیستم قدرت دارای میرایی کافی برای کاهش دامنه نوسانات مزاحم می‌باشد. هر چند که سیستم قدرت دارای میرایی ذاتی می‌باشد و نوسانهای دینامیکی پس از گذشت مدت زمانی که به سیستم و شرایط کاری آن وابسته است، میرا می‌شوند. اما حلقه‌های کنترلی نصب‌شده روی سیستم قدرت می‌توانند با تداخل در نوسانهای فرکانس پایین این میرایی را کاهش داده و یا حتی منفی نمایند. به عنوان مثال می‌توان به حلقه تنظیم‌کننده ولتاژ و سیستم‌های تحریک‌مدرن که دارای پاسخ سریع و بهره زیاد هستند اشاره نمود [۲-۳]. امروزه میرایی کم و حتی منفی عارضه عمومی سیستم‌های قدرت جدید است.

میرایی بستگی به ساختار شبکه، شرایط کاری، بار و اثر کنترل‌کننده‌های خودکار نصب شده دارد. برای اینکه سیستم قدرت دارای پایداری نسبی مناسب برای سیگنال‌های کوچک باشد لازم است به طریقی میرایی سیستم را برای مدهایی که سهم زیادی در نوسانهای مزاحم دارند افزایش داد. طراحی و نصب سیم‌پیچ‌های میرا کننده اولین اقدام در این جهت بود، بطوری که استفاده از آن سبب افزایش میرایی نوسانهای خود ساخته‌ای شد که موقع سنکرون کردن ماشینها پیش می‌آمد، اما گسترش روز افزون سیستم قدرت و لزوم استفاده حداکثر از ظرفیتهای تولید و شبکه انتقال مسایل دینامیکی جدیدی مثل نوسانهای با فرکانس کم و نوسانات پیچشی حاصل از تشدید زیر سنکرون (SSR)، ناپایداری ولتاژ و ... را ایجاد کرد.

در راستای مقابله با نوسانات فرکانس پایین یکی از روشهای کارآمد و معمول استفاده از یک سیگنال اضافی در سیستم تحریک ژنراتور است. وظیفه تولید و اعمال این سیگنال اضافی را پایدار ساز سیستم قدرت (PSS) به عهده دارد.

## ۲-۱ پایدارساز سیستم قدرت:

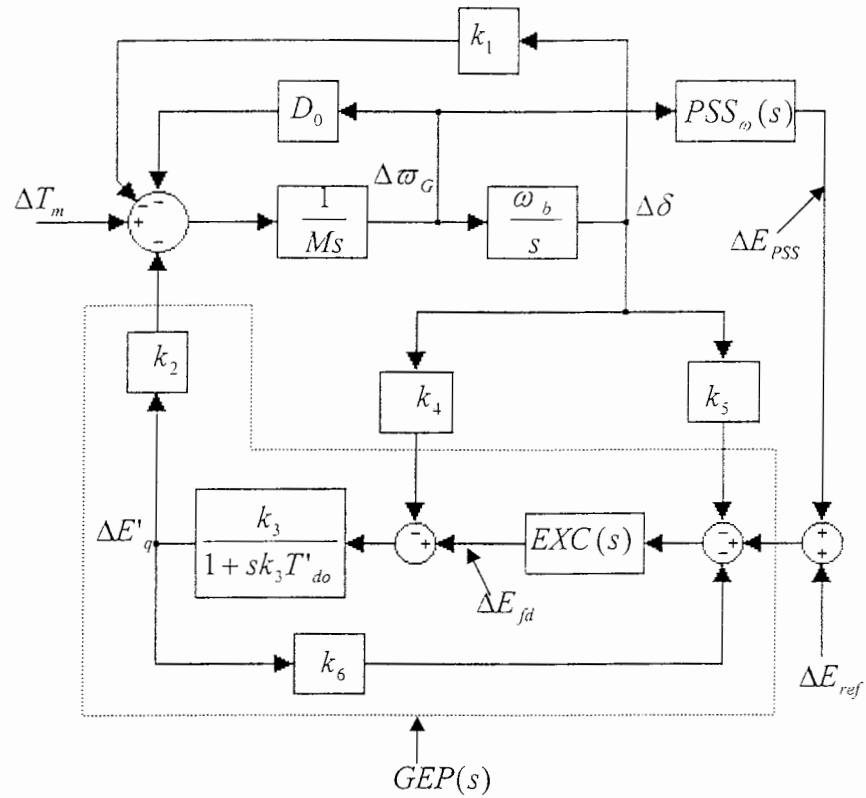
نقش اصلی پایدارساز سیستم قدرت، افزودن میرایی به نوسانهای روتور ژنراتور با استفاده از سیگنالهای پایدارساز کمکی است. این نوسانها معمولاً در محدوده فرکانسی در حدود ۰/۲ تا ۲/۵ هرتز رخ می‌دهند، و میرایی کم این نوسانات ممکن است توانایی انتقال توان را محدود کند. برای فراهم کردن میرایی لازم پایدارساز یک مولفه گشتاور الکتریکی را که با تغییرات سرعت روتور همفاز است، تولید می‌کند. جزئیات مختلف تحقق چنین مؤلفه‌ای به سیگنال ورودی بکار برده شده در پایدارساز بستگی دارد. سرعت شفت، توان شتاب‌دهنده و فرکانس پایانه مرسوم‌ترین سیگنالهای بکار گرفته شده به عنوان ورودی PSS می‌باشند. در هر حال برای هر سیگنال ورودی تابع انتقال پایدارساز باید بهره و مشخصات فاز سیستم تحریک و سیستم قدرت را جبران کند، که مجموعاً تابع انتقالی از خروجی پایدارساز به آن جزء از گشتاور



شکل ۱-۱ پایدارساز با ورودی سرعت - بلوک دیاگرام سیستم

الکتریکی که می‌تواند بوسیله کنترل تحریک مدوله‌شود، را شامل می‌شود. این تابع انتقال در این نوشتار با  $GEP(s)$  نمایش می‌دهیم، که قویاً از بهره تنظیم کننده ولتاژ، سطح قدرت ژنراتور و استحکام سیستم  $ac$  تأثیر می‌پذیرد [۴].





شکل ۱-۲ مدل ساده‌شده سیستم تک ماشینه متصل به شین بینهایت

### ۱-۳ انواع پایداری‌سازهای سیستم قدرت

پایداری‌سازهای سیستم قدرت برحسب سیگنال ورودی بکار برده شده مشخصه‌ها و عملکردهای تقریباً متفاوتی از خود بروز می‌دهند، متداولترین سیگنالهایی که از آنها به عنوان ورودی PSS استفاده می‌شود: سرعت شفت، توان شتاب‌دهنده و فرکانس ترمینال می‌باشد.

## ۱-۳-۱ ورودی سرعت

پایدارساز سیستم قدرت از سرعت شفت به عنوان یک ورودی که با استفاده از آن باید مؤلفه‌های از گشتاور الکتریکی را که با تغییرات سرعت همفاز است، تولید کند تا تأخیرهای  $GEP(s)$  را جبران کند، استفاده می‌کند. بنابراین مشخصه یک پایدارساز ایده‌آل باید با  $GEP(s)$  تناسب معکوس داشته‌باشد، یعنی

$$Ideal PSS_{\omega}(s) = D_{PSS} / GEP(s) \quad ۳-۱$$

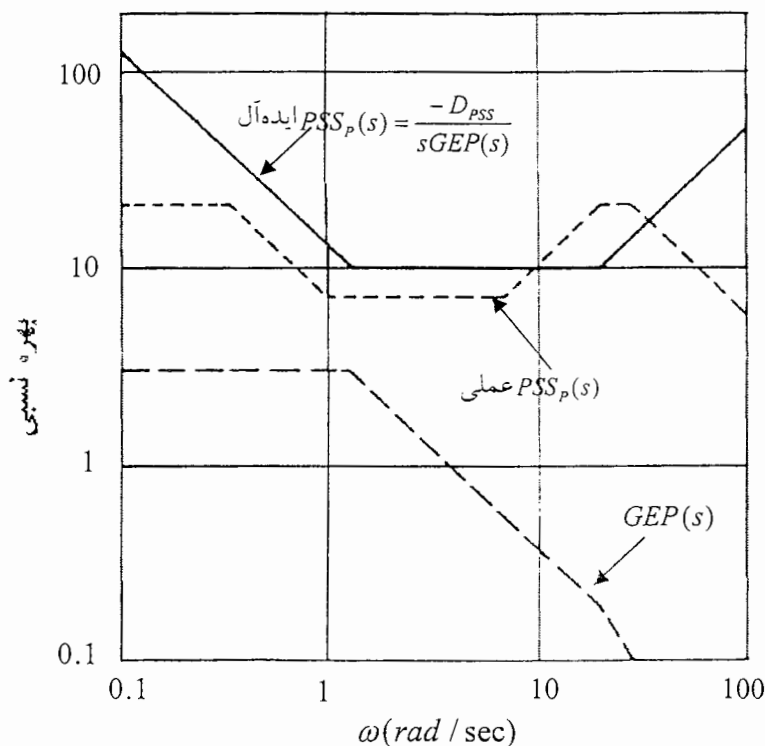
که در آن  $D_{PSS}$  سهم پایدارساز را از میرایی لازم بیان می‌دارد، چنین مشخصه پایدارسازی با وجود بهره بالای اضافی، بدلیل اینکه هم ژنراتور و هم تحریک کننده (بسته به نوع آن) مشخصه‌های بهره و فازی وابسته به فرکانس از خود نشان می‌دهند، غیر عملی است. یک پایدارساز عملی سرعت در محدوده فرکانسی موردنظر باید از بخشهای پسفاز، پیشفاز برای جبران پسفازی در  $GEP(s)$  استفاده کند. بهره در فرکانسهای بالا برای محدود کردن تأثیر نویز و مینیمم کردن تداخل پیچشی باید کم شود، و بدنبال آن فیلترهای پایین‌گذر و احتمالاً حذف باند<sup>۱</sup> نیز لازم می‌شوند. یک بخش پاکساز نیز برای جلوگیری از افستهای ولتاژ حالت ماندگار سیستم بکار می‌رود.

$$عملی PSS_{\omega}(s) = k_s \frac{T_w s (1+T_1 s)(1+T_3 s)}{(1+T_w s)(1+T_2 s)(1+T_4 s)} FILT(s) \quad ۴-۱$$

## ۲-۳-۱ ورودی توان

استفاده از توان شتاب‌دهنده به عنوان سیگنال ورودی پایدارساز سیستم قدرت بیشتر به علت تداخل پیچشی سطح پایین ذاتی آن بوده است [۶]. رایج‌ترین روش تحلیل پایدارساز ورودی توان که از آن استفاده می‌شود، بکاربردن ورودی توان به عنوان مشتق سرعت است [۷]. با این شیوه مفاهیم مشابه

استفاده شده برای تحلیل پایداری ورودی سرعت را برای تحلیل این گونه از پایداریها بکار می‌برند، نتیجه‌ای که چندان هم دور از نظر نیست با این روش بدست می‌آید، عملکرد مشخصه‌های پایداری سیستم قدرت عیناً همانزایی که پایداری سرعت بود، می‌شوند.



شکل ۳-۱ پاسخ فرکانسی پایداریهای ورودی سرعت و توان، ایده‌آل و عملی

### ۳-۳-۱ ورودی فرکانس

استفاده از فرکانس شین ac به عنوان ورودی پایداری، گاهی نتایج متفاوت از نتایجی که به پایداریهای ورودی سرعت مربوط می‌شد می‌دهد. اولین اختلاف آن است که حساسیت سیگنال فرکانس به نوسانات روتور با ضعیف‌تر شدن سیستم انتقال خارجی افزایش می‌یابد و موجب جبران کاهش بهره از خروجی پایداری به گشتاور الکتریکی،  $GEP(s)$ ، ناشی از یک سیستم انتقال ضعیفتر می‌شود. به عبارت دیگر سیگنال فرکانس به مدهای نوسان بین نواحی بزرگ نسبت به مدهای که فقط واحدهای منفرد را در

بر می‌گیرد و نیز مدهای بین واحدهای یک نیروگاه حساستر است. بنابراین به نظر می‌رسد می‌توان سهم میرایی بیشتری برای مدهای بین ناحیه‌ای نسبت به سهم حاصل از سیگنال ورودی سرعت، به دست آورد.

باید توجه داشت که پایدارساز مبتنی بر فرکانس، علی‌رغم مزایای مذکور، دارای چندین عیب است:

- ۱- در حین حالت گذرای سریع، سیگنال فرکانس پایانه، دچار تغییر فاز ناگهانی خواهد شد. که به جهشی در ولتاژ تحریک منجر می‌شود و در کمیت‌های خروجی ژنراتور منعکس می‌شود.
- ۲- معمولاً سیگنال فرکانس دارای نویز سیستم قدرت است که به وسیله بارهای صنعتی بزرگ از قبیل: کوره‌های قوسی به وجود آمده‌است. در بسیاری از موارد این مسأله، مانع استفاده از فرکانس به عنوان سیگنال ورودی می‌شود.
- ۳- فیلتر کردن پیچشی لازم است. از این‌رو، یک پایدارساز مبتنی بر فرکانس محدودیت‌های اساسی پایدارساز ورودی توان را دارد.

### ۴-۳-۱ پایدارسازهای دیجیتالی

امروزه شکلهای دیجیتالی برخی از پایدارسازهای بالا ساخته شده و به صورت تجاری در دسترس قرار گرفته است [۶].

### ۴-۱ ملاحظات بر اهداف اجرایی PSS

استفاده از پایدارساز می‌تواند محدودیت‌های پایداری انتقال توان را برای شرایطی که نوسانها بطور ضعیفی میرا شده یا نوسانها در محدوده فرکانسی ۰/۲ تا ۲/۵ هرتز بی‌اختیار بزرگ می‌شوند، را کاهش دهد. این کار با کنترل تحریک در جهت کمک به میرا کردن مدهای نوسان سیستم انجام می‌شود. و در نهایت این توانایی پایدارساز است که تحت حداقل شرایط پایداری میرایی نوسانها را افزایش دهد. بدین معنی که "شرایط عملکرد" برای پایدارساز مهم است [۵-۶]. در شرایط انتقال ضعیف و اعمال بار سنگین

به سیستم انتقال یک میرایی اضافی ابتدایی، لازم است. برای مثال وقتی می‌خواهیم توان را بر روی خطوط انتقال طولانی، از محل‌های تولید شده دور یا بر روی گره‌های بین سیستم ضعیف انتقال دهیم. بنابراین سیستم‌هایی که معمولاً میرایی مناسب دارند، اغلب می‌توانند از پایدارسازها در طول شرایط غیر نرمال فایده ببرند. احتمالاتی از قبیل قطع برق خطوط یا اتصال کوتاه منبع و اغلب بطور ناگهانی اتفاق می‌افتند.

در پیاده‌سازی پایدارساز این مهم است که پایدارساز بتواند میرایی را برای تحریک‌های کوچک در نزدیکی یک نقطه حالت ماندگار تأمین کند، و مشخصات پایداری گذرا را تغییر ندهد، یعنی توانایی بازیابی از یک اغتشاش سخت را داشته باشد. در حقیقت، این پایداری اغلب تأثیر زین‌آوری روی پایداری گذرا علی‌الخصوص در ابتدای پاسخ به خطا با کشیدن میدان ژنراتور به خارج از حداکثر مقدار خود، خواهد داشت. خروجی پایدارساز معمولاً برای جلوگیری از تأثیر جدی بر روی پایداری گذرا محدود می‌شود. اما تنظیم پایدارساز تأثیر مهمی بر روی عملکرد سیستم زیر اغتشاش بزرگ دارد.

تنظیم تکمیلی کنترل‌های تحریک برای پایدارسازی مدهای نوسان سیستم در بیشتر موضوعات تحقیقی سال‌های اخیر آورده شده است. دو تکنیک اساسی که در طراحی و کاربرد پایدارسازهای سیستم قدرت به‌طور موفقیت آمیزی استفاده شده‌اند: جبران‌سازی فاز و مکان‌هندسی ریشه‌ها می‌باشند. جبران‌سازی فاز شامل تنظیم پایدارساز جهت جبران تأخیر فازی ایجاد شده توسط ژنراتور، سیستم تحریک و سیستم قدرت است، بطوری که مسیر پایدارساز تغییرات گشتاور هم‌فاز با تغییرات سرعت فراهم می‌کند. این سراسرترین روشی است که هم به آسانی فهمیده شده و هم براحتی در میدان پیاده‌سازی می‌شود و در عمل نیز بطور گسترده‌ای از این روش استفاده می‌شود. تجزیه و تحلیل با مکان‌هندسی ریشه‌ها در واقع جابجایی مقادیر ویژه مربوط به مدهای نوسانی سیستم قدرت با تنظیم محل قطب و صفر پایدارساز در صفحه  $s$  می‌باشد. این روش بر خلاف طبیعت حلقه‌باز تکنیک جبران‌سازی فاز، یک بینش اضافی در مورد عملکرد سیستم بدلیل کارکردن مستقیم با مشخصه‌های سیستم حلقه بسته بدست می‌دهد، اما برای بکاربردن بویژه در میدان بسیار پیچیده است.

معمولاً برای طراحی PSS از مدل تک ماشین متصل به باس بی‌نهایت استفاده می‌شود. بدین ترتیب که ابتدا فرکانس طبیعی مدل محاسبه می‌شود و سپس پارامترهای کنترل کننده پیش فاز چنان انتخاب می‌شود که اختلاف فاز لازم را در سیگنال تغییرهای سرعت زاویه روتور به منظور افزایش گشتاور دمپینگ ایجاد و به سیستم تحریک اعمال نماید.

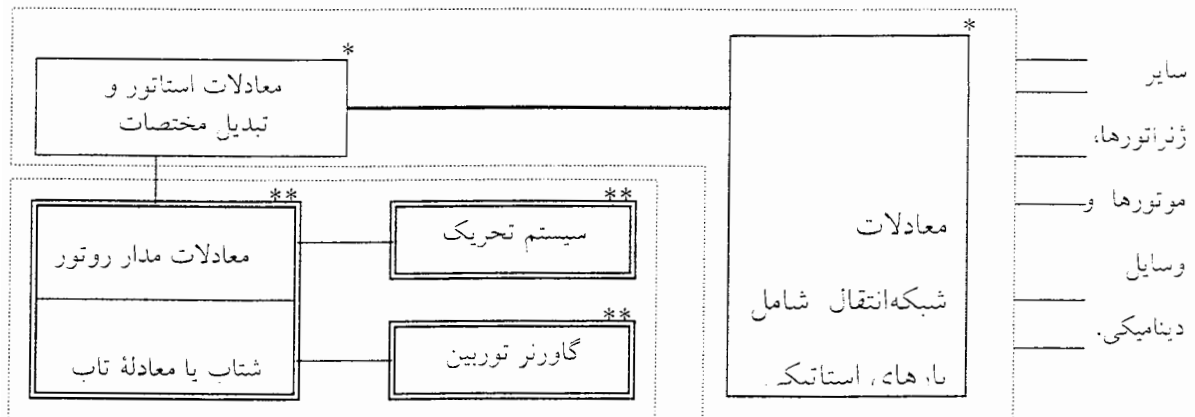
هر چند که PSS های طراحی شده با روش ذکر شده بطور رضایت بخشی عمل می‌کنند ولی بعضی مواقع در یک سیستم غیرمغناطیسی اینک اکثر ماشینها به PSS مجهز هستند باز هم نوسانهای فرکانس پایین مشاهده می‌شود. علت بروز این نوسانات عدم هماهنگی بین PSS های نصب شده است، این ناهماهنگی به خاطر نادیده گرفتن دینامیک بقیه سیستم است که به صورت باس بی‌نهایت مدل می‌شود. PSS که برای ماشین متصل به باس بی‌نهایت طراحی شده عملاً در یک سیستم چند ماشینی کار می‌کند، از آنجایی که پهنای باند PSS های مذکور خیلی کم است و هر کدام فرکانس طبیعی یک ماشین را میرا می‌کنند، لذا از عهده میرا کردن نوسانی با فرکانس دیگر که از طریق خط انتقال بر سیستم تحمیل می‌شود بر نمی‌آیند. این مسئله به‌تداخل بین PSS ها معروف است.

## فصل دوم

# مدلسازی سیستم قدرت

## مدلسازی سیستم قدرت

سیستم قدرت یک سیستم بسیار غیرخطی است که عملکرد دینامیکی آن تحت تأثیر مجموعه وسیعی از تجهیزات تشکیل دهنده آن قرار دارد که هر یک، عکس‌العمل زمانی و مشخصه متفاوتی دارند. می‌توان



\* معادلات جبری

\*\* معادلات دیفرانسیل

شکل ۱۲-۱۸ ساختار مدل کامل سیستم قدرت



گفت تحلیل سیستمهای قدرت عملی شامل حل همزمان معادلات مربوط به موارد ذیل است:

ماشینهای سنکرون، و سیستمهای تحریک مربوط و محرکه‌ها(توربینها)؛

شبکه به هم پیوسته انتقال؛

بارهای استاتیکی و دینامیکی(موتور)؛

سایر وسایل مانند کنورترهای HVDC، جبران‌سازهای استاتیکی توان راکتیو.

شکل ۱-۲ ساختار کلی مدل کامل سیستم را نشان می‌دهد.

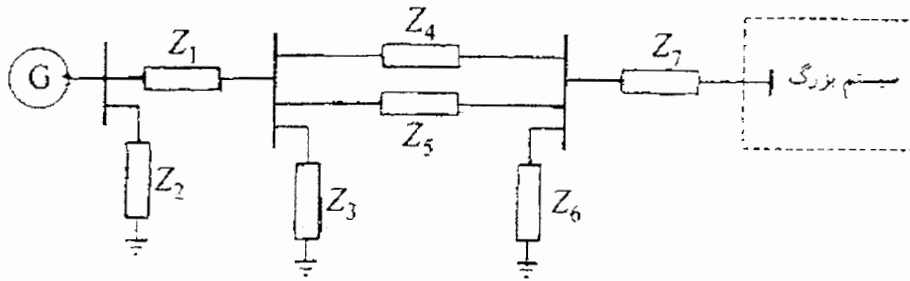
تقریباً مشخصه هر یک از اجزای اصلی سیستم قدرت بر پایداری آن تأثیر می‌گذارد. اطلاع کافی از این مشخصه‌ها برای درک مطالعه پایداری سیستم قدرت ضروری است. بنابراین باید مدل استفاده شده برای هر عنصر برای تحلیل پایداری، مناسب باشد.

ما برای مدل‌سازی سیستم قدرت و اجزا مختلف آن از قسمت Simulink نرم‌افزار Matlab استفاده نموده‌ایم که امکانات جامعی را برای شبیه‌سازی رفتار سیستم‌های مختلف در اختیار کاربران قرار می‌دهد.

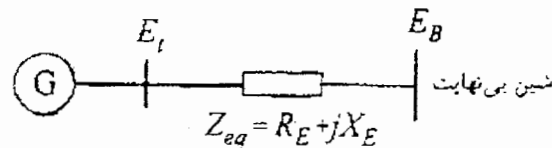
## ۱-۲ مدل‌سازی سیستم قدرت

ما در این قسمت به شرح چگونگی مدل‌سازی یک سیستم تک ماشین که از طریق خطوط انتقال به سیستم بزرگ متصل شده، می‌پردازیم. ترکیب کلی چنین سیستمی در شکل ۲-۲ الف نشان داده شده است. به منظور تحلیل این سیستم را می‌توان با بکارگیری معادل تونن شبکه انتقال خارج ماشین و شبکه مجاور انتقال به صورت شکل ۲-۲ ب تقلیل داد. به علت اندازه نسبی سیستمی که توان آن را ماشین فراهم می‌کند، دینامیک ماشین تقریباً هیچ تغییری در ولتاژ و فرکانس ولتاژ تونن  $E_B$  ایجاد نخواهد کرد. به چنین منبع ولتاژی در اصطلاح شین بی‌نهایت گفته می‌شود.

برای هر وضعیت داده شده سیستم، هنگامی که ماشین دستخوش اغتشاش می‌شود اندازه ولتاژ شین بی‌نهایت  $E_B$  ثابت باقی می‌ماند. ولی، با تغییر وضعیت حالت- ماندگار سیستم ممکن است که اندازه  $E_B$  تغییر پیدا کند. این موضوع نشان دهنده تغییری در نقطه کار شبکه خارجی است.



(الف) ترکیب کلی



(ب) سیستم معادل

شکل ۲-۲ سیستم تک ماشینه متصل به شین بی‌نهایت از طریق خطوط انتقال

## ۲-۲ معادلات ماشین سنکرون

در اینجا مدل فضای حالت سیستم را نخست با کاهش معادلات ماشین سنکرون به شکل مناسب و سپس ترکیب آنها با معادلات شبکه به دست خواهیم آورد. زمان را برحسب ثانیه، زوایا را برحسب رادیان الکتریکی، و سایر متغیرها را در مبنای واحد بیان خواهیم کرد.

