

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ



دانشگاه صنعتی شاهرود

دانشکده مهندسی برق و رباتیک

پایان نامه کارشناسی ارشد مهندسی کنترل

تشخیص خطای مقاوم پنجره **Anti-Pinch** هوشمند برای وسایل نقلیه با

استفاده از  $H_-/H_\infty$

نگارنده

مأده محمدی ازنی

استاد راهنما

دکتر محمد علی صدرنیا

بهمن ۱۳۹۶

شماره ۱۵۴۰، ۱۳۹۶  
تاریخ: ۱۱/۱۱/۹۶

باسمه تعالی



مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای مائده محمدی ازنی با شماره

دانشجویی ۹۴۱۶۰۶۴ رشته مهندسی برق- کنترل گرایش کنترل تحت عنوان: تشخیص خطای مقاوم پنجره

Anti-Pinch هوشمند برای وسایل نقلیه با استفاده از H-H<sub>∞</sub> که در تاریخ ۱۳۹۶/۱۱/۱۱ با حضور هیأت

محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با درجه: <u>جمله</u> ) <input checked="" type="checkbox"/>	مردود <input type="checkbox"/>
نوع تحقیق: <input checked="" type="checkbox"/> نظری <input type="checkbox"/> عملی	

عضو هیات داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
۱- استاد راهنمای اول	محمد علی مهدی	دانشیار	
۲- استاد راهنمای دوم	-	-	-
۳- استاد مشاور	-	-	-
۴- نماینده تحصیلات تکمیلی	حسین قلی زاده	دانشیار	
۵- استاد ممتحن اول	عمریف النی	دانشیار	
۶- استاد ممتحن دوم	عمار ابراهیمی	استادیار	

نام و نام خانوادگی رئیس دانشکده: محمد خاوری  
تاریخ و امضاء و مهر دانشکده:

تیسره: در صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

این اثر را تقدیم می‌کنم به

پدر بزرگوار و مادر مهربانم

آن دو فرشته‌ای که از خواسته‌هایشان گذشتند،

سختی‌ها را به جان خریدند

و

خود را سپر بلای مشکلات و ناملایمات کردند

تا من به جایگاهی که اکنون در آن ایستاده‌ام برسم.

## تقدیر و تشکر

سپاس خدایی را سزاست که از من بی‌نیاز است اما با من دوستی می‌کند و به من محبت می‌ورزد.

سپاس خدایی را سزاست که غیر از او را نمی‌خوانم و اگر بخوانم هم پاسخی نمی‌شنوم.

سپاس خدایی را سزاست که با من بردباری می‌کند انگار که من هیچ گناهی نکرده‌ام.

به گونه ای در من می‌نگرد انگار که هیچ خطایی از من سر نزده است.

با من طوری رفتار می‌کند انگار که هیچ لغزشی نداشته‌ام.

این خدا! خدای من است...!

به راستی ستایش برانگیز است.

به حقیقت دوست داشتنی است.

و به واقع سجده کردنی.

برگرفته از دعای امام سجاده(ع)

همچنین از جناب آقای دکتر محمدعلی صدرنیا که با صبر و حوصله‌ی فراوان، کمک‌های ارزنده‌ای به

بنده کردند، تشکر می‌کنم و آرزوی موفقیت و سلامتی دارم.

## تعهد نامه

اینجانب **مائده محمدی ازنی** دانشجوی دوره کارشناسی ارشد رشته **مهندسی برق** دانشکده **برق و رباتیک** دانشگاه صنعتی شاهرود نویسنده پایان نامه **تشخیص خطای مقاوم پنجره Anti-Pinch هوشمند برای وسایل نقلیه با استفاده از  $H_{-}/H_{\infty}$**  تحت راهنمایی **آقای دکتر محمدعلی صدرنیا** متعهد می شوم:

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

### تاریخ

### امضای دانشجو

### مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزار ها و تجهیزات ساخته شده است ) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

\* متن این صفحه نیز باید در ابتدای نسخه های تکثیر شده پایان نامه وجود داشته باشد .

## چکیده

در این پایان نامه یک راه حل بهینه  $H_-/H_\infty$  برای مساله تشخیص عیب روی مدل سیستم پنجره ضد - پینچ (Anti-Pinch) وسایل نقلیه ارائه می شود. ضد پینچ، با نمونه برداری از پارامترهایی مانند جریان مصرفی، سرعت و موقعیت موتورهای DC، امکان تشخیص وجود مانع را فراهم کرده و از به وجود آمدن آسیبها جلوگیری می کند. در این تحقیق، از روش مبتنی بر سرعت، جهت تشخیص مانع استفاده می شود. به منظور ایمن کردن پنجره ضد پینچ، یک الگوریتم بهینه بر اساس مشاهده گر تشخیص عیب پیشنهاد می شود. در طراحی باقی مانده، الگوریتم پیشنهادی تشخیص عیب، از نرخ تغییر سرعت زاویه ای مدل موتور DC استفاده می کند. با مقایسه سیگنال باقی مانده با آستانه از پیش طراحی شده، وقوع عیب تشخیص داده می شود. مقاومت در برابر اغتشاش و حساسیت در برابر عیب، هر دو به عنوان یک مساله بهینه سازی مبتنی بر الگوریتم چندهدفه (MOPSO) PSO، در نظر گرفته می شوند. در نهایت یک فیلتر بهینه برای حل مساله عیب، از روش  $H_-/H_\infty$  طراحی گردید.

**کلمات کلیدی:**  $H_-/H_\infty$ ، ضد پینچ، باقی مانده، تشخیص عیب، MOPSO.

## فهرست مطالب

فصل اول: مقدمه.....	۱
۱-۱- تاریخچه.....	۲
۲-۱- کلیات و مفاهیم.....	۵
۱-۲-۱- مقدمه.....	۵
۲-۲-۱- تعاریف.....	۵
۳-۲-۱- انواع عیب.....	۶
۱-۳-۲-۱- دسته بندی عیب بر اساس محل وقوع عیب.....	۶
۲-۳-۲-۱- دسته بندی عیب بر اساس مشخصه زمانی.....	۷
۳-۳-۲-۱- دسته بندی عیب بر اساس نحوه مدل کردن.....	۸
۴-۲-۱- تشخیص عیب.....	۹
۱-۴-۲-۱- تشخیص عیب بر مبنای افزونگی سخت افزاری.....	۱۰
۲-۴-۲-۱- تشخیص عیب بر مبنای افزونگی تحلیلی.....	۱۰
۳-۴-۲-۱- روش های بدون مدل.....	۱۰
۴-۴-۲-۱- روش های مبتنی بر مدل.....	۱۱
۵-۴-۲-۱- مساله مقاوم بودن در تشخیص عیب مبتنی بر مدل.....	۱۳
۳-۱- بیان مسئله و ضرورت انجام تحقیق.....	۱۳
فصل دوم: بررسی روش های مختلف تشخیص عیب مبتنی بر مشاهده گر.....	۱۵
۱-۲- مقدمات.....	۱۶
۱-۱-۲- سیستم نامی.....	۱۶
۲-۱-۲- نمایش سیستم های تحت اثر اغتشاش.....	۱۷
۳-۱-۲- نمایش های مدل سیستم ها با عدم قطعیت های مدل.....	۱۷
۴-۱-۲- مدل سازی عیب.....	۱۸
۲-۲- روش های مختلف در تشخیص عیب مبتنی بر مشاهده گر.....	۱۹
۱-۲-۲- فیلتر تشخیص عیب.....	۲۰



۲۲	مشاهده گر تشخیصی.....	۲-۲-۲
۲۲	مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_2$ .....	۳-۲-۲
۲۳	مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_\infty$ .....	۴-۲-۲
۲۶	مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_2/H_\infty$ .....	۵-۲-۲
۲۸	معرفی الگوریتم PSO.....	۶-۲-۲
۳۳	معرفی الگوریتم MOPSO.....	۷-۲-۲
۳۷	فصل سوم: شبیه سازی و آنالیز.....	
۳۸	۱-۳- معرفی مدل.....	
۴۰	۲-۳- مدل تحلیلی از الگوریتم تشخیص عیب.....	
۴۳	۳-۳- شبیه سازی سیستم به کمک حل الگوریتم PSO.....	
۴۷	۱-۳-۳- شبیه سازی سیستم با استفاده از معیار $H_2$ .....	
۴۹	۳-۳-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_2$ .....	
۵۱	۳-۳-۲- شبیه سازی سیستم با استفاده از معیار $H_\infty$ .....	
۵۳	۳-۳-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_\infty$ .....	
۵۴	۴-۳- شبیه سازی سیستم به کمک حل الگوریتم MOPSO.....	
۵۴	۱-۴-۳- شبیه سازی سیستم با استفاده از معیار $H_2/H_\infty$ .....	
۵۸	۳-۴-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_2/H_\infty$ .....	
۶۰	۵-۳- مقایسه‌ی نتایج شبیه‌سازی سیستم با الگوریتم MOPSO و LMI.....	
۶۱	فصل چهارم: نتایج و پیشنهادات.....	
۶۲	۱-۴- نتیجه گیری.....	
۶۲	۲-۴- پیشنهادات.....	
۶۳	منابع.....	

## فهرست اشکال

- شکل (۱-۱) وابستگی زمانی عیب‌ها. (a) ناگهانی. (b) افزاینده. (c) تناوبی..... ۷
- شکل (۲-۱) نامعینی جمعی..... ۸
- شکل (۳-۱) نامعینی ضربی..... ۹
- شکل (۱-۲) دیاگرام بلوکی فیلتر تشخیص عیب..... ۲۱
- شکل (۲-۲) خواص الگوریتم PSO..... ۲۹
- شکل (۳-۲) شمای کلی عملکرد ذرات PSO..... ۳۰
- شکل (۱-۳) شمای کلی پنجره برقی..... ۳۹
- شکل (۳-۳) مدل موتور خطی سازی شده..... ۴۱
- شکل (۳-۳) مدل حلقه بسته سیستم..... ۴۴
- شکل (۴-۳) جعبه ابزار شبیه سازی پنجره ضد پینچ..... ۴۶
- شکل (۵-۳) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته ی نرخ سرعت زاویه‌ای موتور در روش  $H_2$ ..... ۴۸
- شکل (۶-۳) سیگنال باقی مانده به روش  $H_2$  در حضور عیب و اغتشاش..... ۴۸
- شکل (۷-۳) نمودار بود نامعینی جمعی..... ۴۹
- شکل (۸-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H_2$ ..... ۵۰
- شکل (۹-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H_2$  (خط گسسته)..... ۵۰
- شکل (۱۰-۳) تابع هزینه ی الگوریتم PSO در روش  $H_\infty$ ..... ۵۱
- شکل (۱۱-۳) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته ی نرخ سرعت زاویه‌ای موتور در روش  $H_\infty$ ..... ۵۲
- شکل (۱۲-۳) سیگنال باقی مانده به روش  $H_\infty$  در حضور عیب و اغتشاش..... ۵۲
- شکل (۱۳-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H_\infty$ ..... ۵۴
- شکل (۱۴-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H_\infty$  (خط گسسته)..... ۵۴
- شکل (۱۵-۳) تابع هزینه ی الگوریتم MOPSO در روش  $H_-/H_\infty$ ..... ۵۵
- شکل (۱۶-۳) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته ی نرخ سرعت زاویه‌ای موتور در روش  $H_-/H_\infty$ ..... ۵۷
- شکل (۱۷-۳) سیگنال باقی مانده به روش  $H_-/H_\infty$  در حضور عیب و اغتشاش..... ۵۷

شکل (۱۸-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H_1/H_\infty$ ..... ۵۹

شکل (۱۹-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H_1/H_\infty$  (خط گسسته)..  
۵۹.....

## فهرست جداول

- جدول (۱-۳) مقادیر نامی پارامترهای موتور..... ۴۲
- جدول (۲-۳) مقادیر پارامترهای مورد استفاده در الگوریتم PSO..... ۴۶
- جدول (۳-۳) مقادیر پارامترهای مورد استفاده در الگوریتم MOPSO..... ۵۶

# فصل اول

## مقدمه

## ۱-۱- تاریخچه

رشد روزافزون سیستم‌های اتوماتیک و کاربرد آن‌ها در سیستم‌های بزرگتر و پیچیده‌تر و به دنبال آن، افزایش تقاضا برای سیستم‌های ایمنی، منجر به گرایش بیشتر به سمت تکنیک‌های تشخیص عیب، علی‌الخصوص، تکنیک‌های بر پایه مدل در سیستم‌های دینامیکی شده‌است [۱]. عدم تشخیص به‌موقع عیب، منجر به صدمه دیدن و از بین رفتن بخش قابل توجهی از امکانات و اطلاعات و در برخی موارد، نیروی انسانی، خواهد شد.

به‌علت خسارت‌های وارده از بروز عیب، صنایع علاقه‌مند شدند دنبال روشی برای کم کردن وقوع عیب باشند. جلوگیری از وقوع عیب در سیستم‌های کنترلی، امری غیرممکن است. اما اگر بتوان عیب را به‌موقع تشخیص داد و دینامیک آن را شناسایی کرد، می‌توان با اعمال قانون کنترلی مناسب، میزان خسارت را تا حد قابل قبولی کاهش داد. به سیستم‌هایی که دارای چنین توانایی باشند، اصطلاحاً سیستم‌های کنترل تحمل‌پذیر عیب<sup>۱</sup> می‌گویند. در این سیستم‌ها در صورت وقوع عیب، مقداری افت عملکرد نیز قابل قبول می‌باشد [۳ و ۲]

برای طراحی سیستم‌های کنترلی تحمل‌پذیر عیب، دو رویکرد اصلی وجود دارد: رویکرد فعال<sup>۲</sup> و رویکرد غیرفعال<sup>۳</sup>. سیستم‌های غیرفعال کنترل‌پذیر عیب، از یک روش ثابت برای اصلاح کردن همه‌ی عیب‌ها و اشتباهات استفاده می‌کنند. اما از آنجایی که در یک سیستم ممکن است انواع عیب‌ها وجود داشته باشد، برای همین، طراحی سیستم کنترلی تحمل‌پذیر عیب، کاری سخت و در بعضی از موارد غیر ممکن است. در سیستم کنترلی تحمل‌پذیر عیب با رویکرد فعال، اندازه‌گیری‌های مختلف، متناظر با عیب‌های مختلف در نظر گرفته می‌شود و در نهایت، بهترین تصمیم برای برخورد با عیب اتخاذ می‌گردد. این روش نیازمند عیب‌یابی سریع و عکس‌العمل مناسب بعد از رخ دادن عیب است.

---

<sup>۱</sup> Fault tolerant control system

<sup>۲</sup> Active approach

<sup>۳</sup> Passive approach

برای تشخیص عیب، دو رویکرد اصلی وجود دارد: تشخیص عیب مبتنی بر مدل و تشخیص عیب بدون مدل [۴]. روش‌های بدون مدل، نیازی به مدل دینامیکی سیستم برای تشخیص عیب ندارند [۵]. این روش‌ها برای وضعیتی که بدست آوردن مدل سیستم مشکل است و یا دارای پیچیدگی‌های محاسباتی است، مناسب می‌باشد. در این روش، از الگوریتم‌های هوشمند فازی و عصبی استفاده می‌شود. از ویژگی‌های این روش می‌توان به حجم بالای اندازه‌گیری‌های سیستمی اشاره کرد.

روش تشخیص عیب مبتنی بر مدل تحلیلی، یکی از مهم‌ترین روش‌ها در حوزه‌ی تشخیص عیب است [۶]. روش‌های بر پایه مدل سیستم، به‌طور کلی سریع‌تر و دقیق‌تر از رویکرد بدون مدل است. این روش‌ها از ایده‌ی تولید سیگنال باقی‌مانده استفاده می‌کنند که از عدم تطابق بین رفتار تخمین زده شده و سیستم واقعی حاصل می‌شود [۷]. روش‌های گوناگونی برای تولید سیگنال باقی‌مانده وجود دارد که می‌توان به روش‌های مبتنی بر رویکرد <sup>۴</sup>، معادلات برابری <sup>۵</sup> و تخمین پارامتر <sup>۶</sup> اشاره کرد [۸].

اولین اقدامات در زمینه تشخیص عیب به اوایل دهه هفتاد برمی‌گردد [۹]. برخی تحقیقات انجام شده در این زمینه در ادامه معرفی شده است:

- طراحی مقاوم  $H_{\infty}$  کاهش‌یافته به فیلتر ورودی نامعلوم برای یک کلاس از سیستم‌های خطی ناشناخته [۱۰].
- تشخیص عیب مقاوم برای توربین‌های بادی با استفاده از روش مبتنی بر مدل مرجع [۱۱].
- روش تنظیم عضویت برای طراحی سیستم تشخیص عیب مقاوم [۱۲].
- روش LMI به منظور طراحی فیلتر تشخیص عیب مقاوم با استفاده از معیار  $H_{\infty}/H_{-}$  [۱۳].

---

<sup>۴</sup> Observer based approach

<sup>۵</sup> Parity equation approach

<sup>۶</sup> Parameter estimation approach

- طراحی فیلتر برای سیستم های فازی T-S در دامنه فرکانسی محدود با استفاده از معیار  $H_-/H_\infty$  [۱۴].

- یک رویکرد احتمالی جهت شناسایی عیب مقاوم برای یک کلاس از سیستم های غیر خطی [۱۵]. یکی از روش هایی که به طور گسترده برای تولید باقی مانده استفاده می شود، روش های افزونگی تحلیلی مبتنی بر رویتگر است که در فرآیندهای عملی با موفقیت به کار برده شده اند [۱۶].

در روش مبتنی بر رویتگر، از یک مدل نامی خطی، بدون وجود عیب<sup>۷</sup> استفاده می گردد و خروجی آن، با سیستم واقعی مقایسه می شود. در صورتی که از عیب مدل سازی و اغتشاش صرف نظر شود، خروجی مدل نامی با خروجی سیستم واقعی تا زمانی که عیبی در سیستم رخ ندهد، برابر است. اما در عمل، از عیب مدل سازی و اغتشاش نمی توان صرف نظر کرد، بنابراین باید دنبال روشی بود که سیگنال باقی مانده فقط به عیب حساس باشد. برای حساس کردن سیگنال باقی مانده به عیب، با استفاده از روش های مقاوم می توان اثر اغتشاش و حالت های سیستم را در خروجی به حداقل رساند و اثر عیب را افزایش داد. با کم کردن اثر حالت های سیستم در خروجی، در واقع عیب مدل سازی حداقل می شود. به علت اینکه اغتشاشات شناخته شده و ناشناخته در سیستم های عملی وجود دارند، زمانی یک سیستم تشخیص عیب، مقاوم و قابل اطمینان است که در روند طراحی اثرات این پارامترها در نظر گرفته شود. بنابراین روش  $H_-/H_\infty$  برای پیدا کردن این ماتریس پیشنهاد شده است.

در این رساله از روش  $H_-/H_\infty$  برای تشخیص عیب در سیستم های خطی نامتغیر با زمان در حضور اغتشاش استفاده شده است. سپس با استفاده از مدل فضای حالت و الگوریتم بهینه سازی ذرات<sup>۸</sup> ، یک روش مناسب برای تشخیص عیب که حساسیت سیگنال باقی مانده به اغتشاش را به حداقل و حساسیت سیگنال باقی مانده به عیب را به حداکثر برساند، ارائه گردیده است.

---

<sup>۷</sup> Fault

<sup>۸</sup> Particle swarm optimization (PSO)



## ۲-۱- کلیات و مفاهیم

### ۱-۲-۱- مقدمه

تشخیص به موقع عیب برای بهبود اطمینان، امنیت و کارایی بسیاری از فرآیندهای صنعتی و فنی مثل هواپیما، اتومبیل و فرآیندهای شیمیایی حائز اهمیت می باشد [۸]. مکانیزم‌هایی مانند شیشه-بالابره‌های برقی، سانروف، درب‌ها و ... که عملکرد آن‌ها به طور مستقیم به انسان‌ها مربوط است، حائز اهمیت دوچندان می باشد. اگر در این سیستم‌ها عیب به موقع تشخیص داده نشود، ممکن است خسارت‌های سنگین اقتصادی و انسانی به بار آورد. در ابتدا بهتر است برخی از اصطلاحات اساسی و تعاریف مهم مثل عیب و سیستم تشخیص عیب را تعریف کرد.

