بسم الله الرحمن الرحيم

پایان نامه کارشناسی ارشد
دانشکده برق و رباتیک

عنوان:

شناسایی پارامترهای موتور **DC** و طراحی کنترل کننده **PID** برای آن به روش الگوریتمهای ژنتیکی

استاد راهنما: دکتر حداد

ارائه دهنده : آرش معبود

تابستان ۱۳۸۷

تقدیم به

پدر و مادر عزیزم

تشکر و قدردانی

از استاد محترم جناب آقای دکتر حداد ظریف که با راهنمايي های خویش مرا در تهیه این پایان نامه یاری کرده و کاستی های مرا تحمل کرده است، سپاسگزاری نموده و موفقیت روز افزون ایشان را از خدای بزرگ خواستارم.

چکیده:

همان طور که می دانیم امروزه برای طراحی های سیستم های مختلف از روش های جدیدی با عنوان "سیستم های هوشمند" استفاده می کنند. کاربرد روزافزون این روش های هوشمند که عمدتا مبتنی بر الگو برداری از رفتار موجودات زنده و طبیعت می باشد، باعث شد که در این پایان نامه تصمیم به استفاده از این روش در یکی از تجهیزات پرکاربرد در صنعت گرفته شود. روش هوشمند مورد استفاده در این پایان نامه "الگوریتم ژنتیکی" بوده که اصول کار آن بر پایه تولید نسل توسط ژنها و کروموزومهای موجودات زنده می باشد. در این پایان نامه از این الگوریتم برای شناسايي پارامترهای الکتریکی و دینامیکی موتورهای **DC** در صنعت استفاده خواهد شد. همان طور که می دانیم موتورهای **DC** یکی از پرکاربردترین تجهیزات در صنعت و کارخانجات می باشند و برای استفاده بهینه تر از آنها باید بتوان آنها را کنترل کرد. در ادامه پس از شناسايي پارامترهای موتور**DC** به طراحی کنترل کننده **PID** برای آن پرداخته می شود.

فهرست مطالب

فهرست اشکال

فهرست جداول

فصل اول _ مقدمه

تعداد زيادي از مسائل بهينه سازي كه امروزه در صنعت مورد بررسي قرار مي گيرند بسيار پيچيده و نسبتا سخت هستند كه با روشهاي كلاسيك به راحتي قابل حل نيستند. از سال ۱۹۶۰ گرايشهايي براي حل اينگونه مسائل بهينه سازي بوجود آمد كه بسياري از آنها مبتني بر شاخه هاي هوش جمعي مي باشند كه حوزه وسيعي از دانش هوش مصنوعي را شامل می شود. هوش جمعي بر اساس شبيه سازي رفتار اجتماعات زنده و پيشبرد گروه به سمت بهترين وضعيت شكل گرفته است، مثل شبيه سازي ژن هاي موجود در بدن موجودات زنده كه نسل به نسل بدن را نسبت به شرايط محيطي مقاوم تر مي كنند ويا رفتار جمعي حشرات مانند زنبورها، مورچه ها و موريانه ها براي يافتن غذا و يا حركت گروهي حيواناتي مثل ماهيها يا پرندگان .

همانطور که می دانیم در بسیاری از صنایع از موتورهای الکتریکی استفاده می شود که قسمت اعظمی از آنها موتورهای DC هستند .برای راه اندازي و كنترل اين موتورها، روش هاي متعددی وجود دارد که جدیدترین تکنولوژی مورد استفاده در این زمینه اینورترها یا درایوها هستند. اکثر این کنترل کننده ها برای عملکرد صحیح خود نیاز به یک پردازنده دارند تا پارامترهای موتور را به طور دقیق محاسبه کنند که از روشهای گوناگونی برای این امر استفاده می کنند.

هدف این پروژه ارائه یک روش دقیق و با سرعت بالا برای نیل به این نیاز است. بدین منظور از روش الگوريتمهاي ژنتيكي که یک الگوریتم بهينه سازي می باشد استفاده شده است. در ابتدا پارامترهاي ديناميكي و الكتريكي موتور محاسبه مي شود و در ادامه یک کنترل کننده PID که در عین سادگي یک کنترل کننده با راندمان بالا مي باشد براي موتور طراحي مي شود.

فصل دوم _ الگوریتم های ژنتیکی

۲-۱- مقدمه

الگوریتم های ژنتیکی[1] (GA) یکی ازسریع ترین روشها، در به دست آوردن اکسترمم یک تابع است. تنها شرطی که برای به کارگیری این الگوریتم مورد نیاز است، امکان به دست آوردن مقدار تابع در هر نقطه از فضای تعریف متغیرهای آن است. این روش گونه ای از جستجوی تصادفی هوشمندانه است که سعی می شود نقطه های نو به گونه ای هوشمندانه تولید شوند تا به صورت آماری به سمت نقطه بهینه مطلق میل کنند.

ایده الگوریتم ژنتیکی یا GA از دو اصل انتخاب و تولید نسل در طبیعت بهره برده است. با گذشت زمان ساختار ژنتیکی موجودات تغییر کرده و نسلهای جدیدتر با محیط سازگاری بیشتری پیدا می کنند. بدین طریق که امکان زنده ماندن و تولید مثل موجودات قوی تر بیشتر از موجودات ضعیف می باشد. در نسلهای جدید ساختار ژنتیکی موجودات قوی تکرار می شود و موجودات ضعیف از بین می روند. در بعضی موارد نیز جهش بوجود می آید، بدین معنی که از آمیزش دو موجود، موجودی متولد می شود که بر اثر جهش ژنتیکی خیلی بهتر یا بدتر از والدین خود می باشد و به تعبیری یک نابغه یا یک عقب مانده متولد می شود و روی نسلهای بعدی تاثیر می گذارد. طبیعت تضمین می کند که این نوع زاد و ولد موجب تولید موجودات بهتر شود.

الگوریتم ژنتیکی بر این اساس ابزاری ساده، ولی قدرتمند است. GA یک روش بهینه سازی بر اساس جستجو است که توسط جان هولند[2] و شاگردانش در دانشگاه میشیگان ارائه شد. بعضی مسائل دارای

---

[1] Genetic Algorithm

[2] John Holland

راه حل های مشخص می باشند: برای حل مسائل خطی با محدودیتهای خطی، معمولا از روشهای حل LP[1] (برنامه ریزی خطی) استفاده می شود. درحقیقت به این دلیل که بسیاری از روشها در محدوده مسائل خاص کاربرد دارند، پیدا کردن روش مناسب برای حل یک مسئله نیز خود مشکل دیگری است. بعضی روشها به دلیل محدودیت برای حل مسائل خاص خود نیز دچار مشکل می شوند. مثلاً برای بدست آوردن بیشینه یا کمینه یک تابع مشتق پذیر می توان از مشتق گیری استفاده کرد. اما اگر این تابع مشتق پذیر نباشد و یا تعریف صریح نداشته باشد، چه باید کرد ؟

یکی از روشهای حل این گونه مسائل، الگوریتم ژنتیکی است که به دلیل استفاده از جستجوی نقاط بهینه تابع در کل فضای جستجو، هیچ گونه عملیاتی بر روی تابع انجام نمی دهد.

الگوريتم‌هاي ژنتیکی نقش مهمي را به عنوان روش‌هاي جستجو، در حوزه‌هاي هوش مصنوعي، مهندسي، رباتیک و... برعهده دارند. این الگوریتم‌ها جزو روش‌هاي رايانش نرم[2] مي‌باشند و در شاخه رايانش تكاملي[3] طبقه‌بندي مي‌شوند. الگوريتم‌هاي ژنتیکی بر پايه پروسه‌هاي ژنتيكي در ساختارهاي بيولوژيكي و اصول تكامل طبيعي جمعيت‌ها استوار هستند.[1]

بعضی از روشها مانند مزدوج گرادیان برای شروع حل مسئله نیاز به داشتن جواب نزدیک به جواب بهینه می باشند و این خود در بعضی مواقع موجب نیازمندی آنها به روشهای دیگری است که جوابی نزدیک به جواب بهینه تولید کنند. اما در الگوریتم ژنتیکی و روشهایی مانند جستجوی تصادفی شروع حل مسئله از نقاط تصادفی در فضای جستجو صورت می گیرد.

البته لازم بذکر است که سرعت رسیدن به جواب در روشهای مانند الگوریتم ژنتیکی و جستجوی تصادفی کمتر از سیمپلکس و . . . می باشد.

---

1 Linear Programming

2 soft computing

3 evolutionary computing

۲-۲- الگوریتم های ژنتیکی در نگاهی گذرا

فرض کنید هر نقطه x از دامنه تعریف تابع f(x) (که در آن $f(x) \geq 0$ و $\forall x \in D$ و D دامنه تعریف تابع f است)، متناظر با رشته مرتبی به صورت

$$A = \overline{a}_1 \overline{a}_2 \ldots . \overline{a}_n \tag{۲-۱}$$

از فضای S باشد که در آن a_i مقداری از مجموعه T را بتواند بگیرد. مجموعه T می تواند یک مجموعه گسسته و مرکب از m عضو باشد. در حالت کلی x یک بردار است. به سادگی می توان تابعی به نام g در s یافت که :

$$f(x) = g(A) \tag{۲-۲}$$

در نتیجه بیشینه شدن تابع f به معنی بیشینه شدن تابع g است. در حالتی که *f(x)* مقید به شرط $f(x) \geq 0$ نباشد، با کمک هر تابع مطلقاً یکنوای صعودی ϕ، از برد *f* به بازه $[0,+\infty)$، می توان آن را به صورت تابع *F(x)* نشان داد که در آن $\forall x \in D, F(x) \geq 0$. شکل تابع ϕ در سرعت همگرایی الگوریتم تاثیر می گذارد. همچنین اگر هدف کمینه سازی تابع *f(x)* باشد، با کمک هر تابع مطلقاً یکنوای نزولی ϕ از برد f به $[0,+\infty)$ می توان آن را به بیشینه سازی برگرداند.

فرض کنید $2p$ رشته به صورت زیر موجود باشد:

$$A = \overline{a}_1^i \ldots . \overline{a}_n^i \tag{۲-۳}$$

برای اجرای الگوریتم، نخست باید $g(A_i)$ ، $i = 1,\ldots,2p$ را به دست آورد، سپس به کمک عملگرهای گزینش، برش و جهش، که در ادامه به تشریح آنها پرداخته خواهد شد، مجموعه رشته های C_i ، $i = 1,\ldots,2p$ را به دست می آورند و جایگزین رشته های A_i در تکرار بعدی می کنند.[2]

۲-۳- مفاهیم بنیادی الگوریتمهای ژنتیکی

الگوريتم‌هاي ژنتیکی (GA) الگوريتم‌هاي جستجوي همه‌منظوره‌اي هستند که بر اساس مکانيزم وراثت و گزينش طبيعي در موجودات زنده (نظريه داروين ۱۸۵۹) کار مي‌کنند. اين الگوريتم‌ها با جمعيتي از اجزاء منفرد که نماينده پاسخ‌هاي ممکن براي يک مساله هستند سر و کار دارند. به هر يک از اجزاء منفرد بسته به اين که تا چه حد پاسخ مناسبي براي مساله باشند، يک امتياز برازندگي نسبت داده مي‌شود.

GA با يک جمعيت اوليه از پاسخ‌ها شروع به کار مي‌کند و با اعمال عملگرهاي ژنتیکی، که بر اساس پروسه‌هاي ژنتيک موجود در طبيعت مدل شده‌اند، به سمت پاسخ‌هاي بهتر پيش مي‌رود. هر دوره توليد پاسخ‌هاي جديد يک نسل ناميده مي‌شود. در هر نسل پاسخ‌هاي نسبتاً بهتر براي توليد پاسخ‌هاي جديد که متناظر با فرزندان در پروسه ژنتيک طبيعي هستند، مورد استفاده قرار مي‌گيرند و جانشين پاسخ‌هاي نامناسب‌تر مي‌شوند. در اين پروسه يک تابع ارزيابي يا برازندگي که نقش محيط را در پروسه ژنتيک طبيعي بازي مي‌کند، براي تمايز ميان پاسخ‌هاي خوب و بد به کار مي‌رود.[2]

۲-۴- مفاهیم الگوریتم ژنتیکی

هدف الگوریتم ژنتیکی به دست آوردن جواب بهینه یا نزدیک به بهینه می باشد. برای این کار :

۱- GA با پارامترهای کد شده کار می کند نه با خود پارامتر

۲- GA جستجو را بوسیله جمعیتی از نقاط انجام می دهد، نه یک نقطه. فقط به ذکر این نکته بسنده می شود که تعداد این نقاط نیز مهم است، به این صورت که تعداد جمعیت زیاد، محاسبات را زیاد کرده و الگوریتم کندتر به جواب بهینه نزدیک می شود و تعداد

جمعیت کم، فضای جستجو را به خوبی پوشش نداده و ممکن است الگوریتم در یک اکسترمم محلی گیر کند.

۳- GA از مقدار تابع درنقاط مختلف فضای جستجو استفاده می کند و نیازی به دانستن مشتق تابع یا اطلاعات دیگری از تابع ندارد.

۴- الگوریتم ژنتیکی از قوانین آماری استفاده می کند نه از قوانین محاسباتی. [1],[2]

۲-۵- واژگان ژنتیک **[1]**

از آنجا كه GA تركيبي از مفاهيم ژنتیک طبيعي و علوم كامپيوتر است، واژگان مورد استفاده در آن نيز برگرفته از از اين دو ريشه است و در نوشتار مختلف از هر یک از اين واژگان استفاده مي‌شود. واژگان ژنتیک طبيعي و معادل آن در الگوریتم‌هاي ژنتیک را به همراه شرح مختصر آنها دربر دارد.

شرح	واژگان الگوریتم ژنتیکی	واژگان ژنتیک طبيعي
کوچکترين جزء تشکيل دهنده يک ساختار که يک خصوصيت را دربر دارد.	Feature , Character	Gene
مقدار و يا محتواي يک ژن خاص	Value	Allele
رشته‌اي از ژن‌ها که معرف يک جواب خاص است و به آن جزء منفرد نيز گفته مي‌شود.	String	Chromosome
موقعيت يک ژن در يک کروموزوم	Position	Locus
جمعيت، متشکل از تعداد ثابتي از اجزاء منفرد	Population	Population
فضاي مبتني بر ساختار ژنتيکي که الگوريتم در آن فضا کار مي‌کند.	Structure	Genotype
فضاي مبتني بر ساختار واقعي اجزاء	Point in Search Space	Phenotype
هر مرحله از تکرار الگوريتم که يک نسل جديد را توليد مي‌کند.	Iteration	Generation

جدول 2-۱ واژگان ژنتیک

برخی از مفاهیم که در پیاده سازی الگوریتم های ژنتیکی به آنها برخورد می کنیم در زیر آورده شده است :

- ✓ ژن و کروموزوم : ژن کوچکترین واحد سازنده GA می باشد. در حقیقت ژنها برای نمایش شکل کد شده پارامترها می باشد به رشته ای از ژنها، کروموزوم گفته می شود. برای مثال وقتی متغیر X به صورت باینری کد می شود، هر " 0 " یا " 1 " یک ژن محسوب می شود و

به رشته ای از این " 0 " و " 1 " ها که متغیر x را می سازند و در حقیقت شکل کد شده متغیراند، کروموزوم گفته می شود.

- ✓ تابع معیار : تابع معیار تابعی است که هدف مساله بهینه سازی آن است که وسیله لازم برای ارزیابی هر کروموزوم را فراهم می کند. تابع معیار به هر کروموزوم یک عدد نسبت می دهد که این عدد میزان خوب بودن یا مناسب بودن آن کروموزوم را نشان می دهد. برای مثال وقتی هدف، کمینه کردن خطای پاسخ پله یک سیستم پیوسته است، توابع معیار زیر مناسب می باشند :

$\frac{1}{\int |e(t)| dt}$ و $\frac{1}{\int |e^2(t)| dt}$ که $e(t)$ مقدار خطا در هر لحظه می باشد.

- ✓ جمعیت و نسل : جمعیت تعداد نقاطی از فضای جستجو است که GA با آنها به سمت جواب بهینه می رود. برای این کار عملگرهای مختلفی به جمعیت اعمال می شوند و جمعیتی به وجود می آید که جایگزین جمعیت قبل می شود و با اعمال دوباره عملگرها این روند ادامه می یابد. این تکرارها نسلهای GA را بوجود می آورند.
- ✓ والدین و فرزندان : در هر نسل افراد جمعیت به صورت جفت جفت و بر اساس احتمالی متناسب با تابع معیارشان انتخاب شده تا عملگرها بر روی آنها اعمال شود. به این جفت ها والدین گفته می شود. پس از اعمال عملگرها بر روی والدین، یک جفت موجود جدید تولید می شود که " فرزندان " آنها می باشند.

۲-۶- ساختار الگوریتم ژنتیکی

در حالت كلي براي حل هر مساله با استفاده از GA پنج جزء اساسي زير مورد نياز است :

- ❑ يک نمايش ژنتيکي از پاسخ‌هاي مسأله
- ❑ روشي براي توليد جمعيت اوليه پاسخ‌ها
- ❑ يک تابع ارزياب براي تعيين برازندگي هر يک از پاسخ‌ها
- ❑ عملگرهاي ژنتيکي براي دستکاري ساختار ژنتيکي فرزندان در مرحله بازتوليد
- ❑ مقاديري براي پارامترهايي که در GA به‌کار مي‌رود. (اندازه جمعيت، احتمالات مربوط به اعمال عملگرهاي ژنتيکي و...)

بر اين اساس مراحل يک GA به شرح زير خواهد بود :

۱- ايجاد جمعيت اوليه و ارزيابي هر يک از اجزاء جمعيت اوليه

۲- عمليات بازتوليد تا رسيدن به شرط توقف در نظر گرفته شده. اين مرحله شامل تکرار گام‌هاي زير است:

- ✓ يک يا چند تن از والدين براي بازتوليد انتخاب مي‌شوند. اين انتخاب به صورت تصادفي انجام مي‌شود، ولي والدين با بالاترين برازندگي مورد توجه قرار مي‌گيرند.
- ✓ انتخاب عملگرهای ژنتيکي براي والدين.
- ✓ فرزندان ارزيابي مي‌شوند و در يک نسل انباشته مي‌شوند و سپس جايگزين اعضاي نامناسب جمعيت مي‌شوند.

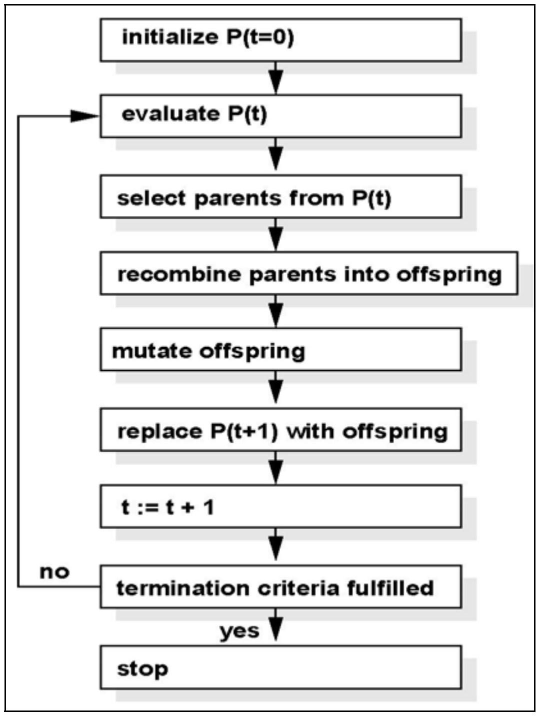
فرض شود P(t) نماينده جمعيت در زمان t است. پس از انتخاب والدين از جمعيت ابتدا عمل بازترکيب بين آنها انجام مي‌شود. بازترکيب به نوعي مبادله خصوصيات ژنتيکي بين افراد جمعيت است. سپس عملگر جهش بر روي هر يک از فرزندان به طور جداگانه اعمال مي‌شود که طي آن يکي از ژن‌ها به طور تصادفي تغيير مي‌کند تا گونه جديدي را به وجود آورد. با جايگزيني فرزندان جديد در جمعيت

قبلي، جمعيت جديد P(t+1) در نسل جديد به وجود مي‌آيد. الگوريتم زماني متوقف مي‌شود که يا تعداد مشخصي از نسل‌ها توليد شده باشند و يا سطح برازندگي مطلوب بدست آمده باشد.

اگر اجزاء يک GA به صورت مناسب انتخاب شوند، پروسه باز توليد به طور پيوسته جمعيت را بهبود می بخشد و نهايتاً جمعيت به پاسخي نزديک به پاسخ بهينه مطلق همگرا مي‌شود. بنابراين اين الگوريتم ابزاري بسيار کارآمد براي جستجو در فضاهاي بزرگ و پيچيده (يعني فضاهاي جستجو با نقاط بهينه زياد) به شمار مي‌روند.

۲-۷- یک الگوریتم ژنتیکی ساده

الگوریتم ژنتیکی به روشهای مختلف و با عملگرهای متنوع پیاده سازی شده است. دیاگرام بلوکی که برای الگوریتم ژنتیکی در این پروژه به کار گرفته شده ، در شکل ۲-۱ آورده شده است.



شکل 2-۱ساختار یک الگوریتم ژنتیکی

انجام عملیات الگوریتم ژنتیکی نیاز به پیاده سازی سه مرحله دارد :

۱- مقدار دهی اولیه

۲- تولید مثل

۳- جایگزینی

در مقدار دهی اولیه ابتدا نقاطی از فضای جستجو به صورت تصادفی انتخاب می شود و سپس هر یک از آنها کد می شوند (تشکیل کروموزوم). هر یک از این کروموزوم ها را می توان یک جواب برای مسئله بهینه سازی در نظر گرفت. تعداد کروموزم های انتخاب شده برای مسئله، به جمعیت در نظر

گرفته شده بستگی دارد. برای هر کروموزوم مقدار تابع معیار محاسبه شده و به آن اختصاص می یابد. مجموعه این کروموزوم ها نسل اولیه را تشکیل می دهند.

در مرحله تولید مثل با استفاده از نسل فعلی، نسل جدید تولید می شود. برای این کار یک جفت کروموزوم از نسل فعلی و با احتمالی متناسب با مقدار تابع معیار آن کروموزوم، کروموزوم جدید (فرزندان) را تولید می کند. این فرزندان در نسل بعدی قرار مي گيرند. انتخاب، برش و جهش آن قدر ادامه می یابد تا فرزندان کافی برای پر شدن نسل جدید تولید شوند. سپس برای هر یک از فرزندان مقدار تابع معیار محاسبه شده و به آن اختصاص می یابد.

در مرحله جایگزینی، کروموزومهای جدید (فرزندان) جایگزین کروموزومهای قبلی (والدین) می شوند، یعنی نسل جدید جایگزین نسل قبلی می شود.

مراحل تولید مثل و جایگزینی آنقدر تکرار می شوند تا الگوریتم به اندازه کافی به جواب بهینه نزدیک شود. میزان نزدیک شدن به جواب بهینه که شرط پایان الگوریتم ژنتیکی است، به وسیله برنامه نویس مشخص می شود. برای این کار معمولاً معیارهایی در نظر گرفته می شود که باید برآورده شوند. [1]

۲-۸- نمایش ژنتیکی

روش‌هاي مختلفي براي نمايش ژنتيكي پاسخ‌هاي یک مساله وجود دارد. معمول‌ترین آنها رشته ارقام باينري، حقيقي، صحيح و ... است. از آنجا که تئوري‌هاي اوليه الگوريتم‌هاي ژنتیکی بر مبناي نمايش باينري بوده است، در تحقيقات و کاربردهاي مختلف از اين شيوه نمايش بيشتر استفاده شده است. در اين شيوه نمايش معمولاً پارامترهاي عددي به یک رشته باينري تبديل شده و از اتصال آنها یک رشته باينري که معرف یک کروموزوم است، بدست مي‌آيد.