معادلات شتاب عبارتند از:

$$\Delta \dot{\omega}_r = \frac{1}{2H} (T_m - T_e - k_D \Delta \omega_r) \quad ۱-۲$$

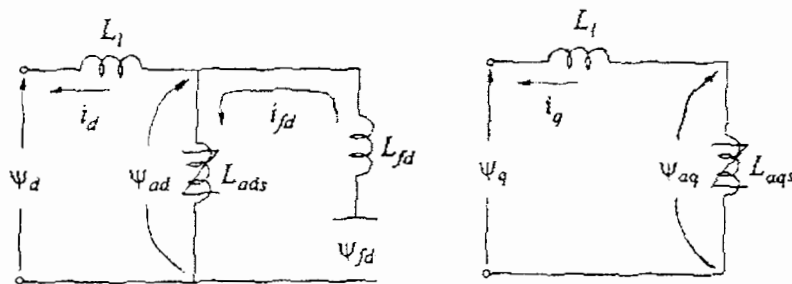
$$\dot{\delta} = \omega_0 \Delta \omega_r$$

که در آن  $\omega_0 = 2\pi f_0$  رادیان الکتریکی بر ثانیه است، و معادله دینامیک مدار تحریک عبارت است از:

$$\psi_{fd} = \omega_0 (e_{fd} - R_{fd} i_{fd})$$

$$= \frac{\omega_0 R_{fd}}{L_{adu}} E_{fd} - \omega_0 R_{fd} i_{fd} \quad ۲-۲$$

که در آن  $E_{fd}$  ولتاژ خروجی تحریک است. معادلات بالا دینامیک ماشین سکرون را با  $\delta, \Delta \omega_r$  و  $\psi_{fd}$  به عنوان متغیرهای حالت، توصیف میکنند. ولی در معادله دینامیک مدار تحریک مشتقات متغیرهای حالت به صورت توابعی از  $T_e$  و  $i_{fd}$  ظاهر شده است، به منظور بدست آوردن معادلات کامل سیستم به صورت فضای حالت لازم است  $T_e$  و  $i_{fd}$  را برحسب متغیرهای حالت تعیین شده توسط معادلات شار دور ماشین و معادلات شبکه بیان کنیم.



شکل ۲-۲

با نادیده گرفتن سیم‌پیچ‌های میرا کننده، مدارهای معادل ارتباطی بین شار دور ماشین و جریانها در شکل ۲-۲ نشان داده شده‌است.

$$\psi_d = -L_l i_d + \psi_{ad} \quad ۳-۲$$

$$\psi_q = -L_l i_q + \psi_{aq} \quad ۴-۲$$

$$\psi_{fd} = \psi_{ad} + L_{fd} i_{fd} \quad ۵-۲$$

در معادلات بالا  $\psi_{aq}$  و  $\psi_{ad}$  شار دور (متقابل) فاصله هوایی بوده و  $L_{ads}$  و  $L_{aqs}$  مقادیر اشباع‌شده اندوکتانس‌های متقابل است.

جریان تحریک را میتوان به صورت زیر بیان کرد:

$$i_{fd} = \frac{\psi_{fd} - \psi_{ad}}{L_{fd}} \quad ۶-۲$$

شار دورهای متقابل محورهای  $d$  و  $q$  را نیز میتوان به صورت زیر نوشت:

$$\psi_{ad} = L'_{ads} \left( -i_d + \frac{\psi_{fd}}{L_{fd}} \right) \quad ۷-۲$$

$$\psi_{aq} = -L_{aqs} i_q$$

که در آن :

$$L'_{ads} = \frac{1}{\frac{1}{L_{ads}} + \frac{1}{L_{fd}}}$$

گشتاور فاصله هوایی برابر است با:

$$T_e = \psi_{ad} i_d - \psi_{aq} i_q \quad ۸-۲$$

با کمی ساده سازی معادلات ولتاژ استاتور عبارتند از:

$$\begin{aligned} e_d &= -R_a i_d + (L_l i_q - \psi_{aq}) \\ e_q &= -R_a i_q - (L_l i_d - \psi_{ad}) \end{aligned} \quad 9-2$$

معادله شبکه برای سیستم شکل ۲-۲ ب عبارت است از:

$$\begin{aligned} \tilde{E}_t &= \tilde{E}_B + (R_E + jX_E) \tilde{I}_t \\ (e_d + j e_q) &= (E_{Bd} + j E_{Bq}) + (R_E + jX_E)(i_d + j i_q) \end{aligned} \quad 10-2$$

که از تجزیه مؤلفه‌های d و q خواهیم داشت:

$$\begin{aligned} e_d &= R_E i_d - X_E i_q + E_{Bd} \\ e_q &= R_E i_q + X_E i_d + E_{Bq} \end{aligned} \quad 11-2$$

که در آن

$$\begin{aligned} E_{Bd} &= E_B \sin \delta \\ E_{Bq} &= E_B \cos \delta \end{aligned} \quad 12-2$$

با استفاده از معادلات ۲-۹ برای حذف  $e_d, e_q$  در معادله‌های ۲-۱۱ و بکارگیری عبارت داده شده با معادلات ۲-۷ برای  $\psi_{aq}, \psi_{ad}$  عبارات زیر را برای  $i_d, i_q$  بر حسب متغیرهای حالت  $\delta, \psi_{fd}$  به دست می‌آوریم:

$$\begin{aligned} i_d &= \frac{X_{Tq} \left[ \psi_{fd} \left( \frac{L_{ads}}{L_{ads} + L_{fd}} \right) - E_B \cos \delta \right] - R_T E_B \sin \delta}{D} \\ i_q &= \frac{R_T \left[ \psi_{fd} \left( \frac{L_{ads}}{L_{ads} + L_{fd}} \right) - E_B \cos \delta \right] + X_{Td} E_B \sin \delta}{D} \end{aligned} \quad 13-2$$

که در آن

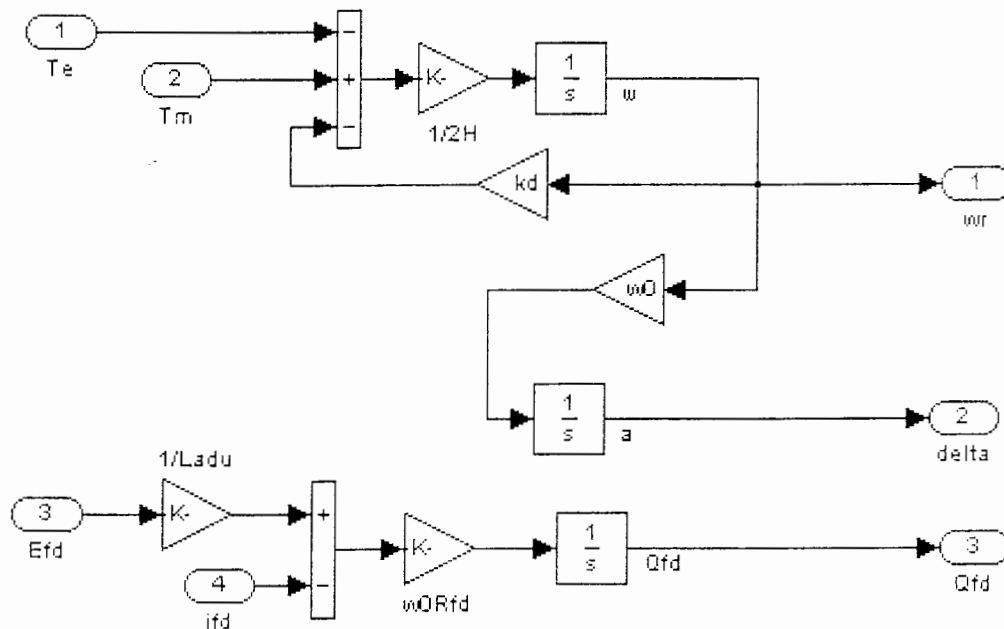
$$R_T = R_a + R_E$$

$$X_{Tq} = X_E + X_{qs}$$

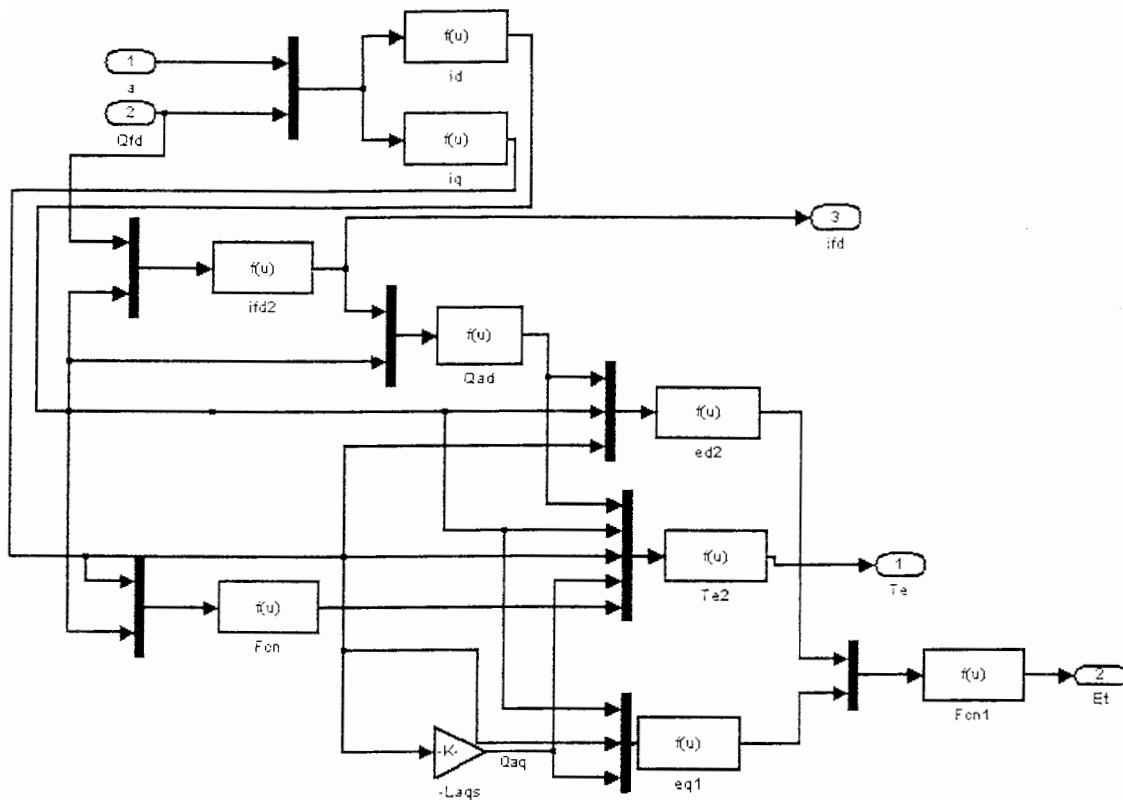
$$X_{Td} = X_E + X'_{qs}$$

$$D = R_T^2 + X_{Tq} X_{Td}$$

راکتانس‌های  $X'_{qs}$  و  $X_{ds}$  مقادیر اشباع شده هستند که در مبنای واحد برابر با اندوکتانسهای متناظر میباشند. با استفاده از معادلاتی که ذکر کردیم میتوان برای حذف  $i_{fd}$  و  $T_e$  از معادله حالت دینامیکی مدار روتور و بیان آنها بر حسب متغیرهای مورد نظر استفاده کرد. شکل‌های ۲-۳ و ۲-۴ بترتیب نمودار بلوکی معادلات حالت سیستم و معادلات جبری لازم را که از روی معادلات بالا بدست آمده‌اند، نمایش می‌دهند.



شکل ۲-۳ بلوک دیاگرام معادلات حالت سیستم

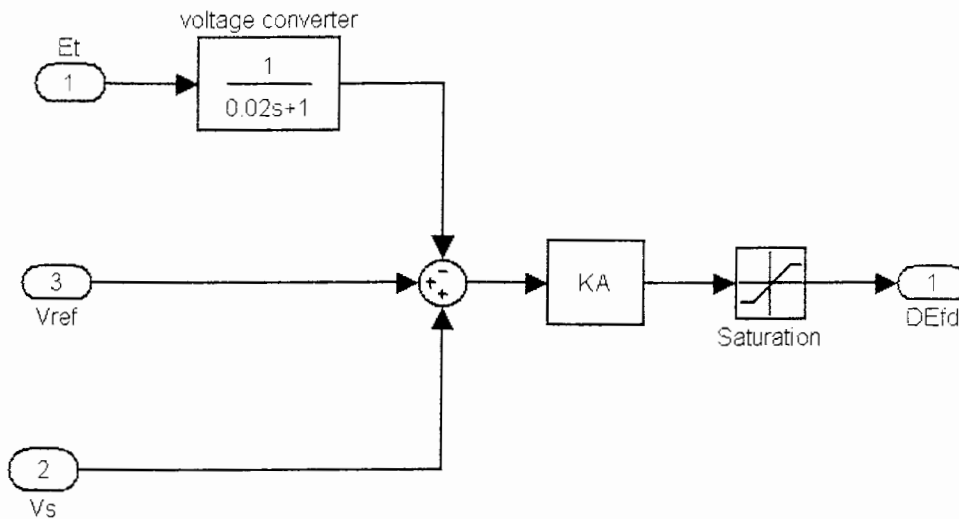


شکل ۲-۴ بلوک دیاگرام معادلات جبری لازم برای تکمیل کردن معادلات حالت سیستم

### ۳-۲ نمایش سیستم تحریک

وظیفه اصلی سیستم تحریک آن است که به منظور ثابت نگه داشتن ولتاژ پایانه ماشین هنگام تغییر خروجی، جریان تحریک ماشین سنکرون را در محدوده ظرفیت پیوسته ژنراتور، تأمین و به طور خودکار تنظیم نماید. علاوه بر این باید سیستم تحریک قادر باشد با تنظیم تحریک منطبق با توانایی‌های لحظه‌ای و کوتاه مدت ژنراتور، در مقابل اغتشاشهای گذرا عکس‌العمل نشان دهد. سیستم‌های تحریک انواع مختلفی دارند که پرداختن به مشخصات و عملکرد آنها در این مجال نمی‌گنجد و علاقمندان می‌توانند به مرجع [۲۶] مراجعه فرمایند. ما برای بررسی‌های خویش مدل یک سیستم تحریک ترستوری تغذیه از

شین همراه با یک تنظیم کننده خودکار ولتاژ (AVR) را در نظر گرفتیم که مدل بلوک دیاگرام آن در شکل ۲-۵ آورده شده است.



شکل ۲-۵ مدل سیستم تحریک همراه با AVR

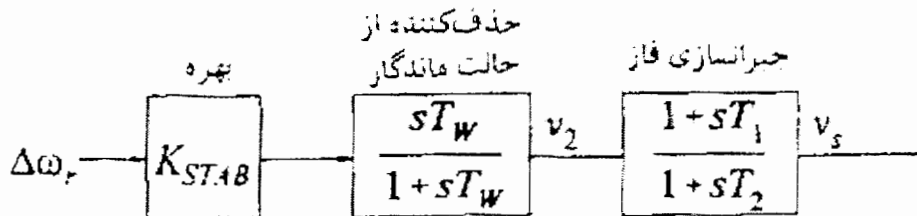
#### ۲-۴ نمایش PSS کلاسیک

نقش اصلی یک پایدارساز سیستم قدرت، افزودن میرایی به نوسانهای روتور ژنراتور به وسیله کنترل تحریک آن با استفاده از سیگنال‌های پایدارساز کمکی است. برای فراهم آوردن میرایی، باید پایدارساز یک مؤلفه گشتاور الکتریکی را هم‌فاز با انحرافهای سرعت روتور ایجاد کند. یکی از سیگنال‌های مناسب برای کنترل تحریک ژنراتور، انحراف سرعت می‌باشد. مدل PSS در شکل ۲-۶ شامل سه بلوک است: بلوک جبرانسازی فاز، بلوک حذف کننده اثر سیگنال حالت ماندگار و بلوک بهره. بلوک جبرانسازی فاز، پیشفازی مناسب را برای جبرانسازی پس‌فاز بین ورودی تحریک کننده و گشتاور الکتریکی ژنراتور را فراهم می‌آورد. شکل بلوکی از مرتبه اول را نشان می‌دهد ولی در عمل ممکن است از دو یا چند بلوک مرتبه اول برای



دستیابی به جبران‌سازی مطلوب فاز استفاده شود. در برخی از نمونه‌ها از بلوک‌های مرتبه دوم با ریشه‌های مختلط نیز استفاده شده است.

همانطور که در فصل اول نیز ذکر شد معمولاً گستره فرکانسی مورد علاقه ۰/۱ تا ۰/۲ هرتز می‌باشد، و مدار پیش‌فاز باید جبران‌سازی را بر روی تمامی این گستره فرکانسی فراهم کند. مشخصه فازی که باید جبران گردد، با تغییر وضعیت سیستم تغییر می‌کند، از این‌رو، مصالحه‌ای صورت می‌گیرد و مشخصه قابل قبولی برای وضعیت‌های مختلف سیستم انتخاب می‌شود. معمولاً مقدار زیر جبران‌سازی نیز لازم است تا PSS، علاوه بر افزایش چشمگیری در گشتاور میراکننده، به افزایش اندکی در گشتاور سنکرون کننده منجر گردد.



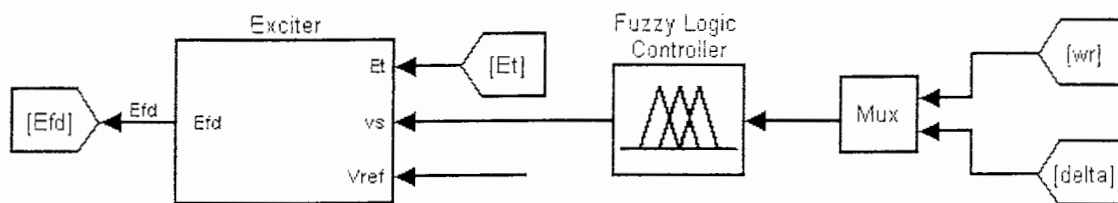
شکل ۲-۶ پایدارساز کلاسیک سیستم قدرت

بلوک حذف کننده اثر حالت ماندگار به صورت فیلتری بالا گذر، با ثابت زمانی  $T_w$  به اندازه کافی بزرگ عمل می‌کند و اجازه می‌دهد تا سیگنال‌های متناظر با نوسانهای  $\omega_r$  بدون تغییر عبور کنند، بدون آن تغییرات ماندگار در سرعت، ولتاژ پایانه را تغییر می‌دهد. این بلوک اجازه می‌دهد تا PSS فقط به تغییرات سرعت پاسخ دهد. از نظر تابع این بلوک، مقدار  $T_w$  حساس نبوده و می‌تواند در گستره ۱ تا ۲۰ ثانیه باشد. نکته اصلی این است که باید مقدار  $T_w$  به اندازه کافی بزرگ باشد تا سیگنال‌های پایدارساز را در فرکانسهای مورد علاقه بدون تغییر عبور دهد، ولی آنقدر بزرگ نباشد که به هنگام پارگی سیستم، به انحراف نامطلوب ژنراتور منجر شود.

بهره پایدارساز  $K_{SRTAB}$  مقدار میرایی ایجاد شده از PSS را تعیین می‌کند و بطور ایده‌آل، باید بهره در مقداری متناظر با حداکثر میرایی تنظیم گردد ولی معمولاً مقدار آن بوسیله ملاحظات دیگری محدود می‌شود.

## ۲-۵ نمایش PSS فاززی

این پایدارساز پس از مشاهدات و بررسی‌هایی که با استفاده از پایدارساز کلاسیک انجام می‌گیرد، که در نهایت منجر به طراحی پایدارساز فاززی می‌شود به جای پایدارساز کلاسیک می‌نشیند (شکل ۲-۷ را ببینید)، برخلاف پایدارسازهای کلاسیک که تنها از یک ورودی بهره می‌گیرند، پایدارسازهای فاززی می‌توانند سیگنال کنترل خویش را با توجه به دو ورودی و حتی بیشتر تنظیم کنند. پایدارسازهای فاززی نیز بر اساس سیستم استنتاجی که از آن استفاده می‌کنند انواع مختلفی دارند، از رایج‌ترین سیستم‌های فاززی می‌توان به سیستم ممدانی و سیستم تاکاگی - سوگنؤ (TSK) اشاره کرد. در فصل ۳ به تفصیل در مورد سیستم‌های فاززی و نحوه طراحی آنها اشاره خواهیم کرد.



شکل ۲-۷ PSS فاززی و سیستم تحریک

## ۶-۲ نمایش PSS فازی سخت افزاری

در مدل‌سازی سیستم این بلوک نیز دقیقاً به جای پایدارساز کلاسیک می‌نشیند و همانند پایدارساز فازی از دو ورودی فرکانس و زاویه استفاده می‌کند، در واقع این بلوک یک Sfunction است که ارتباط بین کامپیوتر و سخت افزار را فراهم می‌سازد. جزئیات تحقق این بلوک در فصل ۴ آورده شده است.

فصل سوم

**منطق فازی**

**و**

**کنترل فازی**

## منطق فازی و کنترل فازی

منطق فازی تکنولوژی جدیدی است که شیوه‌های مرسوم برای طراحی و مدل سازی یک سیستم را که نیازمند ریاضیات پیشرفته و نسبتاً پیچیده است با استفاده از مقادیر و شرایط زبانی و یا به عبارتی دانش فرد خبره و با هدف ساده، دقیق و کارآمدتر شدن طراحی سیستم تا اندازه زیادی تعدیل و تکمیل می‌نماید.

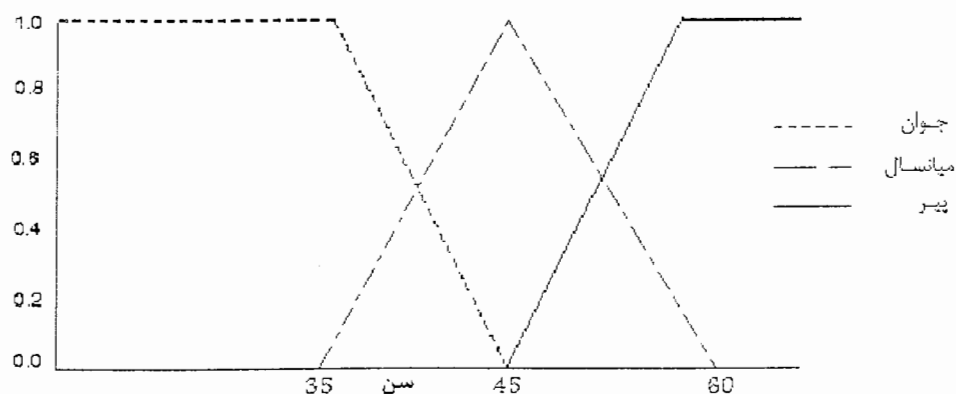
علیرغم اینکه منطق فازی بر پایه ریاضیات پیشرفته و پیچیده قرار دارد اما یادگیری آن بسیار آسان است. از نظر تئوری هر سیستمی که توسط منطق فازی طراحی شده باشد توسط سایر تکنیکهای طراحی مرسوم نیز قابل پیاده‌سازی است و ممکن است نسبت به شیوه منطق فازی پیچیده و مشکل‌تر باشد.

در واقع مدل‌های فازی مکمل مدل‌های کنترل خطی می‌باشند و بنابراین هر مدلی که با استفاده از شیوه های طراحی مرسوم طراحی شده باشد با مدل فازی هم قابل پیاده‌سازی است اما در تعدادی از موارد مدل‌های قدیمی ساده‌تر، سریعتر و کاراتر از مدل‌های مبتنی بر منطق فازی می‌باشند.

## ۳-۱ منطق فازی

منطق فازی با تعریف مفهوم مجموعه‌های فازی آغاز می‌شود. یک مجموعه فازی مجموعه‌ای است بدون مرز و قید مشخص که عناصر آن می‌توانند حتی با داشتن یک مقدار عضویت جزئی نیز به آن مجموعه تعلق داشته باشند.

در منطق کلاسیک، یک عنصر یا متعلق به یک مجموعه هست و یا نیست. به عبارت دیگر درجه عضویت هر عنصر در یک مجموعه مفروض صفر یا یک است، اما در منطق فازی درجه عضویت هر عنصر به تابع تعلق آن بستگی دارد، یک تابع تعلق منحنی است که یک نقطه در فضای ورودی را به یک مقدار تعلق (درجه عضویت) بین ۰ و ۱ نگاشت می‌دهد. بنابر این می‌توان گفت که درجه عضویت هر عنصر عددی بین صفر و یک است. یکی از مثال‌های مشهور برای بیان این مفهوم، دسته بندی سن افراد به سه مجموعه فازی جوان، میانسال و پیر است. این مجموعه‌های فازی توسط توابع تعلقشان مشخص می‌شوند



شکل ۳-۱ توابع تعلق مجموعه‌های فازی جوان، میانسال و پیر

تابع تعلق یک مجموعه فازی، درجه عضویت یک عنصر در آن مجموعه را مشخص می‌کند. مثلاً با توجه به شکل فوق فردی با سن ۳۸ سال دارای درجه عضویت ۰/۲ در مجموعه جوان و ۰/۸ در مجموعه میانسال می‌باشد، یعنی این فرد ۲۰٪ جوان و ۸۰٪ میانسال است.

بیشترین اهمیت در تحقق منطق فازی نحوه استدلال در منطق فازی است که برتری آن را نسبت به منطق بولی می‌رساند. به عبارت دیگر اگر ما مقادیر فازی را در مقدار ۱ (کاملاً درست) و ۰ (کاملاً غلط) فرض کنیم، عملگرهای منطقی استاندارد بدست خواهند آمد. به عنوان مثال جدول درستی شکل ۲-۳ را برای سه عملگر استاندارد AND, OR و NOT بررسی می‌کنیم.

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

**AND**

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

**OR**

A	not A
0	1
1	0

**NOT**

شکل ۲-۳ عملگرهای AND, OR در منطق بولی

حالا سعی می‌کنیم بطور خلاصه توضیح دهیم که منطق فازی چگونه این عبارات را که با درجه عضویت بیان می‌شوند تفسیر می‌کند، و این جدول درستی چگونه باید اصلاح پیدا کند؟ برای مقادیر ورودی که می‌توانند بین مقادیر ۰ یا ۱ باشند، چه تابعی باید استفاده شود تا این جدول درستی برای اینگونه مقادیر نیز ارزش داشته باشد.