### ۱-۲-۲- تعاریف

- عیب: انحراف غیرمجاز حداقل یکی از مشخصه‌های سیستم از شرایط قابل قبول یا استاندارد می باشد [۸].

- شکست<sup>۹</sup> : یک وقفه‌ی موقت در توانایی یک سیستم برای اجرای یک عمل مورد نیاز تحت شرایط عملکردی ویژه می باشد.

- بد عمل کردن<sup>۱۰</sup> : یک بی‌نظمی متناوب در برآورده کردن عملکرد مطلوب سیستم می باشد.

- اغتشاش<sup>۱۱</sup> : یک ورودی ناشناخته و کنترل نشده است که به سیستم وارد می شود.

- باقی مانده<sup>۱۲</sup> : یک شاخص عیب، مبتنی بر انحراف بین اندازه‌گیری‌ها و محاسبات مبتنی بر روی معادلات می باشد [۱۷].

---

<sup>۹</sup> Failure

<sup>۱۰</sup> Malfunction

<sup>۱۱</sup> Disturbance

<sup>۱۲</sup> Residual

- تشخیص عیب<sup>۱۳</sup> : تشخیص وقوع عیب [۵].

- جداسازی عیب<sup>۱۴</sup> : در صورت وقوع عیب، مکان عیب مشخص می‌شود.

- شناسایی عیب<sup>۱۵</sup> : اندازه و رفتار متغیر با زمان عیب مشخص می‌شود.

### ۱-۲-۳- انواع عیب

عیب را می‌توان به چندین روش مختلف دسته‌بندی کرد. حالت اول بر اساس محل وقوع عیب به

سه دسته عیب سنسور، محرک و اجزا دسته‌بندی می‌گردد [۷].

### ۱-۲-۳-۱- دسته بندی عیب بر اساس محل وقوع عیب

عیب بر اساس محل وقوع، به سه دسته تقسیم می‌شود:

- عیب سنسور: این نوع عیب باعث می‌شود اطلاعات نادرستی از محیط بیرون به سیستم منتقل شود و می‌تواند موجب ناپایداری سیستم گردد. بنابراین تشخیص این نوع عیب بسیار مهم است. بایاس، تنزل عملکرد و ... نمونه‌هایی از عیب سنسور می‌باشند.

- عیب محرک: در بسیاری از سیستم‌های الکتروشمیایی و الکترومکانیکی، سیگنال‌های کنترل به‌طور مستقیم به سیستم قابل اعمال نیستند. محرک‌هایی مورد نیاز است که سیگنال‌های کنترل را به سیگنال‌های تحریک مناسب، مثل گشتاورها و نیروها برای راه‌اندازی سیستم تبدیل کنند. عیب محرک معمولاً به نوع محرک بستگی دارد. مثل اشباع‌شدگی، نشتی، خاموشی و ...

- عیب اجزا: عیب‌هایی که در اجزای سیستم رخ می‌دهند و نتوان آن‌ها را در گروه عیب سنسور و عیب محرک جای داد، در این گروه قرار می‌گیرند. در حقیقت این نوع عیب، موجب بی‌اعتبار شدن

---

<sup>۱۳</sup> Fault Detection

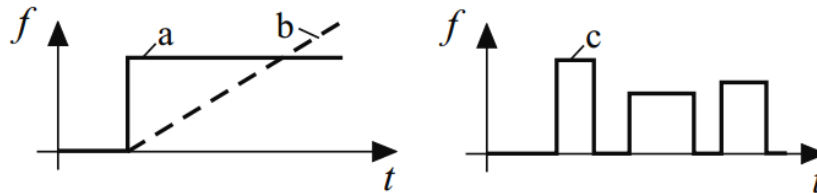
<sup>۱۴</sup> Fault Isolation

<sup>۱۵</sup> Fault Identification

معادلات نامی می‌شوند. ارائه مدل ریاضی این نوع عیب بسیار مشکل می‌باشد. این نوع عیب معمولاً به صورت یک تغییر در معادلات حالت سیستم که می‌تواند یک تغییر پارامتری یا یک تغییر ساختاری باشد، قابل ارائه است. علت ایجاد این عیب، فرسودگی، پارگی، کهنگی اجزا و ... است. نشی در تانک‌های پلیمر و شکستگی در جعبه‌دنده، نمونه‌هایی از عیب اجزا هستند.

### ۱-۲-۳-۲- دسته‌بندی عیب بر اساس مشخصه زمانی

عیب بر اساس مشخصه زمانی به چهار دسته تقسیم می‌شود [۸]. به‌طور شماتیک، این نوع عیب، در شکل (۱-۱) نشان داده شده است.



شکل (۱-۱) وابستگی زمانی عیب‌ها [۸]. (a) ناگهانی، (b) افزایشنده، (c) تناوبی.

- عیب ناگهانی<sup>۱۶</sup> : این نوع عیب‌ها بر اثر وقوع نقص فنی و خرابی تجهیزات به وقوع می‌پیوندند و اثرات مخربی دارند. اما به‌راحتی تشخیص داده می‌شوند. شکستن جعبه‌دنده و یا ازکار افتادن بال هواپیما نمونه‌هایی از این نوع عیب می‌باشند.

- عیب نرم<sup>۱۷</sup> : عیب نرم به عیبی گفته می‌شود که در طول زمان، به‌کندی و به‌علت کهنه و فرسوده شدن اجزای سیستم رخ می‌دهد. به‌علت تغییرات رفتاری آهسته تشخیص آن‌ها مشکل است.

- عیب تناوبی<sup>۱۸</sup> : این عیب‌ها در فاصله‌های زمانی مشخص در سیستم اثر می‌گذارند. مانند حباب‌های هوا در لوله‌های گاز.

<sup>۱۶</sup> Abrupt Faults

<sup>۱۷</sup> Incipion Faults

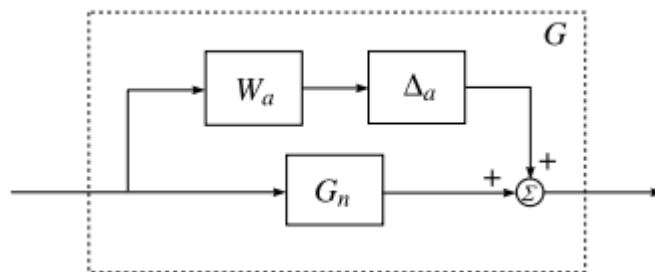
- عیب موقتی: این نوع عیب به طور موقت در سیستم ظاهر می‌شوند و پس از مدتی رفع می‌شوند.

### ۱-۲-۳-۳- دسته بندی عیب بر اساس نحوه مدل کردن

حالت سوم دسته بندی بر اساس نحوه مدل کردن آن‌ها می‌باشد که به دو دسته تقسیم می‌شوند [

۸].

- عیب جمع شونده<sup>۱۹</sup> : این عیب‌ها به صورت جمع شونده در ورودی یا خروجی سیستم ظاهر می‌شوند. مثل آفست سنسورها. نامعینی جمعی در شکل (۲-۱) نشان داده شده است.



شکل (۲-۱) نامعینی جمعی [۸]

$$G(s) = G_n(s) + \Delta_a(s)W_a(s) \quad (1-1)$$

$$|\Delta_a(s)| \leq 1$$

$G(s)$  از رابطه‌ی (۱-۱) حاصل می‌شود که در آن  $W_a(s)$  تابع وزنی،  $\Delta_a(s)$  نامعینی،  $G_n(s)$  تابع

تبدیل نامی سیستم و  $G(s)$  تابع تبدیل سیستم نامعین می‌باشد.

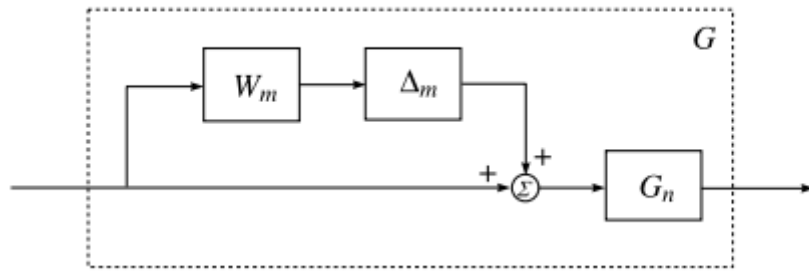
- عیب ضرب شونده<sup>۲۰</sup> : این عیب‌ها سبب ایجاد تغییر در پارامترهای سیستم می‌شوند و به صورت

ضرب شونده پدید می‌آیند. نامعینی ضربی در شکل (۳-۱) نشان داده شده است.

<sup>۱۸</sup> Intermittent Faults

<sup>۱۹</sup> Additive Faults

<sup>۲۰</sup> Multiplicative Faults



شکل (۳-۱) نامعینی ضربی [8]

$$G(s) = G_n(s) (1 + W_m(s) \Delta_m(s)) \quad (2-1)$$

$$|\Delta_m(j\omega)| \leq 1$$

$G(s)$  از رابطه‌ی (۲-۱) حاصل می‌شود که در آن  $W_m(s)$  تابع وزنی،  $\Delta_m(s)$  نامعینی،  $G_n(s)$  تابع

تبدیل نامی سیستم و  $G(s)$  تابع تبدیل سیستم نامعین می‌باشد.

#### ۱-۲-۴- تشخیص عیب

اعتماد به اپراتورهای انسانی در مدیریت رویدادهای غیرطبیعی امکان پذیر نمی‌باشد. عدم تشخیص به‌موقع عیب، به ایجاد مشکلات اقتصادی و انسانی منتهی می‌شود. هدف از به‌کارگیری روش‌های تشخیص عیب، بهبود عملکرد و بالاترین قابلیت اطمینان سیستم می‌باشد.

#### ۱-۲-۴-۱- تشخیص عیب بر مبنای افزونگی سخت‌افزاری<sup>۲۱</sup>

این روش از روش‌های قدیمی تشخیص عیب است. مبنای این روش، مقایسه و بررسی سیگنال‌های تولید شده توسط سخت‌افزارهای مختلف که عموماً چندین سنسور می‌باشد، است. به عنوان مثال، سه سنسور، یک سیگنال را اندازه‌گیری می‌کنند که اگر خروجی یکی از این سه سنسور با خروجی سنسور دیگر تفاوت داشته باشد نشان دهنده وجود عیب است.

<sup>۲۱</sup> Hardware redundancy

این روش، به آنالیزهای سنگین ریاضی نیازی ندارد. اما ایراد اصلی آن، هزینه بالا و فضای زیادی که برای نصب نیاز دارد، می‌باشد و در سیستم‌هایی که به امنیت بالا نیاز دارد مثل سیستم حمل و نقل هوایی و نیروگاه‌های هسته‌ای استفاده می‌شوند.

#### ۱-۲-۴-۲- تشخیص عیب بر مبنای افزونگی تحلیلی<sup>۲۲</sup>

از اوایل ۱۹۷۰ به دلیل هزینه‌های زیاد و محدودیت‌های افزونگی سخت‌افزاری که از نظر وزن و مکان به سیستم تحمیل می‌کرد، روش‌های جدیدتری در حوزه تشخیص عیب ارائه شد. مبنای این روش بررسی رفتار حقیقی سیستم در مقابل مدل ریاضی برای تشخیص تناقض می‌باشد. هرگونه عیبی در سیستم به عنوان باقی‌مانده بیان می‌شود. اگر باقی‌مانده برابر صفر باشد یعنی عیبی رخ نداده است و اگر مخالف صفر باشد نشان دهنده‌ی وقوع عیب است. مزیت این روش نسبت به روش افزونگی سخت‌افزاری این است که به فضا و هزینه‌ی کمتری نیاز دارد. اما دارای قابلیت اطمینان کمتری نسبت به روش افزونگی سخت‌افزاری است.

#### ۱-۲-۴-۳- روش‌های بدون مدل

در این روش، برای تشخیص عیب از مدل ریاضی استفاده نمی‌شود. در این روش با به‌کارگیری اطلاعات موجود در یک پایگاه داده عیب را تشخیص می‌دهند. مشکل اصلی این روش، نیاز آن به داده‌های غنی از سیستم تحت نظارت می‌باشد، ولی برای سیستم‌هایی که دارای مدل ریاضی پیچیده هستند و یا اینکه به‌دست آوردن مدل ریاضی آن بسیار مشکل است، از این روش استفاده می‌شود. روش مبتنی بر آنالیز سیگنال و بررسی حد در این دسته قرار می‌گیرند.

- روش مبتنی بر آنالیز سیگنال: در این روش، یک یا چند سیگنال قابل اندازه‌گیری از سیستم با استفاده از روش‌های مختلفی مانند تحلیل طیف فرکانسی، تحلیل دامنه و تحلیل روند تغییرات

---

<sup>۲۲</sup> Analytical redundancy

سیگنال و ... مورد تحلیل قرار گرفته و هرگونه تغییر غیرمجاز در خصوصیات سیگنال، نشان دهنده وقوع عیب در سیستم است [۱۸].

- روش بررسی حد: در این روش، مقدار متغیرهای سیستم که با اندازه‌گیری به‌دست آمده است، با حدود تعیین شده توسط یک کامپیوتر مقایسه می‌شوند. تجاوز از آستانه، نشان دهنده وقوع عیب است.

روش بررسی حد، ساده‌ترین روش تشخیص عیب است و به راحتی قابل اجرا می‌باشد. ولی این روش زمانی قابل اجرا است که عیب از حدود تعیین شده بسیار بزرگ‌تر باشد که در این صورت، امکان ایجاد خسارت در سیستم وجود دارد و حتی امکان دارد بسیاری از عیب‌ها تشخیص داده نشوند و در بعضی از موارد ممکن است که اغتشاش به عنوان عیب در نظر گرفته شود. همچنین این روش برای سیستم‌هایی که دارای رفتار متغیر هستند مناسب نمی‌باشد [۱۹].

#### ۱-۲-۴- روش‌های مبتنی بر مدل

اکثر روش‌های تشخیص عیب بر اساس روش‌های مبتنی بر مدل هستند. در این روش از یک مدل نامی برای توصیف سیستم استفاده می‌شود، رفتار مدل نامی با سیستم واقعی مقایسه می‌شود، سپس با تحلیل این مقایسه عیب تشخیص داده می‌شود [۱۹]. برای تشخیص عیب از این روش باید دو مرحله زیر را طی کرد.

مرحله اول، تولید باقی‌مانده : هدف این مرحله تولید سیگنال باقی‌مانده است. این سیگنال تفاوت میان متغیرهای اندازه‌گیری شده از سیستم با تخمین‌های محاسبه شده است. در حالت ایده‌آل تا زمانی که عیبی رخ ندهد مقدار سیگنال باقی‌مانده صفر یا نزدیک صفر خواهد بود، ولی اگر مقدار سیگنال باقی‌مانده صفر نباشد، به معنای رخ دادن عیب است. تولید باقی‌مانده مهم‌ترین بخش فرآیند تشخیص عیب می‌باشد. زیرا اگر این سیگنال درست تولید نشود، عیب به درستی تشخیص داده نمی‌شود.

مرحله دوم، ارزیابی باقی مانده<sup>۲۳</sup> : بعد از تولید سیگنال باقی مانده، این سیگنال باید مورد ارزیابی قرار گیرد که به وسیله این ارزیابی باید تشخیص داده شود که اختلاف بین سیستم نامی و سیستم واقعی به علت وجود عیب بوده است یا به علت عوامل دیگر.

روش های رایج تولید باقی مانده به سه دسته کلی تقسیم می شوند:

- روش مبتنی بر مشاهده گر: اگر یک سیستم رویت پذیر باشد، رویتگرها برای تخمین حالت های سیستم با استفاده از داده های اندازه گیری می توانند استفاده شوند. رویتگرهای تشخیص عیب، از عیب بین خروجی اندازه گیری شده و خروجی تخمین زده شده به عنوان باقی مانده استفاده می کنند. اگر  $\hat{y}$  به عنوان خروجی تخمین زده شده و  $y$  به عنوان خروجی واقعی سیستم در نظر گرفته شود، باقی مانده از رابطه (۳-۱) بدست می آید.

(۳-۱)

$$r = \hat{y} - y$$

رویتگرها به مدل دقیق سیستم نیاز دارند و به این علت که این مدل در دسترس نیست، طراحی مقاوم برای رویتگرهای تشخیص عیب مهم است [۲۰].

- روش معادلات برابری: روش معادلات برابری یک روش ساده و مستقیم برای تولید سیگنال باقی مانده است [۱۰]. اگر  $G(s)$  سیستم واقعی باشد و  $\hat{G}(s)$  تخمینی از سیستم واقعی باشد، باقی مانده براساس معادلات برابری از رابطه زیر بدست می آید. در این رابطه  $u$  ورودی کنترل است.

$$r = (\hat{G}(s) - G(s))u \quad (۴-۱)$$

با وجود اینکه روش مبتنی بر معادلات برابری از لحاظ محاسباتی ساده ترین روش است اما به دلیل اینکه برای استفاده از این روش، پارامترهای مدل، شناخته شده و ثابت فرض می شوند، پس به علت عدم قطعیت های سیستم در معرض عدم صحت مدل هستند. برای غلبه بر این مشکل باید از

---

<sup>۲۳</sup> Residual evaluation



معادلات برابری مقاوم استفاده کرد. روش معادلات برابری برای تشخیص و جداسازی عیب‌های جمع شونده مناسب است [۸].

- روش تخمین پارامتر: این روش برای سیستم‌هایی که پارامترهای آن دقیقاً مشخص نیستند به کار می‌رود. این روش برای تشخیص عیب‌های ضرب شونده بسیار مناسب است [۸]. اساس این روش بر مبنای انحراف مقادیر پارامترها از مقادیر نامی آن‌ها است. سیگنال باقی‌مانده از تفاضل مقدار پارامتر در مدل اصلی با مقدار پارامتر تخمین زده شده، بر اساس مقادیر ورودی-خروجی، به دست می‌آید.

### ۱-۲-۴-۵- مساله مقاوم بودن در تشخیص عیب مبتنی بر مدل

در روش تشخیص عیب مبتنی بر مدل، داشتن یک مدل از سیستم لازم و ضروری است. ولی به دست آوردن یک مدل دقیق از سیستم کاری مشکل است. حتی اگر مدل دقیق از یک سیستم داشته باشیم، برخی از پارامترها به طور تقریبی در دسترس هستند. از طرفی سیستم‌های واقعی همواره در معرض ورودی‌های ناخواسته به نام اغتشاش هستند. در نتیجه حتی اگر عیب هم در سیستم وجود نداشته باشد، باقی‌مانده مخالف صفر است و ممکن است برخی از سیگنال‌هایی که ناشی از عیب نیستند را به عنوان عیب تشخیص دهیم. بنابراین در طراحی باید همیشه به دنبال روشی بود که سیگنال باقی‌مانده فقط به سیگنال عیب حساس باشد و هیچ حساسیتی نسبت به سیگنال‌های دیگر نداشته باشد.

### ۱-۳- بیان مسئله و ضرورت انجام تحقیق

در صنعت خودرو، سیستم‌های الکترونیکی همانطور که راحتی مشتری را فراهم می‌کنند، وسایل نقلیه را با نیازهای جدید روبرو می‌کنند [۲۱]. از جمله‌ی این نیازها، استفاده از پنجره‌ی برقی اتوماتیک است که لزوم نگاه داشتن انگشت بر روی دکمه‌ی فرمان حرکت پنجره را تا پایان کورس حرکتی آن، مرتفع می‌سازد.