در مسائل بهینه‌سازي با دامنه پيوسته، استفاده از نمايش حقيقي مناسب‌تر است. در اين شيوه نمايش، رشته‌اي از اعداد حقيقي كه هر يک از آنها نماينده يک پارامتر خاص مساله است، يک كروموزوم را تشكيل مي‌دهد. مزيت بارز نمايش حقيقي نسبت به باينري درجه تفكيک بي‌نهايت در اين نمايش است كه امكان جستجوي پيوسته در فضاي جستجو را مي‌دهد، برخلاف نمايش باينري كه در آن درجه تفكيک براي نمايش n بيت محدود به ۲n است. همچنين به دليل آنكه نيازي به تبديل كرموزوم‌ها به پارامترهاي فضاي مساله وجود ندارد، راندمان الگوريتم افزايش مي‌يابد و چون از نمايش مميز شناور داخلي كامپيوتر استفاده مي‌شود، حافظه كمتري مورد نياز است. در انتخاب عملگرها براي نمايش حقيقي آزادي عمل بيشتري وجود دارد. در كل، شيوه نمايش بايد متناسب با نوع مساله انتخاب شود و روش ارزيابي و عملگرهاي ژنتيكي قابل استفاده را نيز در نظر گرفت. [1]

۲-۹- توليد جمعيت اوليه [1]

پس از تصميم‌گيري درباره شيوه نمايش، توليد جمعيت اوليه اولين قدم در اجراي الگوريتم ژنتيکی است. روش‌هاي مختلفي براي اين كار وجود دارد. معمول‌ترين شيوه توليد تعداد مشخصي از كروموزوم‌ها به صورت يكنواخت در فضاي جستجو است. براي نمايش باينري، رشته‌هاي باينري با احتمال مساوي صفر يا يک بودن و براي نمايش حقيقي، اعداد حقيقي با توزيع يكنواخت در يک بازه توليد مي‌شوند. همچنين مي‌توان با توليد تعداد بيشتري از جواب‌ها به صورت تصادفي، جمعيت اوليه را با استفاده از جواب‌هاي بهتر تشكيل داد.

[1] initial generation

جمعیت اولیه را همچنین می‌توان از روي نتايج قبلي، دانش خبره انساني و يا به صورت ابتكاري توليد كرد كه مي‌تواند سرعت الگوريتم را افزايش دهد. اما در اين صورت احتمال از دست دادن تنوع ژنتيكي و يا به وجود آمدن گرايش‌هاي ناخواسته نيز وجود دارد.

در حالت كلي جمعيت كوچک شانس پيدا كردن جواب‌هاي خوب را كاهش مي‌دهد و جمعيت بزرگ هزينه‌هاي محاسباتي را افزايش مي‌دهد.

۲-۱۰- توابع هدف و برازندگی[1]

تابع هدف براي بدست آوردن معياري از عملكرد پاسخ در فضاي مساله به‌كار مي‌رود. در مسائل كمينه‌سازي بهترين پاسخ‌ها آنهايي هستند كه مقدار تابع هدف مربوط به آنها كمترين است. اين معيار ناخالص معمولاً به عنوان مرحله مقدماتي در تعيين عملكرد نسبي پاسخ‌ها در GA در نظر گرفته می شود. بنابراين تابع ديگري كه تابع برازندگي ناميده مي‌شود براي تبديل مقادير تابع هدف به معيار برازندگي نسبي به كار مي‌رود.

$$F(x) = g(f(x)) \qquad (۲-۴)$$

در رابطه (۲-۴) f تابع هدف، g تبديلي كه مقدار تابع هدف را به يک عدد نامنفي تبديل مي‌كند و F برازندگي نسبي می باشد. دو روش كلي براي اين تبديل وجود دارد: روش تناسبي و روش رتبه‌بندي.

در روش تناسبي براي هر پاسخ x_i با مقدار هدف $f(x_i)$ ، عدد برازندگي $F(x_i)$ طبق رابطه زير نسبت داده مي‌شود :

[1] fitness function

$$F(x_i)=\frac{f(x_i)}{\sum_{i=1}^{N_{ind}} f(x_i)} \qquad (۵-۲)$$

که در آن N_{ind} اندازه جمعیت است. معمولا پیش از این کار یک تبدیل دیگر به منظور مقیاس کردن و تنظیم مقادیر تابع هدف به صورت زیر به‌کار مي‌رود :

$$F(x)=af(x)+b \qquad (۶-۲)$$

که در آن a، عامل مقیاس‌گر است که براي مسائل کمینه‌سازي، منفي و براي مسائل بیشینه‌سازي مثبت است و b یک مقدار جابجاساز است که براي جلوگیري از منفي شدن مقادیر برازندگي به کار مي‌رود.

از معایب این روش امکان همگرایي زودهنگام به دلیل غالب شدن پاسخ‌هاي با برازندگي بالاتر در همان مراحل اولیه است. همچنین در صورتي‌که انحراف معیار جمعیت کوچک باشد، این روش گرایش کوچکي به سمت پاسخ‌هاي بهتر ایجاد مي‌کند.

در روش رتبه‌بندي به هر عضو جمعیت، مقدار برازندگي متناسب با رتبه آن عضو در جمعیت طبق رابطه (۲-۷) نسبت داده مي‌شود.

$$F(x_i)=2-SP+2(SP-1)\frac{x_i-1}{N_{ind}-1} \qquad (۷-۲)$$

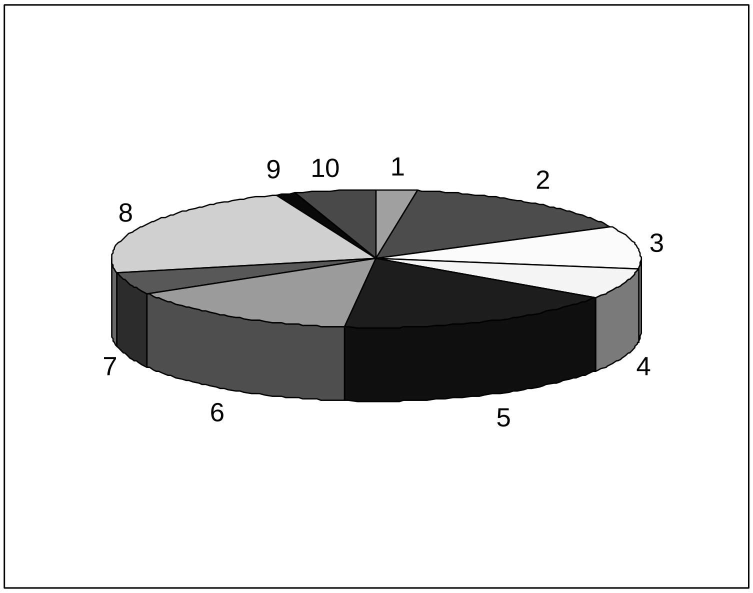
در رابطه (۲-۷)، SP عددي بین ۱و۲ است. [2]

۲-۱۱- انتخاب ژنتیکی

این عملگر ساز و کار بقای موجودی قوی تر در طبیعت را تقلید می کند، یعنی به موجود بهتر شانس بیشتر و به موجود بدتر شانس کمتری برای بقا می دهد. در هر مرحله با دو انتخاب، دو موجود به عنوان والدین در نظر گرفته می شوند.

در مرحله انتخاب، پاسخ‌ها براي مرحله بازتولید به طور تصادفي و با در نظر گرفتن احتمال بیشتر براي پاسخ‌هاي بهتر، انتخاب مي‌شوند. بدین منظور روش‌هاي مختلفي به کار مي‌رود که پرکاربردترین آنها روش رولت گردان است.

در روش رولت گردان از ایده این بازي براي انتخاب استفاده مي‌شود. همانطور که در شکل (۲-۲) مشاهده مي‌شود ابتدا به هر یک از اعضاء جمعیت یک قطاع دایره با زاویه متناسب با برازندگي آن نسبت داده مي‌شود. سپس به طور تصادفي به تعداد مورد نیاز روي دایره علامت‌گذاري مي‌شود و عضو مربوط به قطاع علامت‌گذاري شده براي بازترکیب انتخاب مي‌شود. براي پیاده‌سازي این روش ابتدا مجموع مقادیر برازندگي تمام اعضاء جمعیت (Sum) بدست مي‌آید و سپس اعضاء جمعیت به بازه‌هاي متصل به هم در بازه $[0 , Sum]$ نگاشت مي‌شود. براي انتخاب هر عضو عددي تصادفي در بازه $[0 , Sum]$ تولید مي‌شود و عضوي که عدد تولید شده در بازه مربوط به آن قرار دارد انتخاب مي‌شود.



شکل 2-۲روش رولت گردان

در حقیقت اگر f_i تابع معیار موجود i ام و مجموع تابع معیار موجودات آن نسل $\sum f_i$ باشد، احتمال انتخاب شدن آن موجود برای والد شدن $\frac{f_i}{\sum f_i}$ می شود. برای انتخاب در این حالت مي توان یک چرخ رولتی را مانند شکل در نظر گرفت که به هر موجود به نسبت تابع معیار قطاعی از چرخ اختصاص داده می شود.

موجوداتی که قطاع بزرگتری به آنها اختصاص یابد، احتمال بیشتری برای انتخاب شدن دارند. [1]

۲-۱۲- بازترکیب[1]

عملگر اصلي براي توليد پاسخ‌هاي جديد در GA، باز تركيب يا تقاطع است. بازترکیب مانند عملگر متناظر در طبيعت، پاسخ‌هاي جديدي توليد مي‌كند كه داراي بعضي از اجزاء از ساختار والدین است.

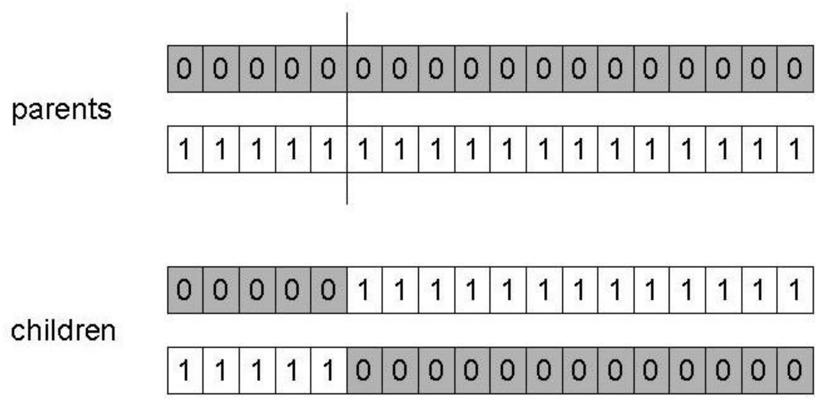
1 crossover

این عملگر به صورت تصادفي بر روي اعضاء انتخاب شده عمل مي‌کند. براي هر نمايش ژنتيکي دسته خاصي از عملگرها به کار مي‌روند. [1]

۲-۱۲-۱- عملگرهاي باز ترکيب براي نمايش باينري و گسسته

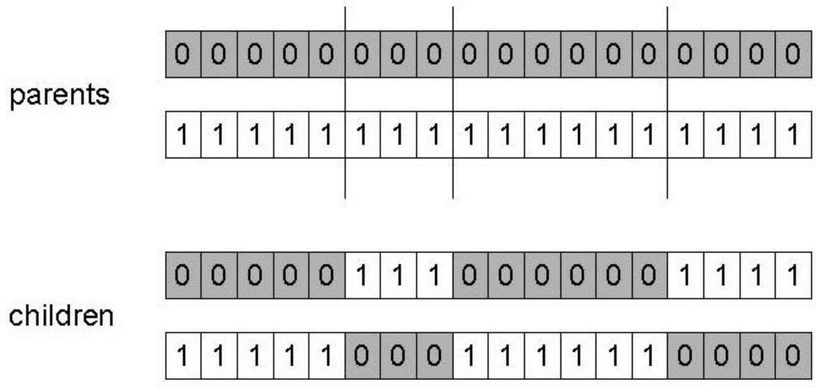
عملگرهايي که براي بازترکيب اعضاء يک جمعيت باينري و يا گسسته استفاده مي‌شوند، فرزنداني توليد مي‌کنند که هر يک از ژن‌هاي خود را از يکي از والدين به ارث مي‌برند. اين‌کار به روش‌هاي مختلفي امکان‌پذير است. ساده‌ترين روش، تقاطع يک نقطه‌اي است که در شکل ۲-۳ نشان داده شده است. در اين روش يک نقطه از ساختار رشته‌اي والدين به‌طور تصادفي انتخاب مي‌شود و با انفصال هر يک از والدين در آن نقطه و جابجايي دنباله‌ها، فرزندان توليد مي‌شوند.

در روش تقاطع چند نقطه‌اي چند نقطه تقاطع به‌طور تصادفي و به تعداد مشخص انتخاب مي شوند.



شکل 2-۳ روش تقاطع يک نقطه‌اي

سپس عمل انفصال و جابجايي به طور متوالي در هر نقطه تقاطع انجام مي‌شود. اين پروسه در شکل ۲-۴ نشان داده شده است.



شکل 2-٤روش تقاطع چند نقطه‌اي

یک تغییر ساده در روش تقاطع یک نقطه‌اي این است که قبل از جابجايي دنباله‌ها ترتیب ژن‌ها در والدین به طور مشابه به هم ریخته شود و پس از جابجايي مجددا به صورت اولیه ردیف شوند. این روش تقاطع به هم ریخته نامیده مي‌شود.

در پاره‌اي از موارد به خصوص زماني که جمعیت به سمت همگرايي پیش مي‌رود، به دلیل شباهت زیاد والدین ممکن است عملگر تقاطع فرزندان جدیدي را تولید نکند. براي رفع این مشکل، عملگر جانشین کاهش یافته به کار مي‌رود که با حذف قسمت‌هاي مشابه والدین، عملگر تقاطع را بر روي قسمت‌هاي متفاوت اعمال مي‌کند.

۲-۱۲-۲- عملگرهاي باز ترکیب براي نمایش حقیقي

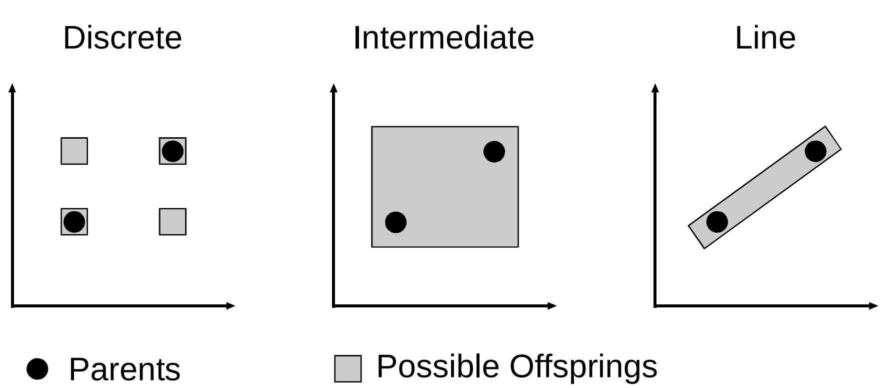
براي نمایـش حقیقي، عملگـرهاي مختلفـي به کار مي‌رود. عملگـر بازترکیب گسسته براي نمایش حقیقي مشابه عملگر تفاطع یکنواخت در نمایش باینري است که در آن مقادیر ژن‌ها بین والدین به طور تصادفي مبادله مي‌شود تا فرزندان جدیدي به وجود آیند.

در بازترکیب میانه‌اي پاسخ‌هاي جدید در اطراف و بین پاسخ‌هاي فعلي ایجاد مي‌شوند. پاسخ‌هاي جدید بر اساس رابطه (۲-۸) بدست مي‌آید.

$$O_1 = P_1 + \alpha(P_2 - P_1)$$
$$O_2 = P_2 + \alpha(P_1 - P_2) \qquad (۲-۸)$$

در این رابطه P_1 و P_2 کروموزوم‌هاي والدین و O_1 و O_2 کروموزوم‌هاي فرزندان به صورت برداري، و $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_L]^T$ که در آن α_iها اعدادي هستند با توزیع تصادفي یکنواخت در یک بازه که معمولا $[-0.25 \, , 1.25]$ است انتخاب مي‌شوند.

در صورتي که α_i ها یکسان باشند، بازترکیب خطي انجام مي‌شود که در این صورت فرزندان تنها روي خطي بین والدین و در محدوده انحراف α_i تولید مي‌شوند. در شکل ۲-۵ مقایسه‌اي بین روش‌هاي مختلف بازترکیب حقیقي در فضاي دوبعدي انجام شده است.



شکل 2-۵ روش‌هاي بازترکیب حقیقي

۲-۱۳- اپراتور جهش[1]

در پروسه تکامل طبيعي، جهش پروسه‌اي است که طي آن يک ژن از يک کروموزوم تغيير مي‌کند و يک ساختار جديد به وجود مي‌آيد. در GA، عملگر جهش به طور تصادفي و با احتمال پايين بر روي کروموزوم‌هاي فرزندان اعمال مي‌شود. نقش جهش به عنوان يک عملگر پس‌زمينه، تضمين جستجوي تمام فضاي جواب مساله و امکان بازيابي پاسخ‌هاي خوبي است که طي انتخاب و بازترکيب از دست داده شده‌اند. اين عملگر نيز براي نمايش باينري و حقيقي متفاوت است.

در نمايش باينري جهش با تغيير تصادفي يکي از ژن‌ها از صفر به يک و برعکس اعمال مي‌شود و در نمايش‌هاي گسسته، مقدار ژن مورد نظر را به يکي از مقادير ممکن تغيير می دهد.

$$100011010100111 \overset{IF\ P>P_{mutation}}{\Rightarrow} 100011110100111$$

اپراتورجهش برای نمایش باینری٦-2شکل

عملگر جهش در نمايش حقيقي با اعمال تغييرات تصادفي در محدوده هر متغير اعمال مي‌شود. يکي از روش‌هاي متداول استفاده از رابطه (۲-۹) است:

$$x_i' = x_i + S \times R \times SF \times \delta \quad (۲-۹)$$

که در آن x_i مقدار ژن انتخاب شده براي اعمال جهش، x_i' مقدار ژن پس از اعمال جهش، S=±۱ با احتمال مساوي براي تعيين جهت تغيير، R محدوده مقادير ژن، SF ثابتي در بازه $[0,1]$ براي

[1] mutation

انقباض محدوده تغيير و

$$\delta = \sum_{i=1}^{m} \alpha_i 2^{-i}$$
(۱۰-۲)

که در آن α_i با احتمال $1/m$، یک و در غیر این صورت صفر است و m نیز ۲۰ در نظر گرفته ميشود. [1]

۲-۱۴- جايگزيني

پس از اعمال عملگرهاي ژنتيکي، فرزندان بدست آمده بايد وارد جمعيت شوند. چند راهبرد مختلف براي جايگزيني وجود دارد. حالتي که تمام عناصر جمعيت با فرزندان توليد شده جايگزين شوند الگوريتم تناسبي يا بدون همپوشاني ناميده ميشود و در غير اينصورت الگوريتم داراي همپوشاني خواهد بود. پارامتر شکاف نسل، G، که عددي در بازه $[0,1]$ است براي بيان ميزان همپوشاني به کار ميرود و نشان دهنده نسبت جايگزيني فرزندان در نسل جديد است. معمولاً فرزندان جديد جايگزين اعضاء نامناسبتر جمعيت ميشوند و اگر تعداد فرزندان براي جايگزيني بيشتر از تعداد مورد نياز باشد بايد پس از ارزيابي بهترينها را براي جايگزيني انتخاب کرد.

يکي از روشهاي سودمند براي جايگزيني، راهبرد انتخاب نخبگان است که در اين حالت تعداد کمي (معمولا یک يا دو) از بهترين پاسخهاي هر نسل مستقيماً وارد نسل بعدي ميشوند. به اين ترتيب هميشه تابع هزينه در حال کاهش خواهد بود.

۲-۱۵- شرايط توقف الگوريتم **[1],[2]**

از آنجا که GA یک روش جستجوي تصادفي است، تعيين شرط همگرايي بهطور مرسوم براي آن مشکل است. چرا که برازندگي جمعيت براي تعدادي از نسلها ممکن است ثابت باشد و پس از آن یک

پاسخ بهتر پيدا شود. شرايط معمول براي توقف الگوريتم عبارتند از :

- تعداد نسلها مشخص شده باشد.
- برازندگي متوسط جمعيت به یک حد از پيش تعيين شده برسد.
- تغيير چنداني در پاسخ‌ها رخ ندهد و جمعيت همگرا شود.

۲-۱۶- ويژگي‌هاي الگوريتم‌هاي ژنتيکی

در حالت کلي ۳ روش اصلي براي جستجو وجود دارد: روش‌هاي مبتني بر گراديان، شمارشي و تصادفي.

در روش‌هاي مبتني بر گراديان جستجو براي يافتن نقاط با شيب صفر در تمام جهات انجام مي‌شود و داراي ديد محلي است و از اين‌رو ممکن است به جاي يافتن پاسخ بهينه مطلق، به یک پاسخ بهينه محلي همگرا شود. شرط لازم براي اعمال اين روش‌ها وجود مشتقات و بنابراين پيوستگي فضاي جستجو است.

در روش‌هاي شمارشي تابع هدف به ازای تمام نقاط فضاي جستجو محاسبه مي‌شود. اين روش ها نيز اگرچه ساده هستند، اما راندمان محاسباتي بسيار پاييني دارند و براي فضاهاي بزرگ قابل اعمال نيستند.

روش‌هاي تصادفي به جستجوي تصادفي در فضاي جستجو مي‌پردازند و بهترين نقاط ارزيابي شده را به عنوان کانديداهاي بهينه ذخيره مي‌کنند. اين روش‌ها در فضاهاي بزرگ چندان بهتر از روش‌هاي شمارشي نيستند، مگر آنکه هدايت شده باشند که مستلزم داشتن اطلاعاتي از فضاي جستجو است.

GA روشي تصادفي نيست بلكه از انتخاب تصادفي بهره مي‌گيرد. تفاوت‌هاي اصلي GA با روش‌هاي ذكر شده به شرح زير است:

- GA بر روي نمايش كد شده پارامترها كار مي‌كند، نه خود آنها، و بنابراين از شباهت‌هاي موجود در ساختار ژنتيكي براي يافتن پاسخ بهينه مطلق استفاده مي‌كند.
- در GA به جاي یک پاسخ، جمعيتي از پاسخ‌ها در نظر گرفته مي‌شود و بنابراين كل مساله در هر زمان در نظر گرفته مي‌شود، نه فقط یک عنصر.
- GA از اطلاعات مربوط به بازدهي (تابع هدف) استفاده مي‌كند و از مشتقات و ساير اطلاعات كمكي استفاده نمي‌كند.
- در GA از قواعد تصادفي استفاده مي‌شود كه نياز به وجود دانش درباره فضاي جستجو را مرتفع مي‌سازد.

فصل سوم - موتورهای DC

۳-۱- مقدمه

ماشين جريان مستقيم DC يک وسيله تبديل انرژي با فراگيري زياد از نقطه نظر کاربرد مي باشد. اين ماشين مي تواند گشتاور راه اندازي بالا و متغير مورد نياز بارها را تأمين کند. در حال حاضر ماشين DC در حوزه وسيع تغييرات سرعت و برگشت سريع انرژي قابل استفاده مي باشد. مشخصه هاي اصلي اين ماشين را مي توان بيشتر توسط مدارهاي فيدبک تغيير داد. باتوجه به اين خصوصيات برجسته، ماشين DC داراي قابليت انعطاف بالايي بوده و بنابراين کاربرد زيادي در صنعت دارد. اين ماشينها عليرغم هزينه اوليه بالايي که دارند، در صنعت بخصوص براي کارهاي سخت همانند محرکهاي کارخانه فولاد سازي بطور وسيعي مورد استفاده قرار مي گيرند.