یک جواب عملگر min است، که رابطه  $A \text{ AND } B$  را وقتی که مقادیر A و B بین ۰ و ۱ باشند با استفاده از رابطه  $\min(A,B)$  حل می‌کند. با استدلال یکسانی می‌توان از عملگرهای max برای OR

A	B	$\min(A,B)$
0	0	0
0	1	0
1	0	0
1	1	1

**AND**

A	B	$\max(A,B)$
0	0	0
0	1	1
1	0	1
1	1	1

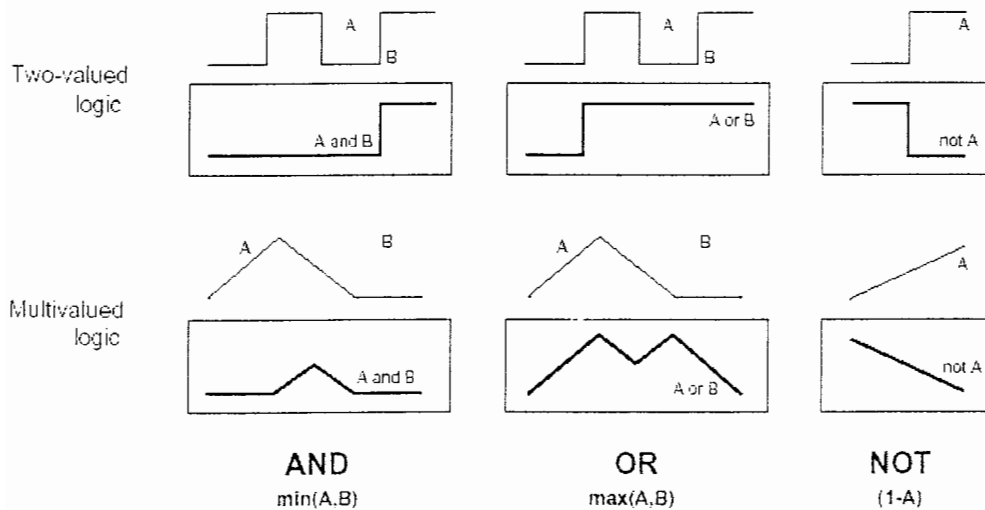
**OR**

A	$1 - A$
0	1
1	0

**NOT**

شکل ۳-۳ عملگرهای AND, OR در منطق فازی

و  $1-A$  برای NOT استفاده کرد. به مقادیر جدول درستی شکل ۳-۳ در این حالت توجه کنید که کاملاً بدون تغییر مانده است. علاوه بر این برای مقادیر دیگری که توابع تعلق می‌توانند به خود بگیرند این جدول درستی صحیح خواهد بود. برای بررسی بیشتر این موضوع به مثال تصویری شکل ۳-۴ توجه کنید.



شکل ۳-۴ مثال تصویری برای مقایسه عملگرهای AND, OR در منطق فازی و بولی

از روی این سه تابع می‌توان دریافت که ما هر ساختاری از مجموعه‌های فازی را می‌توانیم با استفاده از عملگرهای منطقی AND, OR و NOT فازی حل کنیم. و علاوه بر این در منطق فازی این توابع تنها راه حل برای این عملگرها نیستند، بلکه توابع و تعاریف دیگری نیز وجود دارد که ما از بیان آنها در این مختصر صرف‌نظر می‌کنیم که بررسی دقیق آنها در مراجعی که آورده شده است، بطور مفصل انجام گردیده است.

### ۳-۲ سیستم‌های فازی

قلب یک سیستم فازی یک پایگاه قواعد بوده که از قواعد اگر- آنگاه فازی تشکیل شده است. این قواعد اگر- آنگاه عبارتهایی هستند که برای فرموله کردن عبارات شرطی در این منطق بکار می‌رود.

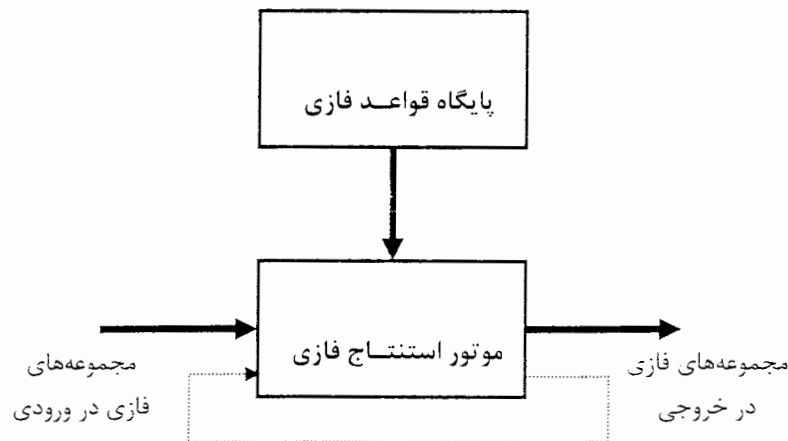


یک قاعده اگر- آنگاه تنها، به شکل زیر در نظر گرفته می‌شود.

۱-۳ اگر  $x_1, A_1$  و ... و  $x_n, A_n$  است، آنگاه  $y, B$  است.

به شکلی که  $A_i$  و  $B$  مقادیر زبانی هستند که با مجموعه‌های فازی بر روی محدوده‌های  $X$  و  $Y$  تعریف می‌شوند. قسمت اگر آن فرض یا مقدمه و قسمت آنگاه آن نتیجه نامیده می‌شود.

می‌توان گفت نقطه شروع ساخت یک سیستم فازی بدست آوردن مجموعه‌ای از قواعد اگر- آنگاه



شکل ۳-۵ ساختار اصلی سیستم‌های فازی خالص

فازی از دانش افراد خبره یا دانش حوزه مورد بررسی می‌باشد. مرحله بعدی ترکیب این قواعد در یک سیستم واحد است. سیستم‌های فازی مختلف از اصول و روشهای متفاوتی برای ترکیب این قواعد استفاده می‌کنند.

بنابراین سوال اساسی این است، چه نوع سیستم‌های فازی معمولاً استفاده می‌شود؟ در کتب و مقالات معمولاً از سه نوع سیستم فازی صحبت به میان می‌آید: (۱) سیستم‌های فازی خالص، (۲) سیستم‌های فازی TSK<sup>۱</sup> و (۳) سیستم‌های با فازی‌ساز و غیر فازی‌ساز، بطور خلاصه این سه نوع سیستم را شرح می‌دهیم.

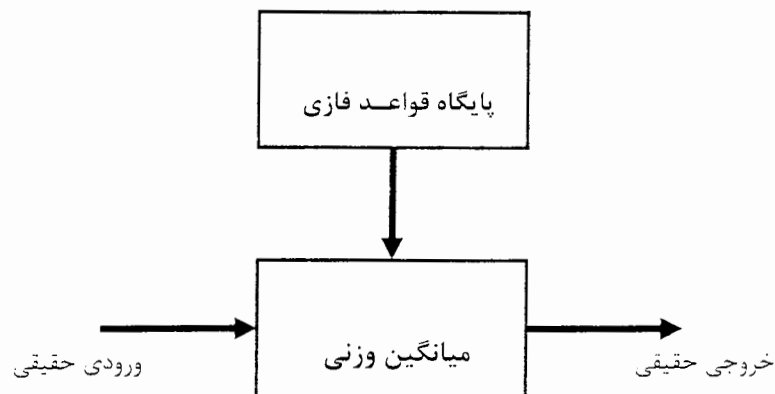
ساختار اصلی یک سیستم فازی خالص در شکل ۳-۵ نشان داده شده است. پایگاه قواعد فازی

مجموعه‌ای از قواعد اگر- آنگاه فازی را نشان می‌دهد. به عنوان مثال، برای پایدارساز فازی در این پروژه، پایگاه قواعد فازی شامل قواعد ۳-۴ تا ۳-۱۲ می‌باشد. موتور استنتاج فازی این قواعد را به یک نگاهت از مجموعه‌های فازی در فضای ورودی به مجموعه‌های فازی در فضای خروجی بر اساس اصول منطق فازی ترکیب می‌کند. در شکل ۳-۵ اگر خط نقطه چین وجود داشته باشد، چنین سیستمی، سیستم فازی دینامیک نامیده می‌شود.

مشکل اصلی در رابطه با سیستمهای فازی خالص این است که ورودی‌ها و خروجی‌های آن مجموعه‌های فازی می‌باشند(واژه‌هایی در زبان طبیعی). در حالی که در سیستمهای مهندسی، ورودی‌ها و خروجی‌ها متغیرهایی با مقادیر حقیقی می‌باشند. برای حل این مشکل، تاکاگی- سوگنو و کانگ نوع دیگری از سیستم‌های فازی را معرفی کردند که ورودی و خروجی‌های آن متغیرهایی با مقادیر واقعی هستند.

سیستم فازی TSK یکی از انواع سیستم‌های فازی است که از قواعدی به شکل زیر بنا شده است.

$$2-3 \quad \text{اگر } x_1, x_1 \text{ است و } \dots \text{ و } x_n, x_n \text{ است، آنگاه } y' = c_0' + c_1'x_1 + \dots + c_n'x_n$$



شکل ۳-۶ ساختار اصلی سیستم‌های فازی TSK

که  $C_l^i$  مجموعه‌های فازی،  $c_l^i$  ضرایب ثابت و  $l=1,2,\dots,M$  می‌باشند. در این رابطه قسمت‌های اگر قواعد همانند قواعد اگر- آنگاه سیستم‌های فازی معمولی می‌باشند، اما قسمت‌های آنگاه، ترکیب خطی از متغیرهای ورودی هستند. به ازای یک ورودی  $x = (x_1, \dots, x_n)^T \in U \subset R^n$  خروجی  $f(x) \in V \subset R$  سیستم فازی TSK، به صورت متوسط وزن‌دار شده  $y^l$ ‌های رابطه ۳-۲ محاسبه می‌گردد، یعنی،

$$f(x) = \frac{\sum_{l=1}^M y^l w^l}{\sum_{l=1}^M w^l} \quad 3-3$$

که وزنهای  $w^l$  از رابطه زیر محاسبه می‌شوند.

$$w^l = \prod_{i=1}^n \mu_{C_l^i}(x_i)$$

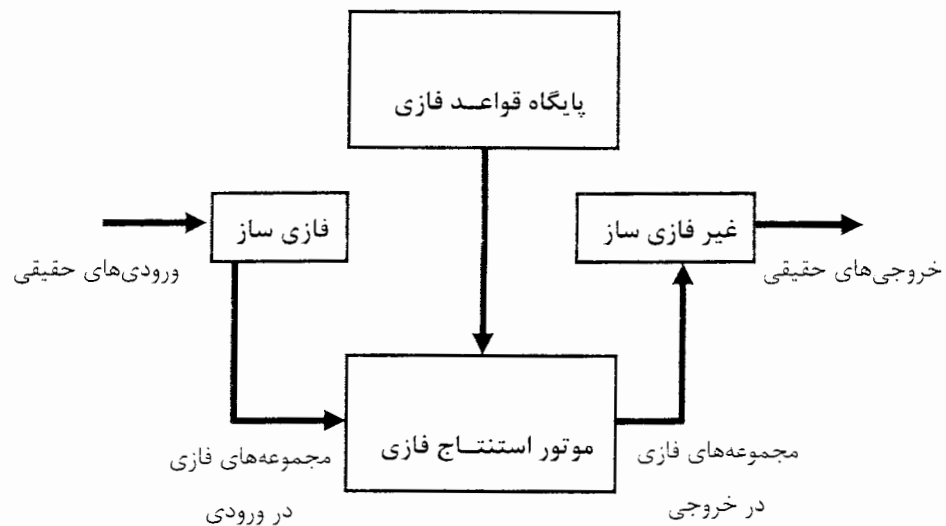
مشاهده می‌شود که سیستم فازی TSK هنوز به عنوان یک نگاشت از  $U \subset R^n$  به  $V \subset R$  مطرح می‌گردد. معنی فیزیکی رابطه ۳-۳ این است که وقتی  $x$  به محدوده فازی مقید می‌گردد که بوسیله بخش اگر قاعده توصیف می‌شود، خروجی یک تابع خطی از متغیرهای ورودی می‌باشد. بنابراین سیستم فازی می‌تواند تا اندازه‌ای همانند یک تابع قطعه قطعه خطی تصور گردد، که تغییرات آن از یک قطعه به قطعه دیگر ملایم می‌باشد. اگر  $c_l^i = 0$  برای  $i=1,2,\dots,n$  و  $c_l^0$  معادل با مرکز  $\bar{y}^l$  از مجموعه فازی  $B^l$  در قاعده اگر- آنگاه فازی ۱-۳ باشد، آنگاه سیستم فازی TSK، با سیستم فازی که بوسیله موتور استنتاج ضرب، فازی‌ساز منفرد و غیر فازی‌ساز میانگین مراکز تعریف می‌گردد، یکسان خواهد بود.

مقایسه ۳-۴ با ۳-۱۳ نشان می‌دهد که بخش آنگاه قواعد فازی از یک عبارت توصیفی با مقادیر زبانی به یک رابطه ریاضی ساده تبدیل شده است. این تغییر، ترکیب قواعد فازی را آسانتر می‌سازد. در حقیقت سیستم فازی TSK یک میانگین وزنی از مقادیر بخش‌های آنگاه قواعد می‌باشد. ساختار اصلی سیستم فازی TSK در شکل ۳-۶ نشان داده شده است.

مشکلات عمده سیستم فازی TSK عبارتند از: (۱) بخش آنگاه قاعده یک فرمول ریاضی بوده و

بنابراین چهارچوبی را برای نمایش دانش بشری فراهم نمی‌کند. (۲) این سیستم دست ما را برای اعمال اصول مختلف منطق فازی باز نمی‌گذارد و در نتیجه انعطاف پذیری سیستم‌های فازی در این ساختار وجود ندارد، برای حل این مشکلات ما از نوع سومی از سیستم‌های فازی یعنی سیستم‌های فازی با فازی سازها و غیر فازی سازها استفاده می‌کنیم.

به منظور استفاده از سیستم‌های فازی خالص در سیستم‌های مهندسی، یک روش ساده اضافه کردن یک فازی ساز در ورودی که متغیرهای با مقادیر حقیقی را به یک مجموعه فازی تبدیل کرده و یک فازی ساز که یک مجموعه فازی را به یک متغیر با مقدار حقیقی در خروجی تبدیل می‌کند، می‌باشد. نتیجه یک سیستم فازی با فازی ساز و غیر فازی ساز بوده که شکل ۳-۷ نشان داده شده است. این سیستم فازی معایب سیستم فازی خالص و سیستم TSK را می‌پوشاند.



شکل ۳-۷ ساختار اصلی سیستم‌های فازی با فازی‌ساز و غیر فازی‌ساز

به عنوان نتیجه گیری برای این بخش، لازم است بر روی یک مشخصه سیستم‌های فازی تأکید نماییم. سیستم‌های فازی از یک سو نگاهی به صورت چند ورودی و یک خروجی از یک بردار با مقادیر حقیقی به یک اسکالر با مدار حقیقی بوده (نگاشت چند خروجی را می‌توان با ترکیب چند نگاشت یک خروجی بوجود آورد) که روابط دقیق ریاضی این نگاشت‌ها را می‌توان بدست آورد و از

سویی دیگر سیستم‌های فازی، سیستم‌های مبتنی بر دانش بوده که از روی دانش بشری به شکل قواعد اگر- آنگاه ساخته می‌شوند. بنابراین می‌توان گفت، سیستم‌های فازی برای فرموله کردن دانش بشری استفاده می‌شوند. این دانش را می‌توان به صورت صریح و روشن در قالب کلمات بیان کرد، این دانش خود آگاه را ما خیلی ساده می‌توانیم از افراد خبره بخواهیم تا به صورت عبارتهای اگر- آنگاه فازی در اختیار ما بگذارد تا آنرا در سیستم‌های فازی قرار دهیم.

پس از اینکه قواعد اگر- آنگاه تهیه گردید این قواعد در یک موتور استنتاج فازی، با هم ترکیب می‌شوند. موتورهای استنتاج حاصلضرب و مینیمم از موتورهای استنتاج فازی رایج برای ترکیب قواعد هستند.

### ۳-۳ کنترل فازی

وقتی سیستم‌های فازی به عنوان کنترل کننده مورد استفاده قرار می‌گیرند به آنها کنترل کننده‌های فازی گفته می‌شود. اگر سیستم‌های فازی به منظور مدل‌سازی فرایند بکار روند و کنترل کننده‌ها بر پایه این طراحی می‌گردند، آنگاه این کنترل کننده‌ها نیز به نام کنترل کننده‌های فازی تلقی می‌شوند. بنابراین کنترل کننده‌های فازی کنترل کننده‌های غیر خطی با یک ساختار خاص هستند. کنترل فازی کاربردهای موفقیت‌آمیزی از تئوری فازی را در مسائل ارائه می‌نماید.

کنترل فازی می‌تواند به دو دسته کنترل فازی غیر تطبیقی و کنترل فازی تطبیقی دسته‌بندی شود. در کنترل فازی غیر تطبیقی، ساختار و پارامترهای کنترل کننده‌های 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 بر حسب شرایط در طی اعمال زمان حقیقی تغییر خواهد کرد. کنترل فازی غیر تطبیقی ساده‌تر از کنترل فازی تطبیقی بوده اما به اطلاعات بیشتری از مدل فرایند و قواعد ذاتی مدل نیاز دارد. از طرفی دیگر، کنترل فازی تطبیقی جهت به اجرا آمدن بر هزینه بوده ولی در عین حال به اطلاعات کمتری نیاز خواهد داشت و در نتیجه ممکن است بهتر و مؤثرتر به اجرا درآید.

### ۳-۴ کنترل فازی در مقایسه با کنترل کلاسیک

کنترل فازی و کنترل کلاسیک دارای شباهت‌ها و تفاوت‌هایی می‌باشند. این دو در موارد زیر مشابه می‌باشند:

این دو سعی در حل نوع یکسانی از مسائل را دارند، که همان مسائل کنترل می‌باشد. بنابراین بایستی به جنبه‌های یکسانی که در هر مسأله کنترل مشترک می‌باشد توجه داشته باشند. به عنوان مثال پایداری و عملکرد.

ابزارهای ریاضی مورد استفاده جهت تحلیل سیستم‌های کنترل طراحی شده مشابه می‌باشند، چراکه به مطالعه جنبه‌های یکسانی (پایداری، همگرایی و ...) برای یک سیستم می‌پردازند.

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 می‌نماید و کنترل کننده‌ها برای مدل طراحی می‌گردند، در حالیکه کنترل کننده فازی با استفاده از تجربیات شخص خبره (بصورت قواعد اگر- آنگاه فازی) شروع می‌گردد و کنترل کننده‌ها بوسیله ترکیب این قواعد طراحی می‌گردند. بدین معنی که اطلاعات لازم جهت ساخت این دو نوع کنترل کننده فازی متفاوت است.

نقطه شروع طراحی یک کنترل کننده فازی بدست آوردن مجموعه‌ای از قواعد اگر آنگاه فازی با استفاده از دانش افراد خبره و دستورالعمل‌های موجود است، بدین معنی که دانستن مدل ریاضی فرآیند تحت کنترل یک شرط لازم برای طراحی کنترل کننده‌های فازی نخواهد بود. با این حال ما در مرحله تحلیل سیستمیای کنترل فازی حلقه بسته بصورت تئوری، ما بایستی حداقل یک مدل اجمالی از فرآیند تحت مطالعه داشته باشیم. علاوه بر آن اگر بخواهیم کنترل کننده فازی را چنان طراحی کنیم که پاره‌ای از ملاک‌ها نظیر پایداری نمایی را تضمین نماید، لذا بایستی یک مدل ریاضی برای فرآیند موجود فرض کنیم، بطوریکه بر اساس آن خواص سیستم طراحی شده مورد تحلیل ریاضی قرار گیرد.

### ۳-۵ نحوهٔ ایجاد کنترل کننده‌های فازی در MATLAB

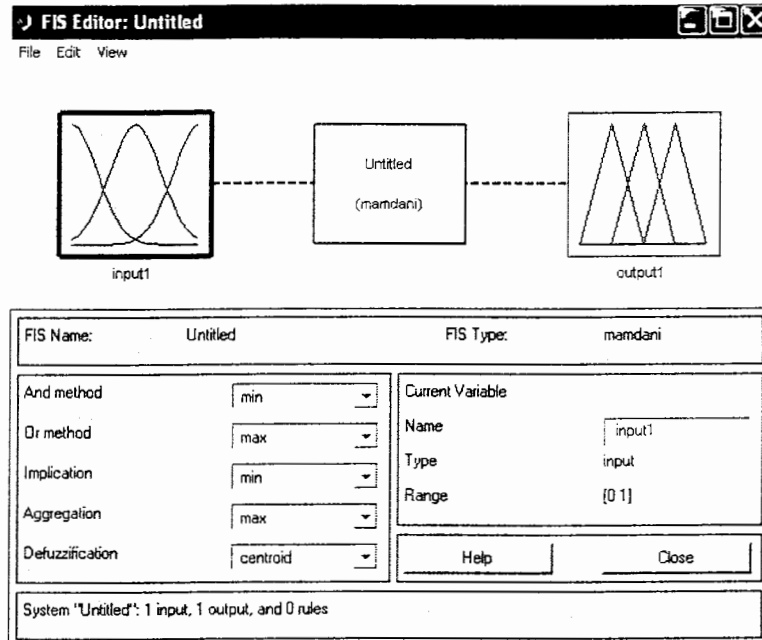
همچنانکه بیان شد کنترل کننده‌های فازی، کنترل کننده‌هایی هستند که بیشتر بر دانش فرد خبره تکیه دارند تا یک مدلسازی ریاضی از فرآیند تحت کنترل. بنابراین ایجاد کنترل کننده‌های اینچنینی نیاز به آگاهی کاملی از سیستم تحت کنترل دارد به شکلی که هر چه این آگاهی در مورد شرایط و نحوه برخورد کنترل کننده در مواجهه با سیستم مورد نظر، کامل‌تر و پخته‌تر باشد. این سعی زودتر و بهتر به نتیجه خواهد رسید.

بدست آوردن چنین شناختی برای افرادی که آشنایی مبسوطی در مورد سیستم ندارند میتواند با ارزش باشد. مراجعه به افراد متخصص که دانش فرد خبره را در اختیار ما بگذارند، به عنوان راه حلی که می‌تواند بسیار راه گشا باشد، پیشنهاد می‌شود. ولی در صورت نبود چنین فردی راه حل دیگری نیز وجود دارد و آن شبیه‌سازی سیستم مورد نظر با یک کنترل کننده خطی مرسوم مثل PID می‌باشد. در این روش ما پس از ایجاد یک کنترل کننده مطلوب که پاسخ مطلوب و دلخواه ما را فراهم کند با تغییر مقادیر ورودی‌ها و زیر مشاهده خروجی کنترل کننده می‌توان تا اندازه‌ای که بتوان با کمی سعی و خطا در قسمت رابط گرافیکی (GUI) فازی در Matlab در تعداد و شکل توابع تعلق ورودی و خروجی و همچنین قواعد استفاده شده در پایگاه قواعد فازی می‌توان به نتیجه دلخواه رسید.

در ادامه طرز ایجاد یک کنترل کننده فازی در Simulink به اختصار توضیح داده می‌شود:

اولین گام ایجاد یک فایل fis می‌باشد که حاوی اطلاعات کنترل کننده، شامل توابع تعلق ورودی و خروجی و قواعد اگر- آنگاه است. برای اینکار در پنجره Command دستور fuzzy را تایپ می‌کنیم. این دستور رابط گرافیکی فازی را که برای ایجاد کنترل کننده‌ها و سیستم‌های فازی در نظر گرفته شده فعال می‌کند، در شکل ۳-۸ تصویری از این رابط نشان داده شده است. این رابط تمامی لوازمی که ما برای ایجاد یک کنترل کننده فازی نیاز داریم فراهم می‌کند. برای داشتن اطلاعات بیشتر در ارتباط با این رابط می‌توانید به مرجع [۳۰] مراجعه کنید. پس از ایجاد فایل و ذخیره آن برای استفاده

در قسمت Simulink از منوی گزینه `File / Export / to workspace` را انتخاب می‌کنیم و کنترل کننده ایجاد شده را با نام خاصی به Workspace منتقل می‌کنیم.



شکل ۳-۸ - رابط گرافیکی فازی (GUI)

گام بعدی استفاده در قسمت Simulink است. برای اینکار پس از ایجاد سیستم تحت کنترل از Toolbox فازی یک کنترل کننده فازی به مدل سیستم اضافه می‌کنیم. با دوبار کلیک بر روی این کنترل کننده در صفحه Simulink پنجره‌ای باز می‌شود که باید نامی را که کنترل کننده با آن به Workspace منتقل شد را وارد کنیم. در صورتی که ورودی بیشتر از یکی بود که معمولاً اینگونه است از یک بلوک MUX برای وارد کردن ورودی‌ها به کنترل کننده استفاده می‌کنیم.