با توجه به عدم رعایت مسائل ایمنی، حوادث متعددی ناشی از کار با پنجره برقی اتومبیل‌ها گزارش شده است. لذا عدم تشخیص به‌هنگام عیب، موجب آسیب دیدن قسمتی از بدن که در مسیر حرکت پنجره قرار گرفته است می‌شود [۲۲]. از این رو سیستم کنترل پنجره ضد پینچ جهت رفع این مشکل، بسیار مورد توجه واقع شده است.

به‌طور کلی، روش‌های متداول در تشخیص شرایط پینچ، به دو دسته تقسیم شده است. در روش اول، تخمین‌گر پینچ بر این فرض استوار است که سرعت پنجره شدیداً کاهش می‌یابد و حرکت در یک سرعت ثابت در شرایط عملیاتی آرام قرار می‌گیرد. مقدار محاسبات مورد نیاز در این نوع تخمین‌گر نیز، کم است اما عملکرد آن در حضور نویزهای اندازه‌گیری موجب خرابی می‌شود.

نوع دیگری که شرایط پینچ را تشخیص می‌دهد، زمانی است که گشتاور موتور اعمالی بیش از حد از پیش تعیین شده شود که نیاز به یک سنسور جریان اضافی برای جلوگیری از هشدار غلط<sup>۲۴</sup> می‌باشد. با این حال، این نیز به عنوان راه حل‌های عمومی در نظر گرفته نمی‌شود چرا که عملکرد آن متکی بر اعتبار یک حد جریان است.

در این تحقیق، یک الگوریتم بهینه به منظور تشخیص شرایط پینچ برپایه‌ی روی‌تگر تشخیص عیب، برای سیستم کنترل پنجره‌ی ضد پینچ در نظر گرفته شده است. در این الگوریتم از اطلاعات نرخ سرعت زاویه‌ای به عنوان یک عیب تحت شرایط پینچ استفاده می‌شود. با مقایسه سیگنال باقی‌مانده با آستانه از پیش طراحی شده، وقوع پینچ تشخیص داده می‌شود. روی‌تگر تشخیص عیب، در برابر اغتشاش‌ها، مقاوم و نسبت به عیب، حساس طراحی می‌شود.

---

<sup>۲۴</sup> False alarm

## فصل دوم

بررسی روش‌های مختلف تشخیص عیب مبتنی بر

مشاهده‌گر

## ۲-۱-۱- مقدمات

### ۲-۱-۱- سیستم نامی<sup>۲۵</sup>

بسته به اینکه چه اهدافی را در مدل‌سازی دنبال می‌کنیم، انواع مختلفی از سیستم برای توضیح یک فرآیند وجود دارد که در بین آنها مدل‌های خطی تغییر ناپذیر با زمان (LTI) ساده‌ترین و در عین حال پر کاربردترین آنهاست. در این بررسی یک سیستم را در عدم حضور اغتشاش و عیب، سیستم نامی می‌نامیم و فرض می‌شود که سیستم نامی یک سیستم LTI باشد [۲۳].

چنانچه می‌دانیم دو نمایش استاندارد ریاضی از سیستم‌های LTI وجود دارد: ماتریس انتقال و مدل فضای حالت. ماتریس انتقال، یک نمایش ورودی-خروجی از رفتار دینامیکی یک سیستم LTI در حوزه فرکانسی است. از این پس برای این حالت نمایش، از نام گذاری  $G_{yu}(s) \in LH_{\infty}^{m \times k_u}$  استفاده می‌شود که نشان دهنده‌ی ماتریس انتقال از بردار ورودی  $u \in R^{k_u}$  به بردار خروجی  $y \in R^m$  است. یعنی:

$$y(s) = G_{yu}(s)u(s) \quad (1-2)$$

همچنین فرض می‌شود  $G_{yu}(s)$  یک ماتریس انتقال گویای حقیقی است<sup>۲۶</sup>.

فرم استاندارد نمایش حالت یک سیستم پیوسته در زمان LTI به صورت زیر است:

$$\dot{x} = Ax(t) + Bu(t), x(0) = x_0 \quad (2-2)$$

$$y(t) = Cx(t) + Du(t) \quad (3-2)$$

که در آن  $x \in R^n$  بردار حالت،  $x_0$  شرط اولیه سیستم،  $u \in R^{k_u}$  بردار ورودی و  $y \in R^m$  بردار خروجی است.

<sup>۲۵</sup> Nominal System

<sup>۲۶</sup> Real Rational Matrix

ماتریس‌های  $A, B, C, D$  ماتریس‌های ثابت حقیقی با بعد مناسب هستند. روابط (۲-۲) و (۳-۲) را تحقق فضای حالت  $G_{yu}(s) = C(sI - A)^{-1}B + D$  می‌نامند و به صورت زیر نمایش می‌دهند:

$$G_{yu}(s) = (A, B, C, D) \text{ یا } G_{yu}(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (۴-۲)$$

### ۲-۱-۲- نمایش سیستم‌های تحت اثر اغتشاش

اغتشاشات طبیعی، تغییرات غیرمنتظره در فرآیندهای تحت مشاهده، نویزهای فرآیند و اندازه‌گیری، اغلب با بردارهای ورودی ناشناخته مدل می‌شوند. در اینجا این ورودی‌ها را با  $d, v$  و  $n$  نشان می‌دهیم. برای مدل ورودی-خروجی داریم [23]:

$$y(s) = G_{yu}(s)u(s) + G_{yd}(s)d(s) + G_{yv}(s)v(s) \quad (۵-۲)$$

که در آن  $G_{yd}(s)$  معلوم است و ماتریس انتقال اغتشاش نامیده می‌شود،  $d \in \mathbb{R}^{k_d}$  نمایش دهنده‌ی بردار ورودی ناشناخته است،  $v \in \mathbb{R}^{k_v}$  یک فرآیند تصادفی پایدار است که فرض می‌شود همان نویز سفید با توزیع نرمال حول صفر باشد. برای مدل فضای حالت داریم:

$$\dot{x} = Ax(t) + Bu(t) + E_d d + n \quad (۶-۲)$$

$$y(t) = Cx(t) + Du(t) + F_d d + v \quad (۷-۲)$$

### ۳-۱-۲- نمایش‌های مدل سیستم‌ها با عدم قطعیت‌های مدل

چنانچه می‌دانیم عدم قطعیت مدل‌سازی به اختلاف بین مدل سیستم و آنچه در واقع وجود دارد اشاره می‌کند که می‌تواند به علت تغییرات درون فرآیند یا شرایط محیطی اطراف فرآیند بوجود آید.

نمایش عدم قطعیت‌های مدل، خود یک موضوع تحقیق است که هر روزه بیشتر مورد توجه قرار می‌گیرد. در میان تمام نمایش‌هایی که از عدم قطعیت مدل وجود دارد، ما نمایش فضای حالت به شرح زیر را انتخاب می‌کنیم [۲۳]:

$$\dot{x} = \bar{A}x(t) + \bar{B}u(t) + \bar{E}_d d \quad (۸-۲)$$

$$y(t) = \bar{C}x(t) + \bar{D}u(t) + \bar{F}_d d \quad (۹-۲)$$

$$\bar{A} = A + \Delta A, \bar{B} = B + \Delta B, \bar{C} = C + \Delta C \quad (۱۰-۲)$$

که  $\Delta A, \Delta B, \Delta C, \Delta D, \Delta E$  و  $\Delta F$  نرم محدود به شرح زیر می‌باشند:

$$\begin{bmatrix} \Delta A & \Delta B & \Delta E \\ \Delta C & \Delta D & \Delta F \end{bmatrix} = \begin{bmatrix} E \\ F \end{bmatrix} \Delta(t) \begin{bmatrix} G & H & J \end{bmatrix} \quad (۱۱-۲)$$

که  $E, F, G$  و  $J$  ماتریس‌های معلوم با بعد مناسب هستند و  $\Delta(t)$  نامعلوم اما محدود است:

$$\bar{\sigma}(\Delta) \leq \delta_\Delta \quad (۱۲-۲)$$

ارزنده است اشاره کنیم که معادلات (۸-۲) و (۹-۲) با عدم قطعیت نرم محدود به شرح (۱۱-۲) می‌توانند به شکل زیر نوشته شوند:

$$\dot{x} = Ax(t) + Bu(t) + E_d d + E_p \quad (۱۳-۲)$$

$$y(t) = Cx(t) + Du(t) + F_d d + F_p \quad (۱۴-۲)$$

$$q = Gx(t) + Hu(t) + Jd + K_p, p = \tilde{\Delta} q, \tilde{\Delta} = (I + \Delta K)^{-1} \Delta \quad (۱۵-۲)$$

با این شرط که  $(I + \Delta K)$  وارون پذیر باشد.

## ۲-۱-۴-مدل‌سازی عیب

روش‌های مختلفی در مدل‌سازی عیب‌ها وجود دارد که از میان آن‌ها بسط مدل پرکاربردترین

است [۲۳]:

$$y(s)=G_{yu}(s)u(s)+G_{yd}(s)d(s)+G_{yf}(s)f(s) \quad (۱۶-۲)$$

که در آن  $f \in \mathbb{R}^{k_f}$  یک بردار نامعلوم است که نشان دهنده‌ی همه‌ی عیب‌های ممکن است و در حالت بدون عیب برابر صفر است و  $G_{yf}(s) \in LH_\infty$  یک ماتریس انتقال معلوم است.

و در فرم فضای حالت داریم:

$$\dot{x}=Ax(t)+Bu(t)+E_d d+ E_f f \quad (۱۷-۲)$$

$$y(t)=Cx(t)+Du(t)+F_d d+F_f f \quad (۱۸-۲)$$

که در آن  $E_f$  و  $F_f$  ماتریس‌های معلوم هستند. آنگاه داریم:

$$G_{yf}(s)=F_f+C(sI-A)^{-1}E_f \quad (۱۹-۲)$$

ماتریس‌های  $E_f$  و  $F_f$  محل وقوع عیب و تاثیر آن را روی اجزای سیستم مشخص می‌کنند. اما همانطور که می‌دانیم عیب‌ها به سه دسته تقسیم می‌شوند:

عیب سنسور  $f_s$ ، عیب عملگر  $f_A$  و عیب فرآیند  $f_p$ . که از میان آن‌ها صرفاً به عیب سنسور می‌پردازیم. برای این منظور ماتریس  $F_f$  به صورت  $F_f=I$  مدل می‌شود؛ یعنی:

$$y(t)=Cx(t)+Du(t)+F_d d+f_s \quad (۲۰-۲)$$

## ۲-۲- روش‌های مختلف در تشخیص عیب مبتنی بر مشاهده‌گر

در این بخش روش‌های مختلف برای تشخیص عیب مبتنی بر روی‌نگر را معرفی می‌کنیم. پیش از معرفی این روش‌ها، مشاهده‌گر لوئنبرگر<sup>۲۷</sup> را شرح می‌دهیم که پایه‌ی بقیه مطالب می‌باشد.

---

<sup>۲۷</sup> Lunberger Observer





$$r=y-\hat{y}=y-C\hat{x}-Du \quad (25-2)$$

با معرفی متغیر  $e=x-\hat{x}$  داریم:

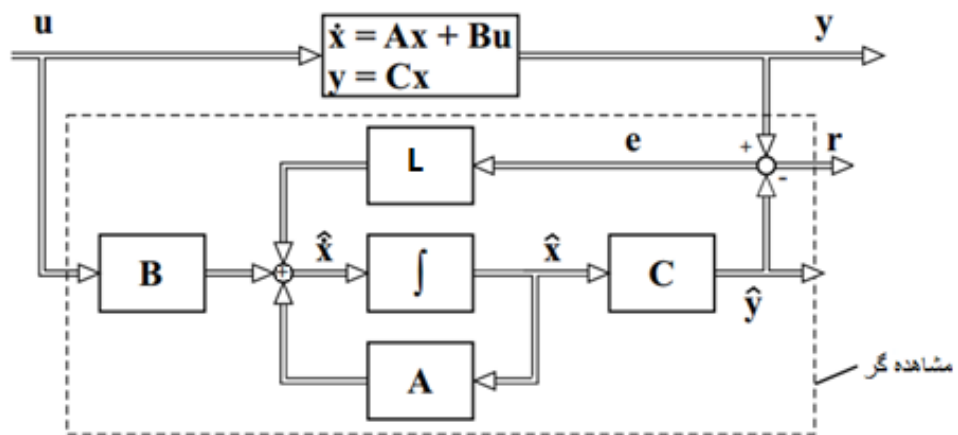
$$\dot{e}=(A-LC)e \quad (26-2)$$

$$r=Ce \quad (27-2)$$

یادآوری می‌کنیم که  $L$  به گونه‌ای انتخاب می‌شود که  $A-LC$  پایدار باشد. در این صورت:

$$\lim_{t \rightarrow \infty} (x(t)-\hat{x}(t))=0 \quad (28-2)$$

بلوک دیاگرام یک مشاهده‌گر مرتبه کامل به منظور تولید سیگنال مانده مطابق شکل (۱-۲) است.



شکل (۱-۲) دیاگرام بلوکی فیلتر تشخیص عیب [23]

### ۲-۲-۲- مشاهده‌گر تشخیصی<sup>۳۳</sup>

این نوع مشاهده‌گر بسیار شبیه به مشاهده‌گر لوئنبرگر است و به همین دلیل بسیار مورد توجه

است. مبنای این مشاهده‌گر به شکل زیر است [۲۶ و ۲۳]:

$$\dot{z}=Gz+Hu+Ly \quad (29-2)$$

<sup>۳۳</sup> Diagnostic Observer

$$\hat{y} = \bar{w}z + \bar{v} + \bar{Q} \quad (30-2)$$

که در آن  $z \in \mathbb{R}^s$  و  $s$  مرتبه مشاهده گر است که می تواند برابر، کوچکتر و یا بزرگتر از  $n$  باشد. مشاهده گرهای مرتبه بالا نقش بسزایی در بهینه سازی سیستم های تشخیص عیب ایفا می کنند.

با فرض  $G_{yu} = C(sI - A)^{-1}B + D$ ، ماتریس های  $G, H, L, \bar{v}, \bar{w}, \bar{Q}$  و  $T \in \mathbb{R}^{s \times n}$  باید شرایط مشهور به شرایط لوتنبرگر را برآورده کنند:

- پایدار  $G$
- $TA - GT = LC, H = TB - LD$
- $C = \bar{w}T + \bar{v}C$  و  $\bar{Q} = -\bar{v}D + D$

که در این صورت تخمین درست از  $y$  خواهیم داشت:

$$\lim_{t \rightarrow \infty} (y(t) - \hat{y}(t)) = 0 \quad (31-2)$$

با مقایسه این روش و فیلتر تشخیص عیب که در بخش قبل به آن اشاره شد، نتیجه می گیریم که:

- در مشاهده گر تشخیصی دارای مرتبه آزادی طراحی بیشتری هستیم.
- در عین حال مراحل طراحی پیچیده تر است.

### ۳-۲-۲- مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_2$ :

برای هر سیستم، سیگنال باقی مانده، هم متاثر از اغتشاش و هم متاثر از عیب است. لذا همواره این نگرانی وجود دارد که آنچه از سیگنال مانده به عنوان عیب تشخیص داده می شود اثر اغتشاش باشد. بنابراین تلاش می شود که اثر ورودی های ناشناخته روی سیگنال مانده کمتر شود [۲۷ و ۲۳].

$$r(s) = \hat{N}_d(s)d(s) + \hat{N}_f(s)f(s) \quad (32-2)$$

$$\hat{N}_d(s) = C(sI - A + LC)^{-1}(E_d - LF_d) + F_d \quad (33-2)$$

$$\hat{N}_f(s) = C(sI - A + LC)^{-1}(E_f - LF_f) + F_f \quad (34-2)$$

باتوجه به روابط (۲-۳۲) تا (۲-۳۴) به دنبال طراحی ماتریس بهره مشاهده گر  $L$  به نحوی هستیم که  $\hat{N}_d(s)$  تا حد امکان کوچک باشد. به عبارت دیگر داریم:

$$\text{Min}_l \left\| C(sI - A + LC)^{-1}(E_d - LF_d) \right\|_2 \quad (2-35)$$

قضیه ۱-۲: برای سیستم  $C(sI - A + LC)^{-1}(E_d - LF_d)$  فرض کنید:

۱-  $(C, A)$  تشخیص پذیر باشد.

۲-  $F_d$  دارای رnk سطری کامل<sup>۳۴</sup> باشد.

۳- رnk ماتریس  $\begin{bmatrix} A - j\omega I & E_d \\ C & F_d \end{bmatrix}$  به ازای  $\omega \in [0, \infty)$  کامل باشد.

آنگاه:

$$\text{Min}_l \left\| C(sI - A + LC)^{-1}(E_d - LF_d) \right\|_2 = (\text{trace}(CXC^T))^{1/2} \quad (36-2)$$

به ازای

$$L = XC^T + E_d F_d^T \quad (37-2)$$

بدست می آید که  $X \leq 0$  پاسخ معادله ریکاتی زیر است:

$$(A - E_d F_d^T C)X + X(A - E_d F_d^T C)^T - XC^T C X + E_d E_d^T - F_d^T F_d E_d^T = 0$$

برای اثبات رجوع کنید به [۲۳].

---

<sup>۳۴</sup> Full row rank

## ۴-۲-۲- مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_\infty$ :

پیش از پرداختن به این موضوع لازم است به پاره‌ای مقدمات ریاضی بپردازیم. همچنین از ارائه برخی تعاریفها و اثباتها خودداری کرده و تنها به ارجاع خواننده به متون دیگر اکتفا می‌کنیم.

قضیه ۲-۲: فرض کنید  $H \in \text{dom}(\text{Ric})$ <sup>۳۵</sup> و  $X = \text{Ric}(H)$  آنگاه:

۱-  $X$  متقارن حقیقی است؛

۲-  $X$  در معادله ریکاتی زیر صدق می‌کند؛

$$A^*X + XA + XRX + Q = 0 \quad (۳۸-۲)$$

۳-  $A + RX$  پایدار است.

اثبات: رجوع کنید به [۲۹ و ۲۸].

قضیه ۳-۲: فرض کنید  $H$  هیچ مقدار ویژه‌ی موهومی نداشته باشد و  $R$  نیمه معین مثبت یا نیمه

معین منفی باشد؛ آنگاه  $H \in \text{dom}(\text{Ric})$  است اگر و تنها اگر  $(A, R)$  پایدارپذیر باشد [28][29].

نتیجه ۱-۲: فرض می‌کنیم  $\gamma > 0$  و  $G(s) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in RH_\infty$  و

$$H = \begin{bmatrix} A + BR^{-1}D^*C & BR^{-1}B^* \\ -C^*(I + DR^{-1}D^*)C & -(A + BR^{-1}D^*C)^* \end{bmatrix}$$

که در آن  $R = \gamma^2 I - D^*D$  سپس بندهای زیر معادند:

$$\|G\|_\infty < \gamma - 1$$

$$\bar{\sigma}(D) < \gamma - 2$$

۳-  $\bar{\sigma}(D) < \gamma$  و  $H \in \text{dom}(\text{Ric})$

<sup>۳۵</sup> برای تعاریف فوق رجوع کنید به [۲۹ و ۲۸]

$$\text{Ric}(H) \geq 0 \text{ و } H \in \text{dom}(\text{Ric}) \text{ و } \bar{\sigma}(D) < \gamma - ۴$$

$$\text{و } \bar{\sigma}(D) < \gamma - ۵ \text{ وجود دارد } X = X^* \geq 0 \text{ که}$$

$$x(A + BR^{-1}D^*C) + (A + BR^{-1}D^*C)^* X + X BR^{-1}B^* X + C^*(I + DR^{-1}D^*)C = 0$$

و  $A + BR^{-1}D^*C + BR^{-1}B^* X$  هیچ مقدار ویژه‌ای روی محور موهومی ندارد.

$$\text{و } \bar{\sigma}(D) < \gamma - ۶ \text{ وجود دارد } X = X^* \geq 0 \text{ که}$$

$$x(A + BR^{-1}D^*C) + (A + BR^{-1}D^*C)^* X + X BR^{-1}B^* X + C^*(I + DR^{-1}D^*)C < 0$$

$$\gamma\text{-وجود دارد } X = X^* \geq 0 \text{ که } \begin{bmatrix} XA + A^*X & XB & C^* \\ B^*X & -\gamma I & D^* \\ C & D & -\gamma I \end{bmatrix}$$

اثبات: رجوع کنید به [۲۸ و ۲۹].