ماشينهاي DC داراي ميدان چند قطبي (قطبهاي متناوب N و S) مي باشند. در ماشينهاي جريان مستقيم معمولي هسته استاتور مورق نيست اما هسته آرميچر مورق است تا تلفات جريان گردابي کاهش يابد. مدار مغناطيسي ماشينهاي جريان مستقيم که در سيستم هاي کنترل به کار مي روند به طور کامل مورق است. اين کار براي به حداقل رساندن اثر ميرائي جريان گردابي براى سريع تر کردن پاسخ لازم در سيستم هاي کنترل است.

در حال حاضر توليد سالانه ماشينهاي جريان مستقيم حدود ۴۰ درصد توليد و فروش ماشينهاي الکتريکي است. و اين به خاطر آن است که تمام وسائط نقليه براي ذخيره انرژي الکتريکي از باتري استفاده مي کنند. در اين وسائط نقليه و اتومبيل ها موتورهاي DC براي راه اندازي موتور، جهت برف پاک کن، کولر و چرخاندن ساير دستگاه ها و لوازم وسيله نقليه مورد استفاده قرار مي گيرد. براي اين

موارد استعمال، سالانه ميليونها موتور جريان مستقيم ساخته مي شود. در كاربردهاي صنعتي كه كنترل دقيق سرعت همراه يا بدون گشتاور لازم است موتور DC بي نظير است. بنابراين موتورهاي DC عموماً در صنايع نورد فولاد و آلومينيوم، بيلهاي الكتريكي، آسانسور يا بالابرهاي الكتريكي، لوكوموتيوهاي راه آهن و تجهيزات متحرک زميني بزرگ به كار گرفته مي شوند.

در ماشين هاي DC، پيچک ميدان، يک پيچک متمركز شده روي قطب هاي برجسته مي باشد كه به بدنه استاتور پيچ شده است و پيچک آرميچر سيم پيچي توزيع شده اي است كه در شيارهاي محيط رتور استوانه اي جا داده شده است. [3]

۳-۲- اصول اساس عملكرد موتورهاي **DC**

موتور DC اساساً يک مبدل گشتاور نيرو است كه انرژي الكتريكي را به انرژي مكانيكي تبديل مي كند. گشتاور نيروي موجود روي ميل گردان موتور با شار ميدان و جريان آرميچر تناسب مستقيم دارد. رابطه ميان گشتاور نيروي ايجاد شده، شار ϕ و جريان i_a عبارت است از:

$$T_m = K_m \phi i_a \qquad (۳-۱)$$

در اين رابطه T_m گشتاور نيروي موتور (نيوتن متر)، ϕ شار مغناطيسي (وبر)، i_a جريان آرميچر (آمپر)، و K_m ثابت تناسب است.

وقتي يک رسانا در ميدان مغناطيسي حركت مي كند، علاوه بر ايجاد گشتاور، ولتاژي در دو سر آن ايجاد مي شود. اين ولتاژ كه ضد emf نام دارد با سرعت ميل گردان متناسب است و ميل به مخالفت با شار جريان دارد. رابطه بين ضد emf و سرعت ميل گردان عبارت است از:

$$e_b = K_m \phi \omega_m \qquad (۳-۲)$$

در این رابطه e_b ضد emf (ولت) و ω_n سرعت میل گردان (rad/s) موتور است. روابط (۳-۱) و (۳-۲) اساس کار موتور DC اند.[۴]

۳-۳- رده بنديهاي اصلي موتورهاي DC

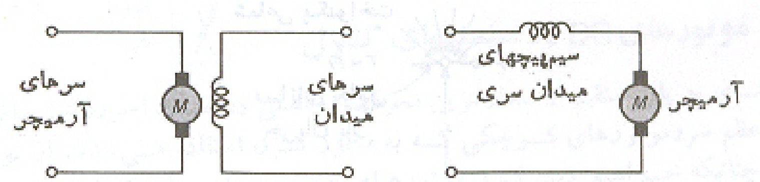
موتورهاي DC را برمبناي ميدان مغناطيسي ايجاد شده و طراحي و ساخت آرميچر مي توان به چندين دسته وسيع رده بندي كرد. براساس ميدان مغناطيسي ايجاد شده موتورهاي DC در دو رده موتورهاي با شار مغناطيسي متغير و موتورهاي با شار مغناطيسي ثابت قرار مي گيرند. در موتورهاي با شار مغناطيسي متغير ميدان مغناطيسي را سيم پيچهاي ميدان كه به منابع خارجي متصل اند توليد مي كنند. اين نوع موتورها خود در دو رده فرعي قرار دارند: موتور با ميدان سري كه در آنها ميدان با آرميچر سري است، و موتور با ميدان تحريک جداگانه كه در آنها سيم پيچ ميدان از آرميچر جداست.

در موتورهاي DC با ميدان سري چون شار مغناطيسي در موتور با جريان ميدان كه متغير است، متناسب می باشد رابطه بين گشتاور نيرو و سرعت معمولاً غير خطي است. بنابراين موتورهاي DC با ميدان سري تنها در كاربردهاي خاصي كه گشتاور نيروي زياد با سرعت كم لازم است به كار مي رود. در اين موتور وقتي كه سرعت افزايش مي يابد گشتاور نيرو معمولاً به سرعت افت مي كند.

در موتور DC با تحريک جداگانه، چون شار مغناطيسي از جريان آرميچر مستقل است، اين موتور را مي توان در گستره وسيعي از خارج كنترل كرد.

موتور DC با شار مغناطيسي ثابت را نيز موتور DC با آهنرباي دائم (PM) [1] مي نامند. در اين موتور ميدان مغناطيسي را آهنربايي دائم ايجاد مي كند و ثابت است. همين امر موجب مي شود منحنيهاي گشتاور نيرو – سرعت موتور نسبتاً خطي باشد.

[1] Permanent Magnet



شکل ۳-۱(الف) آرمیچر و اتصال میدان موتور DC سري. (ب) موتور DC با میدان متغیر جداگانه

موتورهاي DC با آهنرباي دائم را مي توان براساس نوع کموتاسیون و طراحي آرمیچر به رده هاي فرعي تر نیز تقسیم بندي کرد. موتورهاي DC متداول جاروها و کموتاتورهاي مکانیکي دارند، اما در برخي از موتورهاي DC عمل کموتاسیون الکترونیکي انجام مي شود و این نوع موتورها را موتور DC بي جارو می نامند. [5]

۳-۴- مشخصات کاري موتوري هاي DC

براي موتورهاي DC ولتاژ تغذیه معمولا ثابت است و متغیرها عبارتند از : سرعت، گشتاور و غیره، سه مشخصه کاري مهم موتورهاي DC در زیر آمده است.

۱) مشخصه سرعت – جریان آرمیچر

۲) مشخصه گشتاور – جریان آرمیچر

۳) مشخصه سرعت – گشتاور

موضوع این بخش شرح هر یک از این مشخصه هاي کاري براي انواع مختلف موتورهاي DC مي باشد.

۳-۵- موتور شنت DC [5]

در این موتور براي ولتاژ تغذیه ثابت، جریان میدان ثابت است. بازاء یک مقدار کوچک از جریان آرمیچر، اثر ضد مغناطیسي عکس العمل آرمیچر تقریبا ناچیز است و بنابراین شار فاصله هوايي تقریبا ثابت

است. براي مقادير زياد جريان آرميچر (يا جريان بار)، اثر ضد مغناطيسي عكس العمل آرميچر شار فاصله هوايي را اندكي كاهش مي دهد.

سرعت یک موتور DC برابر است با:

$$\omega = \frac{E_a}{K_a \phi} \qquad (۳-۳)$$

$$E_a = V_t - I_a r_a \qquad (۴-۳)$$

$$\Rightarrow \omega_m = \frac{V_t - I_a r_a}{K_a \phi} \qquad (۵-۳)$$

۳-۵-۱- مشخصه سرعت – جريان:

براي ولتاژ تغذيه V_t و جريان ميدان I_f ثابت سرعت موتور توسط افت ولتاژ $I_a r_a$ و اثر ضد مغناطيسي عكس العمل آرميچر تغيير مي كند. با افزايش I_a، اثر ضد مغناطيسي عكس العمل آرميچر افزايش مي يابد كه در نتيجه آن شار ميدان كاهش مي يابد و بنابراين سرعت موتور افزايش مي يابد. اما با افزايش I_a افت ولتاژ $I_a r_a$ افزايش مي يابد و صورت كسر $(V_t - I_a r_a)$ كاهش مي يابد. بنابراين سرعت موتور كاهش مي يابد با افزايش I_a از سرعت بي باري ω_{mo} اندكي كاهش مي يابد. از آنجائيكه I_a در بي باري خيلي كوچک است سرعت موتور شنت در بي باري ω_{mo} از رابطه (۶-۳) بدست مي آيد.

$$\omega_{mo} = \frac{V_t}{K_a \phi} \qquad (۶-۳)$$

در حالتي كه اثر عكس العمل آرميچر (AR) ناچيز است، مخرج كسر رابطه (۳-۶) ثابت است، در نتيجه سرعت كاهش بيشتري با افزايش I_a دارد. [5]

۳-۵-۲- مشخصه گشتاور – جريان

عبارت $T_e = K_a \phi I_a$ نشان مي دهد كه چون شار ϕ در يک موتور شنت ثابت است، پس گشتاور بطور خطي با جريان آرميچر I_a افزايش مي يابد. در هر حال براي I_a بزرگ، شار خالص بواسطه اثر عكس العمل آرميچر كاهش مي يابد. بنابراين مشخصه هاي جريان گشتاور از خط مستقيم خارج مي شود.

۳-۵-۳- مشخصه سرعت – گشتاور

مشخصه سرعت – گشتاور را نيز مشخصه مكانيكي تحت شرايط كار پايدار مي نامند كه به طريق زير بدست مي آيد. از رابطه (۳-۵) داريم:

$$\omega_m = \frac{V_t - I_a r_a}{K_a \phi}$$

$$T_e = K_a \phi I_a$$

$$I_a = \frac{T_e}{K_a \phi}$$

با جايگزين كردن مقدار I_a در رابطه ۳-۶ داريم:

$$\omega_m = \frac{1}{K_a\phi}(V_t - \frac{T_e r_a}{K_a\phi})$$

(۷-۳)

$$= \frac{V_t}{K_a\phi} - r_a \frac{T_e}{K^2{}_a\phi^2} = \omega_m - r_a \frac{T_e}{K^2{}_a\phi^2}$$

از رابطه (۳-۷) دیده مي شود که با افزایش T_e سرعت کاهش مي یابد. باید توجه داشت که براي T_e بزرگتر، I_a بزرگتري مورد نیاز است و این خود باعث کاهش شار فاصله هوایي ϕ مي شود که ناشي از اشباع و اثر عکس العمل آرمیچر مي باشد. از آنجائیکه با افزایش T_e، ϕ کاهش مي یابد به $\frac{T_e}{\phi^2}$ سرعت افزایش مي یابد و سرعت با افزایش گشتاور شدیداً کاهش مي یابد.

اگر اثر آرمیچر ناچیز باشد، $(K_a\phi)^2$ در رابطه (۳-۷) ثابت باقي مي ماند در نتیجه سرعت با T_e به کندي کاهش مي یابد. [5]

۳-۶- موتور سري DC

براي یک موتور سري جریان میدان برابر جریان آرمیچر مي باشد.

۳-۶-۱- مشخصه جریان – سرعت

اگر اثرات اشباع و عکس العمل آرمیچر ناچیز باشند پس شار اصلي ϕ مستقیماً متناسب با جریان آرمیچر I_a است. بنابراین مي توان نوشت $\phi = CI_a$ که C مقدار ثابتي است.

$$E_a = K_a\phi\omega_m = V_t - I_a(r_a + r_s) \quad (۸-۳)$$

$$\omega_m = \frac{V_t}{K_a\phi} - \frac{I_a(r_a + r_s)}{K_a\phi} \quad (۹-۳)$$

یا

$$\omega_m = \frac{V_t}{K_a C I_a} - \frac{(r_a + r_s)}{K_a C} \quad (۱۰-۳)$$

از رابطه (۳-۱۰) دیده مي شود که با صرفنظر از اثرات اشباع و عکس العمل آرميچر، مشخصه سرعت – جريان يک موتور سري به صورت يک هذلولي است.

حال اثرات اشباع و عکس العمل آرميچر در نظر گرفته می شود. با افزايش مقدار I_a، شار ϕ افزايش مي يابد اما به علت اثر ضد مغناطيسي عکس العمل آرميچر و اشباع، شار فاصله هوايي ϕ تمايل دارد که تقريباً ثابت باقي بماند. رابطه (۳-۹) نشان مي دهد که براي ϕ ثابت عبارت $\frac{V_t}{K_a\phi}$ ثابت است و عبارت $\frac{I_a(r_a + r_s)}{K_a\phi}$ بطور خطي با I_a افزايش مي يابد.

در بي باري جريان آرميچر خيلي کوچک است زيرا قدرت ورودي به موتور فقط بايد بر تلفات بي باري غلبه کند، بنابراين افت ولتاژ $I_a(r_a + r_s)$ در بي باري در مقايسه با V_t تقريباً ناچيز است. از رابطه (۳-۹) سرعت بي باري ω_{mo} برابر است با:

$$\omega_{mo} = \frac{V_t}{K_a\phi} = \frac{V_t}{K_a C I_a} \quad (۱۱-۳)$$

از آنجائيکه ω_{mo} با I_a رابطه معکوس دارد، سرعت بي باري موتور سري در اثر جريان بي باري کوچکي بطور خطرناکي افزايش مي يابد. بدين منظور موتور سري بايد هميشه با بار مکانيکي کوپل شده با آن به گردش درآيد و کار کند. [5]

۳-۶-۲- مشخصه گشتاور – جريان:

با صرفنظر از اثرات اشباع و عکس العمل آرميچر داريم:

$$\phi = CI_a \qquad (۱۲\text{-}۳)$$

$$\Rightarrow T_e = K_a \phi I_a = K_a C I^2{}_a = C_1 I_a^2 \qquad (۱۲\text{-}۳)$$

رابطه (۳-۱۳) نشان مي دهد که گشتاور متناسب با مجذور I_a است و بنابراين مشخصه گشتاور – جريان سهمي مي باشد. اما براي I_a بيشتر، شار اصلي متمايل است که تقريباً ثابت باقي بماند. در نتيجه مشخصه گشتاور – جريان براي مقادير بيشتر I_a به خط راست نزديک مي شوند. [5]

۳-۶-۳- مشخصه هاي سرعت – گشتاور:

اگر از اشباع و عکس العمل آرميچر صرفنظر شود از رابطه (۳-۱۳) داريم:

$$I_a = \sqrt{\frac{T_e}{K_a C}} \qquad (۱۴\text{-}۳)$$

با جايگذاري اين مقدار I_a در رابطه ۳-۱۰ داريم:

$$\omega_m = \frac{V_t}{\sqrt{K_a C I_a}} - \frac{r_a + r_s}{K_a C} \qquad (۱۵\text{-}۳)$$

بنابراين با صرفنظر از اثرات اشباع و عکس العمل آرميچر، مشخصه سرعت- گشتاور به صورت هذلولي است.

با در نظر گرفتن اثرات اشباع و عکس العمل آرميچر، گشتاور بيشتري براي جريانهاي زياد مورد نياز است و اين جريانهاي زياد تمايل دارند که شار فاصله هوايي ϕ ثابتي ايجاد کنند. اين داراي اين اثر است که T_e تقريباً متناسب با I_a (يعني $T_e = K_a I_a$) باشد و بدين منظور از رابطه ۳-۱۰ داريم:

$$\omega_m = \frac{V_t}{K_a\phi} - \frac{T_e(r_a + r_s)}{K^2{}_a\phi} \qquad (۱۶-۳)$$

بنابراين به ازاء مقدار مشخص T_e، هنگامي كه شار فاصله هوايي ϕ در اثر اشباع و عكس العمل مغناطيسي تقريباً ثابت باقي مي ماند مشخصه سرعت-گشتاور به خط صاف نزديک مي شود. [5]

۳-۷- راه اندازي موتور DC

در لحظه شروع راه اندازي سرعت موتور صفر است و بنابراين نيروي ضد محركه الكتريكي E_a ($E_a = K_a\phi\omega_m$) نيز صفر مي باشد. در نتيجه براي مدار آرميچر رابطه ولتاژ براي موتور شنت برابر $V_t = 0 + I_a r_a$ و براي موتورهاي سري و كمپوند براي موتور شنت برابر $\frac{V_t}{r_a}$ و براي موتورهاي سري و كمپوند برابر $\frac{V_t}{r_a + r_s}$ مي باشد.

از آنجائيكه مقاومت هاي r_a و $r_a + r_s$ خيلي كوچک مي باشند موتور جريان راه اندازي بزرگي از منبع تغذيه مي كشد. براي مثال يک موتور شنت ۲۵۰ ولت 10Kw كه داراي مقاومت مدار آرميچر $0/2\Omega$ مي باشد اگر كليد مستقيماً بسته شود جريان راه اندازي آرميچر برابر $\frac{250}{0/2} = 1250A$ خواهد بود در حاليكه جريان نامي آرميچر فقط ۴۰A مي باشد.

چنين جريان راه اندازي بزرگي كه توسط موتور كشيده مي شود نتايج زير را دربر دارد:

۱) ايجاد جرقه زيان آور در كموتاتور

۲) آسيب ديدن سيم پيچ آرميچر و خراب شدن عايق در اثر گرماي بيش از اندازه

۳) گشتاور راه اندازي بالا و شتاب سريع که مي تواند به قسمت هاي متحرک موتور و بار آسيب برساند

۴) افت زياد ولتاژ تغذيه

بدين منظور جريان آرميچر بايد به مقداري محدود گردد تا موتور کموتاسيون بدون خطر را همراه داشته باشد. اين جريان مناسب را مي توان با قرار دادن مقاومت خارجي در مدار آرميچر بدست آورد. با شتاب گرفتن موتور نيروي ضد محرکه الکتريکي E_a در آرميچر توليد شده و جريان آرميچر را به مقدار کم کاهش مي دهد. بنابراين مقاومت خارجي وارد شده در مدار آرميچر بايد به تدريج کاهش مي يابد تا آرميچر شتاب بگيرد. اگر اين مقاومت کمکي قرار داده شده از مدار آرميچر برداشته نشود نتايج زير حاصل مي شود:

۱) کاهش سرعت کار موتور

۲) تلفات اضافي انرژي و بنابراين کاهش راندمان

در هنگام راه اندازي موتورهاي کمپوند و شنت بايد احتياط لازم را مبذول داشت. در اين موتور ها در زمان راه اندازي به دو علت زير بايد مقدار تحريک ميدان ماکزيمم باشد:

۱) جريان ميدان بزرگ باعث کار موتور با سرعت کم مي گردد. در نتيجه زمان مورد نياز از حالت سکون تا رسيدن به اين سرعت کم کاهش است بنابراين گرماي کمتري در زمان راه اندازي در آرميچر ايجاد شود.

۲) گشتاور موتور لازم براي غلبه بر گشتاور بار و اصطکاک متناسب با حاصلضرب I_f و I_a است. از آنجائيکه I_f در مقدار ماکزيمم مجاز نگه داشته مي شود جريان آرميچر در زمان راه اندازي به ازاء يک مقدار مشخص گشتاور بار بايد مي نيمم باشد. اين گشتاور راه اندازي مي نيمم، اثر بيشتري در بهبود کموتاسيون موتور و شتاب گرفتن بار دارد. بنابرين رئوستايي که با سيم پيچ

میدان شنت سري شده است در حالت راه اندازي موتورهاي کمپوند و شنت داراي مقاومت صفر است. براي موتورهاي DC کوچک، هيچ مقاومتي راه اندازي در مدار آرميچر مورد نياز نمي باشد. [۴]

۳-۸- کنترل سرعت موتورهاي DC [5]

عبارت «کنترل سرعت» براي تغيير عمدي سرعت که بطور دستي يا اتوماتيک انجام مي شود بکار مي رود. تغييرات طبيعي سرعت ناشي از تغيير بار مشمول عبارت کنترل سرعت نمي باشد.

موتورهاي DC براي رنج وسيع کنترل سرعت بسيار مناسب مي باشند و بنابراين براي بسياري از محرک هاي قابل تنظيم سرعت ضروري هستند.

سرعت يک موتور DC از رابطه (۳-۵) بدست مي آيد و براي راحتي دوباره نوشته شده است.

$$\omega_m = \frac{V_t - I_a r_a}{K_a \phi}$$

که ثابت آرميچر $K_a = \frac{pZ}{2\pi a}$ و ϕ شار ميدان هر قطب باشد

با توجه به رابطه (۳-۵) براي يک موتور DC سه روش کنترل سرعت وجود دارد که عبارتند از:

۱) تغيير مقاومت در مدار آرميچر

۲) تغيير شار ميدان

۳) تغيير ولتاژ ترمينال آرميچر

قبل از شرح این ها باید ابتدا اصطلاحات سرعت پایه، تنظیم سرعت، رنج (حدود) سرعت درایو با قدرت ثابت و درایو با گشتاور ثابت را تعریف کرد.

حال به تعریف برخی اصطلاحات که در کنترل سرعت موتورهای DC کاربرد دارن می پردازیم:

۳-۸-۱- سرعت پایه:

این سرعت سرعت گردش موتور در ولتاژ و جریان نامي آرمیچر مي باشد. سرعت پایه برابر سرعت نامي یا سرعت نوشته شده روي پلاک موتور مي باشد.

۳-۸-۲- تنظیم سرعت:

اگر تغییرات سرعت از بي باري تا بار کامل برابر $\Delta\omega_m$ باشد تنظیم سرعت با نسبت $\Delta\omega_m$ به سرعت نامي (سرعت پایه) ω_m تعریف مي شود:

$$\text{درصد تنظیم سرعت} = \frac{\Delta\omega_m}{\omega_m} \times 100 \qquad (۱۷-۳)$$

۳-۸-۳- رنج (حدود) سرعت:

رنج سرعت به صورت نسبت ماکزیمم سرعت مجاز به مي نیمم سرعت مجاز موتور تعریف مي شود. هنگامي که رنج سرعت یک موتور مشخص شد باید ذکر گردد که آیا این رنج سرعت در بي باري، بار کامل یا کسري از بار کامل است.

۳-۸-۴- درایو با قدرت ثابت:

اگر قدرت محور موتور (گشتاور محور × سرعت) در رنج سرعت معین ثابت بماند، سیستم را درایو قدرت ثابت مي نامند. توجه داشته باشید که در درایو قدرت ثابت، در سرعتهاي کم مقدار گشتاور

بالاتر است و در سرعتهاي زياد گشتاور پائين است. اندازه موتور هميشه توسط بالاترين گشتاور مورد نياز در پاوين ترين سرعت تعيين مي گردد.

۳-۸-۵- درايو با گشتاور ثابت:

اگر گشتاور محور موتور در رنج سرعت معين ثابت باقي بماند سيستم را درايو گشتاور ثابت مي نامند. توجه داشته باشيد كه در درايو گشتاور ثابت قدرت محور با تغييرات سرعت تغيير مي كند.

۳-۹- ساخت مدل رياضي براي موتورهاي **DC [7]**

چون موتورهاي DC در سيستمهاي كنترل كاربرد گسترده دارند، به منظور تحليل اين موتورها ساخت مدل رياضي آنها براي كاربردهاي كنترل ضروري است. مدل رياضي موتورهاي DC با آهنرباي دائم و تحريک جداگانه تنها در مشخصات ميدانهاي مغناطيسي با هم تفاوت دارند.