### ۳-۶ طراحی PSS فازی

برای PSSهای فازی چه از نوع ممدانی و چه از نوع سوگنژ متغیرهای فرکانس و زاویه به عنوان ورودی‌های کنترل کننده انتخاب شده است، و خروجی آن سیگنال خروجی PSS، که همان  $v_s$  است



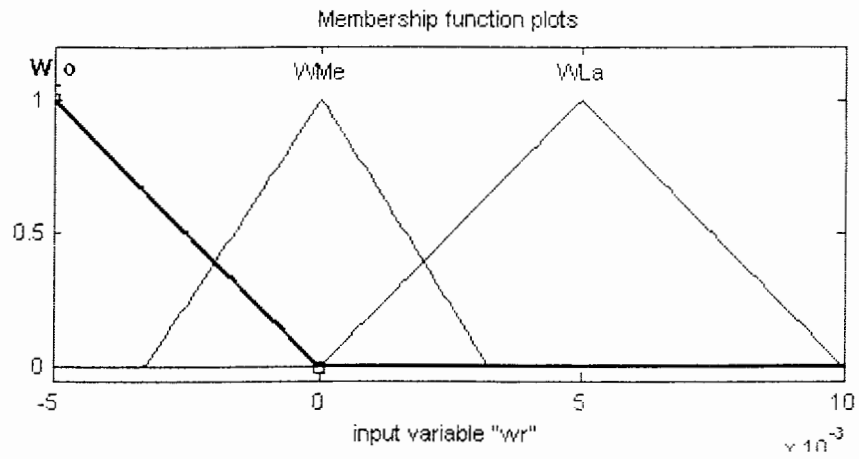
که در شکل ۲-۷ نشان داده شده است. همچنانکه ذکر آن رفت برای ایجاد کنترل کننده فازی در ابتدا سعی می‌کنیم یک شناخت کلی در مورد سیستم بدست آوریم. ما برای اینکار از پایدارساز سیستم قدرت با ورودی فرکانس استفاده کردیم که در شکل ۲-۶ نشان داده شده است. پس از بررسی‌ها و مشاهدات انجام شده کنترل کننده‌ای که برای این مورد در نظر گرفته شده به صورت زیر است:

### ۳-۶-۱ طراحی PSS ممدانی

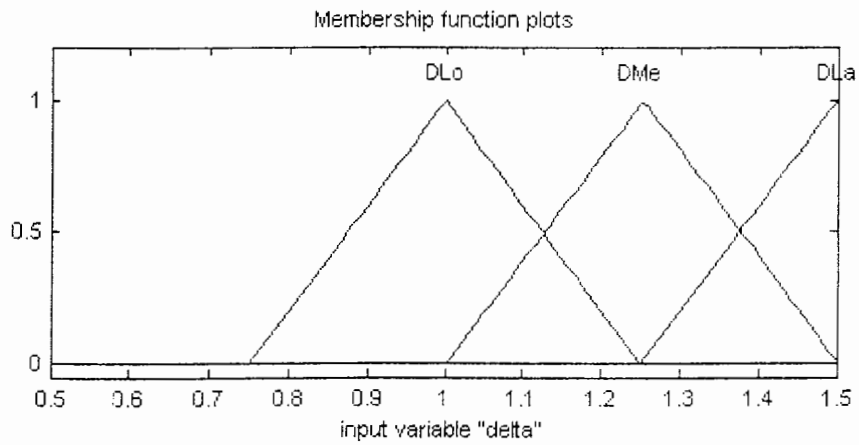
مجموعه توابع تعلق ورودی دارای سه تابع تعلق کم (WLo)، متوسط (WMe) و زیاد (WLa) برای ورودی فرکانس و سه تابع تعلق کم (DL0)، متوسط (DMe) و زیاد (DLa) برای ورودی زاویه و پنج تابع تعلق خیلی کم (PVI0)، کم (PL0)، متوسط (PMe)، زیاد (PLa)، خیلی زیاد (PVIa)، برای خروجی PSS می‌باشد که بترتیب در شکل‌های ۳-۹، ۳-۱۰ و ۳-۱۱ نشان داده شده‌اند.

این کنترل کننده شامل قواعد اگر-آنگاهی به صورت زیر است:

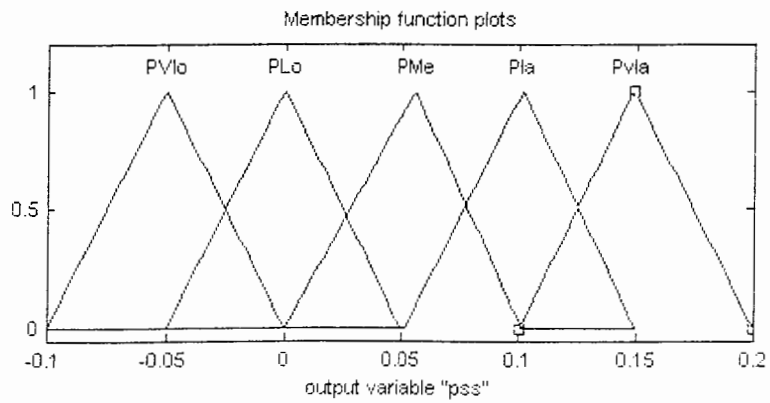
- |      |   |
|------|---|
| ۴-۳  | اگر ورودی فرکانس کم و ورودی زاویه کم ، آنگاه خروجی خیلی کم است.       |
| ۵-۳  | اگر ورودی فرکانس کم و ورودی زاویه متوسط ، آنگاه خروجی کم است.         |
| ۶-۳  | اگر ورودی فرکانس کم و ورودی زاویه زیاد ، آنگاه متوسط خروجی است.       |
| ۷-۳  | اگر ورودی فرکانس متوسط و ورودی زاویه کم ، آنگاه خروجی کم است.         |
| ۸-۳  | اگر ورودی فرکانس متوسط و ورودی زاویه متوسط ، آنگاه خروجی متوسط است.   |
| ۹-۳  | اگر ورودی فرکانس متوسط و ورودی زاویه زیاد ، آنگاه خروجی زیاد است.     |
| ۱۰-۳ | اگر ورودی فرکانس زیاد و ورودی زاویه کم ، آنگاه خروجی متوسط است.       |
| ۱۱-۳ | اگر ورودی فرکانس زیاد و ورودی زاویه متوسط ، آنگاه خروجی زیاد است.     |
| ۱۲-۳ | اگر ورودی فرکانس زیاد و ورودی زاویه زیاد ، آنگاه خروجی خیلی زیاد است. |



شکل ۳-۹ توابع تعلق ورودی فرکانس



شکل ۳-۱۰ توابع تعلق ورودی زاویه



شکل ۳-۱۱ توابع تعلق خروجی PSS

## ۳-۶-۲ طراحی PSS سوگنؤ

همچنانکه در بخشهای قبلی ذکر شد کنترل کننده سوگنؤ فقط در قسمت آنگاه با کنترل کننده ممدانی متفاوت است، بگونه که در قسمت آنگاه یک تابع خطی به اضافه یک عدد ثابت به عنوان نتیجه در نظر گرفته می‌شود. این تفاوت باعث ایجاد فرقهایی در تعریف مجموعه توابع تعلق خروجی گردیده است. براساس بررسی‌ها و مشاهدات انجام شده پنج تابع تعلق خیلی کم (PVlo)، کم (PLo)، متوسط (PMe)، زیاد (PLa)، خیلی زیاد (PVla)، برای خروجی PSS به صورت مقادیر وزنی از این توابع تعلق در نظر گرفته می‌شوند که مقادیر این ثابتها در جدول ۳-۱ لیست می‌شوند.

جدول ۳-۱ مقادیر وزنی طراحی شده برای خروجی PSS سوگنؤ

PVla	PLa	PMe	PLo	PVlo
۰/۲	۰/۱	۰/۰۴	۰	-۰/۰۹

با این توصیف قواعد اگر- آنگاهی که در بخش پیش ارائه شد به صورت زیر تغییر می‌یابد تا شرایط کنترل کننده سوگنؤ را فراهم کند:

- ۱۳-۳ اگر ورودی فرکانس کم و ورودی زاویه کم، آنگاه خروجی  $۰/۰۹$  - است.
- ۱۴-۳ اگر ورودی فرکانس کم و ورودی زاویه متوسط، آنگاه خروجی  $۰$  است.
- ۱۵-۳ اگر ورودی فرکانس کم و ورودی زاویه زیاد، آنگاه خروجی  $۰/۰۴$  است.
- ۱۶-۳ اگر ورودی فرکانس متوسط و ورودی زاویه کم ، آنگاه خروجی  $۰$  است.
- ۱۷-۳ اگر ورودی فرکانس متوسط و ورودی زاویه متوسط ، آنگاه خروجی  $۰/۰۴$  است.
- ۱۸-۳ اگر ورودی فرکانس متوسط و ورودی زاویه زیاد ، آنگاه خروجی  $۰/۱$  است.
- ۱۹-۳ اگر ورودی فرکانس زیاد و ورودی زاویه کم ، آنگاه خروجی  $۰/۰۴$  است.
- ۲۰-۳ اگر ورودی فرکانس زیاد و ورودی زاویه متوسط ، آنگاه خروجی  $۰/۱$  است.
- ۲۱-۳ اگر ورودی فرکانس زیاد و ورودی زاویه زیاد ، آنگاه خروجی  $۰/۲$  است.

## فصل چهارم

**پیاده‌سازی سخت‌افزاری**

**پایدار ساز**

**سیستم قدرت**

## پیاده‌سازی سخت‌افزاری پایدار ساز سیستم قدرت

عباراتی که از مفاهیم و مقوله‌های ذهنی از قبیل خوبی، کمی، زیادی و... در آن استفاده شده است نقشی اساسی در روند پردازش و تصمیم‌گیری‌های مغز انسان دارند و مغز عباراتی را که مقادیر مشخصی ندارند با موفقیت برای محاسبات پیچیده خود جهت تصمیم‌گیری بکار می‌گیرد. البته در بسیاری از موارد عدم قطعیت‌هایی که در عبارات بیانی یا زبانی وجود دارند انعطافی موثر و مفید به روند تلفیق و پردازش اطلاعات در مغز انسان به منظور ارائه نتیجه و یا تصمیم‌گیری خاصی می‌بخشند.

در عالم واقع نمی‌توان برای همه حالاتی که یک پدیده و یا فعل ممکن است به خود بگیرد قانونی مجزا وضع نمود بلکه قوانین حالتی نسبتاً کلی داشته و جز برای تعدادی مورد خاص قابل تعریف نیستند و این باعث گسستگی در جامعیت ارایه قانونی برای معرفی و شمول همه جوانب یک پدیده یا فعل خواهد شد و عدم قطعیت‌ها نقش تقریب‌هایی که ذهن در هنگام اخذ یک تصمیم می‌زند در همین فاصله بین گسستگی‌ها قرار می‌گیرند.

در واقع ذهن انسان بین حالات مختلفی از یک پدیده یا فعل که توسط قوانین نسبتاً کلی مشخص شده‌اند پیوند و رابطه میانگینی برقرار می‌کند و در واقع انعطافی که در قوانین بیانی زبان انسان وجود

دارد، بدست آوردن این حد میانگین در تلفیق قوانین مختلف را ممکن می‌سازد. به علاوه عملکرد استنتاجی و قیاسی فکر انسان نیز بخاطر همین عدم قطعیت و انعطاف‌پذیری در بیان و منطق انسانی است. منطق فازی برای اولین بار به فرموله کردن این قواعد ذهنی با روشهای ریاضی پرداخت و به شکل بسیار موفقیت‌آمیزی در این امر پیشرفت کرد. امروزه ما شاهدیم که بسیاری از دستگاهها و عملیات‌های کنترلی از منطق فازی برای کنترل بهره می‌گیرند و در واقع از سیستم استنتاجی شبیه به مغز انسان برای کنترل سیستم تحت کنترل که توسط یک فرد خیره برای آن قاعده‌بندی شده، بهره می‌گیرد.

#### ۱-۴ ساختارهای سخت‌افزاری برای سیستم‌های منطق فازی

منطق فازی یک تکنولوژی کلی و جهانی است و محدوده کاربردهای آن طیف وسیعی از سیستم‌های کنترلی را شامل می‌شود بنابراین سخت‌افزارهای مورد استفاده، متعدد و مختلف می‌باشند اما ابزارهای نرم‌افزاری امروزی ما را قادر می‌سازند که بتوانیم طراحی و پیاده‌سازی سیستم‌های منطق فازی خود را بر روی بیشتر سخت‌افزارهای موجود، از میکروکنترلرهای ارزان قیمت هشت بیتی با حافظه RAM و ROM کم گرفته تا سیستم‌های کنترل فرایند توزیع شده که شامل چندین پردازنده RISC از انواع ۶۴ بیتی هستند، توسعه دهیم.

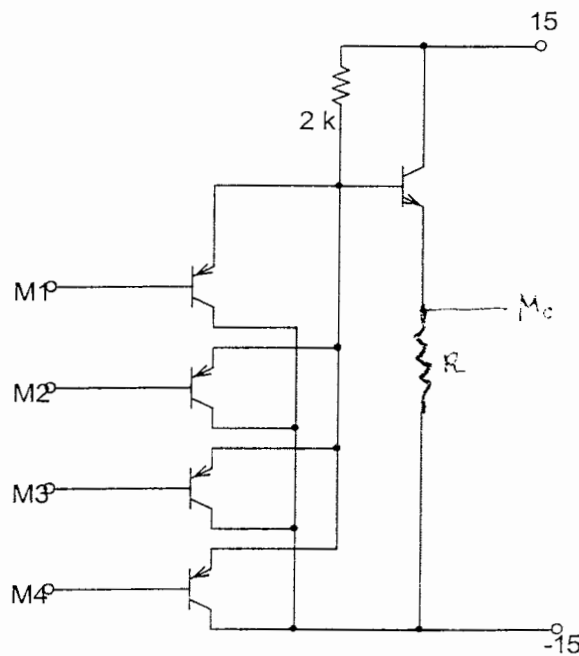
در صفحات بعد بطور خلاصه به معرفی تکنولوژی‌ها و ابزار مورد نیاز جهت کار با سخت‌افزارهای مختلف خواهیم پرداخت.

#### ۲-۴ انواع پردازشگرهای اختصاصی برای منطق فازی

در سال ۱۹۸۰ میلادی اجرای یک نرم‌افزار پیاده‌سازی شده برای سیستم منطق فازی بر روی یک MCU از نوع هشت‌بیتی، در حدود یک ثانیه طول می‌کشید که این زمان برای بیشتر کاربردهای کنترلی از نوع حلقه بسته بسیار کند بود، بنابراین طراحان سخت افزاری شروع به طراحی تراشه‌های مخصوصی برای کاربردهای منطق فازی کردند.

## ۴-۲-۱ نسل اول - پردازشگرهای فازی آنالوگ

از اولین طراحی‌های انجام گرفته ژاپنی در این زمینه مطابق شکل ۴-۱ گیت‌های آنالوگ فازی است، که ولتاژ سیگنالهای ورودی در محدوده‌ای بین  $\pm 5$  ولت، معرف درجه عضویت از  $\mu=0$  تا  $\mu=1$  بودند [۸].



شکل ۴-۱ گیت‌های آنالوگ فازی

با مجتمع ساختن این گیت‌های فازی، تأثیرات گرمایی ترانزیستورها بر یکدیگر اثر گذاشته و دقتی در حدود ۰/۱ درصد و زمان قطع و وصلی در حدود نانو ثانیه بدست می‌دادند، بنابراین در بیشتر مدارات منطقی از این نوع، بیش از پنج گیت بدنبال هم قرار نمی‌گرفت اما مزایای آن عبارت بودند از:

- پردازش‌های فازی سریعتر بودند.
- نسبت به سایر طراحی‌های سخت‌افزاری از پیچیدگی کمتری برخوردار بودند.
- سازگاری الکترومغناطیسی بیشتری نسبت به سایر سخت‌افزارها داشتند.

اما علیرغم این مزایا پردازشگرهای فازی آنالوگ هیچ وقت نتوانستند راه خود را به بیرون از آزمایشگاه‌ها باز کنند، زیرا اشکال عمده برای انجام این کار عدم امکان برنامه سازی آنها بود، به عنوان مثال هر تغییری در شرطها یا توابع عضویت مورد نیاز به تغییر سخت افزار نیاز داشت.

البته تلاشهای زیادی برای غلبه بر این مشکل به عمل آمد برای مثال استفاده از تکنولوژی CMOS و اضافه کردن عناصر اضافه‌ای برای افزودن قابلیت برنامه سازی به مدار بود، اما تاکنون تلاشهای فوق در این زمینه به نتایج چشمگیری نرسیده‌است، اما تحقیقات در این زمینه کماکان ادامه دارد.

البته کاربردها و محصولات متنوع زیادی از منطق فازی وجود دارند، که ضمن داشتن ورودی‌ها و خروجی‌های آنالوگ، نیازی به تغییری در منطق و شرطهای خود (همانند سیستمهای هوشمند خود فراگیر) در هنگام اجرا ندارند. بنابراین در این قبیل کاربردها و محصولات استفاده از شیوه آنالوگ بسیار سریعتر و ارزانتر از استفاده از پردازشگرهای مرسوم است.

#### ۲-۲-۴ نسل دوم - پردازشگرهای فازی دیجیتال

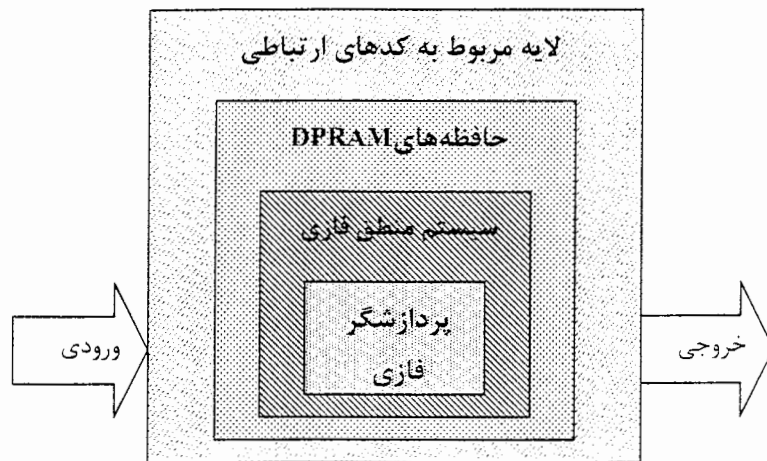
نسل دوم پردازشگرهای فازی، حاصل تجربیات ناشی از توسعه نسل اول است و آنچه که در این نامگذاری ضد و نقیض بنظر می‌رسد، چگونگی دیجیتالی یا دو مقداری ( صفر و یک ) و فازی ( مقادیر پیوسته) بودن آن است! پاسخ به این سوال ساده است، پردازشگرهای فازی دیجیتالی، پردازشگرهای دیجیتالی هستند که دستورات مرسوم آنها با مجموعه قوانین منطق فازی جایگزین شده‌اند.

بعضی از این پردازشگرهای فازی دیجیتالی ساختاری بهینه سازی شده برای عملیات منطق فازی نظیر توابع سخت‌افزاری ALU برای فازی سازی، استنتاج قوانین و غیر فازی سازی، دارند.

نسل دوم از پردازشگرهای فازی تنها از محاسبات منطق فازی پشتیبانی می‌کنند و برای ارتباط با دستگاههای جانبی، I/O، پیش‌پردازش داده‌ها و کدهای فرامین معمولی و انجام سایر وظایف



محاسباتی، نیاز به یک واحد میکروکنترلی میزبان یا (MCU) دارند که از طریق یک حافظه RAM از نوع درگاه دوطرفه (DPRAM) و یا درگاه سریال به آن متصل می‌شوند.



شکل ۲-۴

با توجه به شکل ۲-۴ لایه کدهای ارتباطی، ورودی‌های سیستم‌فازی را دریافت کرده و آنها را پیش‌پردازش و تبدیل به متغیرهای ورودی می‌نماید سپس MCU میزبان متغیرهای ورودی تبدیل شده جهت ورود به سیستم منطق فازی را در داخل DPRAM نوشته و پردازشگر فازی را تحریک و فعال می‌کند که پس از انجام محاسبات و عملیات فازی توسط واحد پردازشگر محاسبات فازی ALU، متغیرهای خروجی آن مجدداً در داخل DPRAM نوشته شده و با صدور یک وقفه به واحد میکروکنترلر میزبان جهت اجرای سایر مراحل عملیاتی تحویل تحویل و اطلاع داده می‌شود.

لایه منطق‌فازی شامل تعاریف مربوط به سیستم فازی مورد نظر نظیر تعاریف توابع عضویت، قوانین فازی و یا غیره می‌باشند.

در مقایسه با پردازشگرهای فازی آنالوگ، پردازشگرهای فازی دیجیتالی سالهاست که به بازار عرضه شده، اما هنوز در کاربردهای صنعتی استفاده چندانی پیدا نکرده‌اند زیرا در این شیوه استفاده از دو تراشه علاوه بر قیمت بالای آن، کار طراحی و پیاده‌سازی آن نیز مشکل است به عنوان مثال اگر

بخواهیم به DPRAM هم بطور مستقیم در طراحی و برنامه سازی خود جهت ارتباطات دسترسی داشته باشیم آن وقت با مشکل کار بر روی سه تراشه مواجه خواهیم شد.

همچنین برنامه سازی بین پردازشگرهای داخلی مثل برقراری و کنترل ارتباط بین پردازشگر فازی و میکروکنترلرهای MCU کار بسیار پیچیده‌ای است و نیازمند برنامه‌سازی و استفاده از وقفه‌های مختلف بر روی میکرو کنترلر MCU دارد.

#### ۴-۲-۳ نسل سوم - پردازشگرهای فازی مجتمع

نسل جدید پردازشگرهای فازی برای جبران نقایص نسل دوم و با هدف مجتمع ساختن قابلیت پردازش فازی در داخل میکروکنترلرها بوجود آمده‌اند.

برای رسیدن به این منظور شیوه‌های متفاوتی وجود دارد بعنوان مثال بعضی از سازندگان تراشه‌های نیمه‌هادی واحد پردازشگر فازی را در کنار میکروکنترلر و بر روی یک تراشه مجتمع ساخته‌اند.

مزایای این طراحی نسبت به حالت استفاده از دو تراشه مجزا عبارتند از عملکرد دو واحد ALU و MCU بطور همزمان و همچنین کاهش چشمگیر سرباری ناشی از نقل و انتقال داده‌ها بدلیل استفاده از ثباتهای مشترک برای رد و بدل کردن داده‌ها می‌باشد و همچنین برنامه‌سازی و ایجاد تغییرات نیز در چنین ساختاری آسانتر است.

شیوه دیگر، گسترش مجموعه دستورات میکروکنترلر استاندارد است، بنحوی که تنها میکروکدها توسعه یابد که این عمل بسته به ساختار MCU سرعت عملیات مربوط به قسمت سیستم فازی را برحسب اندازه سیستم منطق فازی تا دو برابر خلاصه‌تر می‌نماید.

برای شتاب بخشیدن بیشتر به سرعت محاسبات سیستم‌های منطق فازی واحد ALU نیز به دستورات توسعه یافته‌ای نیاز دارد.

چنین توسعه‌هایی عبارت از فازی سازی‌های توابع عضویت MBF استاندارد، دستورات مربوط به استنتاج قوانین و غیرفازی سازی به شیوه COM که همه آنها پشتیبانی شده توسط سخت‌افزار باشند، است.

مزیت دستورات توسعه یافته آن است که یک ALU استاندارد بسیاری از دستورات مورد نیاز جهت محاسبات فازی را شامل می‌شود بنابراین این فضای سیلیکون و اتصالات مورد نیاز جهت اجرای محاسبات فازی در مقایسه با شیوه قبلی مجتمع سازی تراشه‌های هم پردازش، بسیار کمتر و طبعاً سریعتر می‌باشد.

اما از معایب آن ناگزیر به تعویض و استفاده از طراحی مجدد تراشه در صورت نیاز به تغییرات می‌باشد.

#### ۳-۴ میکروکنترلرها و میکروپروسسورهای استاندارد

همانطور که قبلاً اشاره کردیم اجرای یک واحد محاسبه منطق فازی بر روی یک میکروکنترلر هشت بیتی استاندارد با استفاده از الگوریتم‌های موجود در بیست سال پیش چیزی حدود یک ثانیه بطول می‌انجامید.

بنابراین در آن زمان مهندسی که بر روی طراحی و توسعه و بهبود سیستم‌های کنترل فازی تحقیق می‌کردند، به دو گروه عمده توسعه دهندگان سخت افزار و بهبود دهندگان الگوریتم‌های فازی تقسیم شدند، که البته گروه دوم در این راه بسیار موفق‌تر بودند و امروزه زمان لازم برای انجام محاسبات فازی مشابه نسبت به گذشته و بر روی یک میکروکنترلر هشت بیتی استاندارد چیزی در حدود یک میلی‌ثانیه است.

امروزه مترجم‌های منطق فازی کدهای اسمبلی و یا C را از روی زبان برنامه‌سازی منطق فازی FTL و یا مستقیماً از محیط طراحی گرافیکی تولید می‌نمایند. درجه بالای بهینه‌سازی و تنظیم

ساختار اولیه مناسب ناشی از عملکرد این مترجم‌ها موجب بی‌نیاز شدن طراح از نوشتن کدهای بصورت دستی و حذف مراحل زائد و اضافی محاسبات در زمان اجرا می‌شود.