نتیجه فوق، مبنای طراحی در متون مختلف کنترل مقاوم و روش‌های مبتنی بر  $H_\infty$  و تکنیک

$LMI^{۳۶}$  در طراحی مشاهده‌گر تشخیص دهنده‌ی عیب مقاوم است.

با توجه به توضیحات و قضایای فوق، می‌توان به کمک نرم بی‌نهایت مشاهده‌گر مقاوم در برابر ورودی

ناشناخته را به صورت زیر ارائه داد:

قضیه ۲-۴: برای سیستم  $(E_d - LF_d)^{-1}(E_d - LF_d)$  فرض کنید:

۱-  $(C, A)$  تشخیص پذیر باشد.

۲-  $F_d$  دارای رnk سطری کامل باشد.

۳- رnk ماتریس  $\begin{bmatrix} A - j\omega I & E_d \\ C & F_d \end{bmatrix}$  به ازای  $\omega \in [0, \infty)$  کامل باشد.

<sup>۳۶</sup> Linear Matrix Inequality

آنگاه وجود دارد  $\gamma > 0$  که:

$$\|C(sI-A+LC)^{-1}(E_d-LF_d)+F_d\|_{\infty} < \gamma \quad (39-2)$$

به ازای

$$L = -(XC^T + E_d F_d^T)R^{-1}$$

$$R = \gamma^2 I - F_d^T F_d$$

که در آن  $X \leq 0$  پاسخ معادله ریکاتی زیر است [23]:

$$(A - E_d F_d^T C)X + X(A - E_d F_d^T R^{-1} C)^T - X C^T R^{-1} C X + E_d (I - F_d^T R^{-1} F_d) E_d^T = 0 \quad (40-2)$$

## ۲-۲-۵- مشاهده گر مقاوم به ورودی ناشناخته به کمک تکنیک $H_{\infty}$ :

روش‌های ارائه شده تا بدین جا به تضعیف اثر اغتشاش روی سیگنال مانده می‌پرداختند. در این بخش به ارائه روشی می‌پردازیم که به‌طور هم‌زمان اثر اغتشاش روی مانده را کاهش و در عین حال اثر عیب را روی آن افزایش می‌دهد. برای نشان دادن معیاری از اثر عیب روی مانده از  $H_{\infty}$  استفاده می‌کنیم که باید ماکزیمم شود و برای نشان دادن اثر اغتشاش روی مانده از  $H_{\infty}$  استفاده می‌کنیم.

معیار  $H_{\infty}$  و  $H_2$  به صورت زیر تعریف می‌شود [۳۰].

$$H_2 = \|T_{rf}\|_2 = \inf_{\omega} \sigma(T_{rf}(j\omega)) > \beta_0 \quad (41-2)$$

$$H_{\infty} = \|T_{rd}\|_{\infty} = \sup_{\omega} \bar{\sigma}(T_{rd}(j\omega)) < \gamma_0 \quad (42-2)$$

$T_{rf}$  و  $T_{rd}$  توابع انتقال از  $f$  و  $d$  به  $r$  هستند که  $d = [v \ n]^T$ .

باید در نظر داشت که در قید (۲-۴۱)،  $H_2$  در واقع یک نرم نیست. برای کاربرد تشخیص عیب،  $H_2$  به

صورت زیر تعریف می‌شود.

تعریف ۱-۲: برای سیستم داده شده به شکل  $y(s)=G(s)u(s)$  معیار  $H_2$  به صورت زیر است [23]:

اگر تعداد سطرهای  $G(s)$  از ستون‌های آن بیشتر باشد و  $\forall \omega, G(-j\omega)G^T(j\omega) > 0$ :

$$\|G(s)\|_2 = \min_{\omega} \underline{\sigma}(G^T(j\omega)) \quad (43-2)$$

اگر تعداد سطرهای  $G(s)$  از ستون‌های آن کمتر باشد و  $\forall \omega, G^T(-j\omega)G(j\omega) > 0$ :

$$\|G(s)\|_2 = \min_{\omega} \underline{\sigma}(G(j\omega)) \quad (44-2)$$

برای محاسبه این معیار، لم زیر را معرفی می‌کنیم.

لم ۱-۲: فرض کنید  $A, B, P, S, R$  ماتریس‌هایی با بعد مناسب باشند؛ در حالیکه  $P, R$  متقارن،  $R > 0$

و  $(A, B)$  پایدارپذیر باشند. فرض کنید دو فرض زیر برآورده شوند:

۱-  $A$  مقدار ویژه‌ای روی محور موهومی نداشته باشد.

۲-  $P$  نیمه مثبت معین یا نیمه منفی معین باشد در حالیکه  $(A, P)$  هیچ مد مشاهده پذیری روی محور

موهومی نداشته باشد.

آنگاه  $X$  یک پاسخ متقارن حقیقی منحصر به فرد برای معادله زیر است [24]:

$$(A - BR^{-1}S^T)^T X + X(A - BR^{-1}S^T) - XBR^{-1}B^T X + P - SR^{-1}S^T = 0 \quad (45-2)$$

و  $A - BR^{-1}S^T - BR^{-1}B^T X$  پایدار است.

تعریف ۲-۲: برای  $\gamma_1, \gamma_2, \beta$  و رویتگر (۲)، رویتگر تشخیص عیب  $H_2/H_\infty$  نامیده می‌شود اگر  $A -$

LC پایدار باشد و:

$$\|T_m(s)\|_\infty < \gamma_1 \quad (46-2)$$

$$T_m = C(sI - A - LC)^{-1}(B_n - LD_n) + D_n$$

$$\|T_{rv}(s)\|_{\infty} < \gamma_2 \quad (47-2)$$

$$T_{rv} = C(sI - A - LC)^{-1}(B_v - LD_v) + D_v$$

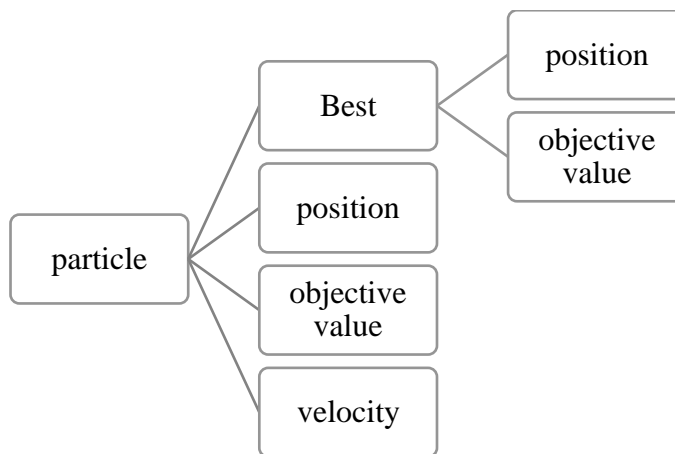
$$\|T_{rf}(s)\|_{\infty} > \beta \quad (48-2)$$

$$T_{rf} = C(sI - A - LC)^{-1}(B_f - LD_f) + D_f$$

قید (۴۶-۲) و (۴۷-۲) نشان دهنده‌ی اثر اغتشاشات در باقی‌مانده  $r$  هستند. درحالی‌که قید (۴۸-۲) حساسیت  $r$  نسبت به عیب را نشان می‌دهد. این سه معیار، مهم‌ترین ویژگی‌های روی‌نگر تشخیص عیب هستند [30].

## ۲-۲-۶- معرفی الگوریتم PSO

الگوریتم PSO جزو الگوریتم‌های فرا ابتکاری است؛ الگوریتم‌هایی که با کمترین اطلاعات، مساله را حل می‌کنند. در الگوریتم PSO، هر ذره<sup>۳۷</sup> چند خاصیت دارد:



شکل (۲-۲) خواص الگوریتم PSO [۳۱]

در برنامه نویسی، Best را به عنوان یک خاصیت مرکب در نظر خواهیم گرفت. موقعیت یا همان position، یک متغیر مستقل است. مقدار تابع هدف یا همان objective value، یک متغیر وابسته به موقعیت می‌باشد. تمام این خواص را در یک ساختار<sup>۳۸</sup> تعریف خواهیم کرد.

<sup>۳۷</sup> Particle



زمانی که ما با یک مساله بهینه‌سازی روبرو هستیم، در واقع تابع هدف را می‌خواهیم بهینه کنیم.

دو حالت را می‌توان متصور شد:

۱- بیشینه سازی<sup>۳۹</sup>؛ که در این صورت تابع هدف ما، تابع برازندگی<sup>۴۰</sup> نامیده می‌شود.

۲- کمینه سازی<sup>۴۱</sup>؛ که در اینجا تابع هدف، تابع هزینه<sup>۴۲</sup> نامیده می‌شود.

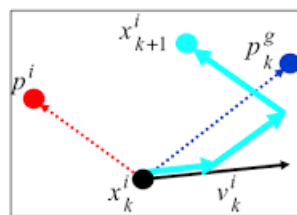
لذا از این پس هر جا از cost صحبت شد، منظور کمینه سازی تابع هدف می‌باشد.

معادلات دینامیک الگوریتم PSO به صورت زیر است:

$$V_i(t+1) = \omega * V_i(t) + c_1 * \text{rand}() * (P_i(t) - X_i(t)) + c_2 * \text{Rand}() * (g(t) - X_i(t)) \quad (۴۹-۲)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (۵۰-۲)$$

که در آن  $c_1$  و  $c_2$  دو ضریب ثابت در بازه ی  $[0 \ 2]$ ؛  $\omega$ ، ضریب اینرسی است و هرچه کمتر باشد بهتر است و الگوریتم سریع تر همگرا می‌شود که مقدار مناسب آن در بازه ی  $[0/4 \ 0/9]$  می‌باشد [31]؛  $\text{rand}()$  و  $\text{Rand}()$  دو عدد تصادفی متفاوت در بازه  $[0 \ 1]$  می‌باشند؛  $X_i(t)$  بیانگر موقعیت کنونی ذره  $i$  ام؛  $P_i(t)$ ، بهترین خاطره شخصی ذره  $i$  ام؛ اندیس  $g$  معرف بهترین خاطره جمعی و  $V_i(t)$ ، بیانگر سرعت کنونی ذره  $i$  ام می‌باشند.



شکل (۳-۲) شمای کلی عملکرد ذرات PSO [۳۱]

<sup>۳۸</sup> Structure

<sup>۳۹</sup> Maximization

<sup>۴۰</sup> fitness function

<sup>۴۱</sup> Minimization

<sup>۴۲</sup> cost function

روابط (۲-۴۹) و (۲-۵۰) مسیر حرکت ذرات را مشخص می‌کنند. رابطه‌ی (۲-۴۹) چگونگی تغییر سرعت و به روز شدن آن را نشان می‌دهد. به همین ترتیب رابطه‌ی (۲-۵۰) چگونگی به روز شدن موقعیت ذرات را نشان می‌دهد. رابطه‌ی (۲-۴۹) از سه قسمت تشکیل می‌شود. بخش اول، جمله‌ی تکانه است و بدین معنی است که سرعت نمی‌تواند به‌طور ناگهانی تغییر کند؛ سرعت همواره نسبت به سرعت فعلی، تغییر می‌کند.

جمله‌ی دوم، بخش شناختی<sup>۴۳</sup> است که نشان دهنده‌ی تفکر خود ذره است-تجربه‌ی حرکت انفرادی. جمله‌ی سوم، بخش اجتماعی<sup>۴۴</sup> است و بیانگر تعامل بین ذرات است-تجربه‌ی حرکت گروهی [۳۱].

ماکزیمم مقداری که سرعت می‌تواند به خود بگیرد، نباید از  $V_{max}$  تجاوز کند و مقدار آن توسط کاربر تنظیم می‌شود. اما در صورتی که سرعت در بُعدی از این حد خارج شد، باید به  $\pm V_{max}$  ، محدود<sup>۴۵</sup> شود. این پارامتر، پارامتر مهمی است؛ به‌طوری که اگر مقدار بزرگی برای آن در نظر گرفته شود، به عامل‌ها اجازه‌ی حرکت به نواحی دورتر را می‌دهد و امکان جستجوی ناحیه‌ی بیشتری از فضای جستجو را فراهم می‌آورد. و در مقابل انتخاب  $V_{max}$  کوچک، امکان جستجوی نواحی دورتر را از ما می‌گیرد و احتمال گرفتار شدن در مینیمم محلی را بالا می‌برد.

در الگوریتم‌های بهینه‌سازی، دو مفهوم را داریم:

- جستجو<sup>۴۶</sup> ، که توانایی تولید پاسخ‌های جدید می‌باشد. یعنی می‌تواند از پاسخ‌ها برای کشف فضاهایی که تا حالا مورد بررسی قرار نگرفته است، استفاده کند.

---

<sup>۴۳</sup> Cognitive part

<sup>۴۴</sup> Social part

<sup>۴۵</sup> clamp

<sup>۴۶</sup> Exploration

- بهره برداری<sup>۴۷</sup> ، توانایی پروراندن پاسخ‌های فعلی می‌باشد. یعنی هر پاسخ یک ایده‌ای هست که ما باید آن را پرورانیم و به آن شاخ و برگ دهیم و مطمئن شویم از آن بهره‌ی کافی را گرفتیم. الگوریتمی که این کار را انجام می‌دهد، می‌تواند از اطلاعاتی که در پاسخ‌ها نهفته است، بیشتر استفاده کند.

مراحل پیاده سازی الگوریتم PSO ساده به صورت زیر است [۳۱].

- ۱- تولید جمعیت اولیه ذرات با موقعیت و سرعت تصادفی.
  - ۲- برآورد تابع هزینه برای هر ذره.
  - ۳- مقایسه‌ی هزینه‌ی هر ذره با بهترین تجربه‌ی خودش (pbest). اگر مقدار کنونی از pbest بهتر بود، Pbest برابر مقدار کنونی می‌شود و  $P_i$  مساوی موقعیت کنونی  $X_i$  قرار داده می‌شود.
  - ۴- بهترین ذره در کل جمعیت، با نماد g مشخص می‌شود.
  - ۵- به روز شدن بهترین تجربه‌ی جمعی.
  - ۶- به روز شدن سرعت و موقعیت ذره‌ها با توجه به روابط (۲-۴۹) و (۲-۵۰).
  - ۷- بازگشت به مرحله‌ی ۲ تا هنگامی که شرط توقف برآورده شود.
- در مورد شرایط توقف می‌توانیم چند حالت را در نظر بگیریم:

- رسیدن به حد قابل قبولی از پاسخ
- سپری شدن تعداد تکرار/ زمان مشخص
- سپری شدن تعداد تکرار/ زمان، بدون مشاهده‌ی بهبود خاصی در نتیجه
- سپری شدن تعداد مشخصی از پاسخ‌ها

برخی از کاربردهای الگوریتم PSO عبارتند از:

---

<sup>۴۷</sup> Exploitation

- حل انواع مسائل بهینه‌سازی پیوسته و گسسته
- مدل‌سازی و کنترل
- پردازش انواع سیگنال‌های دیجیتالی و تشخیص الگو
- مدیریت سیستم‌های تولید و توزیع نیرو
- طراحی و بهینه‌سازی شبکه‌های ارتباطی

## ۲-۲-۷- معرفی الگوریتم MOPSO

مسائل بهینه‌سازی از نظر تعداد توابع هدف و معیارهای بهینه‌سازی، به دو نوع تقسیم پذیر هستند: (۱) مسائل بهینه‌سازی تک هدفه و (۲) مسائل بهینه‌سازی چند هدفه. در مسائل بهینه‌سازی تک هدفه، هدف از حل مسأله، بهبود یک شاخص عملکرد<sup>۴۸</sup> یگانه است که مقدار کمینه یا بیشینه آن، کیفیت پاسخ به دست آمده را به طور کامل منعکس می‌کند. اما در برخی موارد، نمی‌توان صرفاً با اتکا به یک شاخص، یک پاسخ فرضی برای مسأله بهینه‌سازی را امتیازدهی نمود. در این نوع مسائل، ناگزیریم که چندین تابع هدف یا شاخص عملکرد را تعریف نماییم و به طور همزمان، مقدار همه آن‌ها را بهینه کنیم.

بهینه‌سازی چند هدفه، یکی از زمینه‌های بسیار فعال و پرکاربرد تحقیقاتی در میان مباحث بهینه‌سازی است. غالباً بهینه‌سازی چند هدفه<sup>۴۹</sup>، به نام‌های بهینه‌سازی چند معیاره<sup>۵۰</sup> و بهینه‌

<sup>۴۸</sup> Performance Index

<sup>۴۹</sup> Multi-objective Optimization

<sup>۵۰</sup> Multi-criteria Optimization

سازی برداری<sup>۵۱</sup> نیز شناخته می شود. روش های فراوانی تا کنون برای حل این مسائل ارائه شده اند که در حالت کلی می توان آن ها را به دو دسته تقسیم نمود:

- روش های کلاسیک، که اغلب مسأله چند هدفه را به یک مسأله یک هدفه تقلیل می دهند،
- روش های تکاملی، که اغلب مسأله بهینه سازی چند هدفه را واقعا به صورت چند هدفه حل می نمایند.

تفاوت اصلی در MOPSO با PSO در تعیین بهترین ذره در جمعیت و همچنین تعیین بهترین خاطرهی شخصی هر ذره می باشد. در الگوریتم MOPSO یک مفهوم جدید به نام آرشیو نسبت به حالت تک هدفه ارائه شده است که در واقع محل نگهداری پاسخ های نامغلوب است. با تعریف آرشیو در این الگوریتم، مفهوم بهترین ذره در جمعیت نیز تغییر پیدا کرده است. مراحل اصلی الگوریتم MOPSO به صورت زیر خلاصه نمود [۳۲]:

- ۱- ایجاد جمعیت اولیه و مقداردهی اولیهی بردارهای سرعت و مکان هر ذره.
- ۲- محاسبه توابع هزینه برای ذرات.
- ۳- پیدا کردن اعضای نامغلوب جمعیت و ذخیره کردن این ذرات در آرشیو.
- ۴- جدول بندی فضای هدف کشف شده.
- ۵- هر یک از ذرات، از آرشیو یک رهبر انتخاب کرده و به سمت آن حرکت خود را انجام می دهد. همانند PSO تک هدفه، حرکت هر ذره نیاز به به روز شدن سرعت و موقعیت ذره دارد. با این تفاوت که مفاهیم بهترین ذره در کل جمعیت و بهترین خاطرهی شخصی هر ذره متفاوت با حالت تک هدفه است.

---

<sup>۵۱</sup> Vector Optimization

حرکت هر ذره به سوی رهبر انتخاب شده از آرشیو را می توان به صورت مراحل زیر در نظر گرفت.