براي نمايش موتور DC نمودار مدار معادل شكل ۳-۲ به كار برده مي شود. مدل آرميچر را به صورت يک مدار با مقاومت R_a، اندوکتانس L_a سري با آن، و ولتاژ e_b نشان دهنده ضد emf در آرميچر وقتي روتور مي چرخد، درنظر گرفته می شود. شار شکاف هوايي را با $\phi(t)$ مشخص مي شود. متغيرها و پارامترها عبارت اند از:

$i_a(t)$، جریان آرمیچر — L_a، اندوکتانس آرمیچر

R_a، مقاومت آرمیچر — $e_a(t)$، ولتاژ آرمیچر

$e_b(t)$، ضد emf — K_b، ثابت ضد emf

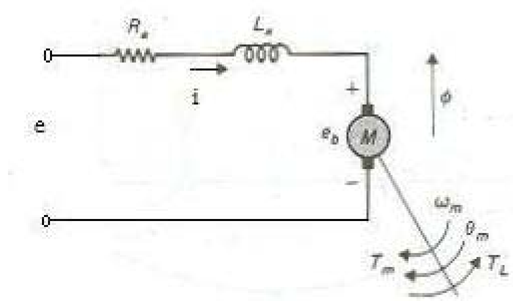
$T_L(t)$، گشتاور نیروي بار — $\phi(t)$، شار مغناطیسي شکاف هوایي

$T_m(t)$، گشتاور نیروي موتور — $\omega_m(t)$، سرعت زاویه اي روتور

$\theta_m(t)$، جابه جایي روتور — J_m، لختي روتور موتور

K_T، ثابت گشتاور نیرو — B_m یا f، ضریب اصطکاک ویسکوز

با توجه به نمودار مداري شکل ۳-۲ کنترل موتور DC به صورت ولتاژ اعمالي $e_a(t)$ به دو سر آرمیچر اعمال مي شود. مستقل از نوع برپایي میدان مغناطیسي توسط ساختار میدان، فرض مي شود شار شکاف هوایي ثابت است و با ϕ مشخص مي شود. براي تحلیل خطي فرض مي شود گشتاور نیروي ایجاد شده در موتور با شار شکاف هوایي و جریان آرمیچر متناسب است.



شکل ۳-۲مدل یک موتور DC با تحریک جداگانه.

$$T_m(t) = K_m\phi\ \ i_a(t) = K_T i_a(t) \qquad (۱۸\text{-}۳)$$

بنابراين در اين رابطه K_T ثابت گشتاور نيرو برحسب Nm/A است.

معادله هاي علت و معلولي سيستم شكل ۳-۲ با شروع از ولتاژ ورودي كنترل $e_a(t)$ عبارت اند از

$$\frac{di_a}{dt} = \frac{1}{L_a} e_a(t) - \frac{R_a}{L_a} i_a(t) - \frac{1}{L_a} e_b(t) \qquad (۳-۱۹)$$

$$T_m(t) = K_i i_a(t) \qquad (۳-۲۰)$$

$$e_b(t) = K_b \frac{d\phi_m(t)}{dt} = K_b \omega_m(t) \qquad (۳-۲۱)$$

$$\frac{d^2\theta_m(t)}{dt^2} = \frac{1}{J_m} T_m(t) - \frac{1}{J_m} T_m(t) - \frac{B_m}{J_m} \frac{d\theta_m(t)}{dt} \qquad (۳-۲۲)$$

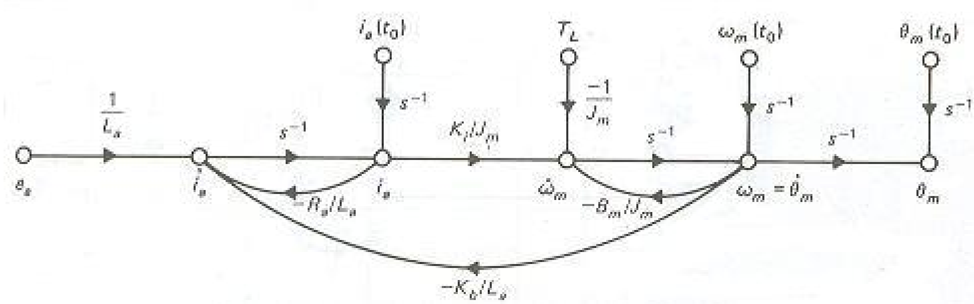
در معادله آخر $T_L(t)$ گشتاور نيروي اصطكاكي بار نظير اصطكاك كولني است.

معادله هاي (۳-۱۹) تا (۳-۲۲) نشان مي دهند كه $e_a(t)$ علت العلل است؛ معادله (۳-۱۹) نشان مي دهد كه $di_a(t)/dt$ معلول بلافصل $e_a(t)$ است، معادله (۳-۲۰) نشان مي دهد كه $i_a(t)$ علت ايجاد $T_m(t)$ است. در معادله (۳-۲۱) ضد emf تعريف مي شود، و بالاخره معادله (۳-۲۲) نشان مي دهد كه چگونه گشتاور نيرو باعث پيدايش جا به جايي زاويه اي $\theta_m(t)$ مي شود.

متغيرهاي حالت سيستم را مي توان به صورت $i_a(t)$، $\omega_m(t)$، و $\theta_m(t)$ تعريف كرد. با جايگذاري مستقيم و حذف همه متغيرهاي غير حالت از معادله هاي (۳-۱۹) تا (۳-۲۲)، معادله هاي حالت سيستم موتور DC به صورت ماتريسي- برداري عبارت است با:

$$\begin{bmatrix} \dfrac{di_a(t)}{dt} \\ \dfrac{d\omega_m(t)}{dt} \\ \dfrac{d\theta_m(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{K_b}{L_a} & 0 \\ \dfrac{K_i}{J_m} & -\dfrac{B_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega_m(t) \\ \theta_m(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} e_a(t) - \begin{bmatrix} 0 \\ \dfrac{1}{J_m} \\ 0 \end{bmatrix} T_L(t) \qquad (۲۳\text{-}۳)$$

توجه شود که در این حالت، $T_L(t)$ به منزله ورودي دوم در این معادلات حالت شمرده مي شود.



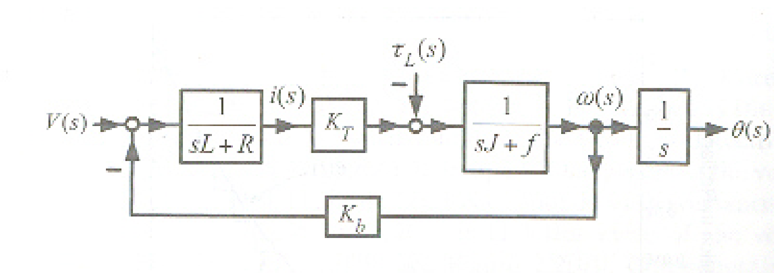
شکل ۳-۳ نمودار حالت موتور DC

نمودار حالت سیستم به کمک معادله (۳-۲۳) در شکل ۳-۳ رسم شده است. تابع تبدیل میان جابجایي موتور و ولتاژ ورودي از نمودار حالت چنین به دست مي آید.

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{K_i}{L_a J_m S^3 + (R_a J_m + B_m L_a)S^2 + (K_b K_i + R_a B_m)S} \qquad (۲۴\text{-}۳)$$

در این رابطه T_L صفر درنظر گرفته شده است.

شکل ۳-۴ نمايش سيستم موتور DC به صورت نمودار بلوکي است. مزيت استفاده از نمودار بلوکي اين است که اين نمودار تصوير واضحي از تابع تبديل ميان هريک از بلوکهاي سيستم عرضه مي کند. چون از مخرج معادله (۳-۲۴) مي توان S را فاکتور گرفت، اهميت تابع تبديل $\Theta_m(s)/E_a(s)$ در اين است که موتور DC اساساً يک وسيله انتگرالگير بين اين دو متغير است. انتظار چنين امري نيز مي رفت زيرا چنانچه e_a يک ورودي ثابت باشد، جابه جايي خروجي موتور مانند خروجي يک انتگرالگير عمل مي کند يعني به صورت خطي نسبت به زمان افزايش مي يابد.



شکل ۳-٤نمودار بلوکي يک سيستم موتور DC

گرچه موتور DC به خودي خود اساساً يک سيستم حلقه باز است، اما نمودار حالت شکل ۳-۳ و نمودار بلوکي شکل ۳-۴ نشان مي دهند که موتور يک حلقه پسخوردي «دروني» دارد که ناشي از ضد emf است. از نظر فيزيکي ضد emf پسخورد سيگنالي را نشان مي دهد که با منفي سرعت موتور متناسب است. چنانکه از معادله (۳-۲۴) ديده مي شود، ثابت ضد emf K_b جمله اي افزون بر مقاومت R_a و ضريب اصطکاک ويسکوز B_m را نشان مي دهد. بنابراين اثر ضد emf معادل با يک «اصطکاک الکتريکي» است که ميل به تصحيح پايداري موتور و نهايتاً پايداري سيستم را دارد.

رابطهٔ بین K_T و K_b. هرچند در یک موتور مفروض ثابت گشتاور نیروي K_T و ثابت ضد emf K_b از نظر کار کردي دو پارامتر جدا از هم اند، اما مقدار آنها باهم رابطه دارند. براي رسیدن به این رابطه توان مکانیکي ایجاد شده در آرمیچر را مي نویسیم:

$$P = e_b(t) i_a(t) \tag{۲۵-۳}$$

توان مکانیکي به صورت زیر نیز بیان مي شود.

$$P = T_m(t) \omega_m(t) \tag{۲۶-۳}$$

در این رابطه $T_m(t)$ برحسب N-m و $\omega_m(t)$ برحسب rad/s است. حال معادله هاي (۳-۲۰) و (۳-۲۱) را در معادله (۳-۲۵) قرار مي دهیم و نتیجه مي گیریم:

$$P = T_m(t)\omega_m(t) = K_b \omega_m(t) \frac{T_m(t)}{K_T} \tag{۲۷-۳}$$

که از آن داریم:

$$K_a(V/rad/s) = K_i(N-m/A) \tag{۲۸-۳}$$

بنابراین مشاهده مي کنیم که در دستگاه SI اگر K_b برحسب V/rad/s و K_i برحسب N-m/A باشد مقدار K_b و K_i یکي هستند.

حال با توجه به لاپلاس معادلات ماتریس (۳-۲۳) خواهیم داشت:

$$LSi(s) = -Ri(s) - K_b \omega(s) + V(s)$$

$$JS\omega(s) = K_T i(s) - B_m \omega(s) - Z_L(s) \tag{۲۹-۳}$$

$$S\theta(s) = \omega(s)$$

آنگاه خواهیم داشت:

$$i(s) = \frac{-K_b \omega(s) + V(s)}{SL + R}$$

$$\omega(s) = \frac{K_T i(s) - Z_L(s)}{SJ + f} \qquad (۳۰-۳)$$

$$\theta(s) = \frac{1}{S} \omega(s)$$

۳-۱۰- كاربردهاي موتور [4],[5]

موتورهاي DC داراي مشخصه گشتاور – سرعت خيلي خوب و داراي حوزه وسيع كنترل سرعت مي باشند. عليرغم تلاشهايي كه براي حصول كنترل سرعت در حوزه وسيع با موتورهاي AC انجام شده و ليكن هنوز فراگيري كاربرد و قابليت انعطاف موتورهاي DC نمي تواند با موتورهاي AC رقابت كند. بدين منظور نيازمندي به موتورهاي DC حتي در آينده كاهش نمي يابد. بررسي اجمالي از كاربرد موتور DC در زير آمده است.

۳-۱۰-۱- موتورهاي شنت:

۱) به ازاء جريان ميدان تحريک معين در موتور شنت، كاهش سرعت از بي باري تا بار كامل همراه كمتر از ۶ تا ۸% است. از اين رو موتور شنت را موتور سرعت ثابت مي نامند. بنابراين براي محرک هاي با سرعت ثابت در صنعت از موتور هاي شنت DC مي توان استفاده كرد. اما اين موتورها نمي توانند با موتورهاي القايي قفس سنجابي سرعت ثابت رقابت كنند زيرا موتور قفس سنجابي ارزانتر و قويتر بوده و نياز به نگهداري و تعميرات كمتري دارد.

۲) هر گاه محرک سرعت ثابت در سرعتهاي پائين مورد نياز باشد، معمولاً مقايسه بين موتورهاي سنكرون و موتورهاي DC شنت صورت مي گيرد. زيرا ساختمان موتور القائي چند فاز با

عملکرد خوب با تعداد زياد قطب مشکل است. براي مورد استعمال با سرعت قابل تنظيم در سرعتهاي پائين انتخاب موتور شنت DC ترجيح دارد.

۳) هنگامي که بار به رنج وسيعي از کنترل سرعت (پائين و بالاي سرعت پايه) احتياج داشته باشد موتور DC بکار رود مانند ماشين تراش و غيره.

۴) در يک موتور شنت اگر سيم پيچ ميدان از آرميچر جدا و متصل به يک منبع ولتاژ خارجي شود تبديل به يک موتور تحريک جداگانه مي شود. اين موتور کنترل مستقل آرميچر و ميدان تحريک را عرضه مي دارد. از آنجائيکه موتورهاي DC تحريک جداگانه به راحتي با رنج وسيع کنترل سرعت گشتاور تطبيق پيدا مي کنند در کاربرد هاي قدرت بالا اين موتورها بطور وسيع در غلطکهاي نورد آلومينيوم و فولاد و سيستم هاي کنترل سرعت وارد – لئونارد مورد استفاده قرار مي گيرند. در کاربردهاي قدرت کم، موتور DC تحريک جداگانه بطور وسيع به عنوان موتور کنترل مورد استفاده قرار مي گيرد.

موتور کنترل موتور کم قدرتي (کمتر از چند صد وات) با تحريک ميدان ثابت و گشتاور T_e متناسب با جريان آرميچر I_a مي باشد.

۳-۱۰-۲- موتورهاي سري:

خصوصيات بر جسته موتورهاي سري اين است که به مجرد اينکه افزايش گشتاور بار لازم باشد سرعت بطور اتوماتيک کاهش مي يابد. کاهش سرعت همراه با افزايش گشتاور بار يا عکس آن فقط تأثير کمي روي ميزان قدرت الکتريکي گرفته شده توسط موتور سري دارد.

۱) از آنجائیکه موتور سري در راه اندازي بارهاي سنگين از خود استقامت خوبي نشان مي دهد و از طرفي مي تواند گشتاورهاي راه اندازي بالايي توليد كند لذا بهترين محرک براي حركت بالابرها، قطارها، خاک بردارها، جرثقيل ها و غيره است. موتورهاي القايي روتور سيم پيچي شده با موتور هاي سري رقابت مي كند اما انتخابی از نظر اقتصادي نيز مطرح است. در هر حال براي استفاده در حمل و نقل فقط موتور سري انتخابي مي شود. بنابراين موتورهاي سري استفاده زيادي در همه انواع وسائط نقليه الكتريكي از قبيل قطارهاي برقي، تراموااهاي شهري، وسائل تغذيه شده از باطري قابل حمل و نقل، موتورهاي راه انداز خود رو و غيره دارند.

۲) موتورهاي سري براي به حركت در آوردن بارهاي با اتصال دائم مانند بادبزن مورد استفاده قرار مي گيرند زيرا گشتاور بار آنها متناسب با مجذور سرعت تغيير مي كند.

۳) براي اجتناب از آلودگي هوا در شهرهاي بزرگ اكنون اتومبيلهائي كه با باتري كار مي كنند در مقياس وسيع معمول شده است. براي چنين اتومبيلهائي موتورهاي سري كه با برشگر كنترل مي شوند بسيار مناسب است.

۳-۱۰-۳- موتورهای کمپوند:

۱) موتور کمپوند با میدان سری قوی مشخصه ای نزدیک به موتور سری دارد. بنابراین این نوع موتورهای کمپوند برای بارهای با گشتاورهای راه اندازی بالا که احتمالا به صفر کاهش می یابند مورد استفاده قرار می گیرند. در موتور سری معمولی، سرعت بی باری به طور خطرناکی بالا می رود اما در موتور کمپوند سرعت بی باری توسط میدان شنت ضعیف محدود می گردد.

۲) موتور کمپوند با میدان سری قوی مشخصه ای نزدیک به موتور سری دارد. میدان سری ضعیف باعث می شود مشخصه گشتاور - سرعت نسبت به موتور شنت معمولی افت زیادی داشته باشد. چنین موتورهای کمپوند با مشخصه پر شیب هنگامی استفاده می شوند که بار با کار نوبه ای در محدوده وسیعی نوسان داشته باشد.

چرخهای طیار به محور چنین موتورهای کمپوند متصل می شوند که نقش متعادل کننده بار را دارند. در زمان بار کم، چرخهای طیار انرژی جنبشی را ذخیره می کنند اما در بار پیک، سرعت موتور کاهش می یابد تا چرخ طیار مقداری از انرژی ذخیره شده را به بار تحویل دهد. این امر باعث کم شدن انرژی دریافتی از منبع تغذیه می شود. محرکهایي که در آنها از چنین موتورهای کمپوندی استفاده می شود عبارتند از غلطکهای نورد، منگنه پرسی، ماشین طراحی یا ماشین فرز و غیره.

۳) هنگامی که ولتاژ تغذیه دو سر ترمینالهای موتور تغییرات محسوسی داشته باشد مانند موتورهای حمل و نقل، در آنصورت استفاده از موتورهای کمپوند ترجیح داده می شود. میدان سری به کاهش تغییرات جریان آرمیچر کشیده شده از منبع کمک می کند که این به علت اندوکتانس سیم پیچ تحریک و نیز اثر نیروی محرکه مغناطیسی میدان سری روس شار اصلی و بنابراین بر روی نیروی ضد محرکه الکتریکی موتور می باشد.

۴) موتور کمپوند نقصانی هرگز به کار نمی رود. در باطن موتور میدانهای شنت وسری با یکدیگر مخالفت می کنند و ممکن است که در بعضی حالات کاری مقدار شار مغناطیسی در فاصله هوايي صفر شود. هنگامیکه این اتفاق می افتد سرعت موتور به طور خطرناکی بالا رفته و جریان آرمیچر به مقدار زیادی افزایش می یابد. سرعت بال و افزایش جریان آرمیچر باعث

تحمیل شرایط سخت راه اندازی به موتور می شود و در نتیجـه ممکـن اسـت موتـور کمپونـد نقصانی آسیب ببیند.

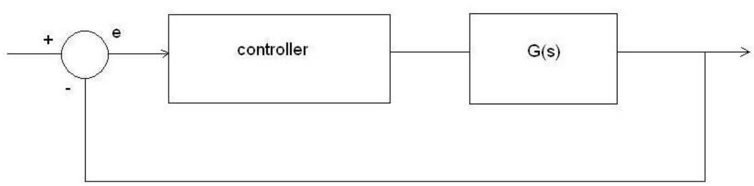
۳-۱۰-۴- موتور DC با آهنربای دائمی :

این نوع موتورها در اتومبیل ها به عنوان موتور راه انداز (استارت)، برف پاک کن، در دمنده ها و تهویه مطبوع، برای بالا و پایین بردن پنجره ها و مسواکهای برقی و غیره به کار می روند. بدون اغراق میلیون ها موتور جریان مستقیم با آهنربای دائمی در هر سال ساخته می شود تا تقاضای میلیونها اتومبیل را برآورده سازد.

فصل چهارم_ کنترل کننده هاي **PID**

۴-۱- تعاریف و مفاهیم

سیستم حلقه بسته زیر را در نظر بگیرید:



شکل ٤-١سیستم حلقه بسته ساده

کنترل کننده تناسبی- مشتق گیر- انتگرال گیـر (PID) ، بـه عنـوان سـاختار اصـلی کنتـرل کننـده استفاده می شود. رابطه ورودی-خروجی یک کنترل کننده PID به صورت زیر است:

$$U(t) = K\left\{c(t) + \frac{1}{T_i}\int_0^t e(c)dc + T_d\,\frac{de(t)}{dt}\right\} \qquad (۴-۱)$$

که در آن u سیگنال کنترل و e سیگنال خطا مي باشد. کنترل کننـده از سـه قـسمت تـشکیل شـده است:

جمله تناسبی P :که سیگنال کنترل متناسب با میزان خطا تولید می کند.

جمله مشتق گیرD : که سیگنال کنترل متناسب با نرخ تغییرات خطا تولید می کند.

جمله انتگرال گیر I : که سیگنال کنترل متناسب با جمع تابع خطا از زمان صفر تا به حال بـه صـورت انتگرال این تابع تغییر می کند.

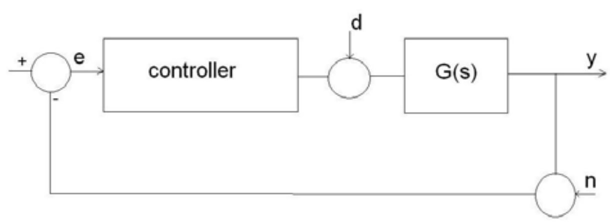
پارامتر های کنترل کننده PID عبارتند از : K,T_i,T_d که K بهره تناسبی، T_i ثابت زمانی انتگرال گیر و T_d ثابت زمانی مشتق گیر نامیده می شوند. حال به ترتیب به بررسی نقش هر یک از این اجـزا پرداخته می شود. [7]

۴-۱-۱- جمله تناسبی کنترل کننده **PID**

اگر تنها جمله تناسبی کنترل کننده در نظر گرفته شود در این صورت سیگنال کنترل برابر خواهد بود با:

$$u(t)=K.e(t) \qquad (۴-۲)$$

در حالت واقعی اغتشاشات محیط و نویز را به سیستم افزوده می شوند:



شکل ٤-٢ سیستم حلقه بسته داراي اغتشاش و نویز

که در آن d اغتشاش و n نویز محیط می باشد. سیگنال خروجی از رابطه زیر بدست می آید:

$$y=\frac{G(s)k}{1+G(s)k}y_d+\frac{G(s)}{1+kG(s)}d-\frac{G(s)k}{1+G(s)k}n \qquad (۴-۳)$$

مقدار بهره حلقه باز برابر است با KG(s) که (Loop Gain) نامیـده شـده و بـا L(s) نـشان داده می شود. هدف این است که y در نهایت مشابه y_d رفتار کند.

$$\frac{y}{y_d}=\frac{G(s)k}{1+G(s)k}=\frac{L(s)}{1+L(s)} \quad (۴-۴)$$

بـا افـزایش بهـره کنتـرل کننـده تناسـبی K مقـدار بهـره حلقـه L(s) افـزایش مـی یابـد و نـسبت $\frac{y}{y_d}=\frac{L(s)}{1+L(s)}$ به عدد واحد نزدیک می گردد(از مزایای بهره تناسبی).

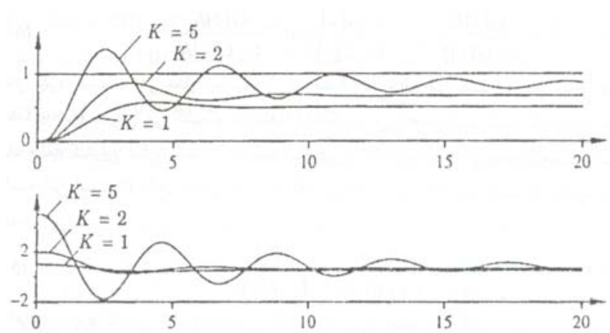
از طرف دیگر باید تاثیر اغتشاش d بر روی خروجی تضعیف شود:

$$\frac{y}{d}=\frac{G(s)}{1+G(s)k}=\frac{G(s)}{1+L(s)} \quad (۵-۴)$$

مجددا با افزایش بهره حلقه K می توان مخرج کسر L(s) را بزرگ نمـوده و بـدین ترتیـب تـابع بـه سمت صفر میل می کند (تضعیف اغتشاش). امـا بهـره حلقـه کنترلـی را نمـی تـوان بـدون توجـه بـه پارامترهای دیگر افزایش داد. طبق رابطه زیر با افزایش بی رویه k و بهره حلقـه بـاز L(s) ایـن تـابع تبدیل به عدد واحد نزدیک می شود. یعنی اثر نویز به صورت ۱۰۰% در خروجی مـشاهده مـی شـود و توانايي تضعيف تاثیر نویز بر روی خروجی را ندارد. از طرف دیگر با افزایش k پایـداری سیـستم حلقـه بسته کاهش می یابد. پس به این دو علت باید مقدار بهینه ای برای k انتخاب شود.

$$\frac{y}{n}=\frac{L(s)}{1+L(s)} \quad (۶-۴)$$

در شکل ۴-۳ نشان داده شده که با افزایش k خطای ماندگار سیستم کاهش یافته اما پاسخ ناپایدارتر و نوسانی تر می شود. [7]



شکل ٤-٣ تاثیر K بر پاسخ سیستم وخطای ماندگار(شکل بالا) و پایداری(شکل پایین)

۴-۱-۲- جمله انتگرال گیر

تاثیر اصلی جمله انتگرال گیر در کنترل فرآیندهای صنعتی، کاهش خطای حالت ماندگار[1] می باشد. پاسخ پله واحد سیستمی را در نظر بگیرید که خروجی سیستم آن را به طور کامل تعقیب نکرده و پس از گذشت زمان، میزان خطای اندکی باقی می ماند. کنترل کننده تناسبی به واسطه کوچک بودن خطا تنها تاثیر کمی در خروجی خواهد داشت. در حالی که انتگرال گیر خطا، مقدار این خطای کوچک را به مرور تشدید می کند. بدین ترتیب با افزودن جمله انتگرال گیر به کنترل کننده تناسبی، کنترل کننده PI خواهد بود که در آن خطای حالت ماندگار سیستم به شدت کاهش می یابد.