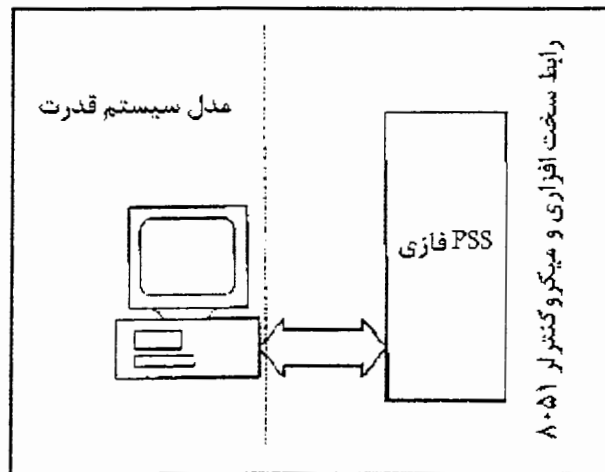
امروزه میکروکنترلرها نیز وارد میدان رقابت با پردازشگرهای فازی خاص شده‌اند با نگاهی به مدت زمان حلقه‌های تکرار در سیستم‌های کنترل مختلف می‌توان دید وسیعتری نسبت به کارایی میکروکنترلرها در ایجاد و پردازش قوانین فازی بدست آورد.

مدت زمان حلقه‌های تکرار در برخی از سیستم‌های کنترلی بر حسب نوع کاربرد آن بشرح زیر متغیر می‌باشد:

- پردازش‌های بیولوژیکی بین یک هزار ثانیه.
- صنایع شیمیایی بین یکصد ثانیه.
- دستگاههای بزرگ مکانیکی بین ۰/۱ الی یک ثانیه.
- سیستم‌های کوچک مکانیکی بین ۱۰ الی یکصد میلی ثانیه .
- مهندسی الکتریکی و الکترونیکی نظیر موتور الکتریکی و یا یک دیسک سخت کامپیوتر بین ۰/۱ الی یک میلی ثانیه

با توجه به مقایسه فوق در می‌یابیم که حتی میکروکنترلرهای هشت بیتی ارزان قیمت نیز قادر به اجرا و کنترل بسیاری از کاربردهای واقعی سیستم‌های منطق فازی است. کارایی میکروکنترلرهای ۱۶ بیتی امکان استفاده آنها را برای سیستم‌های پیچیده‌فازی که سرعت محاسباتی کمتر از یک میلی ثانیه احتیاج دارند، میسر می‌سازد.

از نقطه نظر کارایی سریعترین پردازشگرهای منطق فازی نمی‌تواند سریعتر از میکرو کنترلرها گردد زیرا حتی اگر سریعترین پردازشگرهای فازی هم ساخته شود باز یک راه حل نرم‌افزاری بر روی یک میکرو کنترلر می‌تواند برای بسیاری از کاربردهای عملی از سیستم‌های کنترل فازی راه حلی کارآمدتر و ارزانتر باشد.



شکل ۴-۳

#### ۴-۴ ساخت PSS فازی میکروکنترلری

سیستم قدرت یکی از سیستم بسیار گرانبهایی است که در جهان امروزی نقش بسزایی را به عهده دارد. هر گونه خسارت به این سیستم می‌تواند هزینه‌های بسیاری داشته باشد، لذا در مرحله تحقیق و مطالعه برای بررسی کاربردهای چنین کنترل کننده‌ای در سیستم قدرت از مدل سیستم قدرت که با Simulink در نرم‌افزار Matlab ایجاد شده، استفاده شده است. همچنانکه در شکل ۴-۳ مشخص شده مدل سیستم قدرت که در فصل دوم بدست آمد، داخل یک PC شبیه‌سازی شده و اطلاعات مورد نیاز برای PSS فازی را که همان زاویه و فرکانس سیستم است، بواسطه یک رابط سخت افزاری از طریق درگاه موازی PC (LPT) به میکروکنترلر ۸۰۵۱ ارسال می‌دارد، پس از پردازش این اطلاعات میکروکنترلر اندازه سیگنال کنترلی را به PC برمی‌گرداند.

#### ۴-۴-۱ ایجاد ارتباط با سخت افزار توسط Simulink

در مدل سازی سیستم قدرت بلوکی که به عنوان PSS سخت‌افزاری قرار دارد در حقیقت یک بلوک Sfunction است که توسط Mfile در Matlab نوشته شده است تا در هنگام شبیه‌سازی

اطلاعات مورد نیاز سخت‌افزار را به صورت مطمئنی از کامپیوتر به سخت‌افزار منتقل کند و همچنین مقدار برگشتی از سخت‌افزار را برای استفاده سیستم تحریک برگرداند.

Sfunctionها در قسمت Simulink برای تعریف بلوک‌های دلخواه کاربران در نظر گرفته شده است. نحوه کار بلوک‌ها در یک فایل Simulink بدین شکل است که، Matlab برای ارتباط بین بلوک‌های داخل یک فایل در این قسمت، flagهایی را در هر گام روشن می‌کند با این روش هر flag نماینده یک رخداد در Simulink می‌باشد. برای مثال  $flag=0$ ، مقادیر اولیه هر بلوک را آماده می‌کند و  $flag=3$ ، مقادیر خروجی بلوک را برای دیگر بلوک‌ها می‌فرستد. با این توصیف باید برای هر flag یک برنامه خاصی نوشته شود تا هنگامی که flag مورد نظر روشن می‌شود، کار مورد نیاز کاربر انجام پذیرد، باید توجه شود که Matlab توابع خاصی را برای هر flag به کار می‌برد، و تغییر نام این توابع منجر به اجرا نشدن بلوک ایجاد شده خواهد شد، برای اینکه اشتباهی صورت نگیرد Matlab در قسمت Simulink در جعبه ابزار Sfunction، Mfile آماده‌ای را با عنوان template قرار داده است. پس از ایجاد Mfile مربوطه باید نام تابع اولی که در بالای Mfile قرار دارد و نام خود Mfile باید یکسان باشند، و آنها را می‌توان به دلخواه نامگذاری کرد. و این نام باید در بلوک Sfunction در محلی که باید نام فایل مورد نظر بیاید وارد شود. با این توصیف به شرح برنامه بلوکی که ارتباط نرم‌افزاری بین Simulink و میکروکنترلر را فراهم می‌سازد می‌پردازیم.

نرم‌افزار Matlab جعبه ابزارهایی متنوعی در اختیار کاربران خود قرار می‌دهند که هر گونه شبیه‌سازی و حتی کنترل فرایند را میسر می‌سازد، یکی از این جعبه ابزارهایی که کمتر شناخته شده است جعبه ابزار Data Acquisition است. این جعبه ابزار امکان دسترسی به درگاههای مختلف کامپیوتر را فراهم می‌سازد. از توابعی که در این جعبه ابزار قرار دارند برای دستیابی به درگاه موازی استفاده شده است.

تابع `digitalio()` درگاه مورد نظر را برای ارسال یا دریافت اطلاعات دیجیتال اختصاص می‌دهد.

```
>> dio=digitalio('parallel')
```

Display Summary of DigitalIO (DIO) Object Using 'PC Parallel Port Hardware'.

Port Parameters: Port 0 is port configurable for reading and writing.

Port 1 is port configurable for reading.

Port 2 is port configurable for reading and writing.

Engine status: Engine not required.

DIO object contains no lines.

در گاه موازی با عنوان "Parallel" در Matlab شناخته می‌شود. این درگاه ۲۵ پایه دارد که ۸ تای آنها زمین می‌شوند، ۸ تای آنها با عنوان Port0 برای خواندن و نوشتن داده، ۵ تای آنها با عنوان Port1 برای خواندن داده از درگاه، و ۴ تای باقی مانده برای خواندن و نوشتن در درگاه موازی اختصاص یافته است. برای استفاده از این درگاه باید خطوطی را که داده در آنها منتقل می‌شود و همچنین نوع خط از نظر ورودی یا خروجی بودن داده، باید برای Matlab شناسانده شوند. تابع `addline()` برای این امر در نظر گرفته شده است. برای مثال در دستور زیر تمامی پایه‌های Port0 به صورت ورودی معرفی می‌شوند.

```
>> outputdata= addline(dio,0:7,0,'out')
Index: LineName: HwLine: Port: Direction:
1   'Pin2'  0    0   'Out'
2   'Pin3'  1    0   'Out'
3   'Pin4'  2    0   'Out'
4   'Pin5'  3    0   'Out'
5   'Pin6'  4    0   'Out'
6   'Pin7'  5    0   'Out'
7   'Pin8'  6    0   'Out'
8   'Pin9'  7    0   'Out'
```

و به همین ترتیب ۴ پایه از Port1 برای ورود داده‌ها در نظر گرفته می‌شوند.

```
>> inputdata= addline(dio,0:3,1,'in')
Index: LineName: HwLine: Port: Direction:
9   'Pin15' 0    1   'In'
10  'Pin13' 1    1   'In'
11  'Pin12' 2    1   'In'
12  'Pin10' 3    1   'In'
```

پس از اینکه دستورات بالا اجرا شدند Matlab با شماره Index این خطوط را خواهد شناخت. و برای خواندن و نوشتن داده باید از این شماره‌ها استفاده کرد. تابع putvalue() داده را در خطوط مورد نظر قرار می‌دهد. و تابع getvalue() داده را به صورت باینری از خطوط مورد نظر می‌گیرد.

```
>> putvalue(dio.line(1:8),100);
>> getvalue(dio.line(9:12))
ans =
    1    1    1    1
```

باید دقت کرد که این توابع با داده‌های صحیح مثبت کار می‌کنند، و ملاحظات لازم را برای اینکه ورودی این توابع اعداد منفی نشوند باید به عمل آورده شود. چون داده‌های مربوط به اطلاعات زاویه و فرکانس داده‌های اعشاری می‌باشند و همچنین داده‌های فرکانس شامل داده‌های منفی نیز می‌باشد. برای استفاده از این داده‌ها در این برنامه این مقادیر در ضرایب مشخصی ضرب شده و قسمت اعشاری آنها توسط تابع floor() حذف شده است. این امر جز اینکه باعث از دست رفتن دقت در محاسبات شود تأثیر دیگری در محاسبات فازی ندارد چون در محاسبات فازی درجه عضویت هر داده در محاسبات وارد می‌شود. پس با شیفت توابع تعلق که در سخت‌افزار ایجاد شده‌اند به بازه‌ای که این مقادیر همان درجه عضویت معمولی خود را ایجاد کنند، یا به عبارتی یک تبدیل مختصات خطی مشکل حل می‌شود.

در ضمیمه الف Mfile مربوط به این بلوک که برای ارتباط با میکروکنترلر از طریق درگاه موازی در نظر گرفته شده، آورده شده است. لازم به یادآوری است که خطوط ۱ تا ۸ در این Mfile با عنوان putdata برای فرستادن داده‌ها، خطوط ۹ تا ۱۲ با عنوان getdata برای گرفتن داده‌ها و خطوط ۱۳، ۱۴ و ۱۵ به ترتیب با عناوین RS, RW, AK برای ایجاد ارتباط Handshaking بین کامپیوتر و سخت افزار اختصاص یافته است. اطلاعات زاویه و فرکانس به ترتیب از طریق خطوط putdata در درگاه گذارده می‌شود سپس خطوط RS, RW را برای اعلام به سخت‌افزار به شکل مناسبی فعال می‌کند و منتظر سیگنال AK مبنی بر دریافت داده‌ها از سخت‌افزار می‌ماند، در جدول ۱-۴ مقادیر این خطوط برای رویدادهایی که بین کامپیوتر و سخت‌افزار روی می‌دهد آورده شده است. پس از اینکه



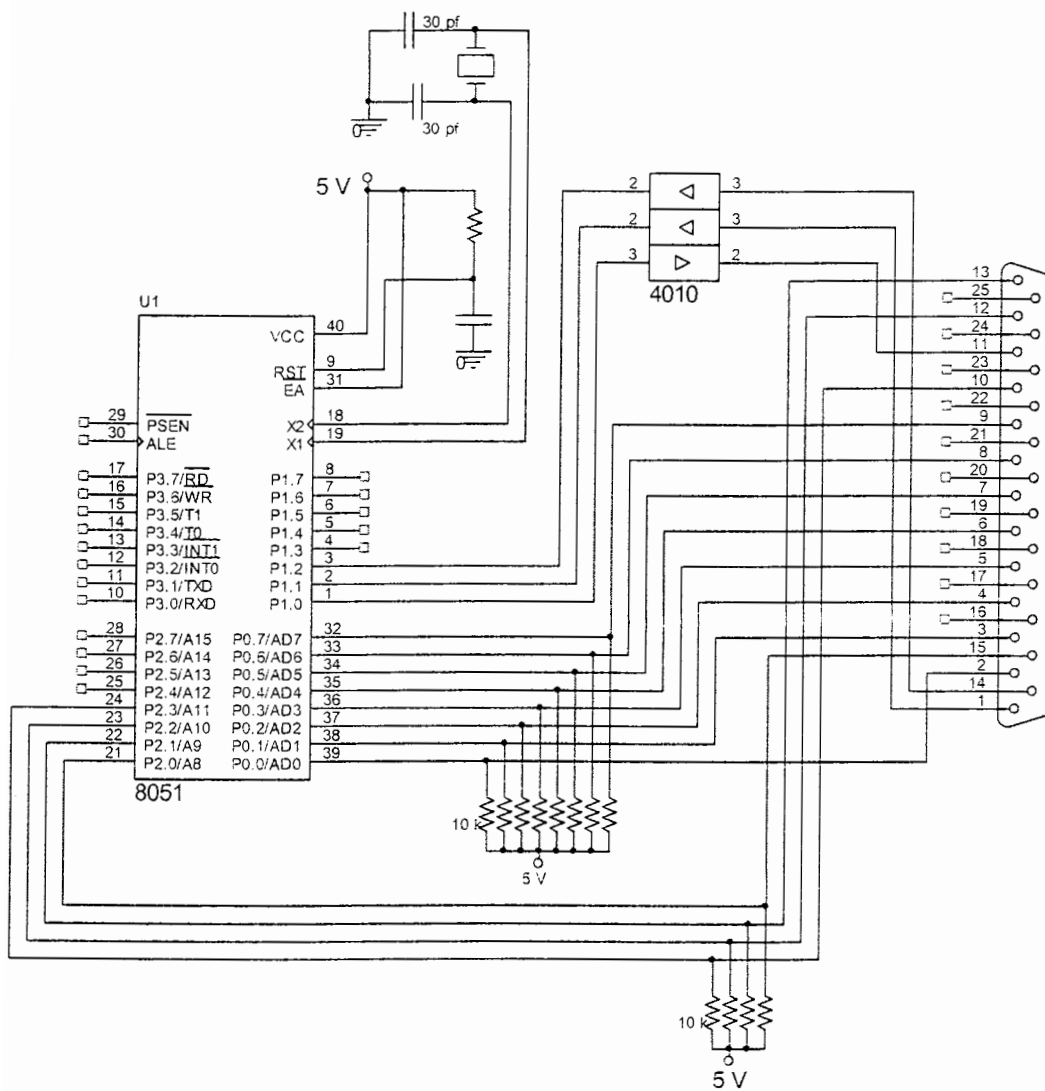
داده‌های مورد نظر به سخت افزار منتقل شد کامپیوتر منتظر سیگنال AK مبنی بر آماده بودن مقدار سیگنال محاسبه شده روی درگاه خواهد ماند، برخلاف اعلانی که Matlab پس از معرفی درگاه موازی به عنوان درگاه مورد نظر برای انتقال داده چاپ می‌کند، تنها Port1 قابلیت خوانده شدن دارد، و این به علت پشتیبانی نکردن تخته‌های مادر<sup>۱</sup> بسیاری از کامپیوترهای شخصی از این امکان است. برای همین تنها ۴ خط getdata برای دریافت داده ارسالی از سخت افزار در نظر گرفته شده است، به همین خاطر برای دریافت هشت بیت در دو مرحله اقدام می‌کنیم، برنامه سخت افزار نیز این امکان را بوجود آورده است که اطلاعات را ۴ بیتی به کامپیوتر ارسال دارد.

RW	RS	رویداد
0	0	داده اول در درگاه آماده است
0	1	داده دوم در درگاه آماده است.
1	0	آمادگی برای دریافت ۴ بیت اول اندازه سیگنال PSS
1	1	آمادگی برای دریافت ۴ بیت دوم اندازه سیگنال PSS

جدول ۴-۱ رویدادهایی که در حالت‌های مختلف RS, RW اجرا می‌شود.

#### ۴-۴-۲ شرح سخت افزار

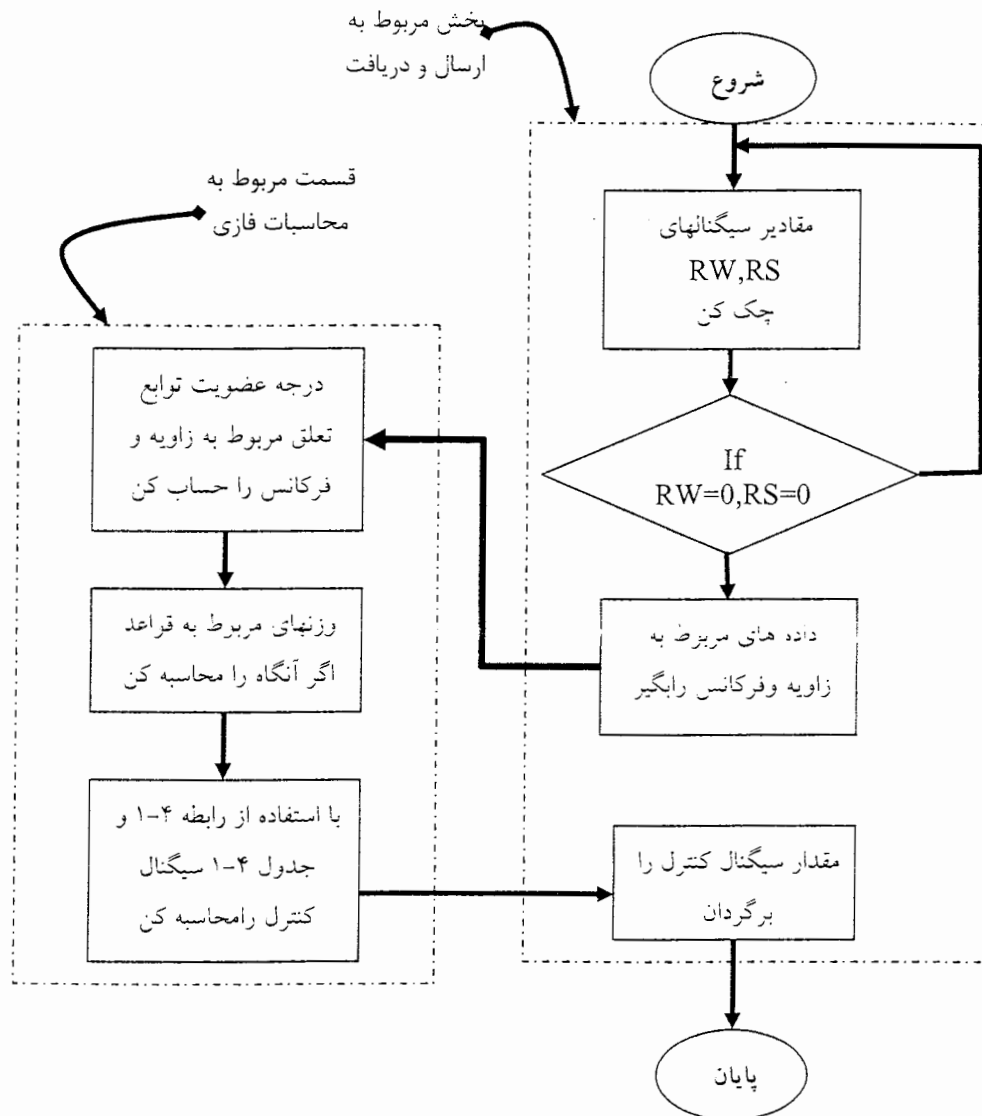
همچنانکه اشاره شد یک راهکار نرم‌افزاری بر روی میکروکنترلرهای ارزان قیمت در بسیاری مواقع کارآمدتر و به صرفه‌تر از پردازشگرهای فازی می‌باشد. برای ایجاد PSS فازی سخت افزاری در این پروژه از میکروکنترلر هشت بیتی ۸۰۵۱ استفاده گردیده است، این انتخاب به علت قیمت مناسب این قطعه و همچنین کوچکی و کم حجم بودن سخت‌افزار تهیه شده با آن که خود موجبات قابل حمل بودن آنرا فراهم می‌آورد، انتخاب شده است. امروزه کاربردهای صنعتی این قطعه به عنوان کنترل‌کننده برنامه‌پذیر فزونی یافته است. این میکروکنترلر ۴۰ پایه امکانات خوبی را برای ایجاد ارتباط بین کامپیوتر و سخت افزار در اختیار قرار می‌دهد به علاوه دارای ۴k بیت ROM و ۱۲۸ بیت RAM و چهار درگاه I/O هشت بیتی است این میکروکنترلر یک پردازنده بولی دارد که عملیات ضرب و تقسیم را در ۴ میکروثانیه انجام می‌دهد.



شکل ۴-۴ مدار سخت‌افزار رابط با کامپیوتر

مدار سخت افزار طراحی شده برای ارتباط با کامپیوتر در شکل ۴-۴ آمده است، خوشبختانه به علت استفاده در گاه موازی از استاندارد TTL نیاز به پیچیده شدن این سخت افزار نبود، و فقط برای سه سیگنال Handshaking از IC بافر استفاده شده است و بقیه پایه‌ها برای حصول اطمینان با یک pullup با اندازه ۱۰k به Vcc متصل شده‌اند. در این سخت افزار درگاه صفر میکرو کنترلر برای دریافت اطلاعات از کامپیوتر و ۴ پایه نخست درگاه ۲ برای ارسال خروجی اختصاص یافته است. پایه‌های P1.0، P1.1، P1.2 از درگاه ۱ به ترتیب برای سیگنال‌های AK، RS، RW در نظر گرفته شده است.

برنامه میکروکنترلر که با کدهای اسمبلی خاص میکروکنترلر ۸۰۵۱ نوشته شده، همانطور که از الگوریتم شکل ۴-۵ پیداست، شامل دو بخش کلی است: الف - دریافت و ارسال داده‌ها. ب - محاسبات فازی. از قبیل فازی سازی، استنتاج و غیرفازی سازی.



شکل ۴-۵ الگوریتم خلاصه شده مربوط به برنامه میکروکنترلر

دریافت و ارسال داده‌ها: همانطور که قبلاً ذکر شد میکروکنترلر برای دریافت و ارسال داده‌ها از درگاههای ۰ و ۲ استفاده می‌کند و همواره سیگنالهای RW, RS را چک می‌کند و پس از دریافت

داده‌ها آنها را در مکانهایی خاصی از حافظه ذخیره می‌کند تا در مواقع لزوم به آنها دسترسی داشته باشد. و سپس سیگنال AK را مبنی بر دریافت یا آمادگی برای ارسال داده فعال می‌کند. برای درک این موضوع به کد اسمبلی مربوط به دریافت و ذخیره سازی داده اول که فرکانس است و قسمتی از کد که مربوط به ارسال داده به کامپیوتر است، توجه کنید.

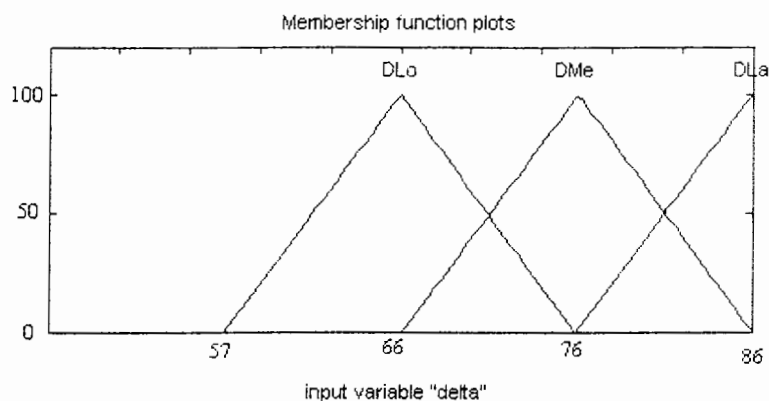
```

JB      RW,$      ;wait for RW signal to read data
JB      RS,$      ;wait for RS signal to get data 1
MOV     A,P0
MOV     R1,#40H
MOV     @R1,A
CLR     AK
.
.
.
JNB     RW,$      ;ready for send data
JB      RS,$      ;ready for put part1 of outdata ot p2
MOV     P2,A      ;put part1 of outdata
CLR     AK        ;AK for put data
SWAP   A          ;make par2 of outdata
JNB     RS,$      ;ready for put part2 of outdata ot p2
MOV     P2,A
SETB   AK

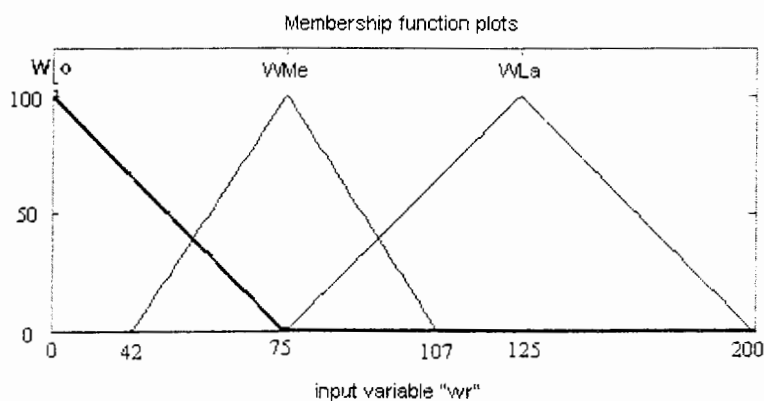
```

**محاسبات فازی:** این بخش تمامی محاسبات لازم برای فازی سازی، استنتاج و غیرفازی سازی را با استفاده از توابعی که با اسمبلی کد شده‌اند، انجام می‌دهد. این قسمت شامل توابعی است که درجه عضویت مقادیر زاویه و فرکانس را در لحظات مختلف محاسبه می‌کند و مقدار خروجی PSS را برحسب این دو داده با رابطه‌ای که با استفاده از سیستم فازی استفاده شده، بدست می‌آورد.