- سرعت هر ذره با استفاده از رابطه ی زیر به روز می شود.

$$V_i(t+1) = \omega * V_i(t) + c_1 * \text{rand}() * (P_i(t) - X_i(t)) + c_2 * \text{Rand}() * (\text{Rep}_h(t) - X_i(t)) \quad (51-2)$$

که در آن  $c_1$  و  $c_2$  دو ضریب ثابت در بازه ی  $[0, 2]$ ؛  $\omega$ ، ضریب اینرسی است و هرچه کمتر باشد بهتر است و الگوریتم سریع تر همگرا می شود که مقدار مناسب آن در بازه ی  $[0.4, 0.7]$  می باشد [31]؛  $\text{rand}()$  و  $\text{Rand}()$  دو عدد تصادفی متفاوت در بازه  $[0, 1]$  می باشند؛  $X_i(t)$  بیانگر موقعیت کنونی ذره  $i$  ام؛  $P_i(t)$ ، بهترین خاطره شخصی ذره  $i$  ام؛  $V_i(t)$ ، بیانگر سرعت کنونی ذره  $i$  ام می باشند. اندیس  $h$ ، مقداری است که از آرشیو به عنوان رهبر انتخاب شده است. برای انتخاب  $h$ ، در ابتدا، فضای هدف توسط فوق جدول بندی، افزای می شود. سپس با توجه به تعداد ذراتی که در هر خانه ی جدول وجود دارد، احتمال انتخاب محاسبه شده و یا به عبارت دیگر یک ضریب شایستگی به آن خانه اختصاص داده می شود. هر خانه، با تعداد ذرات کمتر، ضریب شایستگی بیش تری خواهد داشت. در این پژوهش برای تخصیص این احتمالات به خانه های جدول، روش بولتزمن<sup>۵۲</sup> پیشنهاد شده است که در این روش از رابطه زیر به منظور تخصیص احتمالات به خانه های جدول، استفاده می شود.

$$P_i = \frac{e^{-\beta n_i}}{\sum e^{-\beta n_j}} \quad (52-2)$$

که در آن  $n_i$ ، تعداد ذرات موجود در خانه  $i$  ام است و  $\beta$  فشار انتخاب می باشد. اگر  $\beta = 0$  باشد، تمام خانه ها با احتمال یکسان انتخاب می شوند و اگر  $\beta = 1$  باشد، تنها خانه هایی که شامل یک ذره می باشند، انتخاب خواهند شد.

<sup>۵۲</sup> Boltzman

- مکان هر ذره با استفاده از رابطه زیر به روز می‌شود.

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (۵۳-۲)$$

پس از به‌روز کردن مکان ذرات، در صورتی که این ذرات از محدوده‌ی فضای جستجو خارج شوند، بردار سرعت قرینه می‌شود.

- محاسبه‌ی هزینه‌ی ذراتی که مکان آن‌ها به روز شده است. همچنین اعضای نامغلوب جمعیت کنونی به آرشیو اضافه می‌شوند و اعضای قبلی آرشیو که توسط اعضای جدید مغلوب می‌شوند، از آرشیو حذف می‌شوند. همچنین با توجه به محدودیت‌های سخت‌افزاری در پیاده‌سازی، نیاز است که یک ظرفیت برای آرشیو در نظر گرفته شود. در صورتی که تعداد اعضای آرشیو از این ظرفیت بیش‌تر شود، با استفاده از یک معیار ثانویه، اعضای اضافی حذف می‌شوند. معیار ثانویه در این الگوریتم، طوری در نظر گرفته شده است که میزان نظم در پراکندگی ذرات نامغلوب در نواحی فضای هدف را بهبود می‌بخشد. برای تحقق این موضوع، ذراتی که در نواحی با تراکم بیش‌تر ذرات هستند، دارای احتمال حذف بیش‌تر و ذرات در نواحی خلوت‌تر احتمال حذف کمتری خواهند داشت. برای انجام این عمل همانند انتخاب خانه‌ها، یک دسته احتمال حذف تعیین و با استفاده از چرخه‌ی رولت<sup>۵۳</sup>، اعضای که باید حذف شوند، انتخاب می‌شوند.
- به‌روز کردن بهترین خاطره‌ی شخصی هر ذره، با توجه به اینکه در بهینه‌سازی چندهدفه، پاسخ‌ها نسبت به هم می‌توانند سه حالت متفاوت را داشته باشند.

در مقایسه‌ی موقعیت جدید ذره با بهترین خاطره‌ی شخصی‌اش نیز سه حالت زیر پیش می‌آید:

- اگر موقعیت جدید، بهترین خاطره‌ی شخصی را مغلوب کند، آن‌گاه موقعیت جدید، جای بهترین خاطره‌ی شخصی را می‌گیرد.

---

<sup>۵۳</sup> Roulette wheel

- اگر موقعیت جدید توسط بهترین خاطره‌ی شخصی مغلوب شود، هیچ تغییری رخ نمی‌دهد.
- اگر موقعیت جدید و بهترین خاطره‌ی شخصی نسبت به هم بی‌اثر باشند، یکی از آن‌ها به تصادف به عنوان بهترین خاطره انتخاب می‌شود.

۵- در صورتی که شرط خاتمه محقق نشود، به مرحله چهار رفته و در غیر این صورت الگوریتم خاتمه می‌یابد.



## فصل سوم

### شبیه‌سازی و آنالیز

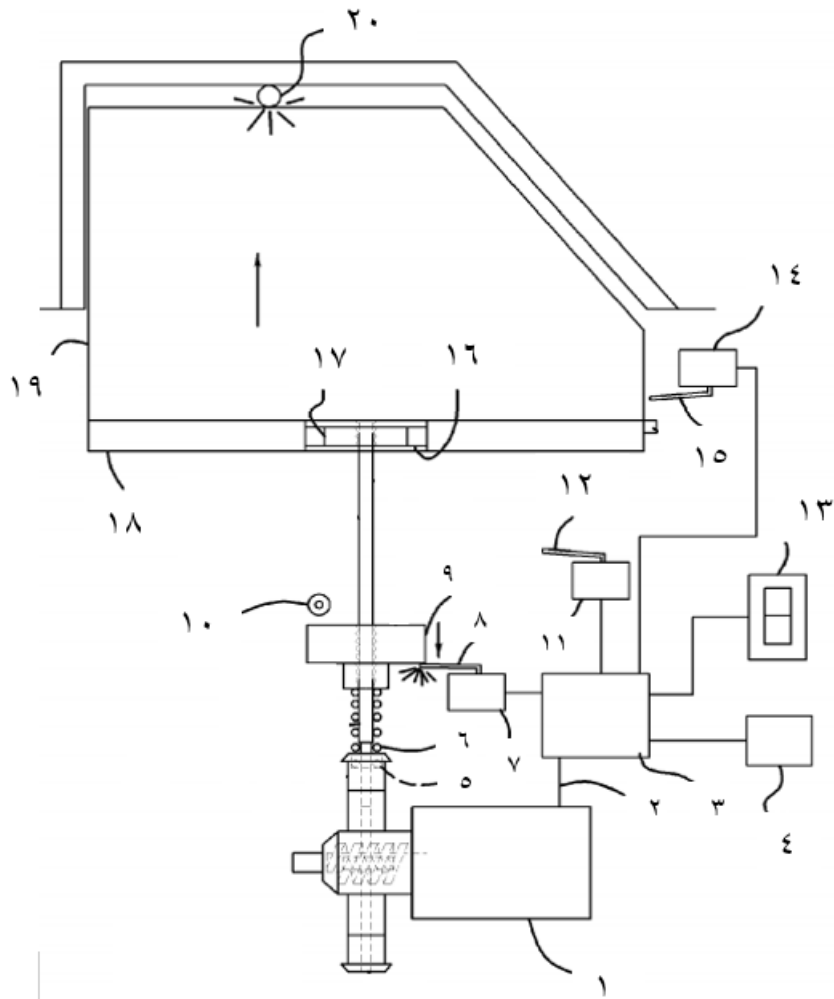
### ۳-۱- معرفی مدل

در بیشتر خودروها، بالا بردن پنجره با استفاده از یک اهرم بسیار دقیق انجام می‌گیرد که همزمان آن را تراز نیز نگه می‌دارد. یک موتور الکتریکی کوچک به یک چرخ‌دنده مارپیچ و چند چرخ‌دنده گرد متصل است تا نیروی بالابر کافی برای بالا بردن پنجره را تولید کند.

یکی از ویژگی‌های مهم پنجره‌های برقی این است که نمی‌توان آنها را با فشار باز کرد؛ چرخ‌دنده مارپیچ مانع این کار می‌شود. بیشتر چرخ‌دنده‌های مارپیچ به خاطر زاویه تماس بین مارپیچ و دندانه‌های چرخ‌دنده گرد، به صورت خودکار قفل می‌شوند. چرخ‌دنده مارپیچ می‌تواند چرخ‌دنده گرد را بچرخاند ولی چرخ‌دنده گرد نمی‌تواند مارپیچ را به چرخش درآورد. اصطکاک بین دندانه‌ها باعث می‌شود چرخ‌دنده‌ها قفل شوند. اهرم بازوی بلندی دارد که به میله‌ای که پایین پنجره را نگه می‌دارد، متصل است. وقتی پنجره بالا می‌رود، انتهای بازو داخل شکافی روی میله می‌لغزد. از طرف دیگر، در انتهای میله، صفحه بزرگی قرار دارد که دندانه‌های چرخ‌دنده داخل آن فرو می‌روند. موتور هم چرخ‌دنده‌ای را که با این دندانه‌ها درگیر است، می‌چرخاند.

اتومبیل‌های نسل جدید، با پنجره‌های برقی مجهز شده‌اند که تلاش مسافر برای حرکت پنجره‌های خودرو را کاهش می‌دهد [۳۳]. اگر شیئی سر راه پنجره قرار بگیرد، به‌طور مثال، دست کودک، پنجره باید در لحظه‌ی برخورد با آن متوقف شود. یکی از روش‌هایی که خودروسازان برای کنترل نیروی پنجره مورد استفاده قرار می‌دهند، طراحی یک مدار است که گشتاور موتور بالابر را ارزیابی می‌کند. اگر گشتاور موتور کم شود، مدار جریان برق ورودی به موتور را برعکس می‌کند و به این ترتیب پنجره به پایین برمی‌گردد. اما مساله اینجاست که مدت زمان بین وقوع عیب و تشخیص عیب چقدر است. در واقع مدت زمان کمتر، مطلوب مساله ما می‌باشد چراکه آسیب کمتری به مانع وارد می‌شود.

شمای کلی پنجره برقی در شکل (۳-۱) نشان داده شده است.



شکل (۱-۳) شمای کلی پنجره برقی. ۱ موتور، ۲ خط ولتاژ ورودی موتور، ۳ مدار الکتریکی، ۴ کلید فلیپ فلاپ، پایه ۵ برای دریافت و نشستن یک انتهای فنر ۶، ۸ محرک کلید ۷ جهت حرکت معکوس پنجره، ۱۵ محرک کلید ۱۴ جهت بالا رفتن پنجره، ۱۲ محرک کلید ۱۱ جهت پایین آمدن پنجره، ۹ پایه، ۱۰ غلتک، ۱۳ کلید باز و بسته کردن پنجره، ۱۶ اقرقه ها، ۱۷ چرخنده، ۱۸ قاب پنجره، ۱۹ پنجره، ۲۰ مانع [۳۴].

سیستم خطی نامتغیر با زمان زیر را در نظر بگیرید:

$$\dot{x}=Ax+Bu+E_f f+E_v v+E_n n \quad (۱-۳)$$

$$y=C_2 x+D_2 u+F_f f+F_v v+F_n n \quad (۲-۳)$$

که  $x$  بردار حالت،  $u$  ورودی کنترل،  $y$  بردار اندازه‌گیری،  $n$  نویز سفید،  $v$  اغتشاش انرژی محدود ناشناخته،  $f$  بردار عیب ورودی می‌باشد.

مولد باقی‌مانده برپایه فیلتر تشخیص عیب به صورت زیر مشخص شده است:

$$\hat{\dot{x}} = (A - LC_2)\hat{x} + (B - LD_2)u + Ly \quad (3-3)$$

$$\hat{y} = C_2\hat{x} + D_2u \quad (4-3)$$

$$r = y - \hat{y} \quad (5-3)$$

$L$  ماتریس گین رویتگر است. با استفاده از (3-1) تا (3-5)، فرم زیر بدست می‌آید:

$$\dot{e} = (A - LC_2)e + (E_f - LF_f)f + (E_v - LF_v)v + (E_n - LF_n)n \quad (6-3)$$

$$r = C_2e + F_ff + F_vv + F_nn \quad (7-3)$$

دینامیک سیگنال  $r$  تنها به  $v, n, f$  بستگی ندارد بلکه تحت تاثیر حالت  $e$  نیز هست. با این حال،  $r$  به‌طور کامل از ورودی  $u$  مستقل است. معادله (7-3) در قالب یک تابع انتقال به صورت زیر بیان می‌شود:

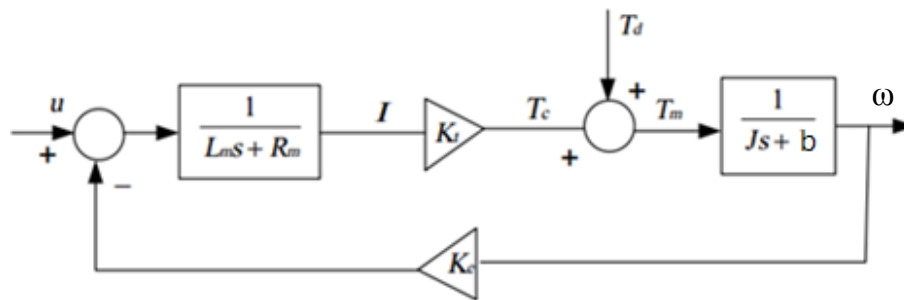
$$r(s) = T_{rf}f(s) + T_{rn}n(s) + T_{rv}v(s) \quad (8-3)$$

که  $T_{rf}$  و  $T_{rv}$  و  $T_{rn}$  به ترتیب توابع انتقال از  $f, v, n$  به  $r$  هستند.

### ۳-۲- مدل تحلیلی از الگوریتم تشخیص عیب

برای تشخیص عیب مبتنی بر مدل تحلیلی، ابتدا مدل فضای حالت دقیق برای سیستم کنترل ضد

پینچ تخمین زده می‌شود که در شکل (3-2) نشان داده شده است.



شکل (۲-۳) مدل موتور خطی سازی شده [۳۰]

$\omega$  سرعت زاویه ای،  $u$  ولتاژ موتور،  $I$  جریان آرماتور،  $T_m$  گشتاور چرخشی،  $T_d$  گشتاور اغتشاش،  $T_c$  گشتاور کنترل،  $L_m$  اندوکتانس آرماتور،  $R_m$  مقاومت آرماتور،  $J$  ممان اینرسی،  $B$  ضریب اصطکاک ویسکوزیته،  $K_e$  ضریب نیروی الکتریکی بازگشتی،  $K_t$  ضریب گشتاور می باشد.

تابع انتقال از گشتاور چرخشی به سرعت زاویه ای برابر است با [۳۰]:

$$\frac{\omega(s)}{T_m} = \frac{1}{Js+b} \quad (۹-۳)$$

$T_v$  گشتاور ارتعاش،  $T_p$  گشتاور پینچ،  $T_w$  گشتاور بار، کلاس بندی شده است. بنابراین گشتاور چرخشی به شرح زیر نوشته شده است:

$$T_m = T_c - T_d = T_c - T_p - T_w - T_v \quad (۱۰-۳)$$

گشتاور ارتعاش با وضعیت جاده متفاوت است و این مشکلی در تعریف سرعت اضافه می کند. برای حل این مشکل، سرعت به اغتشاش انرژی محدود کلاس بندی شده است.

نرخ سرعت زاویه ای موتور به شرح زیر است:

$$\dot{\omega} = -\frac{b}{J}\omega + \frac{1}{J}T_c - \frac{1}{J}(T_p + T_w) + u_v \quad (۱۱-۳)$$

از آنجا که دینامیک الکتریکی موتور سریع تر از مکانیک است، گشتاور کنترل به شرح زیر تخمین زده شده است:

$$T_c \approx \frac{k_t}{R_m} (u - k_e \omega) \quad (12-3)$$

ضریب اصطکاک  $b$  تاثیر کمتری بر روی گشتاور دارد، بنابراین نادیده گرفته شده است. با انتخاب سرعت زاویه ای  $(\omega)$  به عنوان یک متغیر حالت، داریم:

$$\dot{\omega} = -\frac{k_e k_t}{J R_m} \omega - \frac{1}{J} (T_p + T_w) + u_v + \frac{k_t}{J R_m} u \quad (13-3)$$

$T = T_p + T_w$  نیز به عنوان متغیر حالت دوم در نظر گرفته شده است. نرخ گشتاور نیز به عنوان متغیر حالت سوم افزوده می شود:

$$\dot{T} = u_{Td} \quad (14-3)$$

بنابراین داریم [30]:

$$A = \begin{bmatrix} -\frac{k_e k_t}{J R_m} & -\frac{1}{J} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{k_t}{J R_m} \\ 0 \\ 0 \end{bmatrix}, C_2 = [1 \quad 0 \quad 0], D_2 = [0] \quad (15-3)$$

پارامترهای موتور DC با آزمایش بدست آمدند که در جدول (1-3) آمده است [30]:

جدول (1-3) مقادیر نامی پارامترهای موتور [30]

پارامترهای موتور	مقدار
$R_m$	0.85 [ $\Omega$ ]
$L_m$	0.649 [mH]
$K_e$	0.1204 [V/s/rad]
$K_t$	0.1204 [V/s/rad]
$T_n$	$9.3 \times 10^{-3}$ [s]
$J$	$1.586 \times 10^{-4}$ [kg/m <sup>2</sup> ]
$V_c$	12 [V]

با قرار دادن داده های جدول فوق در ماتریس های  $A, B, C_2, D_2$  داریم:

$$A = \begin{bmatrix} -107/530 & -6305/170 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 893/109 \\ 0 \\ 0 \end{bmatrix}, C_2 = [1 \quad 0 \quad 0], D_2 = [0]$$

جهت انجام تشخیص عیب مطابق معادلات (3-1) و (3-2) ماتریس‌های  $E_n, E_v, E_f, F_n, F_v, F_f$  را به صورت زیر انتخاب می‌کنیم [30]:

$$E_n = \begin{bmatrix} 893/109 \\ 0 \\ 0 \end{bmatrix}, E_v = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, E_f = \begin{bmatrix} 893/109 \\ 0 \\ 0 \end{bmatrix}, F_n = [0/05], F_v = [0], F_f = [1]$$

### ۳-۳- شبیه‌سازی سیستم به کمک حل الگوریتم PSO

می‌خواهیم مساله‌ی طراحی کنترل کننده مقاوم را به یک مساله بهینه‌سازی تبدیل کنیم و سپس با استفاده از الگوریتم PSO مساله را حل نماییم.

سیستم را به گونه‌ای طراحی می‌کنیم که در مقابل هر نوع عدم قطعیتی بتواند مقاومت کند.

معادله حالت سیستم خطی را به صورت زیر در نظر می‌گیریم:

$$\dot{x} = Ax + Bu + Ew \quad (۱۶-۳)$$

که در آن  $w = \begin{bmatrix} f \\ v \\ n \end{bmatrix}$ ،  $E = [E_f \quad E_v \quad E_n]$  و معادله خروجی به صورت زیر است:

$$z = C_1 x + F_1 w + D_1 u \quad (۱۷-۳)$$

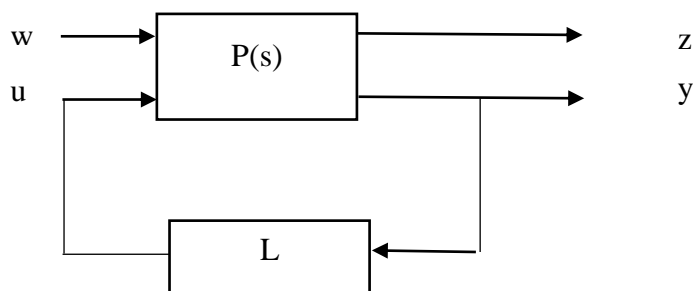
معادله‌ی اندازه‌گیری، مطابق رابطه‌ی (۱۸-۳) می‌باشد:

$$y = C_2 x + F_2 w + D_2 u \quad (۱۸-۳)$$

که در آن  $F_2 = [F_f \quad F_v \quad F_n]$  می‌باشد.

هدف این است که با اعمال یک  $u$  به صورت مناسب، این سیستم را به سمتی هدایت کنیم که

مقادیر  $z$  به صفر میل کند.