در صورت عدم وجود کنترل کننده انتگرال گیر $(T_i = \infty)$، خطای ماندگار حدود ۵۰% است، اما با افزایش بهره کنترل کننده انتگرال گیر (کاهش T_i) این خطا به تدریج کاهش یافته و یا کاملا از بین می رود. البته این کاهش خطا با هزینه ای همراه است و آن، کاهش سرعت پاسخ سیستم می باشد. افزایش جمله انتگرال گیر در فرآیند های صنعتی باعث کند شدن پاسخ سیستم می گردد. [8]

۴-۱-۳- جمله مشتق گیر

اضافه کردن جمله مشتق گیر به کنترل کننده به دو منظور صورت می گیرد: افزایش حاشیه پایداری در سیستم حلقه بسته و افزایش سرعت پاسخ یا پهنای باند سیستم.

در سیستم ها هر تغییری در سیگنال کنترل، با قدری تاخیر در خروجی ظاهر می شود. لذا سیگنال کنترل معمولا با تاخیر خطاها را جبران می سازد. با افزایش یک جمله مشتق گیر می توان تخمینی از

1 steady state error

میزان خطا با توجه به شیب خطا $\frac{de(t)}{dt}$ در زمان آینده بدست آورد و از این اطلاعات جهت رفع تاخیر در پاسخ سیستم استفاده نمود.

افزایش بهره مشتق گیر سرعت پاسخ را افزایش می دهد، ولی چون این افزایش باعث افزایش نوسانات و کاهش پایداری می شود، نباید T_d را خیلی افزایش داد. [8]

۴-۲- موارد استفاده کنترل کننده **PID**

اهداف استفاده از حلقه فیدبک و کنترل کننده در یک سیستم عبارتند از:

- پایداری داخلی
- دنباله روی از سیگنال ورودی[1]
- تضعیف اغتشاش
- کاهش نویز
- عدم حساسیت به تغییر سیستم

تقریبا در کلیه فرآیندهای صنعتی ساختار ساده کنترل کننده PID تمام نیازهای فوق را برآورده می سازد. حتی در برخی از موارد کنترل کننده PI نیز کافی است. در حالت کلی می توان بیان کرد که کنترل کننده PI برای سیستم هایی که با مدل درجه یک تقریب زده می شوند، کافی است. نیاز به جمله انتگرال گیر جهت کاهش خطای ماندگار سیستم حلقه بسته می باشد. اگر این مورد نیز مورد نیاز نباشد کنترل کننده تناسبی کافی خواهد بود.

[1] Signal tracking

همچنین برای سیستم های رتبه دو و یا سیستم هایی که دینامیک غالب آنها رتبه دو می باشد کنترل کننده PID تمام شرایط فوق را برآورده می کند. در این موارد استفاده از کنترل کننده های با ساختار پیچیده تر مورد نیاز نمی باشد. در این گونه سیستم ها توسط جمله مشتق گیر سرعت پاسخ افـزایش یافته و جمله انتگرال گیر خطای ماندگار را صفر می کند. در صورتی که کاهش خطـای مانـدگار مـورد نیاز نباشد، کنترل کننده PD نیز کافی است.

در برخی موارد کنترل کننده PID کافی نمی باشد. این موارد عبارتند از:

- سیستم با دینامیک رتبه بالاتر از دو
- سیستم های با تاخیر زیاد
- سیستمهاي دارای صفر ناپایدار یا سیستمهای غیر مینیمم فاز
- حذف فرکانسهای خاص در خروجی

۴-۳- طراحی کنترل کننده **PID**

۴-۳-۱- مقدمه

در این بخش روشهای مختلف طراحی کنترل کننده PID مورد بررسی قرار می گیرد.

در بحث طراحی اهداف اصلی عبارتند از:

- پایداری داخلی
- تنظیم یا ردیابی
- تضعیف اثر اغتشاش در خروجی
- کاهش تاثیر نویز در خروجی

- عدم حساسیت به تغییر پارامترهای مدل

تضعیف اثر اغتشاش و هدف ردیابی در یک امتداد می باشند و با وجود فیدبک می توان نـشان داد که این دو هدف با افزیش بهره حلقه $L(s)$ تامین می شـود. همچنـین مـی تـوان نـشان داد کـه خاصیت عدم حساسیت به تغییر پارامترها نیز در همـین امتـداد بـوده و لـذا بـا اسـتفاده از حلقـه فیدبک می توان این اهداف را به طور هوزمان برآورده کرد. مساله تنظیم یا ردیابی باید بـه صـورت دقیق تعریف گردد تا بتوان از این تعبیر ریاضی در طراحی کنترل کننده نیز استفاده نمود. سیستم شکل ۴-۲ را در نظر بگیرید.

فرض کنید سیستم در حالت ماندگار به تعادل رسیده باشد ولی در اثر اغتشاش خارجی یک تغییر پلـه در سیستم اعمال گردد. در این حالت پاسخ خطای سیستم در حالت گذرا به صورت زیر است:

$$e(t) = y_d(t) - y(t) \qquad (۷-۴)$$

برای تعریف دقیق میزان خطا و حذف آن توسط فیـدبک کمیتهـای زیـر بـر روی تـابع خطـا تعریـف می شوند:

- میزان فراجهش خطا: $e_{\max}$
- زمان رسیدن به فراجهش: $t_{\max}$
- زمان نشست خطا: t_s
- نسبت تضعیف دامنه نوسانات: d

- انتگرال قدر مطلق خطا:

$$(\text{Integral Absolute Error}) : IAE = \int_0^\infty |e(t)| dt \; : \; \|e\|_1 \qquad (۸-۴)$$

- انتگرال مجذور خطا:

$$(\text{Integral Square Error}) : ISE = \int_0^\infty |e(t)^2| dt : \|e\|_2 \qquad (۹-۴)$$

- انتگرال قدر مطلق خطا در زمان:

$$(\text{Integral Time Absolute Error}) : ITAE = \int_0^\infty t|e(t)| dt \qquad (۱۰-۴)$$

چهار پارامتر اول شكل خطا را تعريف مي كنند، اما سه پارامتر ديگر نوع خطا را تعريف مي كنند. معيار طراحي كاهش خطا مي باشد. اما غالبا بسته به اينكه چه كميتي از خطا كاهش داده خواهد شد، يك روش طراحي انتخاب خواهد شد. ابتدا به تعريف پارامترهاي ۱ تا ۴ پرداخته می شود: [6]

- میزان فراجهش: تفاوت بین ماکزیمم جهش پاسخ نسبت به مقدار ماندگار
- نسبت ميرايي دامنه نوسانات d : عبارتست از نسبت دامنه پیک دوم به پیک اصلي
- زمان نشست t_s : زماني است كه پاسخ پله سيستم به نزديكي 5% (يا 2%) مقدار ماندگار خود رسيده باشد و در حاشيه اين فاصله باقي بماند.
- زمان صعود t_s [1] : زماني است كه پاسخ پله از y_o ۰,۱ به ميزان y_o ۰,۹ می رسد.

شكل نوساني پاسخ پله زماني رخ مي دهد كه سيستم مدار بسته درجه دو باشد. اين حالت زماني اتفاق مي افتد كه يك فرآيند رتبه يك را با كنترل كننده PI يا PD به صورت حلقه بسته كنترل نماييم. در اين حالت تابع انتقال حلقه بسته به صورت زير خواهد بود:

[1] rise time

$$T(s)=\frac{\omega_o{}^2}{s^2+2\xi\omega_o s+\omega_o{}^2} \qquad (۱۱-۴)$$

با استفاده از حل معادلات، پاسخ پله چنین سیستمي داراي معادلات زیر است:

$$y(t)=1-\frac{1}{\sqrt{1-\xi^2}}e^{-\xi\omega_o t}\sin(\omega_o t\sqrt{1-\xi^2}+\varphi) \qquad (۱۲-۴)$$

$$\varphi=tg^{-1}\left(\frac{\sqrt{1-\xi^2}}{\xi}\right) \qquad (۱۳-۴)$$

در این حالت مقادیر ω_o و ζ تابعي از ضرایب کنترل کننده PI یا PD خواهند بود وکمیتهاي مربوط به خطا از روابط زیر محاسبه مي شوند:

$$\%O=e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}}\times 100 \qquad (۱۴-۴)$$

$$d=e^{\frac{-2\xi\pi}{\sqrt{1-\xi^2}}} \qquad (۱۵-۴)$$

$$t_{\max}=\frac{\pi}{\omega_o\sqrt{1-\xi^2}} \qquad (۴-۱۶)$$

$$t_s=\frac{\ln(0.05\sqrt{1-\xi^2}}{\xi\omega_o} \qquad (۴-۱۷)$$

$$t_r=\frac{1}{\omega_o}e^{\frac{\varphi}{tg\varphi}} \qquad (۴-۱۸)$$

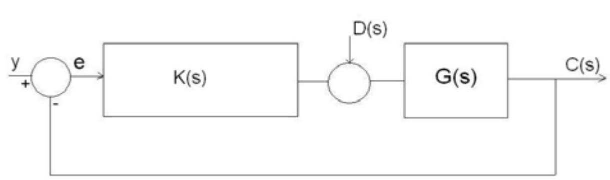
حال اگر از کمیتهاي فوق دو کمیت به مقادیر دلخواه نزدیکتر شوند روابطي براي ω_o و ζ به دست مي آید و چون این کمیتها توابعي از کنترل کننده هاي PI و PD خواهند بود، ضرایب کنترل کننده تعیین خواهند شد. [7]

براي فرآيندهاي صنعتي معمولا ω_o پهناي باند سيستم مدار بسته را مشخص مي سازد و با توجه به نوع و سرعت سيستم حلقه باز انتخاب مي شود. ثابت زماني در يك سيستم حرارتي چند دقيقه و در يك سيستم شيميايي ممكن است چند ساعت باشد. در يك سيستم مكانيكي ثابت زماني به كسري از ثانيه مي رسد و براي سيستم هاي الكتريكي به چند ميلي ثانيه محدود مي شود. اما ميزان ميرايي يا نسبت دامنه ميرايي براي بسياري از فرآيندهاي صنعتي $d=\frac{1}{4}$ انتخاب مي شود. در اين حالت $\zeta=0.2155$ بوده و ميزان فراجهش ۵۰% خواهد بود. درصورتي كه ميزان فراجهش كمتري نياز باشد معمولا $\zeta=0.707$ انتخاب مي شود كه در اين حالت فراجهش حدود ۴% مي شود و ميزان نوسانات بسيار كمخواهد بود، اما سرعت پاسخ كمي كندتر از حالت $d=\frac{1}{4}$ می باشد كه به quarter cycle مشهور است.

۴-۳-۲- ساير روشهاي طراحي [8]

علاوه بر روشهاي ارائه شده، روشهاي تنظيم كنترل كننده بسياري وجود دارند كه در اينجا به روشـهاي انتگرال خطا اشاره مي شود. روشهاي مورد مطالعه در اين بخش ITAE و ASE و IAE مي باشند.

سيستم شكل ۴-۴ را در نظر بگيريد:



شكل ٤-٤ سيستم حلقه بسته ساده

که در آن، سیستم با مدل ساده $\frac{K}{1+\tau.s}$ تخمین زده شده است. همچنین کنتـرل کننـده نیـز PI در نظر گرفته می شود. معیار طراحي نیز ISE فرض مي شود. لذا تابع هزینه برابر است با:

$$J=\int_0^\infty e(t)^2\,dt \tag{۴-۱۹}$$

$$e(t)=-\frac{T_i}{k}\frac{kK/\tau T_i}{s^2+(\frac{1+kK}{\tau})s+\frac{kK}{\tau T_i}} \tag{۴-۲۰}$$

همانگونه که مشخص است تابع خطا e(t) رابطه مـستقیم بـا ضـرایب کنتـرل کننـده یعنـی K و Ti خواهد داشت. ایده اصلي طراحي بدین صورت است که پارامترهـاي K و Ti بایـد بـه گونـه اي یافتـه شوند که تابع هزینه مقدار حداقل خود را بیابد. براي بدست آوردن مقدار کمینـه تـابع J، مـي تـوان از روش مشتق گیري نسبت به پارامتر مجهول استفاده نمود:

$$\frac{\partial J}{\partial K}=0 \tag{۴-۲۱}$$

$$\frac{\partial J}{\partial T_i}=0 \tag{۴-۲۲}$$

نتایج حل این معادلات در جداول ۴-۱ تا ۴-۳ آورده شده اند.

فرض می شود:

$$G(s)=\frac{Ke^{-\tau s}}{1+Ts} \tag{۴-۲۳}$$

همچنین پارامتر θ را به صورت زیرتعریف مي شود:

$$\theta=\frac{\tau}{T} \tag{۴-۲۴}$$

- کنترل کننده تناسبي :

روش	a	b
IAE	۰,۹	۰,۹۸
ISE	۱,۴	۰,۹۲
ITAE	۰,۵	۱,۰۸

جدول ۴-۱پارامترهاي كنترل كننده P با استفاده از روشهاي مختلف براي مدل سه جزيي

كه در آن :

$$K(s)=k \tag{۴-۲۵}$$

$$k=\frac{1}{K}a(\theta)^{-b} \tag{۴-۲۶}$$

- کنترل کننده **PI** :

روش	a	b	c	d
IAE	۰,۹۸	۰,۹۸	۱,۶۵	۰,۷۱
ISE	۱,۳	۰,۹۶	۲,۰۳	۰,۷۴
ITAE	۰,۸۶	۰,۹۸	۱,۴۸	۰,۶۸

جدول ۴-۲پارامترهاي كنترل كننده PI با استفاده از روشهاي مختلف براي مدل سه جزيي

كه در آن :

$$k=\frac{a}{K}\theta^{-b} \tag{۴-۲۷}$$

$$K(s)=k(1+\frac{1}{T_i s}) \tag{۴-۲۸}$$

$$T_i = T.c.\theta^d$$ (۴-۲۹)

کنترل کننده **PID** :

روش	a	b	c	D	e	f
IAE	۱,۴	۰,۹۲	۱,۱۴	۰,۷۵	۰,۴۸	۱,۱۴
ISE	۱,۵	۰,۹۵	۰,۹۲	۰,۷۷	۰,۵۶	۱
ITAE	۱,۳۶	۰,۹۵	۱,۱۸	۰,۷۴	۰,۳۸	۱

جدول ۴-۳پارامترهاي كنترل كننده PID با استفاده از روشهاي مختلف براي مدل سه جزيي

که در آن :

$$k = \frac{1}{K} a(\theta)^{-b}$$ (۴-۳۰)

$$K(s) = k(1 + \frac{1}{T_i s})$$ (۴-۳۱)

$$T_d = T.e.\theta^f$$ (۴-۳۲)

$$T_i = T.c.\theta^d$$ (۴-۳۳)

فصل پنجم- بهينه سازي ضرايب موتور **DC** و كنترل كننده **PID** به کمک روش الگوريتم ژنتيكي

۵-۱- مقدمه

الگوريتم ژنتيكي روشي است براي بهينه سازي مسائل متنوع و مختلـف. بـا اينكـه سـاختار كلـي ايـن الگوريتم براي همه مسائل يكي است، اما خود نيازمند تنظيم مي باشد. تنظـيم صـحيح ايـن الگـوريتم براي هر مساله خود بخشي از حل آن مساله مي باشد و چـه بـسا يـك تنظـيم نادرسـت موجـب شـود همگرايي به جواب نهايي از بين برود يا به جوابهاي نه چندان مناسبی بدست آيند. در اين بخـش ابتـدا به شناسایی پارامترهای دینامیکی یک موتور DC پرداخته می شود و سپس پارامترهای الکتریکـی آن شناسایی خواهد شد. در ادامه نیز کنترل کننده PID مورد نظر برای این موتور DC طراحی می شود.

۵-۲- شناسايي پارامترهای حالت دینامیکی موتور **DC**

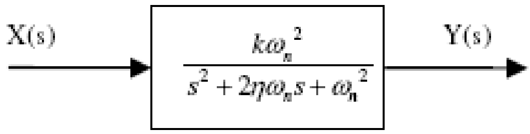
همانطور که در فصل سوم ملاحظه شد معادلات اصلی موتور DC عبارتند از :

$$LSi(s) = -Ri(s) - K_b\omega(s) + V(s)$$

$$JS\omega(s) = K_T i(s) - f\omega(s) - Z_L(s) \qquad (۵-۱)$$

$$S\theta(s) = \omega(s)$$

همچنین می توان حالت دینامیکی یک موتور DC را به صورت شکل ۵-۱ مدل کرد:

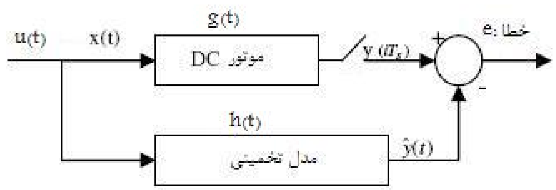


شکل ۵-۱شماتیک کلی حالت دینامیکی موتور DC

با توجه به مدل فوق ملاحظه می شود که تابع تبدیل یک موتور DC را می توان به صورت زیر بیان کرد:

$$H(s)=\frac{\Omega(s)}{E(s)}=\frac{k\omega_n^2}{S^2+2\eta\omega_n S+\omega_n^2} \qquad (۵-۲)$$

درشناسايي پارامترهای حالت دینامیکی موتور DC بدین ترتیب عمل شده که ابتدا یک سیستم مانند شکل (۵-۲) در نظر گرفته می شود:



شکل ۵-۲بلوک دیاگرام حالت دینامیکی موتور DC

سپس یک ورودی (برای مثال تابع پله) به موتور واقعی [G(s)] و مدل تخمینی در نظر گرفته شده از معادله (۵-۲) یعنی [H(s)] اعمال می شود. پاسخ پله این دو سیستم از هم کم شده و تابع خطا را بوجود می آورند.

همانطور که بیان شد تابع معیار الگوریتم ژنتیکی، خطا در نظر گرفته شده است که به صورت مجموع مربع خطاها در الگوریتم ژنتیکی مورد استفاده قرار می گیرد:

$$E^2(K) \triangleq \sum_{n=1}^{N} \left(y(nT) - \hat{y}(nT) \right) \qquad (۳\text{-}۵)$$

سپس با استفاده از این مقادیر، که نسل اول تولید شده هستند، تابع معیار (تابع هزینه) محاسبه می شود. الگوریتم ژنتیکی در نسلهای بعدی تابع معیار یا همان خطا را کاهش داده تا جايي که مقدار آن از حدی که برنامه نویس تعیین کرده کمتر شده و الگوریتم متوقف شود. [11],[17]

۵-۳- شناسايي پارامترهای حالت الکتریکی موتور DC

در ابتدا روابط یک موتور DC به صورت معادلات (۵-۴) در نظر گرفته می شوند:

$$\begin{aligned} L\frac{di}{dt} &= -R_i(t) - K_b\omega(t) + v(t) \\ J\frac{d\omega}{dt} &= -f\omega(t) + K_T i(t) - T_L(t) \\ \frac{d\theta}{dt} &= \omega(t) \end{aligned} \qquad (۴\text{-}۵)$$

براي شناسايي پارامترهای دینامیکی یک موتور DC باید پارامترهاي موتور یعني $K_b = K_T, R, L, J$ و در نهایت f را پیدا کرد. در عمل روش کار بدین صورت است که یک ولتاژ توسط منبع تغذیه به موتور اعمال شده و از طریق اندازه گیري مقادیر ω(t), i(t) , V(t) پارامتر هاي مورد نظر محاسبه مي شوند. براي این امر، دو معادله اول به صورت زیر نوشته می شوند:

$$\begin{bmatrix} di/dt & i(t) & \omega(t) & 0 & 0 \\ 0 & 0 & -i(t) & d\omega/dt & \omega(t) \end{bmatrix} \begin{bmatrix} L \\ R \\ K_T \\ J \\ f \end{bmatrix} = \begin{bmatrix} v(t) \\ 0 \end{bmatrix} \qquad (۵-۵)$$

در این معادله جبري خطي، ماتریس متشکل از f, J, K_T, R, L پارامترهاي مجهول هستند، که از طریق مقادیر اندازه گیري شده و محاسبه شده $V, \frac{di}{dt}, \frac{dw}{dt}, \omega(\mathrm{t}), \theta(t)$ بدست مي آیند.

اگر معادلات موتور دقیقاً مانند رابطه ۵-۴ باشد و مقادیر $V(t), i(t), \theta(t)$ را بتوان دقیقا به دست آورد و همچنین اگر مقادیر $\frac{di}{dt}, \frac{dw}{dt}, \frac{d\theta}{dt}$ به طور دقیق محاسبه شوند، آنگاه معادله ماتریسی را مي توان براي تمام زمانهاي t به کار برد.