چون داده‌های مربوط به زاویه و فرکانس داده‌های اعشاری می‌باشند و این مقادیر برای میکروکنترلر معنایی ندارد توابع تعلق مربوط به زاویه و فرکانس که در شکل‌های ۳-۹ و ۳-۱۰ نشان داده شده‌اند، با یک تبدیل مختصات خطی به صورت شکل‌های ۴-۶ و ۴-۷ در می‌آیند. لازم به یاد آوری است برای حداقل کردن خطا در محاسبات، درجه عضویت هر عنصر در این توابع تعلق، عددی در بازه  $Z \in [0,100]$  انتخاب می‌شود.



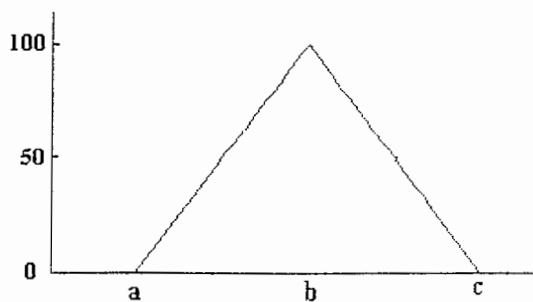
شکل ۴-۶ توابع تعلق تبدیل یافته ورودی زاویه



شکل ۴-۷ توابع تعلق تبدیل یافته ورودی فرکانس

```

if ((x<=a)|(x>=c))
    Memfun=0;
elseif (x>a)&(x<b)
    Memfun =100*(x-a)/(b-a);
else
    Memfun =100*(c-x)/(c-b);
end
    
```



شکل ۴-۸ نحوه ایجاد توابع تعلق در میکروکنترلر

برای درک نحوه ایجاد این توابع تعلق در اسمبلی با توجه به شکل ۴-۸ و کدی را که در Matlab نوشته شده، توجه کنید.

حال کد اسمبلی زیر را بررسی می‌کنیم. چون تمام توابع تعلق که در ایجاد کنترل کننده فازی استفاده شده است مثلثی می‌باشد. این کد اسمبلی پایه ایجاد تمامی توابع تعلق مورد نظر برای ورودی‌های فرکانس و زاویه است. ثابتهای BD,CD و AD به ترتیب مقادیر c, b و a را نمایش می‌دهند. در ابتدا این تابع مقدار متغیر مورد نظر را از مکان 40H حافظه می‌خواند، سپس مقدار را با CD مقایسه می‌کند در صورت بزرگتر بودن این به برچسب ZERODL می‌پرد و درجه عضویت صفر را در مکان حافظه 41H برای این متغیر در این تابع تعلق ذخیره می‌کند، در غیر اینصورت مقدار متغیر را با ثابت BD مقایسه می‌کند، در صورتی که بزرگتر از BD بود مقدار درجه در ناحیه‌ای که شیب آن منفی است محاسبه و در مکان حافظه ذکر شده قرار داده می‌شود، در غیر اینصورت مقدار متغیر با ثابت AD مقایسه می‌شود، اگر مقدار متغیر بزرگتر بود مقدار درجه عضویت در ناحیه‌ای که شیب آن مثبت است محاسبه می‌شود ولی در غیر اینصورت درجه عضویت صفر در مکان حافظه مذکور ذخیره خواهد شد. این مراحل در تمامی توابع تعلق مربوط به زاویه و فرکانس شبیه به یکدیگر انجام می‌شود و مقادیر درجه‌های عضویت در مکان‌های حافظه‌ای که به آنها اختصاص داده شده ذخیره می‌شوند.

```
;***** MEMBERSHIP FUNCTION
```

```
MFUNDL:
```

```
MOV R1,#40H
MOV A,@R1
MOV R1,A
CLR C
SUBB A,#CD
JNC ZERODL
MOV A,R1
CLR C
SUBB A,#BD
JC PRMEMDL
MOV A,#ED
```

```
CLR    C
SUBB   A,R1
MOV    R1,A
CLR    C
SUBB   A,#12
JC     CTNUJDL1
MOV    R7,#RDM
CALL   MATHFUN
SJMP   RETURNNDL
CTNUJDL1:MOV  A,R1
        MOV  B,#RDM
        MUL  AB
        MOV  B,#3
        DIV  AB
        JMP  RETURNNDL
PRMEMDL:MOV  A,R1
        CLR  C
        SUBB A,#BD
        JC   ZERODL
        MOV  R1,A
        CLR  C
        SUBB A,#12
        JC   CTNUJDL2
        MOV  R7,#RDM
        CALL MATHFUN
        SJMP RETURNNDL
CTNUJDL2:MOV  A,R1
        MOV  B,#RDM
        MUL  AB
        MOV  B,#3
        DIV  AB
        SJMP RETURNNDL
ZERODL: MOV  A,#0
RETURNNDL:
        MOV  R1,#41H
        MOV  @R1,A
        RET
```

پس از محاسبه درجه‌های عضویت مربوط به هر کدام از توابع تعلق نوبت به استنتاج و غیرفازی سازی این مقادیر می‌رسد ما برای اینکار و برای سادگی محاسبات و جلوگیری از پیچیده‌گی روابط مربوط به استنتاج از سیستم فازی TSK استفاده کرده‌ایم در این صورت مقدار سیگنال کنترل یا همان مقدار خروجی از میکرو کنترلر با رابطه زیر محاسبه می‌شود.

$$y = \frac{\sum_{l=1}^M y^l w_l}{\sum_{l=1}^M w_l} \quad 1-4$$

که در آن  $y^l$ ، از روی رابطه ۲-۳ و وزنه‌های  $w_l$  با رابطه زیر محاسبه می‌شوند.

$$w^l = \prod_{i=1}^n \mu_{c_i}(x_i)$$

در این صورت با توجه به قواعد اگر- آنگاه ۳-۱۳ تا ۳-۲۲ مربوط به طراحی سیستم فازی TSK و مقادیر  $y^l$  که در جدول ۳-۱ لیست شده اند. باید مقدار  $w^l, y$  محاسبه شوند اما چون محاسبات در میکروکنترلر با دقت پایینی انجام می‌شوند هرچه تعداد قواعد کمتر شود باعث کاهش محاسبات و کمتر شدن خطای محاسبات خواهد شد. با کمی سعی خطا و مشاهده قواعد دریافت شد که حذف بعضی از قواعد تأثیر چندانی در نحوه عملکرد پایدار ساز نمی‌گذارند، اما نبود این قواعد برنامه مربوط به آن را بسیار ساده می‌کند. بنابراین قواعد اصلاح شده به صورت زیر می‌باشد.

۲-۴ اگر ورودی فرکانس کم و ورودی زاویه متوسط، آنگاه خروجی ۰ است.

۳-۴ اگر ورودی فرکانس متوسط و ورودی زاویه متوسط ، آنگاه خروجی ۰/۰۴ است

۴-۴ اگر ورودی فرکانس متوسط و ورودی زاویه زیاد ، آنگاه خروجی ۰/۱ است.

۵-۴ اگر ورودی فرکانس زیاد و ورودی زاویه کم ، آنگاه خروجی ۰/۰۴ است.

۶-۴ اگر ورودی فرکانس زیاد و ورودی زاویه متوسط ، آنگاه خروجی ۰/۱ است

۷-۴ اگر ورودی فرکانس زیاد و ورودی زاویه زیاد ، آنگاه خروجی ۰/۲ است.



با توجه به قواعد بالا مقادیر  $y, w^l$  با استفاده از رابطه ۴-۱ محاسبه می‌شوند. همچنانکه از رابطه معلوم است، برای محاسبه خروجی کفایت مقادیر توابع تعلق را در هر نقطه بدست آورده و رابطه مربوط به  $y$  را محاسبه کنیم

در نهایت لازم به یادآوری است که در ایجاد کد مربوط به این محاسبات باید وسواس لازم را بکاربرد تا از وارد شدن خطای محاسبات جلوگیری شود، تا خطای حاصل از گرد کردن ورودی‌ها که از ابتدای ورود داده‌ها به میکروکنترلر در محاسبات وارد می‌شود و به این سادگیها قابل حذف نیست. باعث واگرا شدن خروجی نگردد. کد اسمبلی مربوط به میکروکنترلر در ضمیمه ب آورده شده است.

با توجه به عملیات‌هایی که سخت‌افزار برای پردازش داده‌ها انجام می‌دهد تا سیگنال کنترل را آماده سازد حداکثر زمانی که میکروکنترلر برای محاسبه سیگنال کنترل از دو ورودی مذکور استفاده می‌کند، بیش از ۰/۵ میلی‌ثانیه نخواهد بود و این زمان نسبت به ثابت زمانی سیستم قدرت که سیستم نسبتاً کندی بشمار می‌آید خیلی کم است و به راحتی می‌توان از این سخت‌افزار استفاده کرد، لازم به یادآوری است که هرچه تعداد توابع تعلق و همچنین قواعدی که برای محاسبه سیگنال کنترل بکار می‌روند بیشتر شود این زمان نیز به همان نسبت افزایش خواهد یافت، و در نتیجه در آن صورت نیاز به سخت‌افزار پیچیده‌تر و میکروکنترلر پیچیده‌تری خواهیم داشت.

## فصل پنجم

### مقایسه و تحلیل نتایج

## مقایسه و تحلیل نتایج شبیه‌سازی

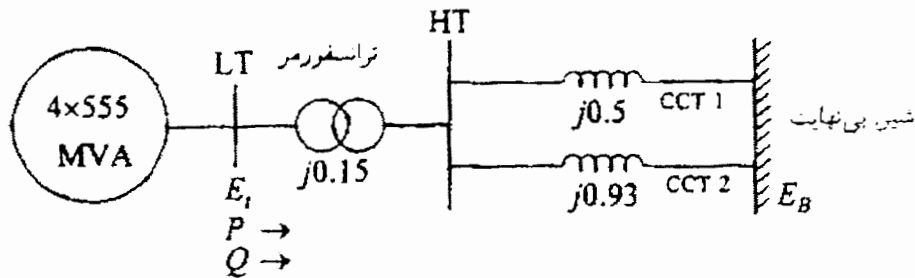
### مقدمه

در این فصل به بررسی نتایج بدست آمده از حالات مختلف شبیه‌سازی شده می‌پردازیم. همچنانکه در فصول قبل دیدیم، با بدست آوردن شناخت کلی در مورد نحوه عملکرد پایدارساز سیستم قدرت کلاسیک در مواجهه با اغتشاشهای محتمل در سیستم موفق به طراحی پایدارسازهای فازی به روش‌های ممدانی و سوگنؤ شدیم. و با طراحی پایدارساز فازی سخت‌افزاری، مدل پایدارساز را از شبیه‌سازی حذف کردیم و از سخت افزار طراحی‌شده به عنوان پایدارساز استفاده کردیم، و با مقایسه‌هایی که در این فصل انجام می‌دهیم کارایی پایدارسازهای طراحی شده در حالت‌های مختلف بررسی خواهیم کرد.

### ۱-۵ توصیف سیستم قدرت استفاده شده:

در بخش‌های مختلف این پایان‌نامه برای نشان دادن نتایج اعمال پایدارسازهای طراحی شده در از مدل سیستم قدرت تک ماشینه متصل به شین بی‌نهایت که در شکل ۱-۵ نشان داده شده است استفاده کرده‌ایم. این سیستم از مرجع [۲۶] گرفته شده است. این سیستم متشکل از چهار واحد

۵۵۵MVA، ۲۴KV، ۶۰Hz است و توان تولیدی از طریق دو مدار انتقال به یک شین بی‌نهایت منتقل می‌شود.



شکل ۱-۵ سیستم تک ماشینه متصل به شین بی‌نهایت

### ۱-۱-۵ پارامترهای ژنراتور

چهار ژنراتور نیروگاه به شکل ژنراتوری معادل که پارامترهای آن بر پایه ۲۲۲۰MVA در مبنای واحد بیان شده است در زیر آورده شده است:

$x_d = 1/81$	$x_q = 1/76$	$x'_d = 0/30$	$x'_q = 0/65$
$x''_d = 0/23$	$x''_q = 0/25$	$x_f = 0/15$	$R_a = 0/003$
$T'_{d0} = 8/0$ (ثانیه)	$T'_{q0} = 1/0$ (ثانیه)	$T''_{d0} = 0/03$ (ثانیه)	$T''_{q0} = 0/07$ (ثانیه)
$H = 3/5$	$k_D = 0$		

### ۲-۱-۵ پارامترهای سیستم تحریک

همان‌طور که در شکل ۲-۵ نشان داده شده است، ژنراتورها با تحریک کننده‌های تریستوری با AVR دارای پارامترهای زیر تجهیز شده‌اند:

$$k_A = 200 \quad T_R = 0.15 \text{ (ثانیه)} \quad E_{F \max} = 0.7 \quad E_{F \min} = -0.4$$

پارامترهای پایدارسازهای سیستم قدرت:

پایدارساز کلاسیک از پارامترهای زیر استفاده می‌کند.

$$k_{STAB} = 9/5 \quad T_{IV} = 1/41 \text{ (ثانیه)} \quad T_1 = 0.154 \text{ (ثانیه)} \quad T_2 = 0.03 \text{ (ثانیه)}$$

$$v_{s \max} = 0.2 \quad v_{s \min} = -0.2$$

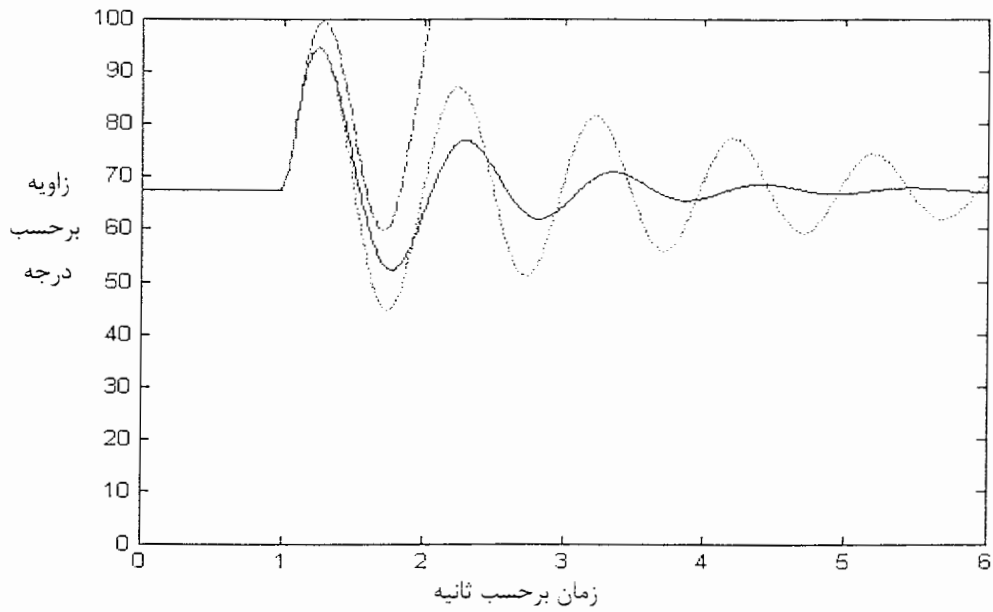
پایدارساز فازی ممدانی از قواعد ۳-۴ تا ۳-۱۵ و توابع تعلق شکل‌های ۳-۹، ۳-۱۰ و ۳-۱۱ را به ترتیب برای ورودی‌های فرکانس و زاویه و خروجی پایدارساز بکار می‌گیرد.

پایدارساز فازی سوگنو از قواعد ۳-۱۶ تا ۳-۲۷ و توابع تعلق شکل‌های ۳-۹ و ۳-۱۰ به ترتیب برای ورودی‌های فرکانس و زاویه و ضرایب جدول ۳-۱ را به عنوان  $\gamma$  بکار می‌برد.

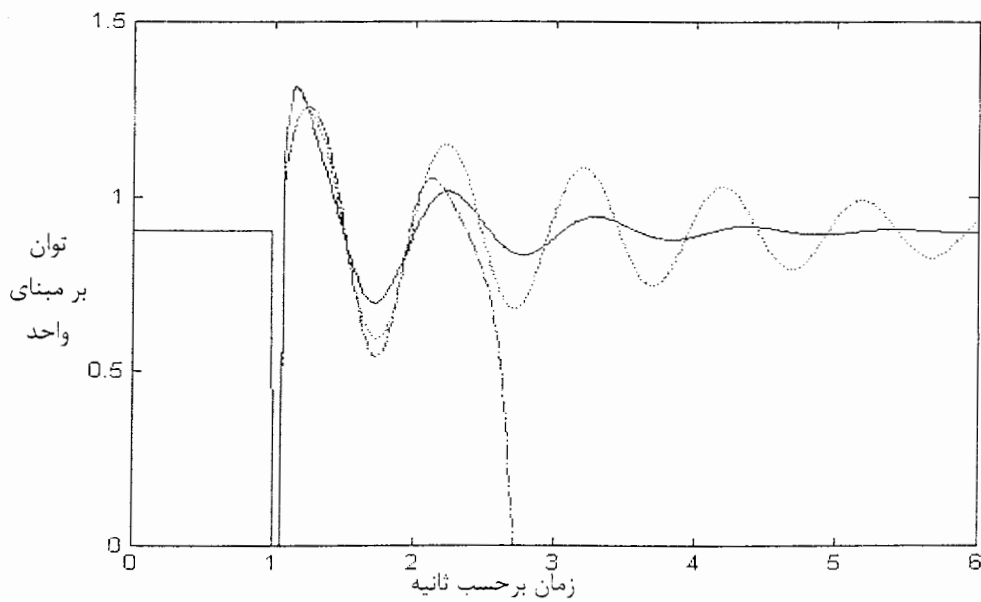
و سرانجام پایدارساز فازی سخت‌افزاری به ترتیبی که در فصل ۴ شرح آن رفت استفاده خواهد شد.

## ۲-۵ شبیه‌سازی‌های انجام شده

نتایج پاسخهای زمانی شبیه‌سازی شده با سه کنترل تحریک متفاوت، بدون AVR و PSS، با AVR و بدون PSS و هم با وجود هر دو در شکل‌های ۵-۲ و ۵-۳ نشان داده شده است که به ترتیب پاسخهای زمانی زاویه و توان الکتریکی می‌باشند.



شکل ۵-۲ پاسخ زاویه روتور به خطای رفع شده در ۰/۰۷ ثانیه



شکل ۵-۳ پاسخ توان الکتریکی به خطای رفع شده در ۰/۰۷ ثانیه

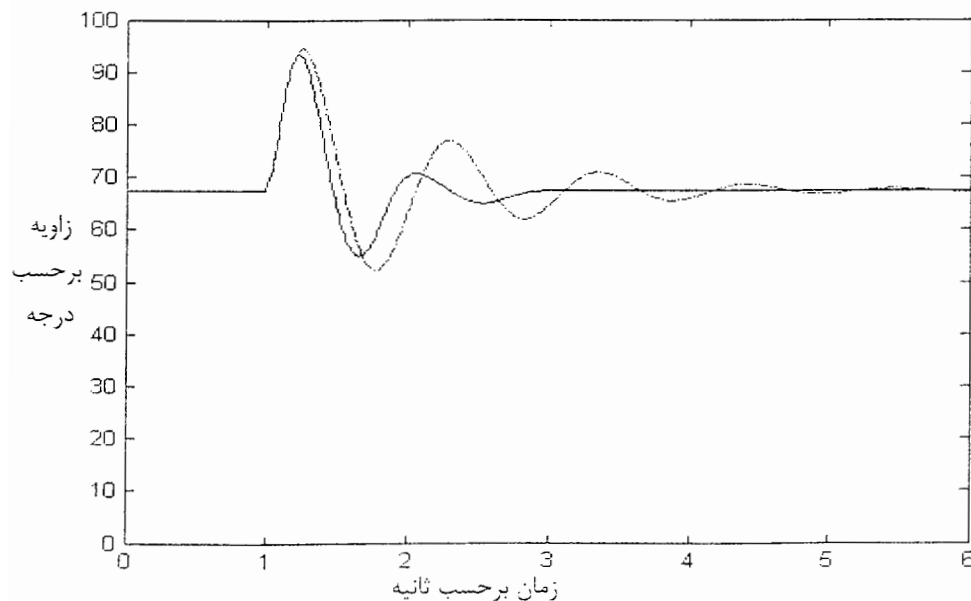
در دو شکل بالا پاسخ مربوط به بدون AVR و PSS با “—” و پاسخ مربوط به بدون PSS با “...” و پاسخ مربوط به حضور هر دو با “—” نمایش داده شده‌اند.

از شکل ۵-۲ مشاهده می‌کنیم که سیستم بدون PSS پایداری گذرا دارد ولی میزان میرایی نوسانهای آن کم است، ولی بدون AVR و یک ولتاژ سقف بالای تحریک کننده سیستم پس از اولین نوسان به علت داشتن میرایی منفی نمی‌تواند به حالت پایدار خود برگردد.

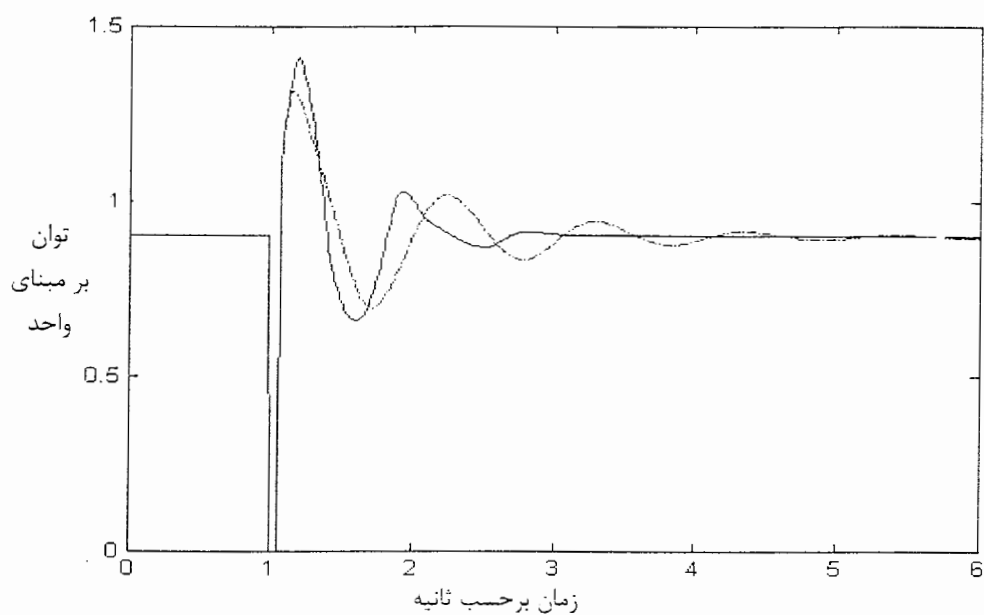
همچنانکه مشاهده می‌شود، با استفاده از PSS نوسانهای روتور بخوبی میرا می‌شوند.

این مشاهدات را شکل ۵-۳ نیز تصدیق می‌کند، همچنانکه در این شکل دیده می‌شود، سیستم بدون AVR و PSS پس از اغتشاش وارده بشدت افت توان پیدا می‌کند و پس از دومین نوسان به علت نداشتن میرایی مناسب از عملکرد طبیعی خارج می‌شود. در حالی که در حضور هر دو به شکل قابل قبولی به خطا پاسخ می‌دهد.

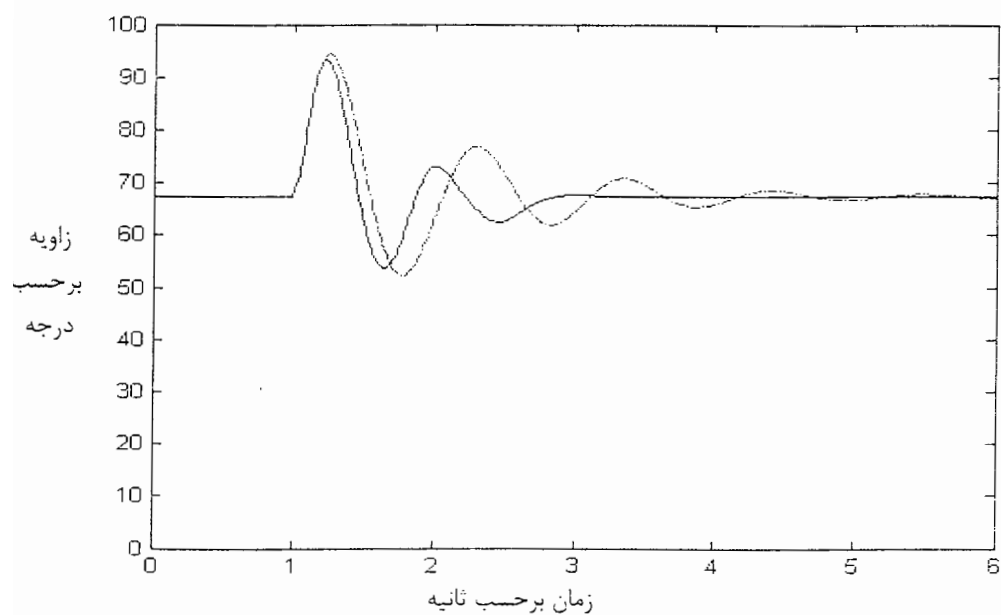
شکل ۵-۶، ۵-۷ و ۵-۸ پاسخ‌های زمانی زاویه مربوط به رفع خطا در ۰/۰۷ ثانیه را با استفاده از سه پایدارساز فازی ممدانی، سوگنؤ و سخت افزاری در مقایسه با پایدارساز کلاسیک نمایش می‌دهند. در همه این شکل‌ها پایدارساز کلاسیک با " - " و بقیه با " — " نمایش داده شده‌اند.