شکل (۳-۳) مدل حلقه بسته سیستم

بنابراین داریم:

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & E & B \\ C_1 & F_1 & D_1 \\ C_2 & F_2 & D_2 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

مدل تابع تبدیل، به صورت زیر می باشد:

$$P(s) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} [E \quad B] + \begin{bmatrix} F_1 & D_1 \\ F_2 & D_2 \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \quad (۱۹-۳)$$

تابع تبدیل حلقه بسته ورودی  $w$  و خروجی  $z$ ، از رابطه (۲۰-۳) بدست می آید:

$$T_{zw}(s) = P_{11} + P_{12}L(I - P_{22}L)^{-1}P_{21} \quad (۲۰-۳)$$

از آنجایی که  $T_{zw}(s)$  به صورت چند متغیره هست، از  $\|T_{zw}(s)\|$  استفاده می کنیم.

بنابراین مساله طراحی کنترل کننده با بهره ثابت را به یک مساله بهینه سازی تبدیل کردیم.

به کمک الگوریتم PSO،  $L$  را به نحوی می یابیم که  $\|T_{zw}(s)\|$  کمینه شود. در واقع  $z$  باید مستقل از

$w$ ، صفر بماند؛ اگر بتوانیم این را تامین کنیم، یک کنترل کننده مقاوم داریم.

هدف طراحی در سیستم انتخابی ما این است که نرخ سرعت زاویه ای موتور ( $\dot{\omega}$ )، را به صفر برسانیم. در

واقع در سیستم انتخابی ما  $Z \propto \dot{\omega}$  می باشد.

بنابراین معادلات (۱۶-۳)، (۱۷-۳) و (۱۸-۳) به صورت زیر بازنویسی می شوند:



$$\begin{bmatrix} \dot{\omega} \\ \dot{T} \\ \dot{T} \end{bmatrix} = \begin{bmatrix} -107/5 & -6305/2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ T \\ \dot{T} \end{bmatrix} + \begin{bmatrix} 893/1 & 1 & 893/1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f \\ v \\ n \end{bmatrix} + \begin{bmatrix} 893/1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ u_v \\ u_{Td} \end{bmatrix} \quad (21-3)$$

$$z = [-107/5 \quad -6305/2 \quad 0] \begin{bmatrix} \omega \\ T \\ \dot{T} \end{bmatrix} + [\alpha \quad 1 \quad 893/1] \begin{bmatrix} f \\ v \\ n \end{bmatrix} + [893/1 \quad 1 \quad 0] \begin{bmatrix} u \\ u_v \\ u_{Td} \end{bmatrix} \quad (22-3)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} \omega \\ T \\ \dot{T} \end{bmatrix} + [893/1088 \quad 0 \quad 0] \begin{bmatrix} f \\ v \\ n \end{bmatrix} + [0 \quad 0 \quad 0] \begin{bmatrix} u \\ u_v \\ u_{Td} \end{bmatrix} \quad (23-3)$$

که در معادله (۲۲-۳)،  $\alpha \in [890 \quad 896]$  عدم قطعیت می‌باشد و  $\alpha_0 = 893/1$ .

جعبه ابزار شبیه‌سازی مورد استفاده برای ساخت سیستم تشخیص عیب مقاوم در شکل (۴-۳)

نشان داده شده است.

در شبیه‌سازی،  $v$  سیگنال ضربه با تاخیر ۰ ثانیه، دامنه ۵۰ و عرض ۰/۳ ثانیه است. علاوه بر این،

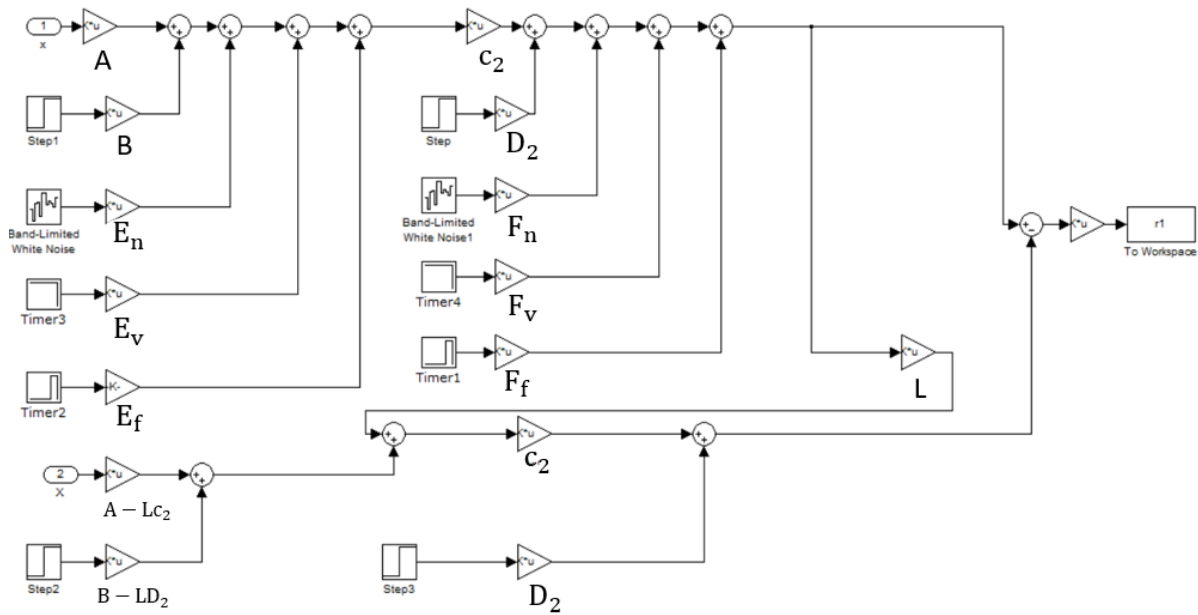
نویز سفید با قدرت ۰/۰۰۰۱۴ شبیه‌سازی شده است و فرکانس نمونه برداری اش ۰/۰۱ می‌باشد.

$f$  سیگنال ضربه با تاخیر ۲/۷ ثانیه، دامنه ۱ و عرض ۱ ثانیه می‌باشد. حال با استفاده از الگوریتم PSO

در نرم‌افزار متلب، گین رویتر را بدست می‌آوریم. جدول (۲-۳)، مقادیر پارامترهایی که در الگوریتم

PSO به کار رفته‌اند را نشان می‌دهد که در هر تکرار الگوریتم،  $\omega$ ، به صورت  $\omega = \omega * \omega_{damp}$  به روز

می‌شود.



شکل (۳-۴) جعبه ابزار شبیه‌سازی پنجره ضد پینچ

جدول (۳-۲) مقادیر پارامترهای مورداستفاده در الگوریتم PSO

مقدار	پارامترهای الگوریتم PSO
۳	تعداد متغیرهای تصمیم
[-۱ ۱]	حد پایین و بالای هر متغیر تصمیم
۲۵	اندازه جمعیت
۳۵	تعداد تکرار الگوریتم
۱	$\omega$
۰/۹۹	$\omega_{damp}$
۲	$c_1$
۲	$c_2$

### ۳-۳-۱- شبیه‌سازی سیستم با استفاده از معیار $H_2$

در این روش به کمک الگوریتم PSO، بهره‌رویتگر را به گونه‌ای بدست آوردیم که معیار  $H_2$  را برآورده کند. درواقع، تابع هزینه در این روش، به صورت رابطه‌ی (۳-۲۴) در نظر گرفته شده است.

$$z = \|T_{zw}\|_2 \quad (۳-۲۴)$$

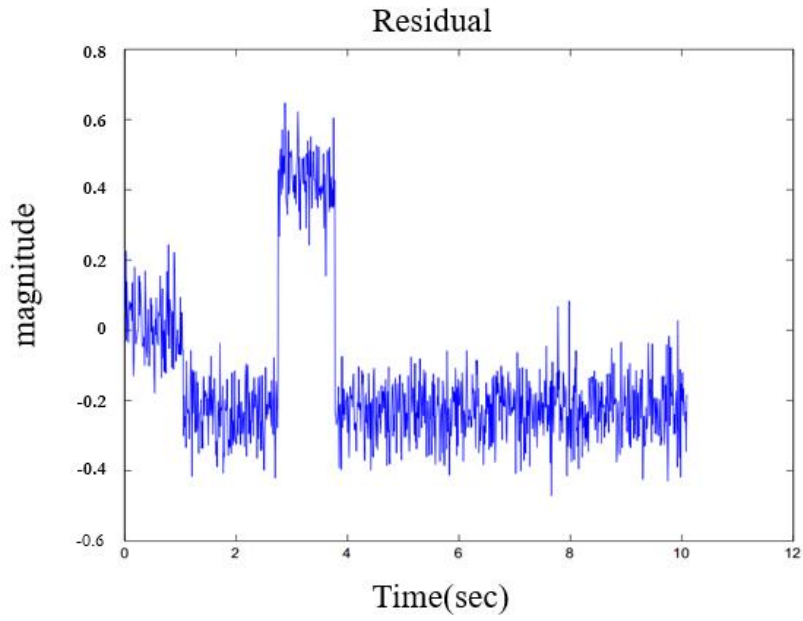
آنگاه بهره‌رویتگر عبارت است از:

$$L = \begin{bmatrix} 0/3418 \\ -0/7512 \\ 0/001 \end{bmatrix}$$

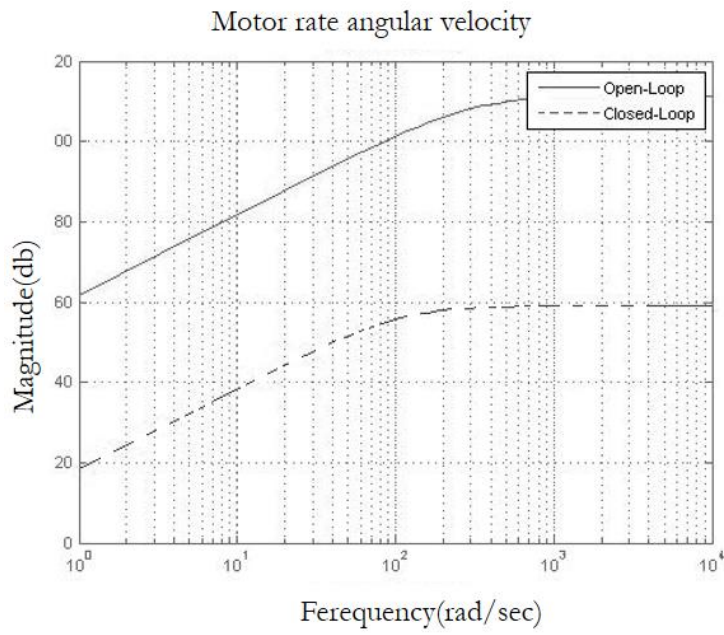
نتایج شبیه‌سازی برای رویتگر بدست آمده به صورت زیر است.

شکل (۳-۵)، پاسخ فرکانسی حلقه باز و حلقه بسته را نشان می‌دهد. همانطور که مشاهده می‌شود، باوجود اینکه در سیستم حلقه بسته، کاهش دامنه در تمامی فرکانس‌ها را داریم اما باز هم در فرکانس‌های بالا، دامنه‌ی فرکانسی زیاد است و این مطلوب مساله ما نیست.

همانطور که در شکل (۳-۶) مشاهده می‌شود، مشخص است که روش  $H_2$ ، اثر اغتشاش روی سیگنال مانده را تضعیف کرده اما در جداسازی عیب از اغتشاش موفق نیست.



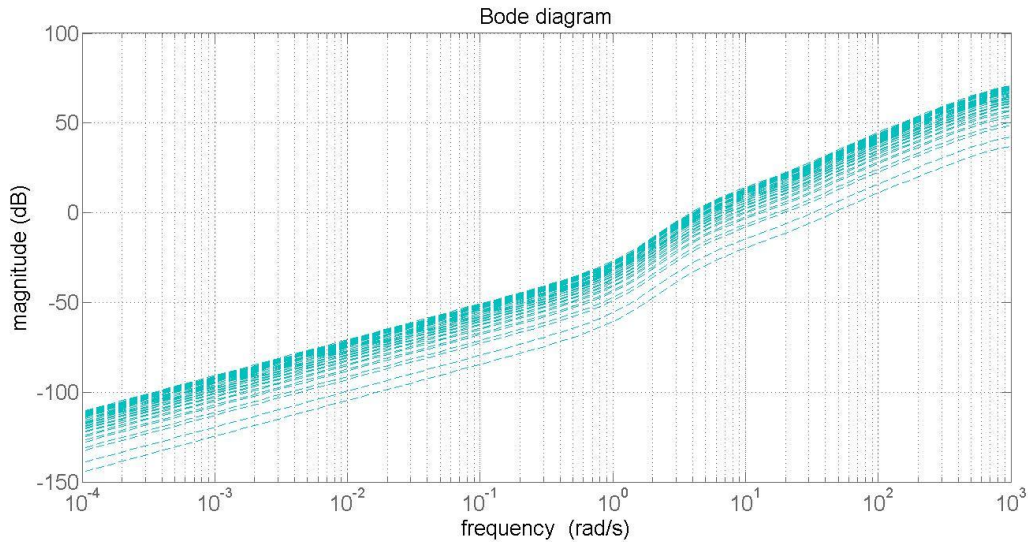
شکل (۵-۳) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته‌ی نرخ سرعت زاویه‌ای موتور در روش  $H_2$



شکل (۶-۳) سیگنال باقی‌مانده به روش  $H_2$  در حضور عیب و اغتشاش

### ۳-۳-۱-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_2$

همانطور که در بخش ۱-۲-۳-۵ اشاره شد، در این شبیه‌سازی، نامعینی از نوع جمعی است.



شکل (۷-۳) نمودار بود نامعینی جمعی

شکل (۷-۳)، نمودار بود نامعینی جمعی را نشان می‌دهد. تابع وزنی  $w_a$ ، همان پاسخ فرکانسی اختلاف سیستم نامعین از نامی آن است. با دستورات متلب، ۲۰ نقطه را از پوش نمودار فوق انتخاب می‌کنیم و  $w_a$  را با یک تابع مرتبه دو تخمین می‌زنیم و در آخر نیز تابع تبدیل آن را با توجه به نقاط انتخاب شده، به دست می‌آوریم:

$$w_a = \frac{5187 s^2 + 2177 s + 0/6778}{s^2 + 3623 s + 204900}$$

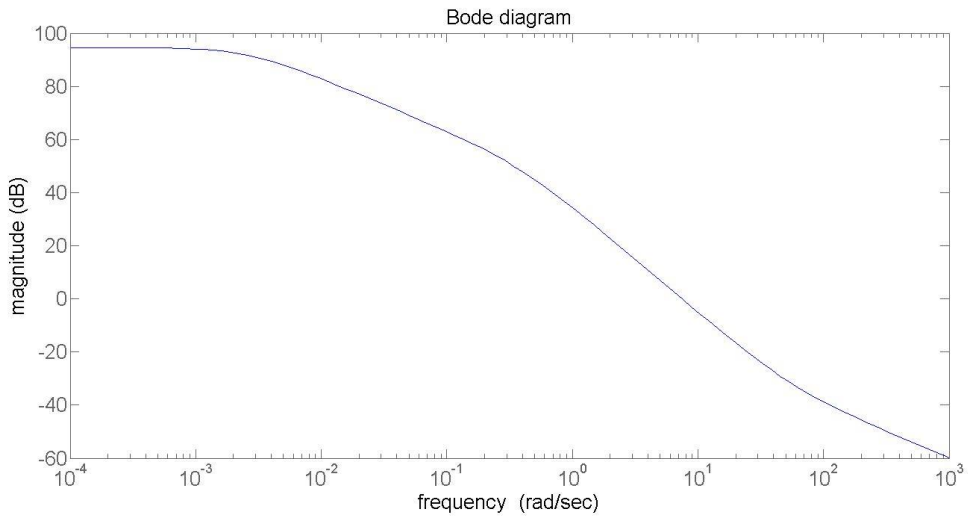
برای رسیدن به سطح مطلوب کاهش اغتشاش، برقرار شدن رابطه‌ی  $\|w_a S\|_\infty < 1$  ضروری

می‌باشد. که  $S$  تابع حساسیت سیستم حلقه بسته می‌باشد.

از این شرط داریم که برای تمامی فرکانس‌ها باید  $\| (I + GL)^{-1} \|_\infty < \frac{1}{w_a}$  شود. که برای رسیدن به

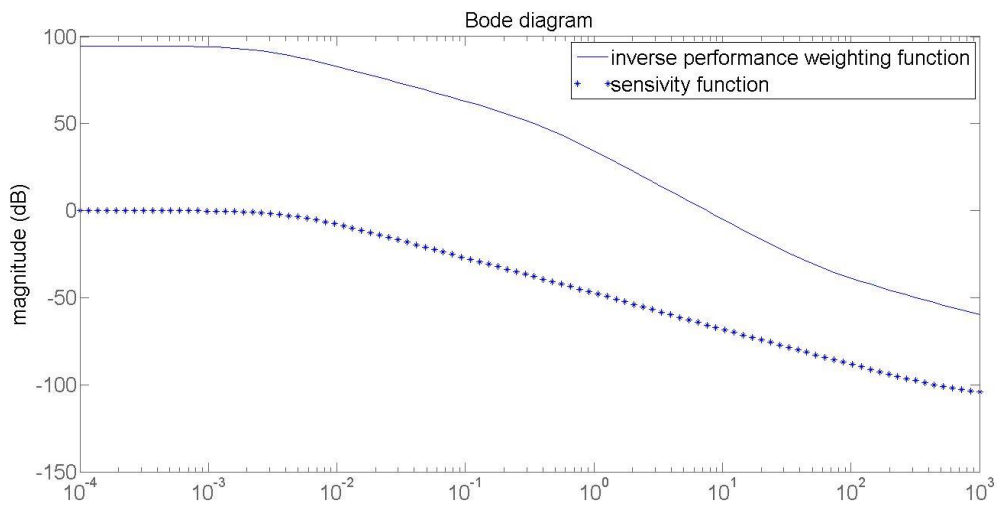
این شرط می‌توانیم بگوییم اگر و فقط اگر  $\left| \frac{1}{w_a} \right| < \sigma[(I + GL)^{-1}]$ .

مقادیر استثنایی  $\frac{1}{w_a}$ ، در شکل (۸-۳) نشان داده شده است.



شکل (۸-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H_2$

حال ، رفتار سیستم حلقه بسته با استفاده از کنترلر طراحی شده، در مرحله‌ی قبل را بررسی می‌کنیم که برای بررسی می‌توان تابع حساسیت حلقه بسته را با معکوس تابع وزنی مقایسه نمود. نتیجه‌ی این مقایسه در شکل (۹-۳) نشان داده شده است. همانطور که دیده می‌شود، در فرکانس‌های پایین، تابع حساسیت، زیر نمودار  $\frac{1}{w_a}$  قرار گرفته است.



شکل (۹-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H_2$  (خط گسسته)

### ۳-۳-۲- شبیه‌سازی سیستم با استفاده از معیار $H_{\infty}$

این بار برای کم کردن اثر اغتشاش از روش  $H_{\infty}$  استفاده می‌کنیم و تابع هزینه را به صورت زیر

در نظر می‌گیریم:

$$Z = \|T_{zw}\|_{\infty} \quad (۳-۲۵)$$

آنگاه بهره رویتگر عبارت است از:

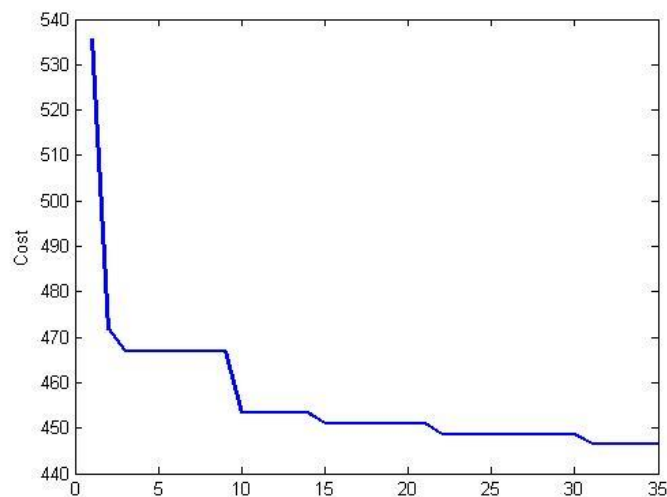
$$L = \begin{bmatrix} 0/1 \\ -2 \\ 0/03 \end{bmatrix}$$

نتایج شبیه‌سازی برای رویتگر بدست آمده به صورت زیر است.