اگر ω(nT) نشان دهنده سرعت در زمان nT و i(nT) نشان دهنده جریان در زمانnT و $\frac{di(nT)}{dt}$ نشان دهنده مشتق جریان در زمان nT باشد، خواهیم داشت:

$$W(nT) \triangleq \begin{bmatrix} \frac{di}{dt}(nT) & i(nT) & \omega(nT) & 0 & 0 \\ 0 & 0 & -i(nT) & \frac{d\omega}{dt}(nT) & \omega(nT) \end{bmatrix} \in R^{2\times5}$$

$$y(nT) \triangleq \begin{bmatrix} v(nT) \\ 0 \end{bmatrix} \in R^2 \qquad (۶-۵)$$

$$K \triangleq \begin{bmatrix} L \\ R \\ K_T \\ J \\ f \end{bmatrix} \in R^5$$

در نتیجه مي توان معادلات بالا را به صورت زیر نوشت:

$$W(nT)K = y(nT) \qquad (۷-۵)$$

W در اينجا ماتريس بازگشتي (Regressor matrix) نام دارد. هدف از معادله بالا، يافتن ماتريس ثابت k براي تمام مقادير n مي باشد. براي اين منظور دو طرف معادله بالا را در $W^T(nT)$ ضرب کرده و خواهيم داشت:

$$W^T(nT)W(nT)K = W^T(nT)y(nT) \tag{۸-۵}$$

$$W^T(nT)W(nT) = \begin{bmatrix} \frac{di(nT)}{dt} & 0 \\ i(nT) & 0 \\ \omega(nT) & -i(nT) \\ 0 & \frac{d\omega(nT)}{dt} \\ 0 & \omega(nT) \end{bmatrix} \begin{bmatrix} \frac{di(nT)}{dt} & i(nT) & \omega(nT) & 0 & 0 \\ 0 & 0 & -i(nT) & \frac{d\omega(nT)}{dt} & \omega(nT) \end{bmatrix}$$

$$= \begin{bmatrix} (di/dt)^2 & idi/dt & \omega di/dt & 0 & 0 \\ idi/dt & i^2 & \omega i & 0 & 0 \\ \omega di/dt & \omega i & \omega^2 + i^2 & -id\omega/dt & -\omega i \\ 0 & 0 & -id\omega/dt & (d\omega/dt)^2 & \omega d\omega/dt \\ 0 & 0 & -\omega i & \omega d\omega/dt & \omega^2 \end{bmatrix}_{|\, t = nT}$$

(۹-۵)

$$W^T(nT)y(nT) = \begin{bmatrix} di(nT)dt & 0 \\ i(nT) & 0 \\ \omega(nT) & -i(nT) \\ 0 & d\omega(nT)/dt \\ 0 & \omega(nT) \end{bmatrix} \begin{bmatrix} \nu(nT) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \nu(nT)di(nT)/dt \\ \nu(nT)i(nT) \\ \nu(nT)\omega(nT) \\ 0 \\ 0 \end{bmatrix}$$

(۱۰-۵)

در نظر داشته باشید که ماتریس $W^T(nt)W(nt) \in \Re^{5\times5}$ یک ماتریس مربعي است. در صورتیکه بتوان دو طرف معادله بالا را در $(W^T(nt)W(nt))^{-1}$ ضرب کرد، حل این معادله آسان خواهد گشت. ولي ماتریس $W^T(nt)W(nt)$ معکوس پذیر نمي باشد. براي مثال فرض کنید که ستون اول این ماتریس را در i(n)– و ستون دوم آنرا در $\frac{di(nt)}{dt}$ ضرب کرده و با هم جمع کنیم. در نتیجه خواهیم داشت:

$$\begin{bmatrix} -i(nT) & di(nT)/dt & 0 & 0 & 0 \end{bmatrix} W^T(nT)W(nT) \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (۱۱\text{-}۵)$$

حال اگر این ماتریس معکوس پذیر باشد باید بتوان دو طرف معادله بالا را در معکوس آن ضرب کرد.

$$\begin{bmatrix} -i(nT) & di(nT)/dt & 0 & 0 & 0 \end{bmatrix} W^T(n)W(nT)(W^T(nT)W(nT))^{-1}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} (W^T(n)W(nT))^{-1}$$

آنگاه

$$\begin{bmatrix} -i(nT) & di(nT)/dt & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (۱۲\text{-}۵)$$

که این یک تناقض است. زیرا مقدار جریان نباید صفر باشد.

روش دیگري که باید بدان عمل کرد بدین صورت است. معادله اول را که براي تمامي مقادیر n برقرار بوده و k ماتریس ثابتي در آن است، براي N بار جمع مي زنیم:

$$\left(\sum_{n-1}^{N} W^T(nT)W(nT) \right) K = \sum_{n=1}^{N} W^T(nT)y(nT) \quad (۱۳\text{-}۵)$$

که در آن :

$$R_W \triangleq \sum_{n=1}^{N} W^T(nT)W(nT) \in R^{5\times 5} \qquad (۱۴-۵)$$

$$R_{wy} \triangleq \sum_{n=1}^{N} W^T(nT)y(nT) \in R^{5\times 1} \qquad (۱۵-۵)$$

که در نهایت می توان روابط (۵-۱۱) تا (۵-۱۳) را بدین صورت نوشت:

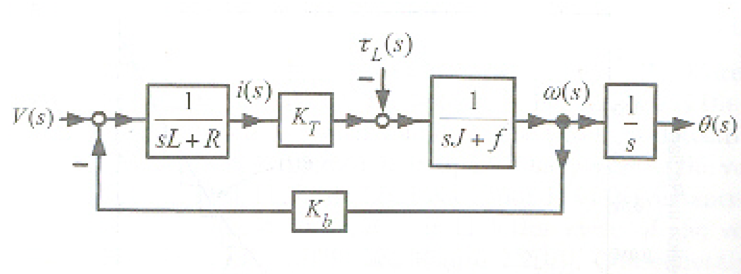
$$R_w K = R_{wy} \qquad (۱۶-۵)$$

اگر ماتریس R_w معکوس پذیر باشد، آنگاه با ضرب دو طرف معادله بالا در $R_w^{-1} = (\sum_{n=1}^{N} W^T(nt)w(nt))^{-1}$ خواهیم داشت:

$$K = R^{-1}{}_W R_{wy} \qquad (۱۷-۵)$$

راه حل این معادله این است که ماتریس R_w باید معکوس پذیر باشد. براي این منظور باید مقادیر مناسبي را به عنوان ورودي موتور انتخاب کرد و هر مقدار دلخواه جوابگو نمي باشد. براي مثال اگر v(t)=0 آنگاه i(t)=0 و در نتیجه ω(t)=0 و در نهایت براي همه مقادیرn، W(nT)=0بوده و $R_w = 0$ خواهد شد که معکوس پذیر نمي باشد. به عنوان یک مثال بدیهي دیگر در نظر بگیرید که $i(t) = i_0$ آنگاه ستون اول ماتریس $W^T(nT)W(nT)$ براي تمام مقادیر n برابر صفر شده و ستون اول R_w نیز براي تمام مقادیر n صفر مي شود و در نتیجه R_w معکوس پذیر نخواهد بود. اگر ورودي مناسب به شکلي انتخاب شود که R_w معکوس پذیر باشد، آنگاه مي توان گفت که سیستم به طور صحیح تحریک شده است.

حال به طراحی الگوریتم ژنتیکی پرداخته می شود. سیستم ذکر شده در فصل قبـل بـه صـورت شـکل ۵-۳ در نظر گرفته می شود:



۳-۵مدل موتور DC

در این سیستم از مقادیر $\frac{di}{dt}, \frac{d\omega}{dt}, \omega(t), i(t)$ به ازای پارامترهای واقعی موتور نمونه برداری مـی شـود. در ادامه با استفاده از مقادیر بدست آمـده و الگـوریتم نوشـته شـده، بـه شناسـايي پارامترهـای موتـور پرداخته می شود. [9],[3]

۵-۴- تنظیم الگوریتم ژنتیکی

در این بخش به تنظیم پارامترهای الگوریتم ژنتیکی برای این مساله پرداخته می شود. [19],[10]

۵-۴-۱- ژن و کروموزوم

هدف رسیدن به معیارهاي بهینه مورد نظر است. بنابراین با دید الگـوریتم ژنتیکـي پارامترهـاي بهینـه کنترلر را جستجو مي کنیم. بنابراین هر پارامتر را يك ژن در نظر مي گیریم.

۵-۲-a- حالت دینامیکی: در این بخش پارامترهای حالت دینامیکی موتور DC به عنـوان پارامترهـای الگوریتم ژنتیکی در نظر گرفته می شوند.(سه پارامتر)

۵-۲-b- حالت الکتریکی: در این بخش پارامترهای حالت الکتریکی موتور DC بـه عنـوان پارامترهـای الگوریتم ژنتیکی در نظر گرفته می شوند.(پنج پارامتر)

۵-۲-c-کنترل کنده PID: در این بخش پارامترهـای کنترلـر PID بـه عنـوان پارامترهـای الگـوریتم ژنتیکی در نظر گرفته می شوند.(سه پارامتر)

۵-۴-۲- ایجاد جمعیت اولیه

جمعيت اوليه را تعدادي كروموزوم با ژنهايي كه به صورت تصادفي توليد مي شوند تشكيل مي دهند. براي توليد يك ژن به صورت تصادفي كافي است يك عدد تصادفي توليد شود. در صورتي كه بازه تعيين شده اي مد نظر باشد مي توان مقدار اوليه را در بازه دلخواه تنظيم كرد. در اين حالت اگر فرض شود كروموزوم مورد نظر از ۵ ژن تشكيل شده است، می توان رشته زير را به عنوان كروموزومي كه به طور تصادفي از فضاي جستجو انتخاب شده است در نظر گرفت: (بازه در نظر گرفته شده بین ۰ تا ۱۰ باشد)

۱,۳۶۳۳, ۵,۴۲۵۶, ۰,۹۸۲۵, ۴,۱۸۴۹, ۹,۷۲۴۳

با در كنار هم قرار دادن تعدادي از اين كروموزوم هاي تصادفي، مي توان جمعيت اوليه را تشكيل داد.

حالت دینامیکی: در این حالت جمعیت از ۳ کروموزوم تشکیل شده و تعداد ژنهای هر کروموزوم ۵۰ انتخاب شده است. در نتيجه مي توان جمعيت را به صورت يك ماتريس 3×50 نشان داد كه درايه هاي آنرا اعدادي با محدوديت هاي ذكر شده تشكيل داده اند.

۵-۳-b- حالت الکتریکی: در این حالـت جمعیـت از ۵ کرومـوزوم تـشکیل شـده و تعـداد ژنهـای هـر کروموزوم ۱۰۰ انتخاب شده است. در نتيجه مي توان جمعيت را به صورت يك ماتريس ۵*۱۰۰ نشان داد كه درايه هاي آنرا اعدادي با محدوديت هاي ذكر شده تشكيل داده اند.

۵-۳-c-کنترل کنده PID: در این حالت جمعیت از ۳ کروموزوم تشکیل شده و تعداد ژنهای هر کروموزوم ۱۰ انتخاب شده است. در نتیجه مي توان جمعیت را به صورت یك ماتریس ۳*۱۰نشان داد که درایه هاي آنرا اعدادي با محدودیت هاي ذکر شده تشکیل داده اند.

در این مساله از اعداد تصادفی با توزیع یکنواخت به عنوان جمعیت اولیه استفاده می شود.

۵-۴-۳- تابع معیار

تابع معیار بر اساس آنچه که میزان خوب یا بد بودن کروموزوم را تعیین مي کند بدست مي آید. در این طرح معیار اصلي خطا در نظر گرفته شده است. تابع معیار در همه بخشها $\int_0^t |e^2(t)| dt$ و یا $\sum_{n=0}^{N} {e_n}^2$

(خطا : e) در نظر گرفته شده است.

محاسبه تابع هدف بدین صورت است که هر کروموزوم به صورت جداگانه در سیستم قرار داده مي شود و سیستم در یك حلقه قرار گرفته و سعي در کاهش خطا دارد. به منظور جلوگیري از ناپایدار شدن سیستم، محدوده اي براي خروجي در نظر گرفته می شود. بدین صورت که اگر مقدار لحظه ای پاسخ از ۳۰۰% پاسخ پایدار نهایي بیشتر شود یك رقم بزرگ به تابع معیار افزوده شده و چون اندازه تابع معیار در حال کاهش است، کروموزوم تولید کننده این مقدار در نسلهاي بعد از بین مي رود.

در ادامه مي توان توابع معیار دیگري نیز تعیین کرد، نظیر: فراجهش، زمان نشست و ... براي مثال مي توان فرا جهش مورد نظر را بین ۵% و ۱۰% در نظر گرفت. در انتها نیز مي توان با در نظر گرفتن اهمیت هر از این معیارها به آنها وزن داد. (اصولا باید معیار پایداري بیشترین وزن را داشته باشد)

۵-۴-۴- عملگرهاي الگوريتم ژنتيكي

- عملگر انتخاب براساس تابع معيار و با استفاده از چرخ رولتي صورت مـي گيـرد. بـدين شـكل عمل مي شود كه مقدار معكوس شده تابع معيار را بدست آورده مـی شـود. سـپس بـراي هـر كروموزوم تابع معيار محاسبه شده و با مقدار قبلي جمع زده می شود. سپس يك سـري عـدد تصادفي با توزيع يكنواخت توليد می شود كه طول بازه آن برابر طـول كـل مقـادير جمـع زده شده می باشد. اين عدد در هر بازه اي از مقادير جمع زده شده قرار گرفـت، ژنهـايي كـه تـابع معيار اين بازه را توليد كرده اند به عنوان انتخاب چرخ رولتی در نظر گرفته مـی شـوند. چـون هدف نهايی مينيمم كردن خطا است، پس هر چه خطا كمتر باشد معكـوس آن بزرگتـر شـده وطبيعي است كه احتمال قرار گـرفتن عـدد تـصادفي در آن بـازه بيـشتر اسـت. يعنـي درهـر انتخاب، هر چه خطا كوچكتر باشد احتمال انتخاب آن بيشتر است و بدين شكل الگـوريتم بـه جواب بهينه نزديكتر مي شود.
- در مرحله بعد مقادير بدست آمده براي تابع معيار به ترتيب پشت سر هم چيده مي شوند.
- در عملگر برش ابتدا ضريب (احتمال) انجام گرفتن برش در نظر گرفته شده و سـپس يكـی از قوانين موجود برای عمل برش به كار گرفته می شود. نتايج حاصل از روشهای مختلف برش در فصل بعد مورد بررسی قرار خواهند گرفت.
- در عملگر جهش ابتدا يكی از ژنهای موجود به صورت تصادفي انتخـاب مـی شـود. سـپس بـا احتمال ضريب جهش يك ژن ديگر به صـورت تـصادفي توليـد مـی شـود كـه جـايگزين ژن انتخاب شده در مرحله قبل گردد. بنابراين عملگر جهش يك ژن(يک پارامتر موتور و يـا يـک پارامتر PID) از يك كروموزوم را تغيير مي دهد. مي توان برنامه را طوري تنظيم كرد كـه در

آخرین نسل جهش وجود نداشته باشد.(به این دلیل که پاسخهای بهینه نهایي حفـظ شـوند و از بین نروند.)

۵-۴-۵- تنظیم پارامترها

برنامه نوشته شده به شکل کلي است و با تغيير يك پارامتر مي توان طول کروموزوم را تغییر داد. همانگونه که ذکر شد در برنامه، طول کروموزوم برای حالت دینامیکی ۵۰، برای حالت الکتریکی ۱۰۰ و برای کنترل کننده PID ، ۱۰ در نظر گرفته شده است.

۵-۴-۶- تنظیم فضاي جستجو

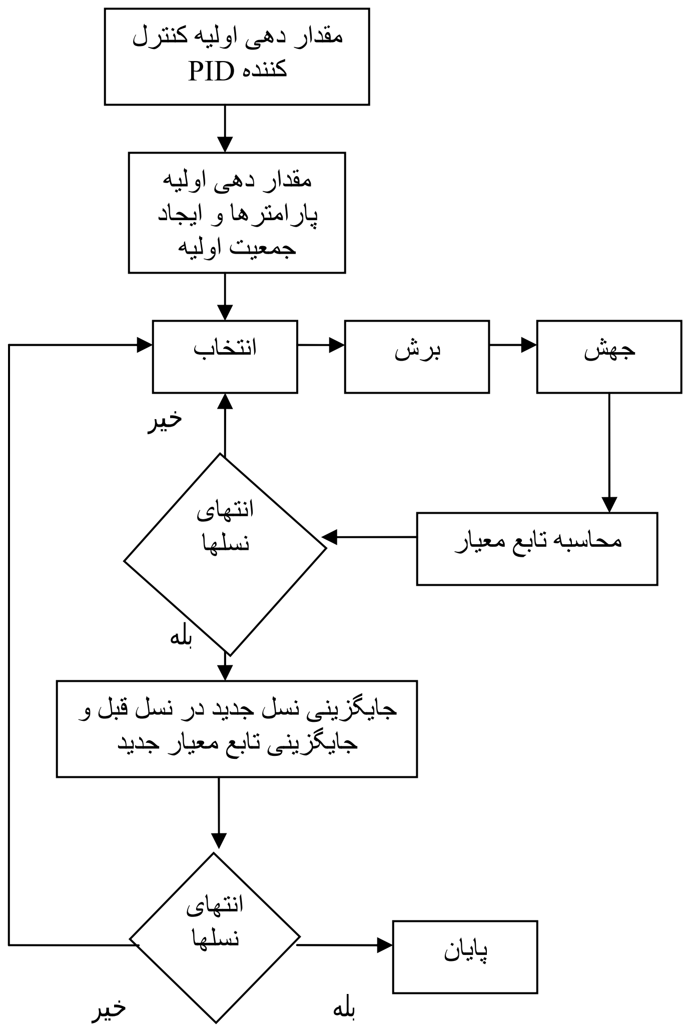
براي تنظيم فضاي جستجوي پارامترها، مي توان به دو شكل عمل كرد:

۱- براي هر يك از پارامترها فضاي جستجوي بزرگي را در نظرگرفت، كه در اين صورت جوابهـاي بدست آمده بسيار مناسب خواهند بود.

۲- فضاي جستجو را محدودتر کنیم، که حسن این روش این است که هم مقادیر بدست آمده بـه مقادیر واقعی نزدیکتر خواهند بود، و هم مي توان اين روش را با روشهاي ديگر به طـور دقيـق مقايسه كرد.

۵-۴-۷- فلوچارت برنامه

فلوچارت برنامه الگوریتم ژنتیکي براي بهینـه سـازي پارامترهـاي کنتـرل کننـده PID در شـکل نشان داده شده است:



شکل ٥-٤ فلوچارت برنامه

فصل ششم- نتایج

۶-۱- مقدمه

در این بخش با درنظر گرفتن معیار خطاي ITAE پارامترها شناسایي و کنترلر طراحي مي شود.

در این فصل الگوریتم ژنتیکی بدین ترتیب بررسی می شود:

ابتدا یک برنامه کلی نوشته شده و پاسخهای آن مشاهده می شود. سپس پاسخ سیتم به ازای ضرایب مختلف الگوریتم مشاهده می شود. همچنین معیارهاي مختلف انتخاب شده و پاسخ هایشان بررسی مي شود تا ضرایب بهینه برای حل این مساله بدست آیند. در انتها نیز نتایج حاصله با نتایج بدست آمده از یک روش دیگر مقایسه شده و مزایا و معایب آن بررسی می گردد.

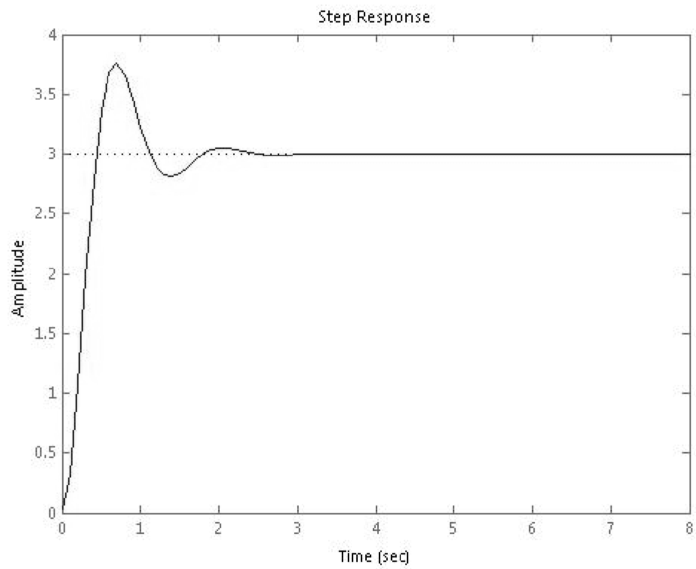
۶-۲- شناسايي پارامترهای حالت دینامیکی موتور **DC** به کمک الگوریتم ژنتیکی

برای شناسايي پارامترهای حالت دینامیکی موتور DC، ابتدا در m-file نرم افزار MATLAB هر یک از پارامترها و عملگرهای الگوریتم ژنتیکی به صورت Function نوشته شده و یک برنامه کلی برای مساله نوشته می شود (تا در طول تست برنامه بتوان به راحتی هر یک از مقادیر را تغییر داد). حال به شرح نحوه اجرای برنامه پرداخته می شود.

در ابتدا رابطه مدل شده یک موتور DC واقعی به صورت زیر در نظر گرفته می شود:

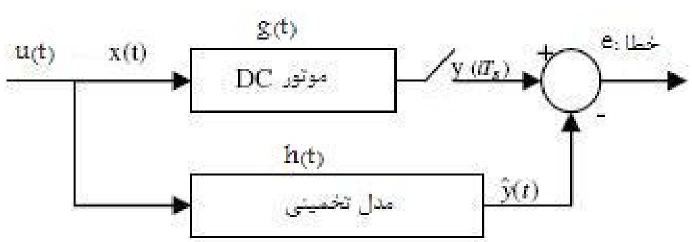
$$\frac{v(s)}{E(s)} = \frac{75}{S^2 + 4S + 25} \qquad (۶-۱)$$

پاسخ پله این سیستم به صورت شکل ۶-۱ می باشد:



شکل ٦-١پاسخ پله موتور DC

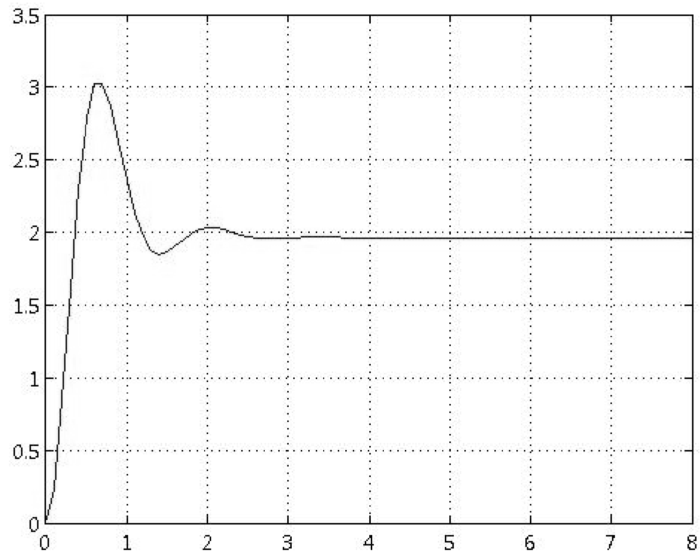
حال سیستم ذکر شده در فصل قبل به صورت شکل ۶-۲ مدل می شود:



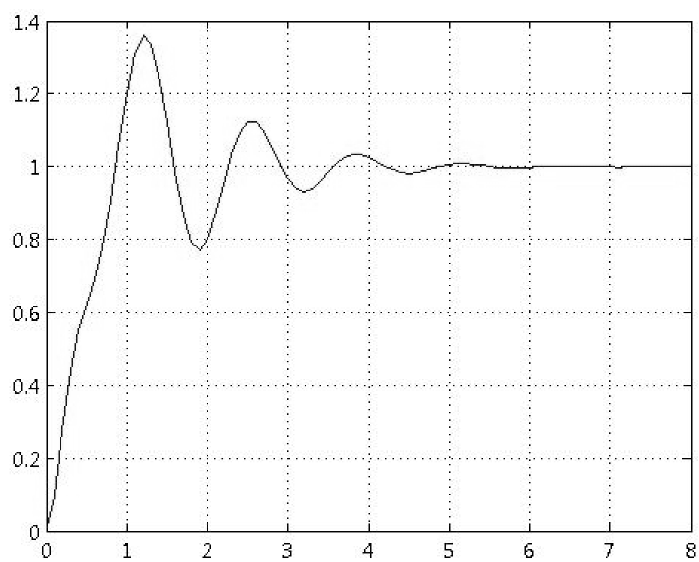
شکل ٦-٢مدل سیستم

با اجرای الگوریتم ژنتیکی پاسخ پله سیستم شناسايي شده به صورت شکل ۶-۳ ، خطای این پاسخ به

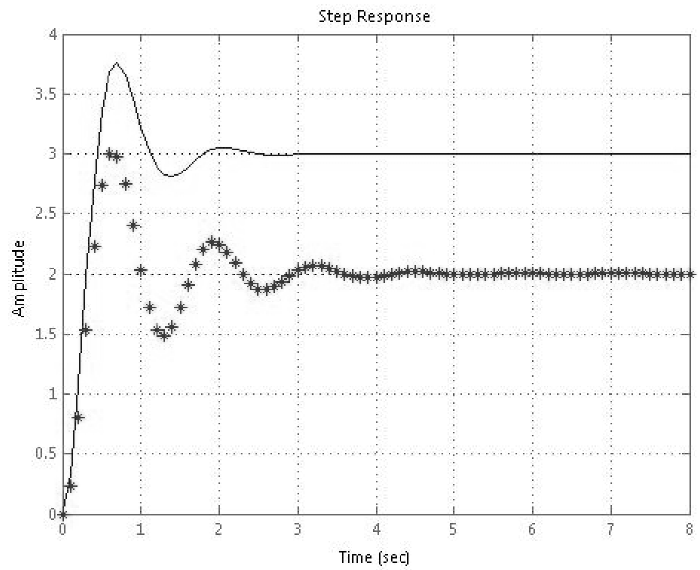
صورت شکل ۶-۴ می باشد و پاسخ پله سیستم واقعی و سیستم شناسايي شده به طور همزمان در شکل ۶-۵ نشان داده شده است:



شکل ٦-٣پاسخ پله سیستم سیستم شناسايي شده



شکل ٦-٤خطای سیستم شناسايي شده



شکل ٦-٥پاسخ پله سیستم سیستم شناسايي شده و موتور DC واقعی به طور همزمان
(* : سیستم شناسايي شده ، - : موتور DC واقعی)

از این پس برای سادگی در مقایسه پاسخها و مشاهده جوابهای بهتر، دو پاسخ به صورت همزمان نشان داده خواهند شد.