شکل ۵-۶- الف پاسخ زاویه به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک و ممدانی

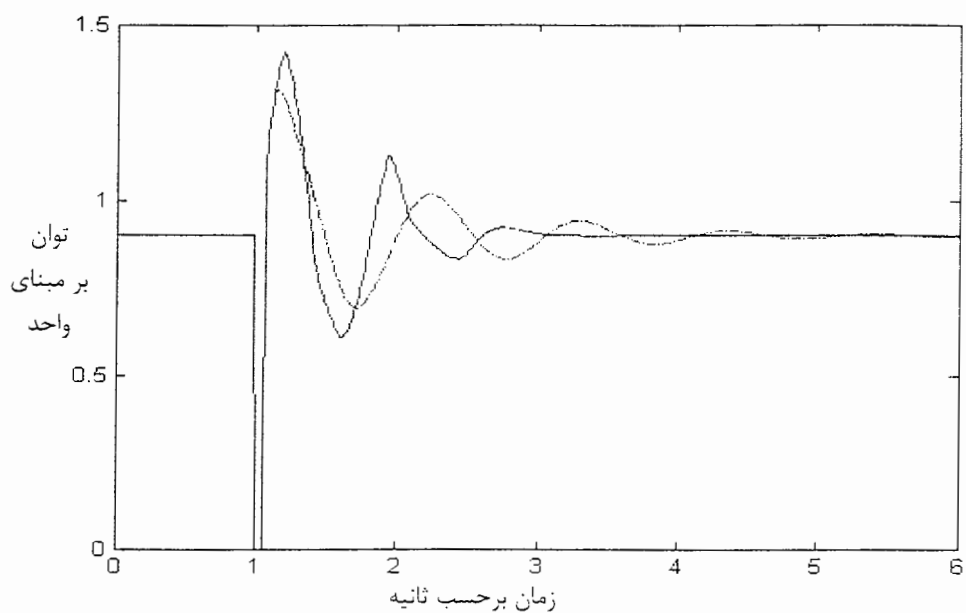


شکل ۵-۶-ب پاسخ توان الکتریکی به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک و ممدانی

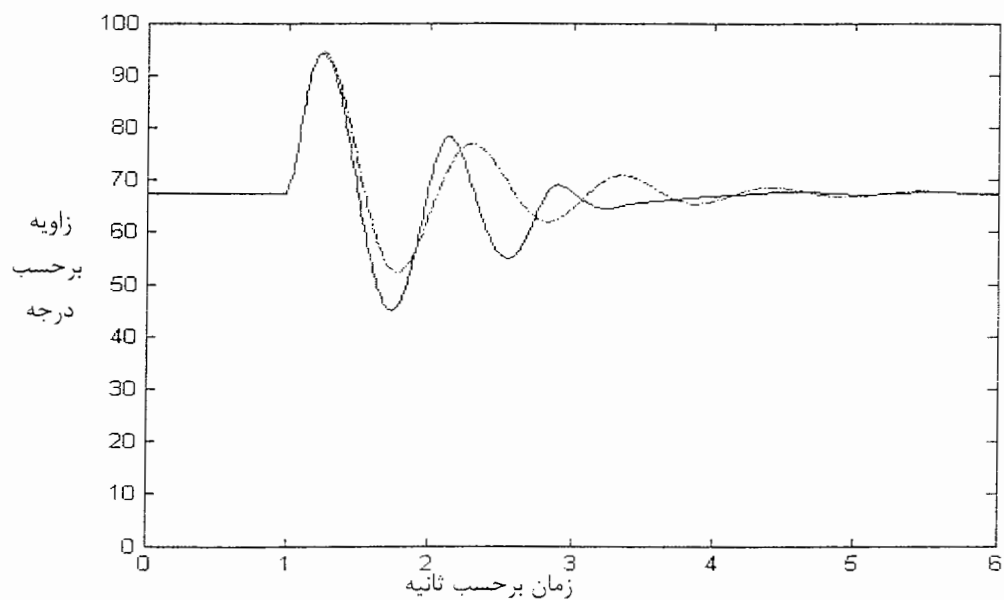


شکل ۵-۷-الف پاسخ زاویه به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک و سوگنؤ

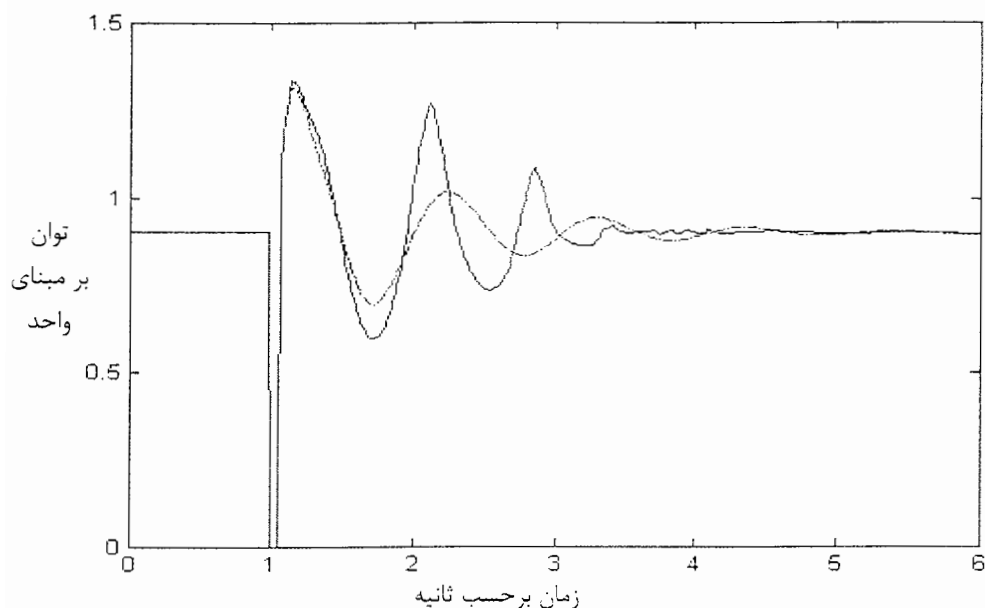




شکل ۵-۷-ب پاسخ توان الکتریکی به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک و سوگنؤ



شکل ۵-۸-الف پاسخ زاویه به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک و سخت‌افزاری

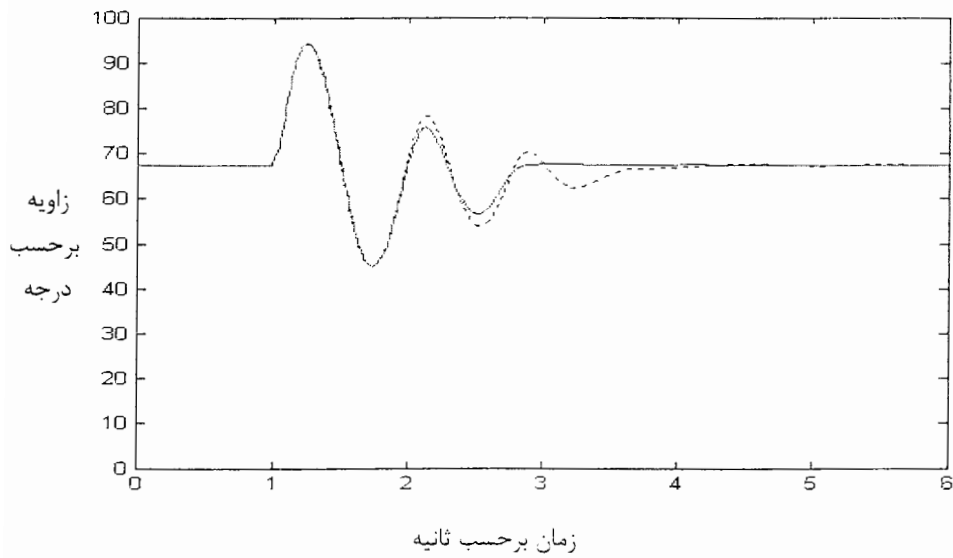


شکل ۵-۸-ب پاسخ توان الکتریکی به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSS کلاسیک وسخت‌افزاری

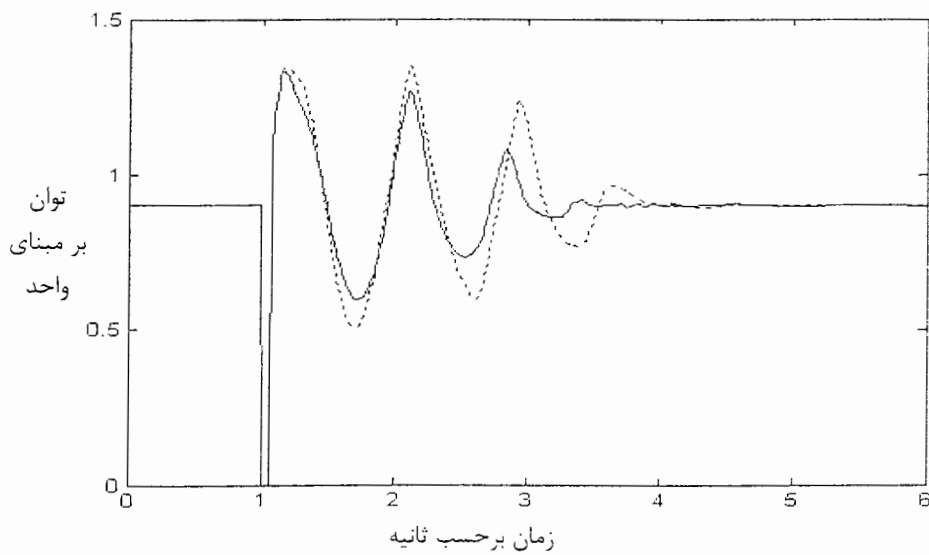
از شکل‌های ۵-۶، ۵-۷ و ۵-۸ مشاهده می‌کنیم که پاسخهای زمانی بدست آمده با استفاده از پایدارسازهای فازی در مواجهه با این خطا به شکل نسبتاً بهتری عملکرده و زودتر میرایی لازم را به سیستم می‌دهد. همچنانکه مشاهده می‌شود پایدارساز سخت‌افزاری نیز به شکل نسبتاً مطلوبی میرایی را به سیستم باز می‌گرداند. گرچه پایدارسازهای ممدانی و سوگنؤ فازی که در کامپیوتر شبیه‌سازی شده‌اند، نسبتاً بهتر عمل کرده‌اند، و با اینکه در طول برنامه نویسی میکروکنترلر سعی در پخش نشدن خطا شده‌است با این همه و با در نظر گرفتن امکانات پردازش میکروکنترلر ۸۰۵۱ و خطای وارد شده در قسمت ارسال داده‌ها و همچنین خطاهای بوجود آمده در قسمت محاسبات فازی این پاسخ قابل قبول می‌باشد. با این حال پاسخ زمانی سخت‌افزار نسبت به سوگنؤ اصلاح‌شده که به علت پرهیز از نداشتن اعداد منفی در سیگنال خروجی میکروکنترلر در قسمت ۴-۴-۲ اجباراً به اصلاح آن اقدام شد نسبتاً بهتر می‌باشد.

مقایسه پاسخ‌های زمانی مربوط به زاویه و توان الکتریکی در حالتی که از فازی سوگنؤ اصلاح‌شده استفاده شده و در حالتی که از سخت‌افزار استفاده شده در شکل‌های ۵-۹ و ۵-۱۰ آورده شده است. در این شکل‌ها نموداری که با "..." نشان داده شده مربوط به حالتی است که از پایدارساز

فازی سوگنژ اصلاح‌شده استفاده گردیده است و نمودار مربوط به پایدارساز سخت‌افزاری با “—” نمایش داده شده است.

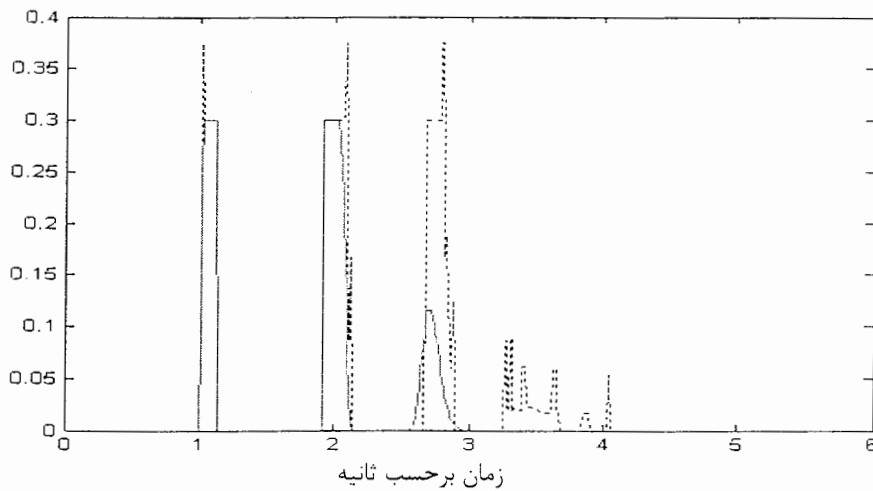


شکل ۵-۹ پاسخ زاویه به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSSهای فازی سوگنژ اصلاح شده و سخت‌افزاری



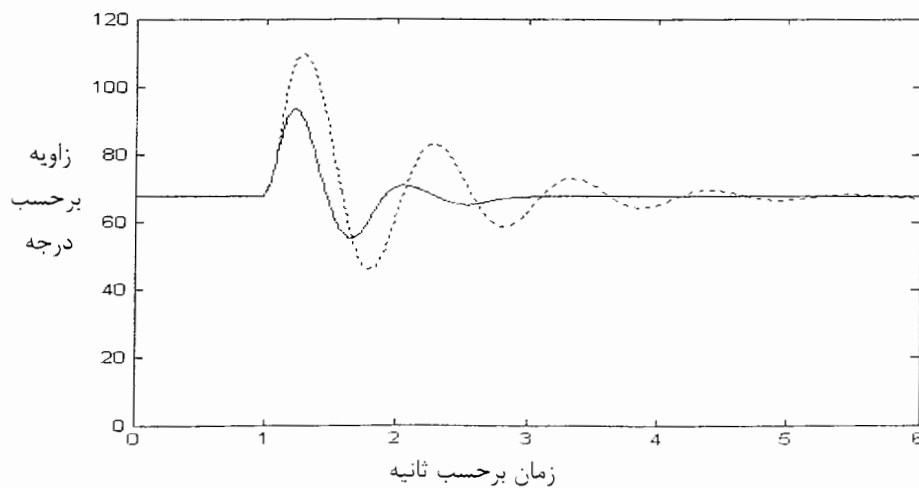
شکل ۵-۱۰ پاسخ توان الکتریکی به خطای رفع شده در ۰/۰۷ ثانیه با استفاده از PSSهای فازی سوگنژ اصلاح شده و سخت‌افزاری

در شکل ۵-۱۱ سیگنال کنترل دو پایدارساز سوگنو اصلاح شده و پایدارساز میکروکنترلی آورده شده است.

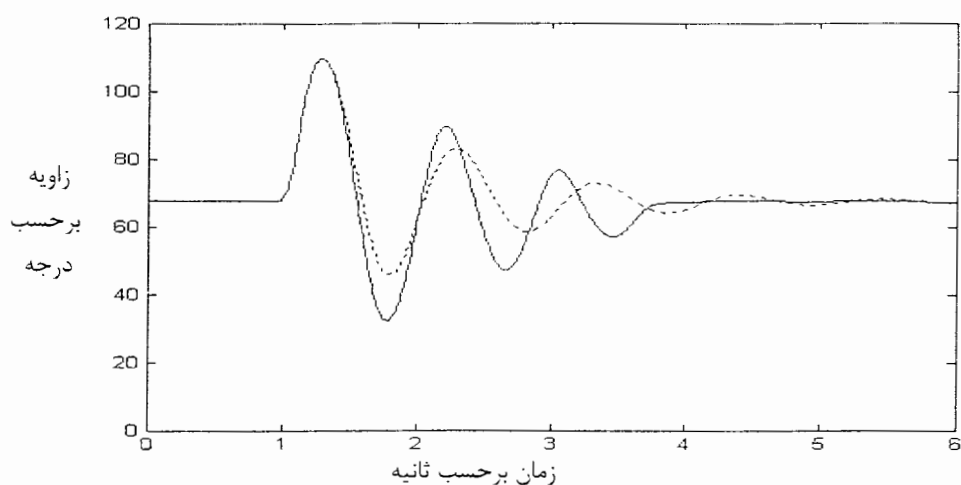


شکل ۵-۱۱ سیگنال‌های کنترل پایدارساز سخت‌افزاری "\_\_\_" و پایدارساز سوگنو اصلاح‌شده "..."

در شکل های ۵-۱۲ و ۵-۱۳ پاسخ زمانی زاویه پایدارسازهای ممدانی و سخت افزاری در مقایسه با PSS کلاسیک به خطای رفع شده در ۰/۱ ثانیه نشان داده شده است. در این شکل‌ها نمودار مربوط به PSS کلاسیک با "..." نشان داده شده است.

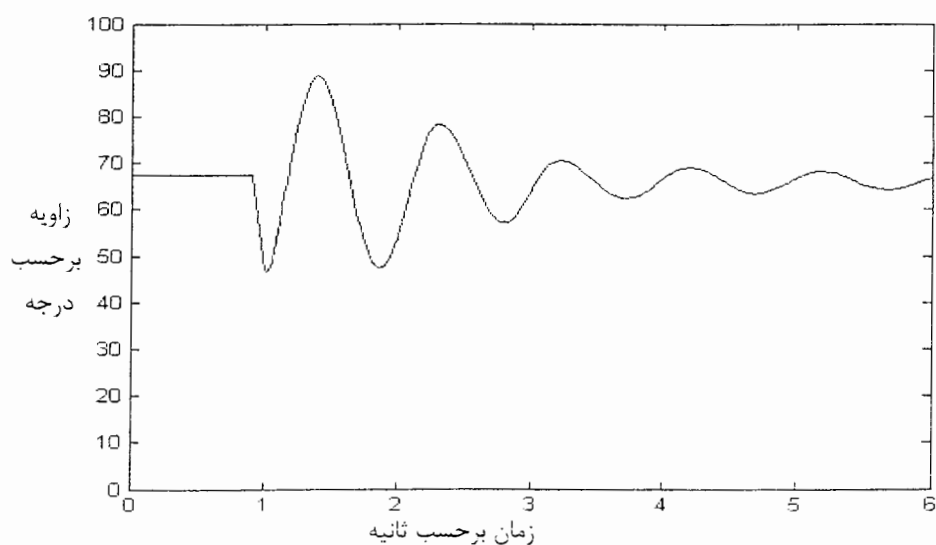


شکل ۵-۱۲- الف پاسخ زاویه به خطای رفع شده در ۰/۱ ثانیه با استفاده از PSS های کلاسیک و ممدانی



شکل ۵-۱۳-الف پاسخ زاویه به خطای رفع شده ۰/۱ ثانیه با استفاده از PSSهای کلاسیک و سخت‌افزاری

پایدارساز سخت‌افزاری به زمان نمونه برداری نیز حساس می باشد، طبق مشاهداتی که انجام شده است زمان نمونه برداری بیشتر از ۰/۰۱ ثانیه پاسخ مطلوبی را به همراه نخواهد داشت. تمامی شکل‌های نشان داده شده نیز با این زمان نمونه برداری بدست آمده‌اند. در شکل ۵-۱۴ پاسخ زمانی زاویه برای خطای رفع شده در ۰/۰۷ ثانیه با زمان نمونه برداری ۰/۱ نشان داده شده است.



شکل ۵-۱۴ پاسخ زاویه پایدارساز سخت‌افزاری به خطای رفع شده در ۰/۰۷ ثانیه با زمان نمونه برداری ۰/۱ ثانیه

همچنانکه از شکل‌های بالا بر می‌آید پاسخ سخت افزار به پارامترهایی از قبیل خطای داده ها و زمان نمونه برداری که آن هم به حجم کدهای نوشته شده برای میکروکنترلر دارد بستگی دارد. بنابراین هرچه تعداد قوانینی که در ایجاد پایدارساز فازی دخیل هستند کمتر باشند، به همان نسبت برنامه سریعتر شده و همچنین خطای انجام محاسبات فازی نیز کاهش خواهد یافت و در نتیجه زمان نمونه برداری سریعتری را لازم خواهد داشت.

با این همه پاسخ‌هایی که سخت افزار در آن شرکت داشته پاسخ‌های نسبتاً قابل قبولی هستند و ما مجبور به مصالحه این چینی هستیم که برای داشتن زمان نمونه‌برداری کمتر و خطای کمتر به جای پایدارساز سوگنو از پایدارساز سوگنو اصلاح‌شده برای برنامه نویسی میکروکنترلر استفاده کنیم.

## فصل ششم

**نتیجه‌گیری**

**و**

**پیشنهادات**

## نتیجه گیری و پیشنهادها

### ۶-۱ نتیجه گیری

در این پایان نامه با بررسی انواع پایداری های گذرا و سیگنال کوچک در تحلیل سیستم های قدرت، با پایدار سازهای سیستم قدرت به عنوان کنترل کننده ای که یک سیگنال اضافی همفاز با تغییرات سرعت روتور را برای فراهم کردن میرایی لازم نوسانات روتور ژنراتور توسط کنترل تحریک ژنراتور به سیستم اعمال می کند، آشنا شدیم. و انواع کلاسیک این کنترل کننده را که در واحدهای تولید برق هم اکنون استفاده می شوند، بررسی کردیم.

سپس به مدلسازی ریاضی سیستم قدرت تک ماشینه پرداختیم و معادلات حاکم با سه متغیر حالت برای یک سیستم تک ماشینه متصل به شین بی نهایت بدست آوردیم، این معادلات حاکی از این است که این سیستم بسیار غیرخطی بوده و عملکرد آن تحت تأثیر اجزاء مختلف آن قرار دارد که بدون داشتن یک آگاهی کافی از معادلات و عملکرد اجزاء آن نمی توان به تحلیل این سیستم پرداخت. و منطق فازی را به



عنوان راه حلی نیمه هوشمند که می‌تواند از دانش فرد خبره برای تحلیل و استنتاج مسائل بهره گیرد معرفی شد. این روش راه حلی ساده و مفید را پیش پای همگان قرار می‌دهد تا با استفاده از تجربیات خویش در زمینه کنترل و رفتار سیستم‌ها بدون درگیر شدن در معادلات ریاضی سیستم و یا فرایند تحت کنترل به کنترل و بهره‌گیری از آن سیستم دست پیداکنند. در ادامه با بررسی انواع سیستم‌های فازی دو کنترل کننده فازی را با روشهای ممدانی و سوگنو طراحی کردیم، با مقایسه کنترل کننده‌های فازی و پایدارساز کلاسیک مشاهده شد که این کنترل کننده‌ها پاسخ نسبتاً مناسب‌تری را در مواجهه به خطا تولید کرده و میرایی نوسانات روتور را بیشتر می‌کنند.

در ادامه با اصلاح قواعد اگر- آنگاه پایدارساز فازی سوگنو این پایدارساز فازی را به علت ساده‌گی محاسبات در مقایسه با همتای ممدانی آن برای نوشتن برنامه سخت‌افزاری گزیدیم، و به ترتیبی که در فصل ۴ توضیح داده‌شد، برنامه‌ای که ارتباط سخت‌افزاری و هم محاسبات فازی را به شکل نسبتاً قابل قبولی ارائه دهد تهیه شد، این ارتباط سخت‌افزاری و انجام محاسبات مربوط به سیستم فازی توسط یک میکروکنترلر ۸۰۵۱ انجام می‌شود، که کاربردهای صنعتی بسیاری به علت ارزانی و امکانات ارزنده‌ای که در کارهای کنترل صنعتی ارائه می‌دهد، پیدا کرده است. با نتایجی که در شبیه‌سازی سیستم قدرت با این پایدارساز بدست آمد، مشاهده شد که این پایدارساز نیز به شکل نسبتاً مطلوبی میرایی نوسانات روتور ژنراتور را فراهم می‌کند، گرچه این میکروکنترلر پردازنده بولی نسبتاً خوبی دارد ولی به علت خطاهایی که به علت ضعف میکروکنترلر در محاسبات ریاضیاتی دارد، پاسخ‌های گذرای سیستم قدرت در هنگام وقوع خطا با این پایدارساز با مقداری اختلاف نسبت به مشابه طراحی شده آن در Matlab بدست آمد. با اینحال همانطور که اشاره شد این پایدارساز نیز قادر است به میزان قابل قبولی میرایی نوسانات روتور را افزایش دهد.

## ۲-۶ پیشنهادها

همچنانکه اشاره شد، ضعف میکروکنترلر ۸۰۵۱ در پردازش محاسبات ریاضی باعث وارد شدن خطا در محاسبات فازی گردید، پیشنهاد می‌شود با انتخاب یک میکروکنترلر مناسب‌تر که امکان محاسبات ریاضی بیشتری را ارائه دهد، توان ایجاد سیستم‌های فازی میکروکنترلری بررسی و مطالعه شوند.