شکل (۳-۱۰) تلقی ما از نحوه عملکرد فیدبک است و خیلی مهم نیست که چه عددی باشد اما

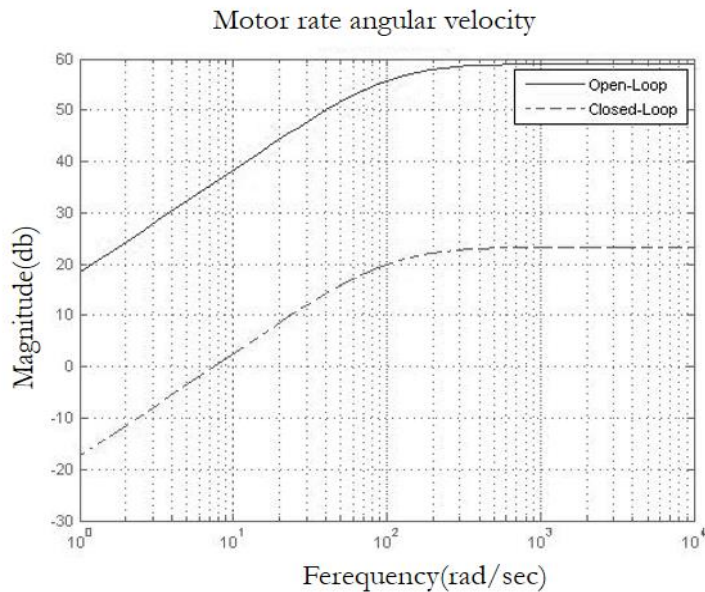
همگرا یا واگرا شدن آن برایمان مهم می‌باشد. همانطور که دیده می‌شود، پاسخ همگرا شده است.

الگوریتم پس از سپری کردن تعداد مشخصی تکرار، متوقف شده است.

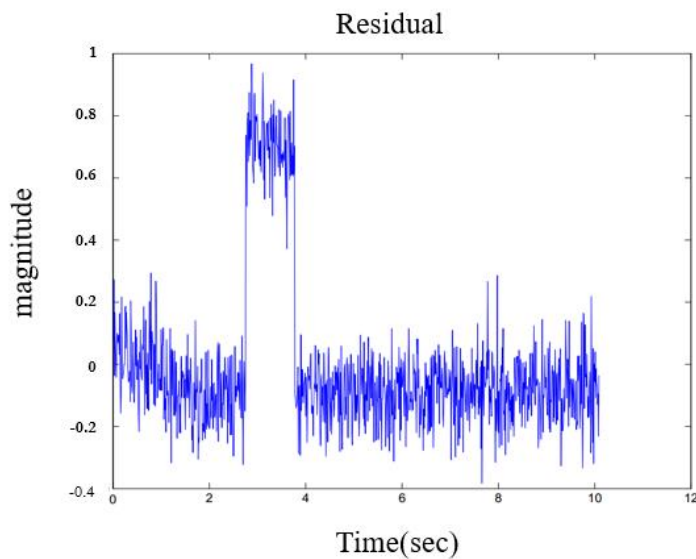


شکل (۳-۱۰) تابع هزینه‌ی الگوریتم PSO در روش  $H_{\infty}$

شکل (۱۱-۳) پاسخ فرکانسی حلقه باز و حلقه بسته را نشان می‌دهد. همانطور که مشاهده می‌شود، با وجود اینکه در سیستم حلقه بسته، کاهش دامنه در تمامی فرکانس‌ها را داریم اما باز هم در فرکانس‌های بالا، دامنه‌ی فرکانسی زیاد است و این مطلوب مساله ما نیست. مشاهده می‌شود که نسبت به روش  $H_2$  موفق‌تر عمل کرده و دامنه فرکانسی حدوداً به ۲۰ رسیده است.



شکل (۱۱-۳) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته‌ی نرخ سرعت زاویه‌ای موتور در روش  $H_\infty$



شکل (۱۲-۳) سیگنال باقی‌مانده به روش  $H_\infty$  در حضور عیب و اغتشاش



همانطور که در شکل (۳-۱۲) مشاهده می‌شود، در این روش همانند روش  $H_2$  ، اثر اغتشاش روی سیگنال مانده به خوبی تضعیف شده اما در عین حال اثر عیب روی سیگنال مانده حتی نسبت به روش  $H_2$  افزایش یافته و حتی نرخ رشد بالایی دارد.

### ۳-۳-۱-۲-۳-۳-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_\infty$

در این بخش از معیار  $H_\infty$  استفاده نموده و همانند بخش ۳-۳-۱-۱ ، با دستورات متلب، ۲۰ نقطه را از پوش شکل (۳-۷) انتخاب می‌کنیم و  $w_a$  را با یک تابع مرتبه دو تخمین می‌زنیم و در آخر نیز تابع تبدیل آن را با توجه به نقاط انتخاب شده، به دست می‌آوریم:

$$w_a = \frac{8414 s^2 + 4642 s + 1/382}{s^2 + 5032 s + 235900}$$

برای رسیدن به سطح مطلوب کاهش اغتشاش، برقرار شدن رابطه‌ی  $\| [w_a S] \|_\infty < 1$  ضروری می‌باشد. که در آن،  $S$  ، تابع حساسیت سیستم حلقه بسته می‌باشد. با توجه به این شرط، داریم که برای تمامی فرکانس‌ها باید:

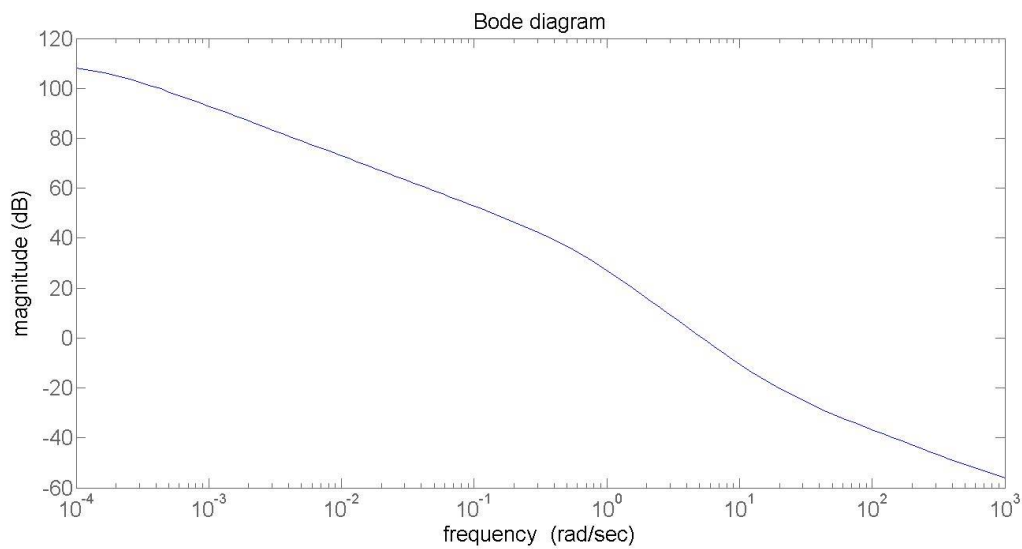
$$\| (I + GL)^{-1} \|_\infty < \frac{1}{w_a}$$

که برای رسیدن به این شرط می‌توانیم بگوییم اگر و فقط اگر:

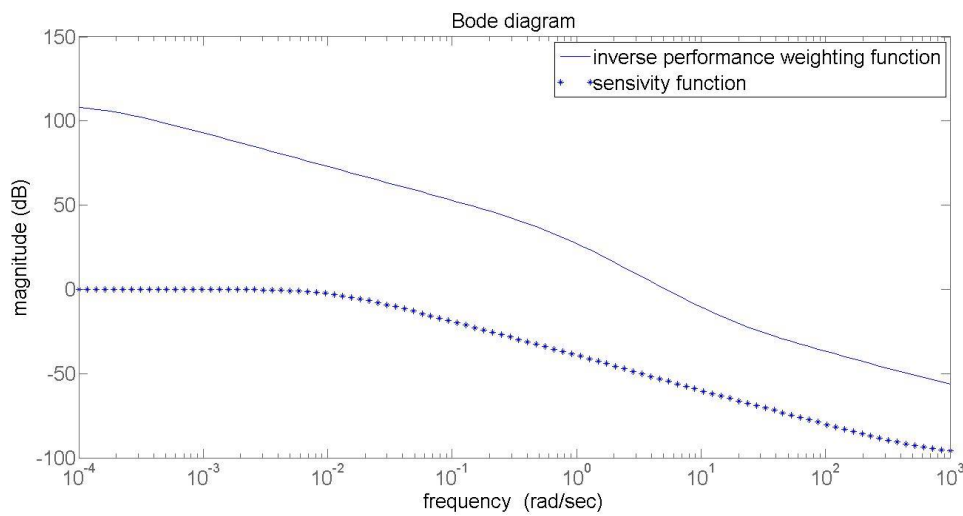
$$\sigma[(I + GL)^{-1}] < \left| \frac{1}{w_a} \right|$$

مقادیر استثنایی  $\frac{1}{w_a}$  ، در شکل (۳-۱۳) نشان داده شده است.

برای بررسی رفتار سیستم حلقه بسته با استفاده از کنترلر طراحی شده، در مرحله‌ی قبل، تابع حساسیت حلقه بسته را با معکوس تابع وزنی مقایسه می‌نماییم. نتیجه‌ی این مقایسه در شکل (۳-۱۴) نشان داده شده است. همانطور که دیده می‌شود، در فرکانس‌های پایین، تابع حساسیت، زیر نمودار  $\frac{1}{w_a}$  قرار گرفته است.



شکل (۱۳-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H_\infty$



شکل (۱۴-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H_\infty$  (خط گسسته)

۳-۴- شبیه سازی سیستم به کمک حل الگوریتم MOPSO

۳-۴-۱- شبیه سازی سیستم با استفاده از معیار  $H_2/H_\infty$

در این بخش با یک مساله بهینه‌سازی چندهدفه روبرو هستیم. هدف حل مساله، کاهش اثر اغتشاش و افزایش اثر عیب می‌باشد که ناگزیریم دو تابع هدف را تعریف نماییم و به طور همزمان، مقدار آن‌ها را بهینه کنیم. بنابراین از الگوریتم MOPSO که عملکرد آن در بخش ۲-۲-۶ بطور کامل شرح داده شد، استفاده می‌کنیم و توابع هزینه را مطابق معیار  $H_-/H_\infty$ ، به صورت زیر تعریف می‌کنیم:

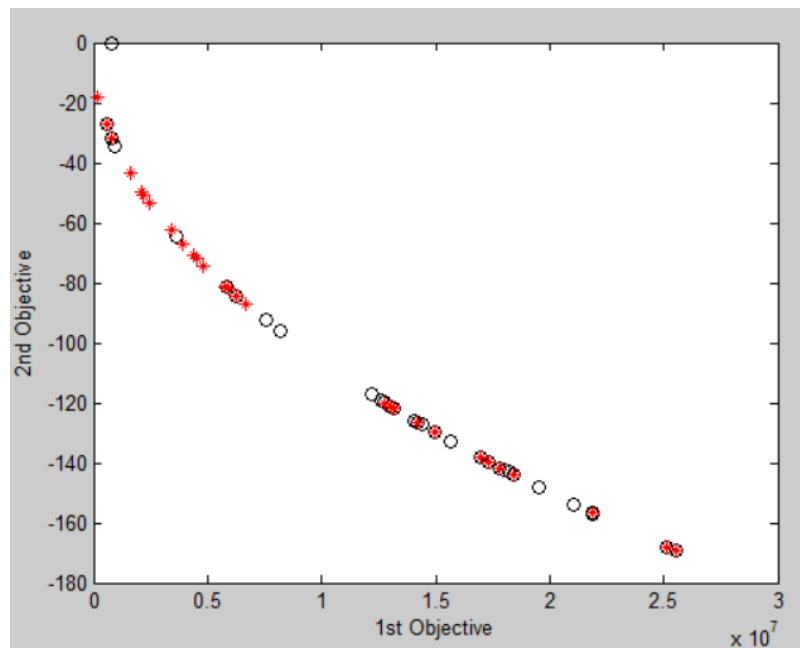
$$Z_1 = \|T_{zw_d}\|_\infty \quad (۲۶-۳)$$

$$Z_2 = \|T_{zw_f}\|_2$$

که در آن  $w_f = [f]$  و  $w_d = \begin{bmatrix} v \\ n \end{bmatrix}$  می‌باشد. آنگاه بهره رویتگر عبارت است از:

$$L = \begin{bmatrix} 0/0459 \\ 0/08533 \\ 1 \end{bmatrix}$$

نتایج شبیه‌سازی برای رویتگر بدست آمده به صورت زیر است.



شکل (۳-۱۵) توابع هزینه‌ی الگوریتم MOPSO در روش  $H_-/H_\infty$

شکل (۳-۱۵) پاسخ‌های الگوریتم MOPSO را نشان می‌دهد. در صفحه‌ی این شکل، پاسخ‌ها، با  $\circ$

و \* نشان داده شده اند که \*ها ، اعضای آرشیو هستند. نمودار افقی و نمودار عمودی، به ترتیب معرف توابع هزینه  $Z_1$  و  $Z_2$  در رابطه‌ی (۳-۲۶) می‌باشند. مشاهده می‌شود که هر یک از ذرات، از آرشیو یک رهبر انتخاب کرده و به سمت آن حرکت خود را انجام داده‌اند.

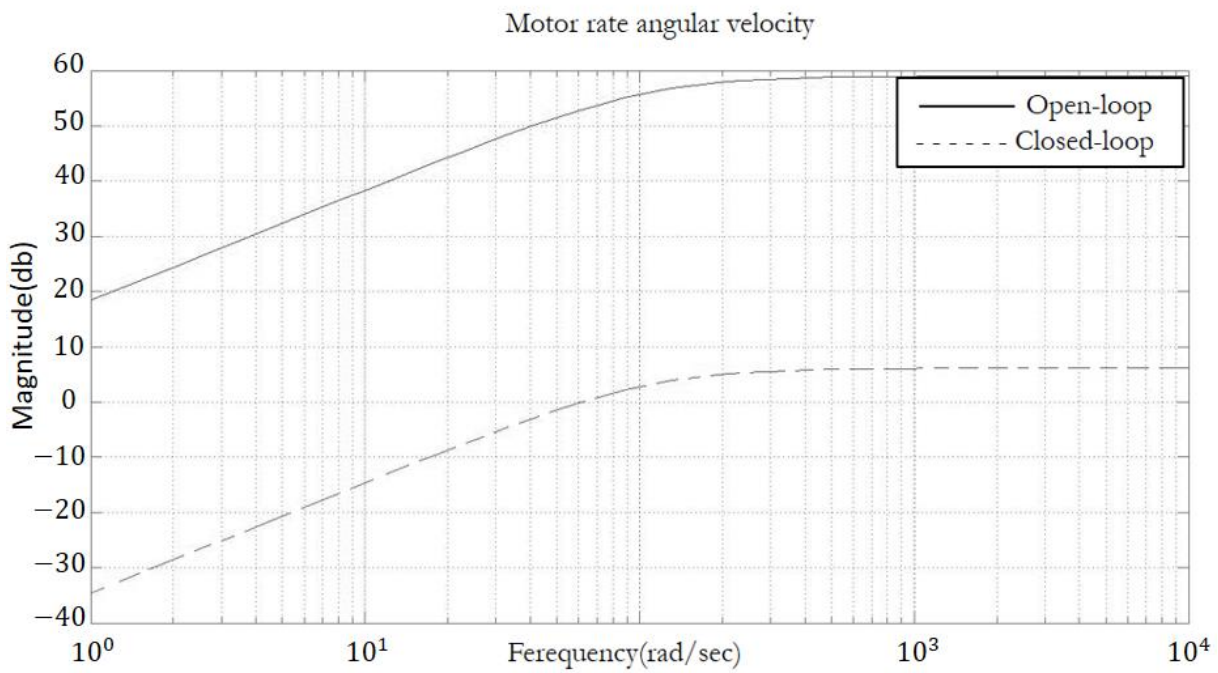
جدول (۳-۳) مقادیر پارامترهای مورد استفاده در الگوریتم MOPSO

مقدار	پارامترهای الگوریتم MOPSO
۳	تعداد متغیرهای تصمیم
[-۱ ۱]	حد پایین و بالای هر متغیر تصمیم
۵۰	اندازه جمعیت
۳۰	اندازه آرشیو
۳۵	تعداد تکرار الگوریتم
۰/۵	$\omega$
۱	$c_1$
۲	$c_2$
۱	فشار انتخاب $\beta$

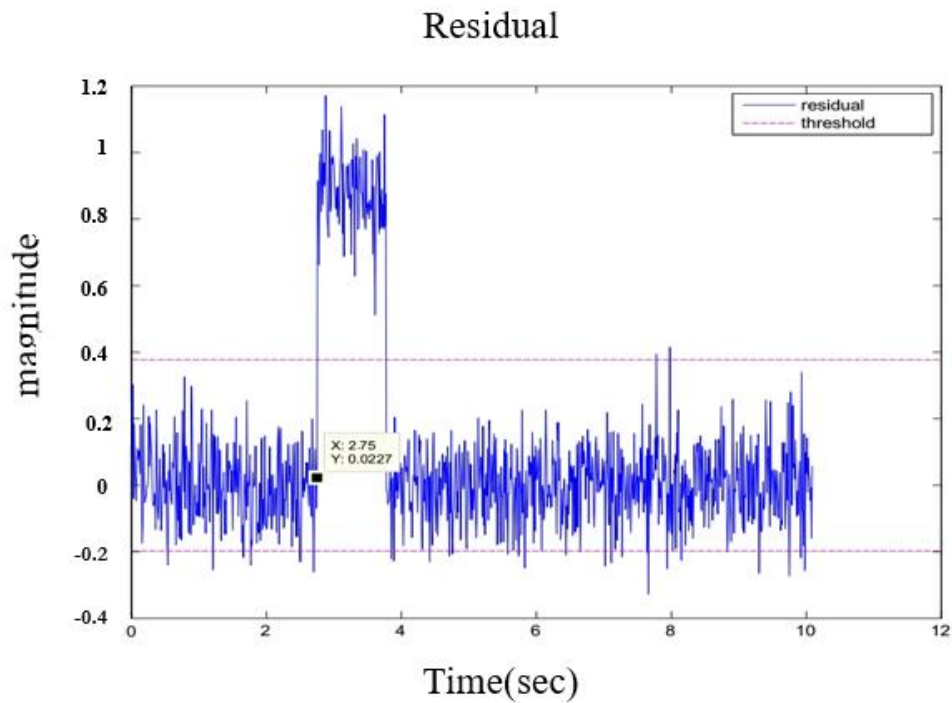
جدول (۳-۴) مقدار پارامترهای الگوریتم MOPSO را که در این شبیه‌سازی، استفاده شده‌اند را نشان می‌دهد.

شکل (۳-۱۶) نشان می‌دهد که در سیستم حلقه بسته، کاهش دامنه در تمامی فرکانس‌ها را داریم. مشخص است که این روش، نسبت به روش  $H_2$  و  $H_\infty$  موفق‌تر عمل کرده و دامنه فرکانسی حدوداً به صفر رسیده است.