در این بخش ضرایب در نظر گرفته شده برای الگوریتم ژنتیکی به قرار زیر می باشند:

- بازه مقدار اولیه پارامترها:
- k =[0,10] , η=[0,10] , ω_n=[0,10]
- معیارهای در نظر گرفته شده: خطا (در صورتیکه تابع هزینه از ۰,۰۱ کمتر شود، الگـوریتم متوقف شود.)
- روش انتخاب: چرخ رولت
- روش بازترکیـب و ضـریب آن: $\begin{matrix} O_1 = P_1 + \alpha(P_2 - P_1) \\ O_2 = P_2 + \alpha(P_1 - P_2) \end{matrix}$ کـه در اینجـا α برابـر بـا ۰,۹ و ضریب احتمال بازترکیب برابر با ۰,٦ می باشد.

- ضریب جهش: ۰,۳

مقادیر بدست آمده از این شناسایي در جدول ۶-۱- نشان داده شده اند:

k	η	ω_n
۱,۰۳۷۸	۱,۴۵۹۷	۳,۶۸۱۲

جدول ٦-١پارامترهای سیستم شناسایي شده

مشاهده می شود که پارامترهای بدست آمده هیچک به مقادیر واقعی نزدیک نیستند و ضرایب الگوریتم ژنتیکی نیاز به تنظیم پارامتر دارد.

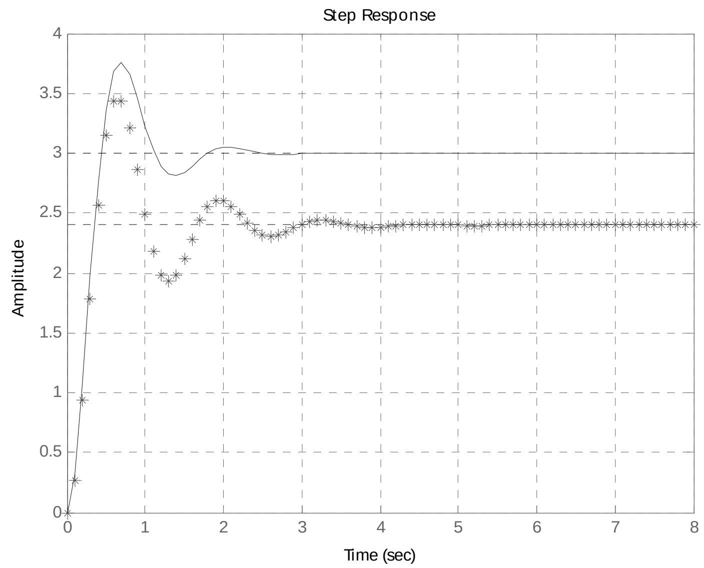
حال با تغیر یک به یک این پارامترها پاسخ ها مشاهده شده و مناسب ترین آنها برای تنظیم الگوریتم ژنتیکی انتخاب می شوند.

۶-۲-۱- بازه مقدار اولیه پارامترها

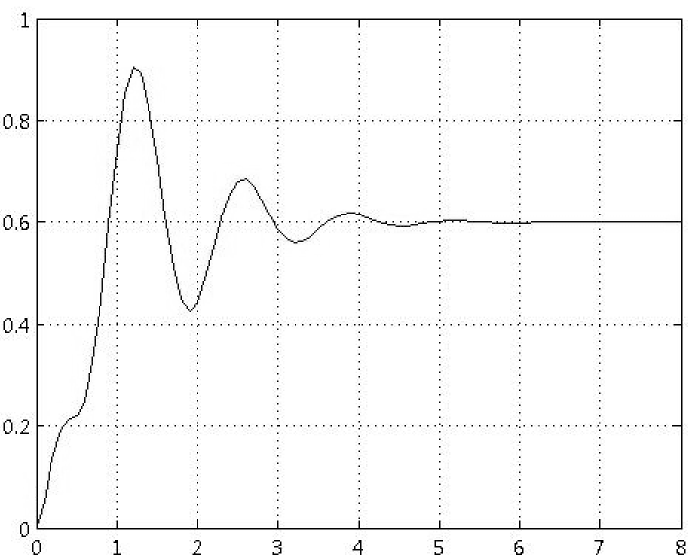
شکل ۶-۳- پاسخ پله سیستم به ازای بازه مقدار اولیه [۱۰,۰] بود. حال با تغییر این بازه به مقادیر وسیعتر و کوچکتر، تغییرات بوجود آمده در خروجی، مشاهده و مقایسه می شوند.

دقت شود که انتخاب مقدار اولیه تاثیر زیادی بر پاسخها و سرعت الگوریتم ژنتیکی در رسیدن به جوابهای بهینه خواهد داشت.می توان از تجربه های قبلی بدست آمده در این زمینه کمک گرفت.برای مثال مقدار پارامتر ضریب میرایي کوچکتر از ۱ بوده و می توان مقدار اولیه آنرا [۱,۰] انتخاب نمود.

انتخاب بازه مقدار اولیه: ω_n=[0,10] , η=[0,1] , k =[0,10]



شکل ٦-٦پاسخ پله سیستم شناسايي شده و موتور DC واقعی به طور همزمان



شکل ٦-٧خطای سیستم شناسايي شده

k	η	ω_n
۱٫۱۱۴۳	۰٫۶۳۷۵	۲٫۴۰۱۳

جدول ٦-٢پارامترهای سیستم شناسايي شده

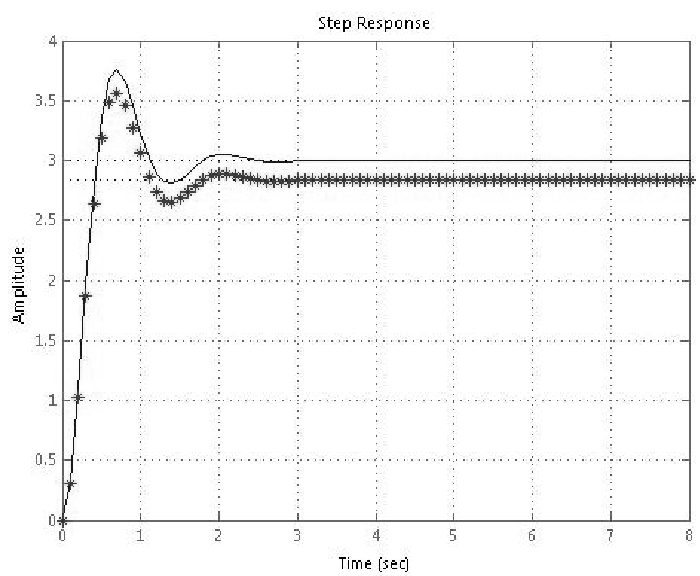
از شکل ۶-۷- مشاهده می شود که خطای سیستم مقداری کاهش یافته است. حال به تنظیم پارامترهای دیگر الگوریتم پرداخته می شود.

۶-۲-۲- ضریب بازترکیب و احتمال بازترکیب

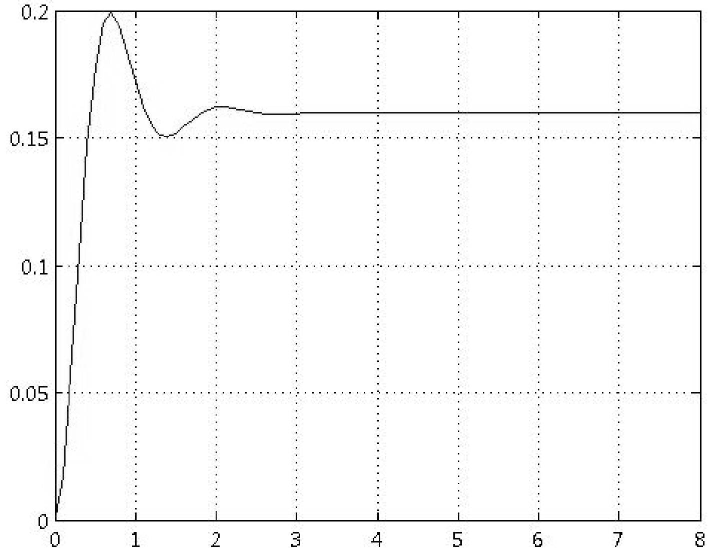
با تغییر مقادیر ضریب بازترکیب و احتمال بازترکیب مشاهده شد که بهترین مقادیر برای این دو پارامتر به قرار زیر می باشند:

α =۰,۷ و ضریب احتمال بازترکیب = ۰,۷

پاسخ حاصل از این مقادیر در اشکال ۶-۸- و ۶-۹- نشان داده شده اند:



شکل ٦-٨پاسخ پله سیستم سیستم شناسايي شده و موتور DC واقعی به طور همزمان



شکل ٦-٩خطای سیستم شناسایي شده

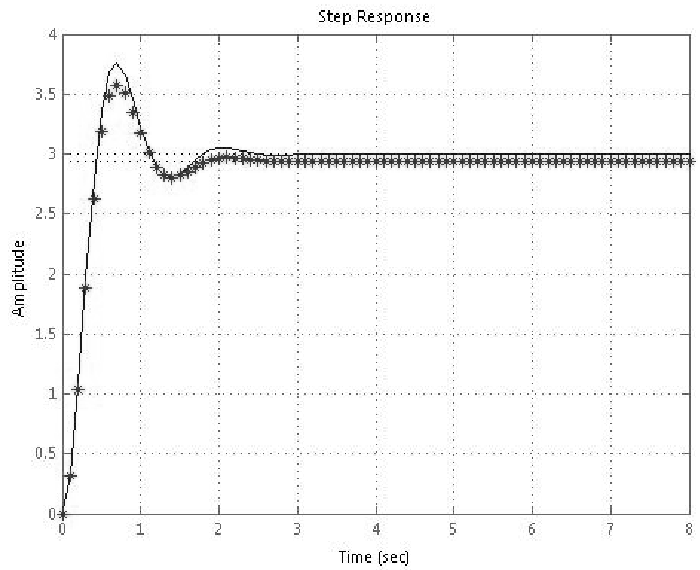
k	η	ω_n
۲,۵۶۷۲	۲۳۷۱.	۴,۳۷۱۲

جدول ٦-٣ پارامترهای سیستم شناسایي شده

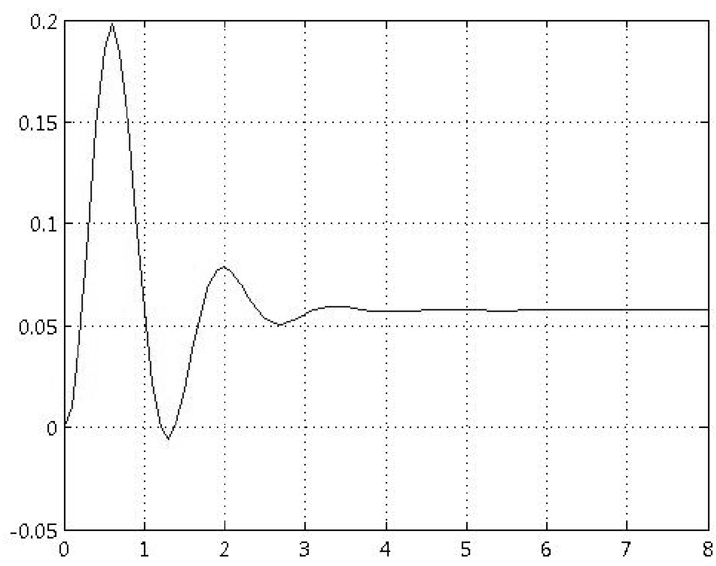
۶-۲-۳- ضریب جهش

تغییر مقدار ضریب جهش نیز تاثیر بسزایي روی پاسخ سیستم خواهد داشت. زیرا انتخاب مقدار بزرگ برای این ضریب باعث تولید اعداد تصادفی بیشتری خواهد شد و ممکن است جوابهای مناسب از بین بروند. انتخاب کوچک این مقدار نیز ممکن است باعث شود الگوریتم ژنتیکی در مینیمم های محلی

گیر کند و نتواند مینیمم های مطلق را پیدا کند. در این قسمت ضریب جهش ۰٫۱۸ = Pm[1] انتخاب شد. پاسخ های حاصل از این این مقدار در اشکال ۶-۱۰- و ۶-۱۱- مشاهده می شود.



شکل ٦-١٠ پاسخ پله سیستم سیستم شناسايي شده و موتور DC واقعی به طور همزمان



شکل ٦-١١خطای سیستم شناسايي شده

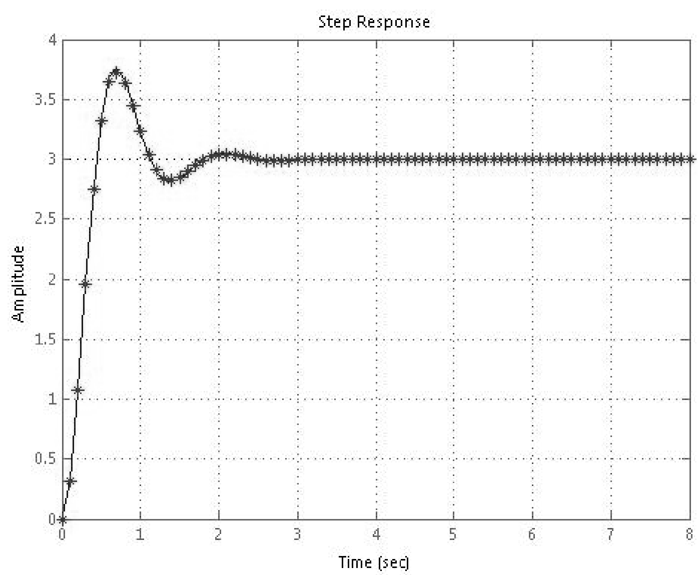
[1] Probability of Mutation

k	η	ω_n
۲,۵۶۷۲	۲۳۷۱.	۴,۳۷۱۲

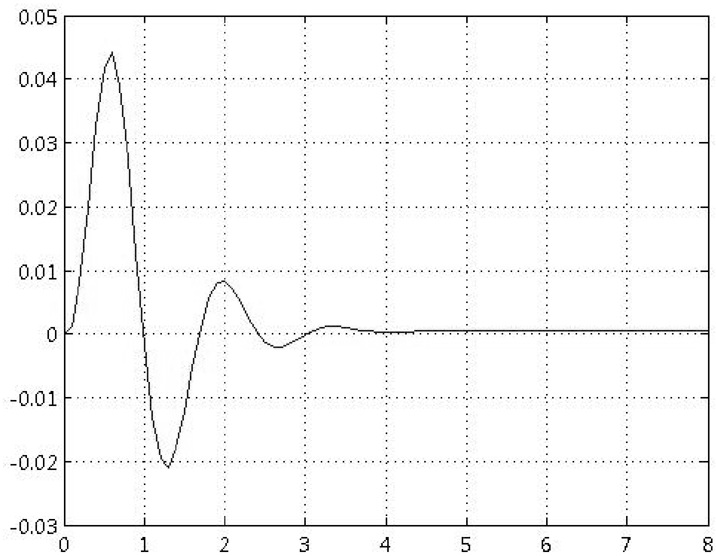
جدول ٦-٤ پارامترهای سیستم شناسايي شده

۶-۲-۴- يافتن پاسخ های بهینه نهايي

در اين مرحله تمام ضرايب بهينه در كنار هم گذاشته شده و سيستم شناسايي شده با كمترين خطـا را نشان می دهند. در اين صورت سيستم شناسايي شده در اشكال ۶-۱۲ و ۶-۱۳- نمايش داده شده اند.



شكل ٦-١٢پاسخ پله سيستم سيستم شناسايي شده و موتور DC واقعی به طور همزمان



شکل ٦-١٣خطای سیستم شناسايي شده

k	η	ω_n
۳,۰۰۱۱	۰,۳۹۹۸	۴,۹۹۹۶

جدول ٦-٥پارامترهای سیستم شناسايي شده

مشاهده می شود که فراجهش تابع خطا حدودا ۰,۰۴۴ بوده و مقدار خطا پس از ۴ ثانیه به صـفر میـل می کند.

بررسی جدول ۶-۵ نشان می دهد که ضرایب بدست آمده در این بخش کاملا نزدیـک مقـادیر واقعـی بوده و تقریبا برابر با آن می باشد. این پاسخها در نسل ۳۱ ام بدست آمده اند. همان طور که بیـان شـد الگوریتم به شکلی تنظیم شده بود که الگوریتم تا ۴۰ نسل اجرا شـود و در صـورت اینکـه مقـدار تـابع هزینه از مقدار تعیین شده کمتر بود (در اینجا ۰,۰۰۱)، الگوریتم متوقف شود که چنین اتفاقی در ایـن مرحله رخ داد.

در جدول ۶-۶ قسمت انتهايي كروموزوم توليد شده در نسل ۳۱ ام نشان داده شـده اسـت كـه در ايـن نسل به دليل اينكه جوابهای بدست آمده شرايط الگوريتم را برآورده كـرده، ايـن پاسـخها باعـث توقـف برنامه شده اند. در اين جدول مشاهده می شود كه روی برخی ژنهـا جهـش اتفـاق افتـاده اسـت(مـثلا شماره های ۵-۸-۱۲-۱۶)، ولي به دليل اينكه به ازای اين ژنهای جهـش يافتـه مقـدار تـابع هزينـه از مقدار تابع هزينه به ازای ژن قبلی بيشتر شده است، الگوريتم ژنتيکی به دنبال ژن هايي رفته كـه تـابع هزينه كمتری دارند.(يعنی همان ژنهای قبلی)

	k	η	ω_n
1	3.0011	0.3998	4.9996
2	3.0011	0.3998	4.9996
3	3.0011	0.3998	4.9996
4	3.0011	0.3998	4.9996
5	1.2378	0.3998	4.9996
6	3.0011	0.3998	4.9996
7	3.0011	0.3998	4.9996
8	3.0011	0.6624	5.2468
9	3.0011	0.3998	4.9996
10	3.0011	0.3998	4.9996
11	3.0011	0.3998	4.9996
12	0.9534	0.3998	4.9996
13	3.0011	0.3998	4.9996
14	3.0011	0.3998	4.9996
15	3.0011	0.3998	2.3796
16	3.0011	0.3998	4.9996
17	3.0011	0.3998	4.9996
18	3.0011	0.3998	4.9996
19	3.0011	0.3998	4.9996
20	3.0011	0.3998	4.9996

جدول ٦-٦كروموزومهای توليد شده در نسل ۳۱ ام

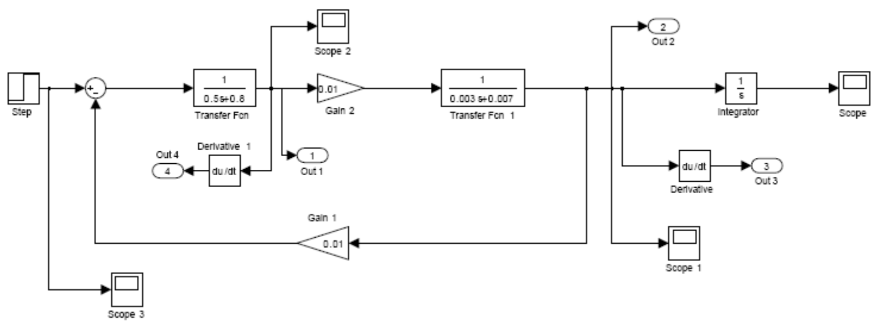
در ضمن می توان الگوریتم را طوری تنظیم کرد که نیازهای دیگر طراح مساله را نیز برآورده سازد. برای مثال ممکن است طراح در نظر داشته باشد که مقدار خطا زودتر از زمان بدست آمده در حالت قبل (شکل ۶-۱۳-) به صفر میل کند(برای مثال در کمتر از ۱ ثانیه) . در این حالت با تنظیم تابع معیار می توان به این خواسته رسید، با این تفوت که الگوریتم ژنتیکی ممکن است در تعداد نسل های بیشتر (و متعاقبا در مدت طولانی تری از زمان اجرای برنامه) به این خواسته دست یابد.

۶-۳- شناسايي پارامترهای حالت الکتریکی موتور **DC** به کمک الگوریتم ژنتیکی

۶-۳-۱- مقدمه

در این بخش برای شناسايي پارامترهای الکتریکی موتور DC از یکی از Toolbox های نرم افزار MALAB به نام gatool (که Toolbox مربوط به الگوریتم ژنتیکی است) استفاده می شود. بدین منظور ابتدا در بخش Simulink، سیستم مورد نظر طراحی می شود. پس از اجرای این مدل، مقادیر مورد نظر در Workspace ذخیره می شوند. سپس در m-file، پارامترهای مجهول برنامه و توابع معیار الگوریتم ژنتیکی نوشته می شوند. پس از Load و Save کردن مقادیر ذخیره شده در Workspace، تنظیمات لازم در gatool انجام شده و toolbox مذکور اجرا می شود. در انتها مقادیر بدست آمده از اجرای الگوریتم ژنتیکی به Workspace انتقال داده می شود (Export) و توسط یک برنامه m-file دیگر، مقادیر بدست آمده جمع آوری شده و پاسخها به نمایش در می آیند. حال به شرح این بخش پرداخته می شود.