در این پایان‌نامه به ساخت یک پردازشگر فازی میکروکنترلری دست پیدا کردیم که با ارتباط با کامپیوتر توانست به شکل نسبتاً مطلوبی پایداری گذرا را برای سیستم شبیه‌سازی شده در داخل کامپیوتر فراهم کند، و باعث افزایش میرایی نوسانات آن شود، با این همه این‌که این‌جا چنین پایداری در عمل نیز به همین شکل پاسخ دهد در پرده ابهام است، پیشنهاد می‌شود، این مسئله با تعریف پروژه‌ای که به بررسی این نوع پایداری روی یک ژنراتور عملی پردازد، بررسی گردد.

همچنانکه در فصل ۴ اشاره شد با افزایش سرعت پردازش میکروکنترلر و تعداد بیت‌های کاری آن توان محاسباتی این نوع کنترل‌کننده‌ها نیز بالا می‌رود، پیشنهاد می‌شود ساخت اینچنین پایداری با یک میکروکنترلر ۱۶ بیتی نیز آزمایش شود.

## مراجع و ماخذ

- [1] C.P. Steinmetz, "Power Control and Stability of Electric Generating Station", AIEETrans, Vol. XXXIX, part II, pp 1215, July-December 1920.
- [2] Y.N. YU, "electric Power System Dynamics", Academic Press, New York, 1983.
- [3] P. M. Anderson and A. A. Fouad , " Power System Control and Stability", Iowa StateUniversity Press, Ames, Iowa, 1977.
- [4] E.V. Larsen, D.A. Swann "Applying Power System Stailizers Part I: Genral Concepts" IEEE. Transactions on Power Apparatus and Systems. Vol. PAS-100, No. 6 June 1981.
- [5] E.V. Larsen, D.A. Swann "Applying Power System Stailizers Part II: Performance Objectives And Tuning Concepts" IEEE. Transactions on Power Apparatus and Systems. Vol. PAS-100, No. 6 June 1981.
- [6] E.V. Larsen, D.A. Swann "Applying Power System Stailizers Part III: Practical Considerations" IEEE Transactions on Power Apparatus and Systems. Vol. PAS-100, No. 6 June 1981.
- [7] F.P. deMello, J.S. Czuba, P.A. Ruche, And J. Willis, "Developments in Application of Stabilizing Measures tHrough Excitation Control" GIGRE Paper 38-05, 1986.
- [8] Constantin Von Altrok " Fuzzy Logic and NeuroFuzzy Application Explained" Prentice Hall 1995.
- [9] Y.N. YU, C.Siggers "Stabilitzation and Optimal Control Signals for a Power System"IEEE. Trans. PAS. PP 1469-1481, July. 1971.
- [10] J.H. Anderson," The Control of a Synchronous Machine Using Optimal control Thory,"Proc. of the IEEE, Vol. 59, PP. 25-35,Jan. 1971.
- [11] Asgharian, R. and McDonald, D.C. "The Design of Turbine Generators OptimalController Including the Effect of Tursional Modes of Osilations," IEEE Trans. EnergyConversion, Vol.3, No.2, PP. 230-234, 1988.
- [12] F. P. de Mello, P. J. Nolan, T. F. Laskowski and J. M. Undrill "Cordinated Application ofStabilizers in Power System" IEEE. Trans. PAS. Vol. 99, PP 892-901, may. 1980.
- [13] J.V.Milanovic and I.A.Hiskens,"Effects of Load Dynamics on Power SystemDamping",IEEE Trans. Power systems, Vol.10, No.2, May 1995.
- [14] Asgharian, R. and Hashemi, A.A." Robust Control Design of a Power System StabilizerUsing $\mu$ Syntheses," Transmitted to IEEE/Pes, 1996, Winter Meting.

- [15] Sanai S.A. "Studies on Robust Control for Power System Stabilization," Doctoral Dissertation, Submitted to the Department of Electrical Engineering Waseda University, March 1997.
- [16] M. K. EL-Sherbiny and et al, "Efficient Incremental Fuzzy Logic Controller For Power System Stabilizer," EMPS. 25:429-441, 1997.
- [17] Y-Y. Hsa and et al, "A Fuzzy Controller for Generator Excitation Control", IEEE Transactions on Systems, Man, Cybernetics, Vol.23, No.2, 1993.
- [18] P. M. Anderson and A. A. Foad, Power Systems Control and Stability, Iowa State University Press, Ames, IA, 1997.
- [20] P. Kundur and P.L. Dandeno, " Practical Application of Eigenvalue Techniques in the Analysis of Power System Dynamic Stability Problems" ,Proc. Fifth Power System Computation Conference, Cambridge, England, September 1975.
- [21] M.L. Crenshaw and etc "Excitation System Models For Power System Stability Studies " IEEE Transactions on Power Apparatus and Systems. Vol. PAS-100, No. 2 February 1981.
- [22] F. P. de Mello, L.N. Mannett, J.M. Undrill, "Practical Approaches to Supplementary Stabilizing from Accelerating Power" IEEE. Trans. Vol. PAS-97, PP 1515-1522, September/October 1978.
- [23] P. Kundur, D.C. Lee, H.M. Zein El-Din, " Power System Stabilizers for Thermal Units: Analytical Techniques and on-Site Validation" ,Paper F80-227-9 presented at IEEE PES Winter Meeting, New York, February 1980.
- [24] P. Kundur and P.L. Dandeno, " Practical Application of Eigenvalue Techniques in the Analysis of Power System Dynamic Stability Problems" ,Proc. Fifth Power System Computation Conference, Cambridge, England, September 1975.
- [25] K.E. Bollinger, A. Laha, R. Hamilton, T. Harras, " Power System Stabilizer Design Using Root Locus Methods" IEEE Trans. Vol. PAS\_94, September/October 1975, pp. 1484-1488.
- [26] P.Sh. Kundur " Power System Stability And Control" Mc Grohil , 1995
- [27] Constantin Von Altk " Fuzzy Logic and NeuroFuzzy Application Explained" Prentice Hall 1995.
- [28] Zimmerman, H.J., " Fuzzy Set Theory – and its Applications" Zweite Revidierte Auflage 1991.

---

[29] N.N., " fuzzy TECH 5 NeuroFuzzy Module Manual", Inform Gmbh Aachen / Inform Software Grop., Chicago 1998.

[30] "Matlab\Help OR web site [www.mathworke](http://www.mathworke) "

صالح

## ضمیمه الف

Mfile مربوط به ارتباط سخت افزاری

```
function [sys,x0,str,ts] = lptport(t,x,u,flag)
%SFUNTMPL General M-file S-function template
%
% The following outlines the general structure of an S-function.
%
switch flag,

    %%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes;

    case 3,
        sys=mdlOutputs(t,x,u);

    %%%%%%%%%%%

    case {1,2,4,9},
        sys=[] ;

    %%%%%%%%%%%
    % Unexpected flags %
```



```

%%%%%%%%%%
otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

% end sfuntmpl

%
%=====
%
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
%
function [sys,x0,str,ts]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs    = 1;
sizes.NumInputs     = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];

%
% str is always an empty matrix
%

```

```

str = [];

%
% initialize the array of sample times
%
ts = [0.1 0];

% end mdlInitializeSizes
%
%=====
=====

% mdlOutputs
% Return the block outputs.
%=====
=====

%
function sys=mdlOutputs(t,x,u)
pp=digitalio('parallel','lpt1');
outline=addline(pp,0:7,0,'out');    % out data 1~8
inline=addline(pp,0:3,1,'in');      % in data 9~12
rdline=addline(pp,4,1,'in');        % AK 13
wrline=addline(pp,0:1,2,'out');     % Write(frommic) 14,15
A=[]; d=[];
putvalue(pp.line(14),1);
putvalue(pp.line(15),1);
%***** get user data *****
if (u(2)<0|u(2)>200)
    data1=0;
else
    data1=floor(u(2));
end
if (u(1)<0|u(1)>200)
    data2=0;
else
    data2=floor(u(1));
end

%***** write to port Data1 *****
putvalue(pp.line(1:8),data1);
putvalue(pp.line(14),0);    % connected to p0.2
putvalue(pp.line(15),0);    % connected to p0.3
AK = getvalue(pp.line(13));
while AK==0
    AK = getvalue(pp.line(13));
end
% ready micro to receive data

```

```

%***** write to port Data2 *****
putvalue(pp.line(1:8),data2);
putvalue(pp.line(15),1);      % sending data

AK = getvalue(pp.line(13));
while AK==1
    AK = getvalue(pp.line(13));
end          % ready micro to receive data
putvalue(pp.line(14),1);      % sending data
putvalue(pp.line(15),0);      % read part 1 of data
AK = getvalue(pp.line(13));
while AK==0
    AK = getvalue(pp.line(13));
end          % ready micro to send data
d=getvalue(pp.line(9:12));
A=[A d];
putvalue(pp.line(15),1);
AK = getvalue(pp.line(13));
while AK==1
    AK = getvalue(pp.line(13));
end          % ready micro to send data
d=getvalue(pp.line(9:12));
A=[A d];

temp=binvec2dec(A);
if temp==0
    sys =inf;
else
    sys=temp; 3/temp;
end
daqreset;

% end mdlOutputs

```

## ضمیمه ب

کد اسمبلی مربوط به سخت افزار

8051 Assembler Version 1.00 08/28/104 10:11:31 Page 1  
E:\My Project\IDE\Myproject\Myproject.a51

```
1 0000      AD    EQU    42
2 0000      BD    EQU    57
3 0000      CD    EQU    72
4 0000      ED    EQU    87
5 0000      RDM   EQU    20
6 0000      AW    EQU    45
7 0000      BW    EQU    75
8 0000      CW    EQU    105
9 0000      EW    EQU    125
10 0000     FW    EQU    175
11 0000     RWM   EQU    10
12 0000     AK    BIT    P1.0      ; TO COMPUTER
13 0000     Rw    BIT    P1.1      ; 0=READE, 1=WRITE
14 0000     RS    BIT    P1.2      ; 0=DATA1, 1=DATA2
15 0000                ORG    00H
16 0000 020080        JMP    MAIN
17 0080                org    80h
18 0080     MAIN:
```

```

19 0080 7A00      MOV  R2,#0
20 0082 7900      MOV  R1,#0
21 0084 75A0FF    MOV  P2,#255
22 0087 7580FF    MOV  P0,#255
23 008A D292      SETB RS
24 008C D291      SETB RW
25 008E D290      SETB AK
26 0090 2091FD    START: JB  RW,S ;wait for RW signal to read data
27 0093 2092FD    JB  RS,S ;wait for RS signal to get data 1
28 0096           ;CALL DELAY
29 0096 E580      MOV  A,P0
30 0098 7940      MOV  R1,#40H
31 009A F7        MOV  @R1,A
32 009B C290      CLR  AK ;AK for take data1
33 009D 3092FD    JNB  RS,S ;wait for RS signal to get data
34 00A0 E580      MOV  A,P0 ;take data2
35 00A2           ;CALL DELAY
36 00A2 7945      MOV  R1,#45H
37 00A4 F7        MOV  @R1,A
38 00A5 D290      SETB AK
39 00A7 11C1      ACALL DEFUZZ
40 00A9 3091FD    JNB  RW,S ;ready for send data
41 00AC 2092FD    JB  RS,S ;ready for put part1 of outdata ot p2
42 00AF           ;CALL DELAY
43 00AF F5A0      MOV  P2,A ;put part1 of outdata
44 00B1           ;CALL DELAY
45 00B1 C290      CLR  AK ;AK for put data
46 00B3 C4        swap a ;make par2 of outdata
47 00B4 3092FD    JNB  RS,S ;ready for put part2 of outdata ot p2
48 00B7 F5A0      MOV  P2,A
49 00B9           ;CALL DELAY
50 00B9 D290      SETB AK
51 00BB 020090    LJMP START
52 00BE 020080    JMP  MAIN
53 00C1           DEFUZZ:
54 00C1 1200D1    LCALL MFUNDL
55 00C4 120121    LCALL MFUNDM
56 00C7 120171    LCALL MFUNWL
57 00CA 12019D    LCALL MFUNWM

```

8051 Assembler Version 1.00 08/28/104 10:11:32 Page 2  
 E:\My Project\IDE\Myproject\Myproject.a51

```

58 00CD 120205      LCALL OUTDATA
59 00D0 22          RET
60 00D1            ;***** MEMBERSHIP FUNCTION OF DELTA LARG
61 00D1            MFUNDL:
62 00D1 7940        MOV  R1,#40H
63 00D3 E7          MOV  A,@R1
64 00D4 F9          MOV  R1,A
65 00D5 C3          CLR  C
66 00D6 9457        SUBB A,#ED
67 00D8 5041        JNC  ZERODL
68 00DA E9          MOV  A,R1
69 00DB C3          CLR  C
70 00DC 9448        SUBB A,#CD
71 00DE 401D        JC   PRMEMDL
72 00E0 7457        MOV  A,#ED
73 00E2 C3          CLR  C
74 00E3 99          SUBB A,R1
75 00E4 F9          MOV  R1,A
76 00E5 C3          CLR  C
77 00E6 940C        SUBB A,#12
78 00E8 4007        JC   CTNUDL1
79 00EA 7F14        MOV  R7,#RDM
80 00EC 1201ED      CALL MATHFUN
81 00EF 802C        SJMP RETURNNDL
82 00F1 E9          CTNUDL1:MOV A,R1
83 00F2 75F014      MOV  B,#RDM
84 00F5 A4          MUL  AB
85 00F6 75F003      MOV  B,#3
86 00F9 84          DIV  AB
87 00FA 02011D      JMP  RETURNNDL
88 00FD E9          PRMEMDL:MOV A,R1
89 00FE C3          CLR  C
90 00FF 9439        SUBB A,#BD
91 0101 4018        JC   ZERODL
92 0103 F9          MOV  R1,A
93 0104 C3          CLR  C
94 0105 940C        SUBB A,#12
95 0107 4007        JC   CTNUDL2
96 0109 7F14        MOV  R7,#RDM
97 010B 1201ED      CALL MATHFUN

```

```

98 010E 800D      SJMP RETURNNDL
99 0110 E9      CTNUDL2:MOV A,R1
100 0111 75F014    MOV B,#RDM
101 0114 A4      MUL AB
102 0115 75F003    MOV B,#3
103 0118 84      DIV AB
104 0119 8002      SJMP RETURNNDL
105 011B 7400      ZERODL:MOV A,#0
106 011D          RETURNNDL:
107 011D 7941      MOV R1,#41H
108 011F F7      MOV @R1,A
109 0120 22      RET
110 0121
111 0121          ;***** MEMBERSHIP FUNCTION OF DELTA MEDIUM
112 0121          MFUNDM:
113 0121 7940      MOV R1,#40H
114 0123 E7      MOV A,@R1

```

8051 Assembler Version 1.00 08/28/104 10:11:32 Page 3  
E:\My Project\IDE\Myproject\Myproject.a51

```

115 0124 F9      MOV R1,A
116 0125 C3      CLR C
117 0126 9448      SUBB A,#CD
118 0128 5041      JNC ZERODM
119 012A E9      MOV A,R1
120 012B C3      CLR C
121 012C 9439      SUBB A,#BD
122 012E 401D      JC PRMEMDM
123 0130 7448      MOV A,#CD
124 0132 C3      CLR C
125 0133 99      SUBB A,R1
126 0134 F9      MOV R1,A
127 0135 C3      CLR C
128 0136 940C      SUBB A,#12
129 0138 4007      JC CTNUDM1
130 013A 7F14      MOV R7,#RDM
131 013C 1201ED      CALL MATHFUN
132 013F 802C      SJMP RETURNNDM
133 0141 E9      CTNUDM1:MOV A,R1
134 0142 75F014    MOV B,#RDM
135 0145 A4      MUL AB

```

```

136 0146 75F003      MOV  B,#3
137 0149 84         DIV  AB
138 014A 02016D      JMP  RETURNNDM
139 014D E9         PRMEMDM:MOV  A,R1
140 014E C3         CLR  C
141 014F 942A       SUBB A,#AD
142 0151 4018       JC   ZERODM
143 0153 F9         MOV  R1,A
144 0154 C3         CLR  C
145 0155 940C       SUBB A,#12
146 0157 4007       JC   CTNUDM2
147 0159 7F14       MOV  R7,#RDM
148 015B 1201ED     CALL MATHFUN
149 015E 800D       SJMP RETURNNDM
150 0160 E9         CTNUDM2:MOV  A,R1
151 0161 75F014     MOV  B,#RDM
152 0164 A4         MUL  AB
153 0165 75F003     MOV  B,#3
154 0168 84         DIV  AB
155 0169 8002       SJMP RETURNNDM
156 016B 7400     ZERODM:MOV  A,#0
157 016D           RETURNNDM:
158 016D 7943       MOV  R1,#43H
159 016F F7         MOV  @R1,A
160 0170 22         RET
161 0171
162 0171           ;***** MEMBERSHIP FUNCTION OF WR(FRECUNCY)
LARG
163 0171           MFUNWL:
164 0171 7945       MOV  R1,#45H
165 0173 E7         MOV  A,@R1
166 0174 F9         MOV  R1,A
167 0175 C3         CLR  C
168 0176 94AF       SUBB A,#FW
169 0178 501D       JNC  ZEROWL
170 017A E9         MOV  A,R1
171 017B C3         CLR  C

```

```

172 017C 947D       SUBB A,#EW

```



```

173 017E 400B      JC  PRMEMWL
174 0180 74AF      MOV A,#FW
175 0182 C3        CLR C
176 0183 99        SUBB A,R1
177 0184 75F002    MOV B,#2
178 0187 A4        MUL AB
179 0188 020199    JMP RETURNWL
180 018B E9        PRMEMWL:MOV A,R1
181 018C C3        CLR C
182 018D 944B      SUBB A,#BW
183 018F 4006      JC  ZEROWL
184 0191 75F002    MOV B,#2
185 0194 A4        MUL AB
186 0195 8002      SJMP RETURNWL
187 0197 7400      ZEROWL:MOV A,#0
188 0199          RETURNWL:
189 0199 7946      MOV R1,#46H
190 019B F7        MOV @R1,A
191 019C 22        RET
192 019D
193 019D          ;***** MEMBERSHIP FUNCTION OF WR(FRECUNCY)
MEDIUM
194 019D          MFUNWM:
195 019D 7945      MOV R1,#45H
196 019F E7        MOV A,@R1
197 01A0 F9        MOV R1,A
198 01A1 C3        CLR C
199 01A2 9469      SUBB A,#CW
200 01A4 5041      JNC ZEROWM
201 01A6 E9        MOV A,R1
202 01A7 C3        CLR C
203 01A8 944B      SUBB A,#BW
204 01AA 401D      JC  PRMEMWM
205 01AC 7469      MOV A,#CW
206 01AE C3        CLR C
207 01AF 99        SUBB A,R1
208 01B0 F9        MOV R1,A
209 01B1 C3        CLR C
210 01B2 940C      SUBB A,#12
211 01B4 4007      JC  CTNUWM1
212 01B6 7F0A      MOV R7,#RWM
213 01B8 1201ED    CALL MATHFUN
214 01BB 802C      SJMP RETURNWM
215 01BD E9        CTNUWM1:MOV A,R1
216 01BE 75F00A    MOV B,#RWM

```

---

```
217 01C1 A4      MUL  AB
218 01C2 75F003  MOV  B,#3
219 01C5 84      DIV  AB
220 01C6 0201E9  JMP  RETURNWM
221 01C9 E9      PRMEMWM:MOV  A,R1
222 01CA C3      CLR  C
223 01CB 942D    SUBB A,#AW
224 01CD 4018    JC   ZEROWM
225 01CF F9      MOV  R1,A
226 01D0 C3      CLR  C
227 01D1 940C    SUBB A,#12
228 01D3 4007    JC   CTNUWM2
```

8051 Assembler Version 1.00 08/28/104 10:11:32 Page 5  
 E:\My Project\IDE\Myproject\Myproject.a51

```

229 01D5 7F0A      MOV  R7,#RWM
230 01D7 1201ED      CALL MATHFUN
231 01DA 800D      SJMP RETURNWM
232 01DC E9      CTNUWM2:MOV  A,R1
233 01DD 75F00A      MOV  B,#RWM
234 01E0 A4      MUL  AB
235 01E1 75F003      MOV  B,#3
236 01E4 84      DIV  AB
237 01E5 8002      SJMP RETURNWM
238 01E7 7400      ZEROWM:MOV  A,#0
239 01E9      RETURNWM:
240 01E9 7948      MOV  R1,#48H
241 01EB F7      MOV  @R1,A
242 01EC 22      RET
243 01ED
244 01ED
245 01ED      ;*****
246 01ED      MATHFUN:
247 01ED E9      MOV  A,R1
248 01EE 8FF0      MOV  B,R7
249 01F0 A4      MUL  AB
250 01F1 A9F0      MOV  R1,B
251 01F3 25F0      ADD  A,B
252 01F5 75F003      MOV  B,#3
253 01F8 84      DIV  AB
254 01F9 F8      MOV  R0,A
255 01FA 74FF      MOV  A,#255
256 01FC 75F003      MOV  B,#3
257 01FF 84      DIV  AB
258 0200 89F0      MOV  B,R1
259 0202 A4      MUL  AB
260 0203 28      ADD  A,R0
261 0204 22      RET
262 0205
263 0205      ;***** OUT DATA
264 0205      OUTDATA:
265 0205 7941      MOV  R1,#41H
266 0207 E7      MOV  A,@R1
267 0208 603F      JZ  OUTZERO
268 020A FA      MOV  R2,A

```

---

269 020B C3	CLR C
270 020C 7943	MOV R1,#43H
271 020E E7	MOV A,@R1
272 020F FB	MOV R3,A
273 0210 7946	MOV R1,#46H
274 0212 E7	MOV A,@R1
275 0213 6034	JZ OUTZERO
276 0215 FC	MOV R4,A
277 0216 7948	MOV R1,#48H
278 0218 E7	MOV A,@R1
279 0219 FD	MOV R5,A
280 021A EB	MOV A,R3
281 021B 2A	ADD A,R2
282 021C	
283 021C 8DF0	MOV B,R5 ; $x=wm*(dm+dl)/dl$
284 021E A4	MUL AB
285 021F A9F0	MOV R1,B

8051 Assembler Version 1.00 08/28/104 10:11:32 Page 6  
 E:\My Project\IDE\Myproject\Myproject.a51

```

286 0221 25F0      ADD  A,B
287 0223 8AF0      MOV  B,R2
288 0225 84        DIV  AB
289 0226 F8        MOV  R0,A
290 0227 74FF      MOV  A,#255
291 0229 8AF0      MOV  B,R2
292 022B 84        DIV  AB
293 022C 89F0      MOV  B,R1
294 022E A4        MUL  AB
295 022F 28        ADD  A,R0
296 0230           ; 100*x/w1
297 0230 75F00A    MOV  B,#10
298 0233 A4        MUL  AB
299 0234 A9F0      MOV  R1,B
300 0236 25F0      ADD  A,B
301 0238 8CF0      MOV  B,R4
302 023A 84        DIV  AB
303 023B F8        MOV  R0,A
304 023C 74FF      MOV  A,#255
305 023E 8CF0      MOV  B,R4
306 0240 84        DIV  AB
307 0241 89F0      MOV  B,R1
308 0243 A4        MUL  AB
309 0244 28        ADD  A,R0
310 0245 240A      ADD  A,#10
311 0247 8002      SJMP RETURNOUT
312 0249 7400      OUTZERO:MOV  A,#0
313 024B           RETURNOUT:
314 024B 22        RET
315 024C

```

Defined Symbols:

```

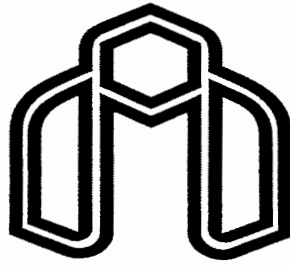
AD          00002A 42
AK          000090 144
AW          00002D 45
BD          000039 57
BW          00004B 75
CD          000048 72
CW          000069 105
ED          000057 87

```

EW	00007D 125
FW	0000AF 175
RDM	000014 20
RS	000092 146
RW	000091 145
RWM	00000A 10

## Defined Labels:

CTNUDL1	0000F1 241
CTNUDL2	000110 272
CTNUDM1	000141 321
CTNUDM2	000160 352
CTNUWM1	0001BD 445
CTNUWM2	0001DC 476
DEFUZZ	0000C1 193
MAIN	000080 128
MATHFUN	0001ED 493
MFUNDL	0000D1 209
MFUNDM	000121 289
MFUNWL	000171 369
MFUNWM	00019D 413
OUTDATA	000205 517
OUTZERO	000249 585
PRMEMDL	0000FD 253
PRMEMDM	00014D 333
PRMEMWL	00018B 395
PRMEMWM	0001C9 457
RETURNDL	00011D 285
RETURNDM	00016D 365
RETURNOUT	00024B 587
RETURNWL	000199 409
RETURNWM	0001E9 489
START	000090 144
ZERODL	00011B 283
ZERODM	00016B 363
ZEROWL	000197 407
ZEROWM	0001E7 487



دانشگاه صنعتی شاهرود

**Shahrood University of Technology**  
**Department of Electric and Robotics**

**Thesis submitted for the degree of Master of Sciences in  
Control Engineering**

Title:

**Design and Implementation  
of  
Fuzzy Power system stabilizer**

Supervisors:

**Dr. Ali Akbar ghareveici**  
**Dr. Ebrahim Hoseini**

By:

**Ebrahim Noormohammadi**

September 2004