مطابق شکل (۳-۱۷) واضح است که در این روش، حساسیت به اغتشاش به خوبی محو شده و  $H_-/H_\infty$  در جداسازی عیب از اغتشاش موفق عمل کرده است. با استفاده از سطح آستانه می‌توان زمان تشخیص عیب را بدست آورد. در نمودار فوق، خطوط خط چین، آستانه را نشان می‌دهند. واضح است بعد از اینکه پینچ در  $2/7$  ثانیه رخ داد، تشخیص در  $50$  میلی ثانیه بعد صورت می‌گیرد.



شکل (۳-۱۶) پاسخ فرکانسی سیستم حلقه باز و حلقه بسته‌ی نرخ سرعت زاویه‌ای موتور در روش  $H_-/H_\infty$



شکل (۳-۱۷) سیگنال باقی‌مانده به روش  $H_-/H_\infty$  در حضور عیب و اغتشاش

### ۳-۴-۱-۱-۱- آنالیز سیستم حلقه بسته با کنترلر L و معیار $H_2/H_\infty$

در این بخش از معیار  $H_2/H_\infty$  استفاده نموده و همانند بخش ۳-۳-۱-۱، با دستورات متلب، ۲۰ نقطه را از پوش شکل (۷-۳) انتخاب می‌کنیم و  $w_a$  را با یک تابع مرتبه دو تخمین می‌زنیم و در آخر نیز تابع تبدیل آن را با توجه به نقاط انتخاب شده، به دست می‌آوریم:

$$w_a = \frac{4845 s^2 + 3189 s + 0/5558}{s^2 + 3206 s + 175000}$$

برقرار شدن رابطه‌ی  $\| [w_a S] \|_\infty < 1$  جهت رسیدن به سطح مطلوب کاهش اغتشاش، ضروری می‌باشد. که S تابع حساسیت سیستم حلقه بسته می‌باشد.  
از این شرط داریم که برای تمامی فرکانس‌ها باید:

$$\| (I + GL)^{-1} \|_\infty < \frac{1}{w_a}$$

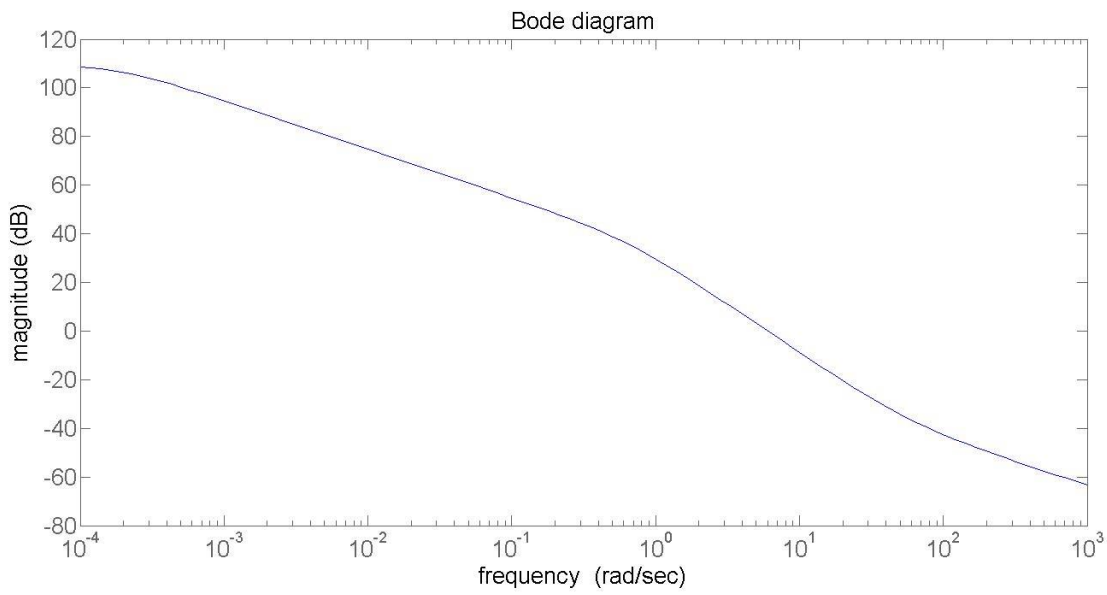
که برای رسیدن به این شرط می‌توانیم بگوییم اگر و فقط اگر:

$$\sigma[(I + GL)^{-1}] < \left| \frac{1}{w_a} \right|$$

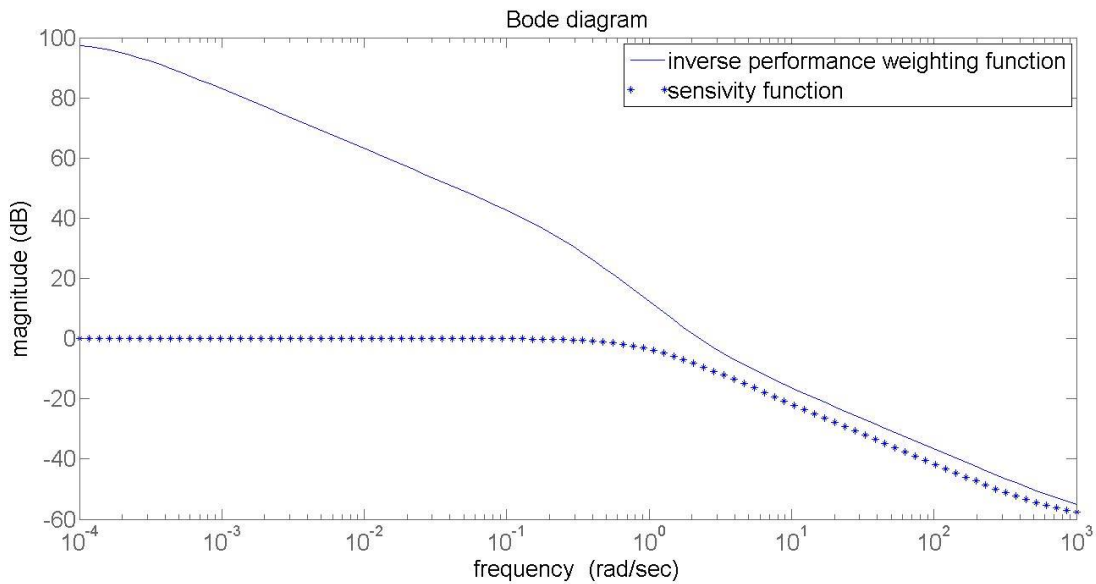
مقادیر استثنایی  $\frac{1}{w_a}$ ، در شکل (۱۸-۳) نشان داده شده است.

حال، رفتار سیستم حلقه بسته با استفاده از کنترلر طراحی شده، در مرحله‌ی قبل را بررسی می‌کنیم که برای بررسی می‌توان تابع حساسیت حلقه بسته را با معکوس تابع وزنی مقایسه نمود.

نتیجه‌ی این مقایسه در شکل (۱۹-۳) نشان داده شده است. همانطور که دیده می‌شود، در فرکانس‌های پایین، تابع حساسیت، زیر نمودار  $\frac{1}{w_a}$  قرار گرفته است.



شکل (۱۸-۳) مقادیر استثنایی  $\frac{1}{w_a}$  با استفاده از معیار  $H/H_\infty$



شکل (۱۹-۳) معکوس تابع وزنی (خط پیوسته)، تابع حساسیت با استفاده از معیار  $H/H_\infty$  (خط گسسته)

### ۳-۵- مقایسه‌ی نتایج شبیه‌سازی سیستم با الگوریتم MOPSO و LMI<sup>۵۴</sup>

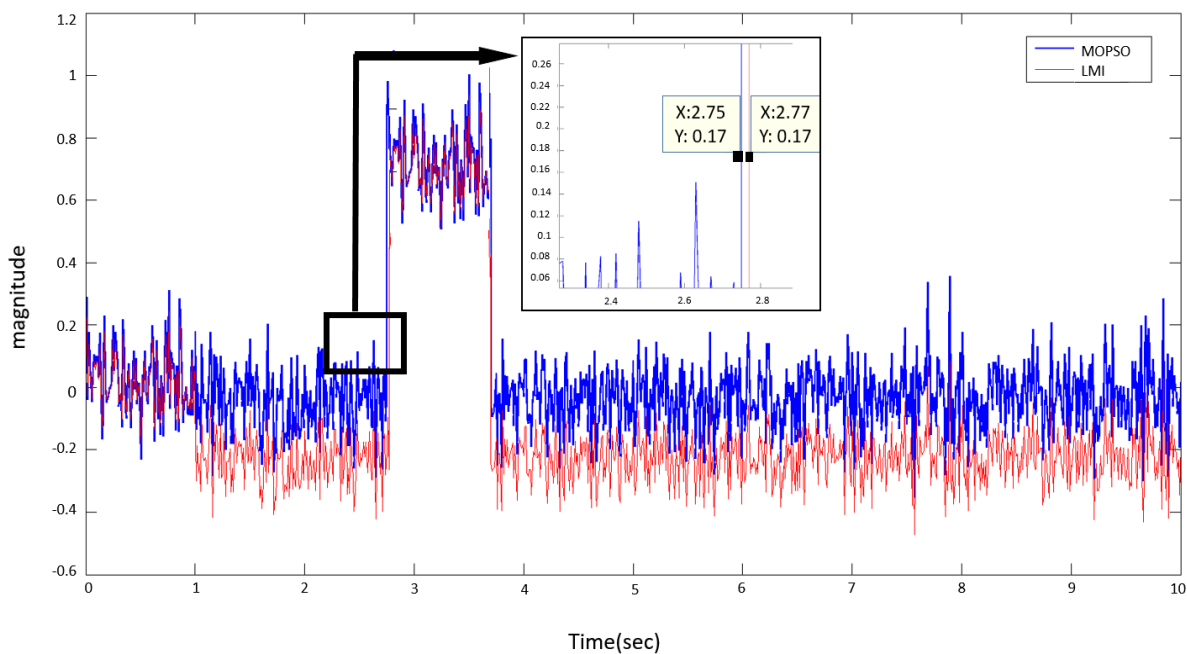
با استفاده از الگوریتم LMI، بهره رویتگر برای این سیستم عبارت است از [۳۰]:

$$L = \begin{bmatrix} 7/3967 \\ -0/0164 \\ 0/0201 \end{bmatrix}$$

شکل (۳-۲۰) سیگنال باقی‌مانده در روش LMI و MOPSO را با استفاده از معیار  $H_2/H_\infty$  نشان می‌دهد.

مقایسه‌ی زمان بدست آمده جهت تشخیص پینچ در این دو روش، نشان می‌دهد که MOPSO

موفق‌تر عمل کرده و توانسته است عیب را ۲۰ میلی ثانیه زودتر از LMI تشخیص دهد.



شکل (۳-۲۰) مقایسه‌ی سیگنال باقی‌مانده در روش MOPSO و LMI با استفاده از معیار  $H_2/H_\infty$

<sup>۵۴</sup> Linear matrix inequality



## فصل چهارم

### نتایج و پیشنهادات

#### ۴-۱- نتیجه گیری

در این تحقیق، روش‌های مختلف، جهت تولید سیگنال باقی‌مانده به روش مبتنی بر مشاهده‌گر بررسی شد. هدف، طراحی روی‌نگر  $H_-/H_\infty$  و استفاده از آن در تشخیص و جداسازی عیب مقاوم سنسور در سیستم ضد پینچ خودرو است.

با توجه به نتایج بدست آمده در فصل قبل، کاملاً مشهود است که در روش  $H_-/H_\infty$ ، عملکرد تشخیص به مراتب بهتر از روش‌های  $H_\infty$  و  $H_2$  می‌باشد بطوریکه در روش  $H_-/H_\infty$ ، تاثیر عیب روی سیگنال مانده افزایش می‌یابد و در عین حال، اثر ورودی ناشناخته تقریباً محو می‌شود.

در این تحقیق، برای حل مساله  $H_-/H_\infty$ ، از روش MOPSO استفاده شد؛ نشان داده شد که الگوریتم ارائه شده به خوبی می‌تواند مقدار بهینه برای بهره‌ی مشاهده‌گر را به دست آورد. نتایج شبیه‌سازی نشان می‌دهد که مشاهده‌گر، قادر است عیب را ۵۰ میلی ثانیه بعد از رخ دادن، تشخیص دهد که نشان می‌دهد باقی‌مانده حساسیت بالا به عیب دارد و الزامات را برآورده کرده است.

#### ۴-۲- پیشنهادات

مساله‌ی تشخیص عیب مقاوم، یکی از موضوعات بسیار مهم دنیای صنعتی امروز می‌باشد که همچنان جای تحقیقات گسترده‌ای دارد و اقدامات بسیار زیاد دیگری در این زمینه می‌توان انجام داد. در ادامه چند پیشنهاد برای بهبود کار ارائه می‌گردد.

۱- بررسی و مقایسه الگوریتم NSGA-II با الگوریتم MOPSO.

۲- استفاده از الگوریتم‌های هوشمند و ترکیب آن با روش  $H_-/H_\infty$ .

۳- بررسی و استفاده از یک مدل غیرخطی و استفاده از مشاهده‌گرهای غیرخطی جهت انجام امر تشخیص.

## منابع:

- [۱] طبسی، ش. (۱۳۹۲). "تشخیص خطای مقاوم سنسور مبتنی بر مشاهده گر روی مدل توربین گازی." پایان نامه کارشناسی ارشد. دانشگاه صنعتی شاهرود.
- [2] He, X., et al. (2017). "Fault-tolerant control for an Internet-based three-tank system: Accommodation to sensor bias faults." *IEEE Transactions on Industrial Electronics* 64(3) : 2266-2275.
- [3] Hwang, I., et al. (2010). "A survey of fault detection, isolation, and reconfiguration methods." *IEEE transactions on control systems technology* 18(3): 636-653.
- [4] Samy, I., et al. (2011). "Survey and application of sensor fault detection and isolation schemes." *Control Engineering Practice* 19(7): 658-674.
- [5] Ma, J. and J. Jiang (2011). "Applications of fault detection and diagnosis methods in nuclear power plants: A review." *Progress in nuclear energy* 53(3): 255-266.
- [6] Mojallal, A. and S. Lotfifard (2017). "Multi-Physics Graphical Model based Fault Detection and Isolation in Wind Turbines." *IEEE Transactions on Smart Grid*.
- [7] Sobhani-Tehrani, E. and K. Khorasani (2009). *Fault diagnosis of nonlinear systems using a hybrid approach*, Springer Science & Business Media.
- [8] Isermann, R. (2005). "Model-based fault-detection and diagnosis—status and applications." *Annual Reviews in control* 29(1): 71-85.
- [9] Simani, S., et al. (2013). *Model-based fault diagnosis in dynamic systems using identification techniques*, Springer Science & Business Media.
- [10] Alif, A., et al. (2010). "Design of Robust  $H_{\infty}$  Reduced-Order Unknown-Input Filter for a Class of Uncertain Linear Neutral Systems." *IEEE Transactions on Automatic Control* 55(1): 6-19.
- [11] Nazir, M., et al. (2017). "Robust fault detection for wind turbines using reference model-based approach." *Journal of King Saud University-Engineering Sciences* 29(3): 244-252.

[12] Zhai, S., et al. (2017). "A set-membership approach to integrated trade-off design of robust fault detection system." *International Journal of Adaptive Control and Signal Processing* 31(2) : 191-209.

[13] Aouaouda, S. and M. Chadli (2017). Robust fault detection filter design for discrete-time fuzzy models. *Recent Advances in Electrical Engineering and Control Applications*, Springer : 215-232.

[14] Chibani, A., et al. (2017). "Fuzzy fault detection filter design for t–s fuzzy systems in the finite-frequency domain." *IEEE Transactions on Fuzzy Systems* 25(5): 1051-1061.

[15] Zhong, M., et al. (2017). "A probabilistic approach to robust fault detection for a class of nonlinear systems." *IEEE Transactions on Industrial Electronics* 64(5): 3930-3939.

[16] Ye, D., et al. (2009). Fault Diagnosis by an Observer-Based Fuzzy Decision System. *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC'09. International Conference on*, IEEE.

[17] Isermann, R. (2006). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, Springer Science & Business Media.

[۱۸] میرآبادی، ا. و سایرین. (۱۳۸۹). "تشخیص و شناسایی خطا در سیستم سوزن با روش الگوریتم فازی."، *مجله مهندسی حمل و نقل*. سال دوم.

[19] Abid, M. (2010). Fault detection in nonlinear systems: An observer-based approach, *Universität Duisburg-Essen, Fakultät für Ingenieurwissenschaften» Elektrotechnik und Informationstechnik» Automatisierungstechnik und komplexe Systeme*.

[20] Liu, L. and E. Collins (2006). "ROBUST FAULT DETECTION AND DIAGNOSIS FOR PERMANENT MAGNET SYNCHRONOUS MOTORS By."

[21] Kunkel, J. (2009). Anti-pinch window control system, Google Patents.

- [22] YAN, J., et al. (2017). "Study on current de-noising and character analysis based on wavelet transform for anti-pinch window system." *Journal of Guangxi University (Natural Science Edition)* 2: 024.
- [23] Ding, S. X. (2008). *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*, Springer Science & Business Media.
- [24] Chen, C.-T. (1998). *Linear system theory and design*, Oxford University Press, Inc.
- [25] Dorf, R. C. and R. H. Bishop (2011). *Modern control systems*, Pearson.
- [26] Edwards, C., et al. (2010). "Fault tolerant flight control." *Lecture Notes in Control and Information Sciences* 399: 1-560.
- [27] Dai, X., et al. (2008). Discrete-time robust fault detection observer design: a genetic algorithm approach. *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on, IEEE*.
- [28] Frank, P. M. and X. Ding (1997). "Survey of robust residual generation and evaluation methods in observer-based fault detection systems." *Journal of process control* 7(6): 403-424.
- [29] Zhou, K. and J. C. Doyle (1998). *Essentials of robust control*, Prentice hall Upper Saddle River, NJ.
- [30] Li, H., et al. (2013). "Robust Fault Detection Algorithm for the Smart Anti-pinch Window of Pure Electric Vehicles." *Research Journal of Applied Sciences, Engineering and Technology* 5(24): 5683-5693.
- [31] Kennedy, J. (2011). Particle swarm optimization. *Encyclopedia of machine learning*, Springer: 760-766.
- [32] Coello, C. A. C., et al. (2004). "Handling multiple objectives with particle swarm optimization." *IEEE Transactions on evolutionary computation* 8(3): 256-279.
- [33] John, N. A., et al. (2016). *Anti-Pinch Mechanism for Power Window*, SAE Technical Paper.
- [34] Schooler, P. T. (2006). *Anti-pinch power window system*, Google Patents.

**Abstract**

In this thesis, an optimal solution  $H_2/H_\infty$  is presented for fault detection of anti-pinch window system. By sampling parameters such as current consumption, the speed and the position of DC motors, an anti-pinch allows detection of an obstacle and prevents the damages. In this research, a speed-based method is used to detect the obstacles. In order to secure an anti-pinch window, an optimal algorithm based on a fault detection observer is suggested. In the residual design, the proposed fault detection algorithm uses the DC motor rate angular velocity. By comparing the residual signal with the pre-designed threshold, the fault is detected. Robustness against disturbances and sensitivity to the faults are considered as an optimization problem based on Multi-Objective PSO algorithm. Finally, an optimal filter for solving the fault problem is designed using the  $H_2/H_\infty$  method.

**Keywords:**  $H_2/H_\infty$ , anti-pinch, residual, fault detection, MOPSO.



Shahrood University Of Technology  
Faculty of Electrical and Robotics Engineering  
M.Sc. Thesis in Control Engineering

**Robust Fault Detection for the Smart Anti-pinch Window of  
Car using  $H_2$  /  $H_\infty$**

By  
Maedeh Mohammadi Azni

supervisor  
Dr. Mohamad Ali Sadrnia

January 2018