سیستم ذکر شده در فصل قبل (شکل ۵-۳) به صورت شکل ۶-۱۴ مدل می شود:



شکل ٦-١٤مدل موتور DC

در این مدل، از پارامترهای الکتریکی یک موتور DC واقعی استفاده شده است. پارامترهای این موتور عبارتند از:

L	۰,۶
R	۰,۲
K_T	۰,۰۱
J	۰,۰۰۳
f	۰,۰۰۷

جدول ٦-٧پارامترهای الکتریکی واقعی موتور DC

برای بدست آوردن تابع معیار در طول این شناسايي، باید تابع خطا را بدست آورد که به صورت معادله ۶-۲ محاسبه می شود:

$$e(nT) \underline{\underline{\Delta}} y(nT) - W(nT)K = y(nT) - \hat{y}(nT) \in R^2 \qquad (٦\text{-}٢)$$

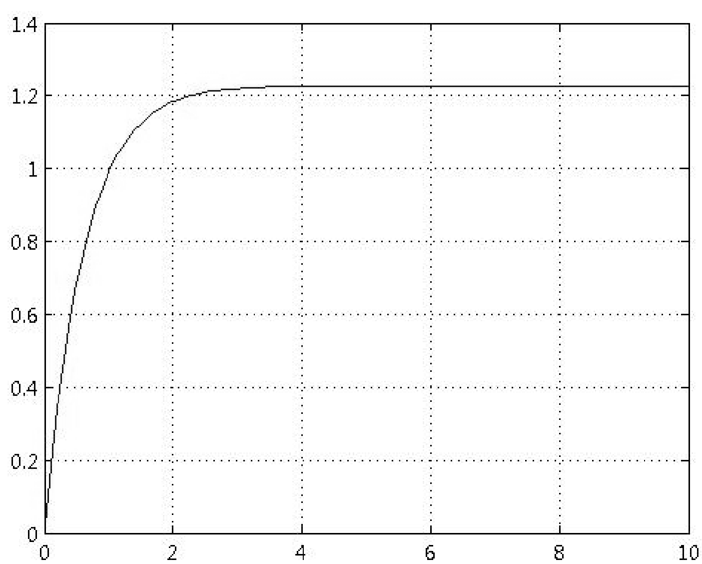
از مجموع مربع خطاها به عنوان تابع معیار استفاده می شود که از رابطه ٦-٣ بدست می آید:

$$E^2(K)\underline{\underline{\Delta}}\sum_{n=1}^{N}\ (y(nT)-W(nT)K)^T(y(nT)-W(nT)K) \qquad (۳-٦)$$

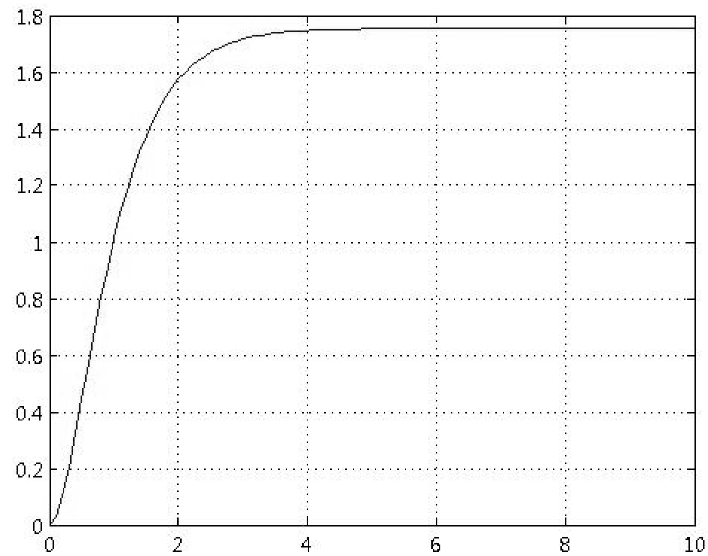
۶-۳-۲- روش اجرای الگوریتم ژنتیکی در شناسايي پارامترهای الکتریکی

برای اجرای الگوریتم بدین صورت عمل می شـود کـه ابتـدا از شـکل ۶-۱۴ مقـادیر عـددی $i(t)$ ، $\frac{di(t)}{dt}$ ، $\omega(t)$ و $\frac{d\omega(t)}{dt}$ بدست می آیند. سپس با استفاده از روابط ۲-۶ و ۳-۶ تابع هزینه (تـابع خطا) ساخته می شود. در ادامه الگوریتم ژنتیکی با استفاده از مقادیر موجود و برنامه نوشـته شـده به شناسايي پارامترهای مجهول می پردازد.

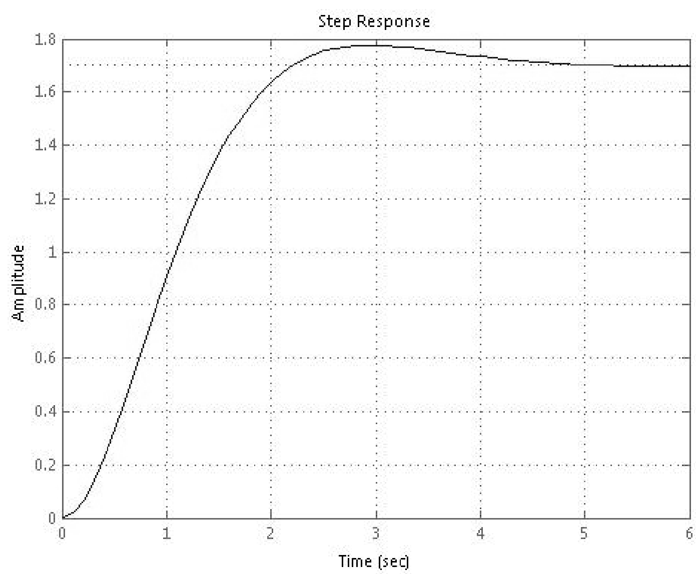
در مدل موتور DC، ولتاژ ورودی تابع پله در نظر گرفته شده است. در ادامه برای مقایـسه جوابهـا، به جز مقادیر بدست آمده از شناسايي، پاخهای پله دو سیستم واقعی و سیـستم شناسـايي شـده و همچنین خطای حاصل از این شناسايي ترسیم خواهند شد.



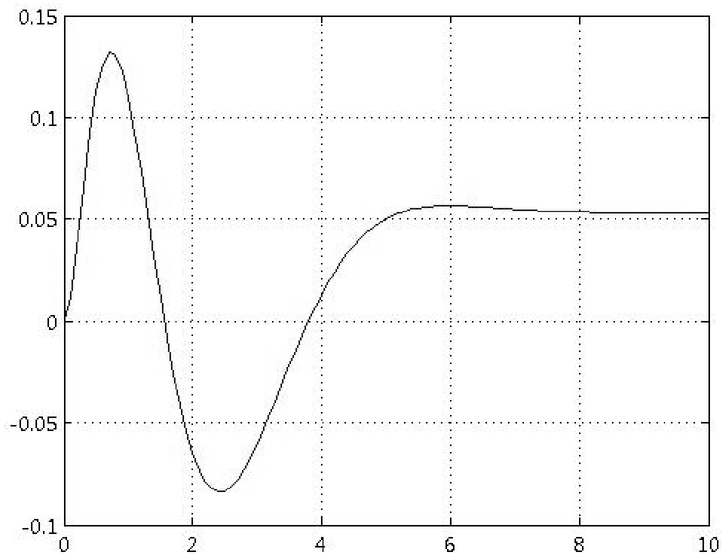
شکل ٦-١٥نمودار جریان موتور DC



شکل ٦-١٦نمودار سرعت موتور DC (خروجی موتور یا به عبارتی پاسخ پله) به ازای مقادیر واقعی



شکل ٦-١٧نمودار سرعت موتور DC (خروجی موتور یا به عبارتی پاسخ پله) به ازای مقادیر شناسايي شده



شکل ٦-١٨خطای سیستم شناسايي شده

L	٠,٥٧۴
R	٠,۶٣١
K_T	٠,٢٢
J	٠,٠٢٥
f	٠,٠٠٩

جدول ٦-٨پارامترهای الکتریکی شناسايي شده موتور DC (مرحله ١)

حال به تنظیم پارامترهای الگوریتم ژنتیکی (برای بهبود پاسخها و کاهش خطا) پرداخته می شود.

۶-٣-٣- روش گزینش

در بین روش های مختلف در gatool نظیر uniform، stochastic uniform، tournament و ... مناسب ترین روش چرخه رولت می باشد.

۶-۳-۴- عملگر بازترکیب

با تغییر روش بازترکیب و ضرایب آن مشاهده شد که بهترین روش برای این مساله بازترکیب دو نقطه ای می باشد.

۶-۳-۵- ضریب جهش

در این toolbox برای عملگر جهش نیز روش های مختلفی وجود دارد. (همانند Gaussian، Uniform و ...) که مناسب ترین روش موجود uniform می باشد.

۶-۳-۶- پارامترهای دیگر **gatool**

به غیر از موارد ذکر شده، gatool دارای تنظیمات دیگری نیز می باشد که به بیان برخی از مهمترین آنها پرداخته می شود:

- Population type: در این پارامتر می توان نوع کروموزوم ها را مشخص کرد که شامل پیوسته (Double vector) و گسسته (Bit string) می باشد. با توجه به مساله گزینه Double vector انتخاب می شود.
- Population size: که تعداد نسل ها را نشان می دهد.
- Initial range: با کمک این پارامتر می توان بازه پارامترهای مجهول را مشخص نمود.
- Stopping criteria: در این بخش با توجه به خواسته طراح می توان شرایط توقف الگوریتم را تنظیم کرد. برای مثال الگوریتم پس از اجرای تولید تعداد نسل مشخصی متوقف شود، یا پس از زمان خاصی متوقف شود.

۶-۳-۷- یافتن پاسخ های بهینه نهايي

با آزمودن تمام پارامترهای الگوریتم ژنتیکی سیستم شناسايي شده با کمترین خطا بدین صورت بدست می آید:

Population type: Duble vector

Population size: 100

Initial range: [0,1]

Selection: Roulete

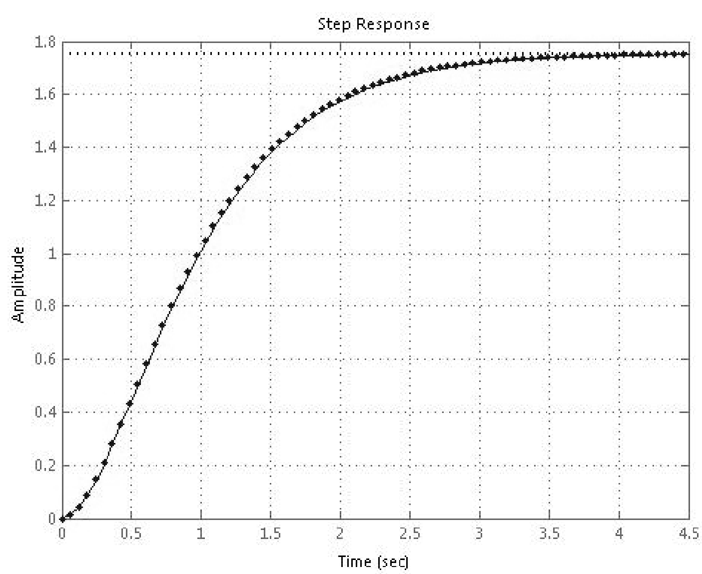
Crossover: Two point

Mutation: Uniform , rate = 0.3

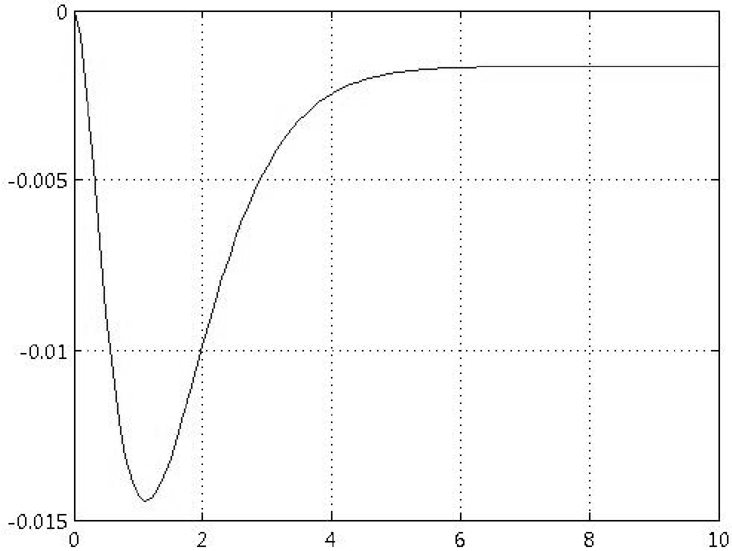
Stopping criteria: generation = 300 , Time limit = Inf

Hybrid: fminsearch

پاسخ حاصل از این روش در اشکال ۶-۱۹ و ۶-۲۰ نشان داده شده اند:



شکل ٦-١٩پاسخ پله سیستم سیستم شناسايي شده و موتور DC واقعی به طور همزمان
("." سیستم شناسايي شده ، "-" سیستم واقعی)



شكل ٦-٢٠خطای سيستم شناسايي شده

L	۰,۴۷۷
R	۰,۷۹۸
K_T	۰,۰۱۱۱
J	۰,۰۰۳۳
f	۰,۰۰۷۷

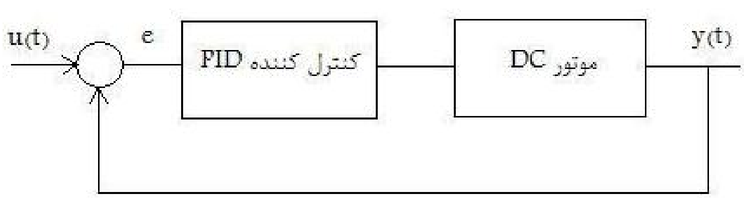
جدول ٦-٩پارامترهای سيستم شناسايي شده

۶-۴- طراحی کنترل کننده **PID** برای موتور **DC** به کمک الگوریتم ژنتیکی

۶-۴-۱- مقدمه

از آنجاییکه برای کار با موتورهای DC در صنعت، باید بتوان آنها را بـه درسـتی راه انـدازی کـرد و بـه کنترل سرعت آنها پرداخت، لذا هدف این بخش طراحی یک کنترل کننده ساده و پرکاربرد برای ایـن منظور است.

در این بخش در ابتدا با درنظر گرفتن معیار خطاي ITAE به عنوان تـابع هزینـه بـه طراحـی کنترلـر پرداخته می شود. در ادامه معیارهای دیگری به تابع هزینه افزوده می شوند. به عنوان مثال برای اینکه در هنگام راه اندازی موتور DC سرعت ابتدايي به طور ناگهاني بالا نرفته و به آرامی به سـمت سـرعت مورد نظر میل کند، می بایست فراجهش را تا حد ممکن کاهش داد. همچنین در برخی کاربردهـا لازم است پس از راه اندازی موتور، سرعت آن سریعا به سرعت دلخواه نهايي برسد. در این موارد بایـد زمـان صعود و زمان نشست را کاهش داد. بدین منظور در خلال برنامه معیارهـاي مختلـف انتخـاب شـده و پاسخ هايشان مقایسه مي شود.



شکل ٦-٢١ بلوک دیاگرام کنترل موتور DC

۶-۴-۲- طراحی کنترلر با معیار خطای ITAE

در این قسمت کنترل کننده فقط با استفاده از معیار خطا طراحی می شود. طبیعی است پاسخ چنین سیستمی فقط یک سری از خواسته های کلی مساله را می تواند برآورده کند. برای مثال اگر سیستم ناپایدار باشد، به کمک کنترلر PID سیستم پایدار شده و همچنین در صورت وجود خطای حالت ماندگار، می توان مقدار آنرا کاهش دهد.

تابع تبدیل در نظر گرفته شده برای موتور DC، همان تابع تبدیل اسفاده شده در بخش قبلی است:

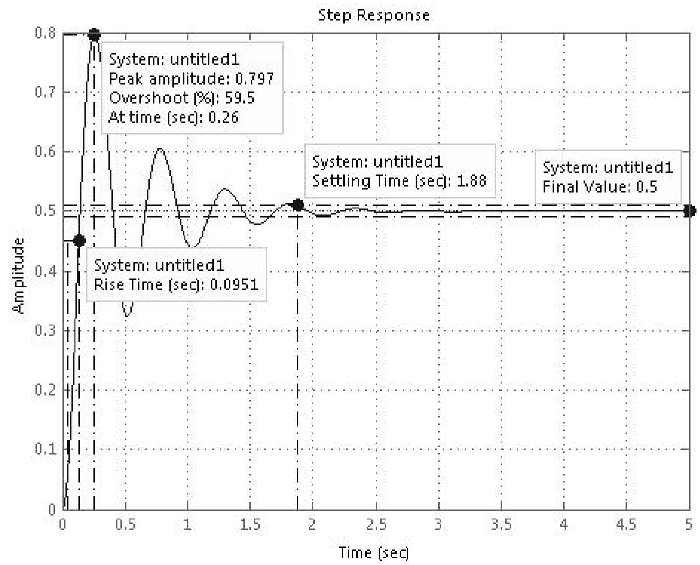
$$G(s)=\frac{v(s)}{E(s)}=\frac{75}{s^2+4s+25} \qquad (۶-۴)$$

در این صورت تابع تبدیل سیستم حلقه بسته از رابطه ۶-۶ بدست می آید:

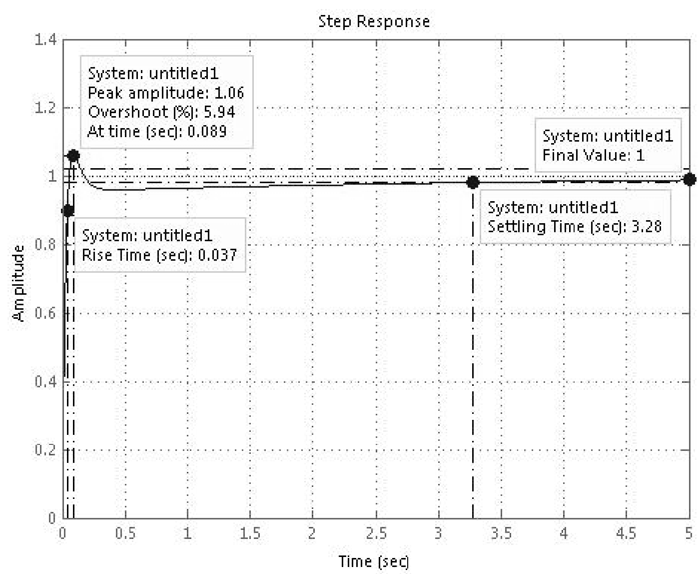
$$C(s)=\frac{Kd.s^2+Kp.s+Ki}{s} \qquad (۶-۵)$$

$$\frac{Y(s)}{U(s)}=\frac{G(s)C(s)}{1+G(s)C(s)}=\frac{75(Kd.s^2+Kp.s+Ki)}{s(s^2+4s+25)} \qquad (۶-۶)$$

پس از اجرای الگوریتم ژنتیکی، پاسخ پله سیستم ترسیم می شود:



شکل ٦-٢٢ پاسخ پله موتور DC بدون کنترلر



شکل ٦-٢٣ پاسخ پله موتور DC با کنترل کننده PID

از اشکال ۶-۲۲ و ۶-۲۳ مشاهده می شود که کنترل کننده PID خطای حالت ماندگار را از مقدار ۰,۵ به صفر کاهش داده است.

K_p	۴,۵
K_i	۷,۲
K_d	۱,۳

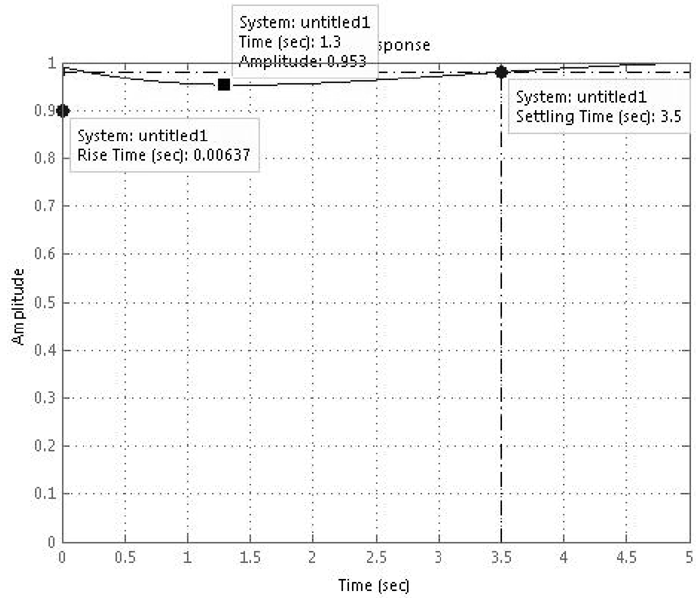
جدول ٦-١٠پاسخهاي به دست آمده با استفاده از معيار خطا به عنوان تابع هزينه

۶-۴-۳- افزودن معیار فراجش به تابع هزینه

در این قسمت معیار فراجهش به عنوان يكي ديگر از پارامترهای تعیین کننده تابع معیار، به تابع هزینه افزوده می شود. با توجه به نیازهای موجود و همچنین خواسته های طراح، می توان به هر یک از این معیارها وزن جداگانه ای اختصاص داد که ارزش هر یک از معیارها را از دیگری جدا کند. در صورت استفاده نکردن از این روش، تمام هزینه ها دارای ارزش برابرخواهند بود.

با توجه به شکل ۶-۲۲ ، ملاحظه می شود که مقدار فراجهش برابر % ۵۹,۵ است. در نهایت هدف این بخش کاهش فراجهش به کمتر از ۵% می باشد. (Overshoot < 5%)

شکل ۶-۲۴ پاسخ پله سیستم را به ازای این تابع معیار نمایش می دهد:



شکل ٦-٢٤پاسخ سیستم به ازای معیارهای خطا و فراجهش

از شکل ۶-۲۴ مشاهده می شود که مقدار فراجهش به % ۴,۷- کاهش یافته است (فروجهش).

K_p	۶,۷۸
K_i	۵,۳
K_d	۳,۵

جدول ٦-١١آخرین نسل پاسخها به ازای معیارهای خطا و فراجهش

۶-۴-۴- افزودن معیار زمان صعود و زمان نشست به تابع هزینه

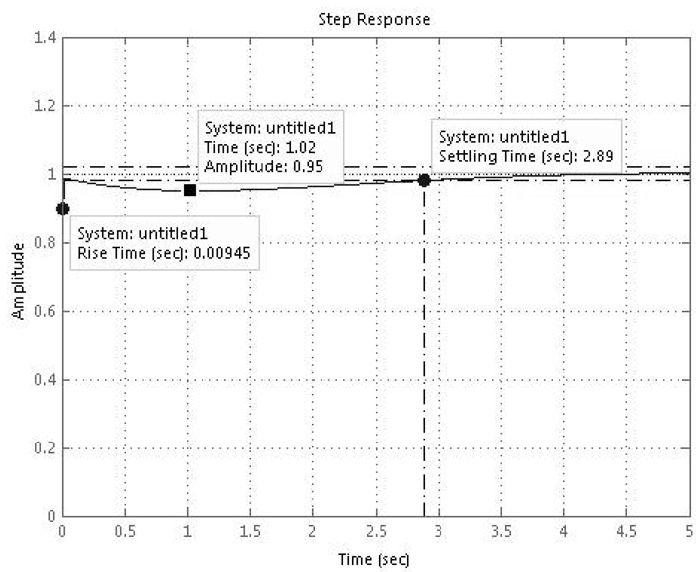
در این مرحله معیار دیگری نیز به الگوریتم افزوده می شود. در ایـن قـسمت هـدف کـاهش زمـان صعود و زمان نشست است تا سیستم سـریعا بـه مقـدار نهـايي خـود میـل کنـد (بـرای مثـال در

کاربردهاي که موتور باید پس از استارت با سرعت به سرعت رفرنس خود میـل کنـد). مقـادیر در نظر گرفته شده برای زمان نست و زمان صعود به صروت زیر می باشد:

$t_r \leq 0.01$

$t_s \leq 3$

حال با توجه به این مقادیر الگوریتم ژنتیکی به طراحی پارامترهای کنترل کننده می پردازد. در این بخش وزن این معیار برابر وزن معیار فراجهش و نصف معیار اصلی یعنی خطا در نظر گرفته شده است. شکل ۶-۲۵ پسخ پله سیستم را نشان می دهد:



شکل ٦-٢٥پاسخ سیستم به ازای معیارهای خطا و فراجهش و زمانهای صعود و نشست

K_p	۵,۷
K_i	۶,۷
K_d	۸,۴۵

جدول ٦-١٢آخرین نسل پاسخها به ازای معیارهای خطا و فراجهش و زمانهای صعود و نشست

نکته مهمی که باید به آن توجه کرد این است ممکن است در محدوده اي كـه بـراي معيارهـا در نظـر گرفته شده، پاسخی برای پارامترهای کنترلـر PID وجـود نداشـته باشـد، لـذا الگـوریتم ژنتیکـی نیـز نمی تواند معیارهای خواسته شده را برآورده کند. براي مثال اگر معيار زمان نشست $t_s \leq 0.5$ انتخاب شود (و معيار فراجهش هم باقي بماند)، الگوريتم ژنتيكي نمي تواند هيچ جوابي پيدا كند، زيـرا جـواب ممکنی برای این محدوده وجود ندارد.

۶-۴-۵- تاثیر پارامتهای الگوریتم ژنتیکی بر روی پاسخها

پس از اعمال معیارهای متفاوت روی تابع هزینه، به بررسی پارامترهای الگوریتم ژنتیکی پرداخته می شود.

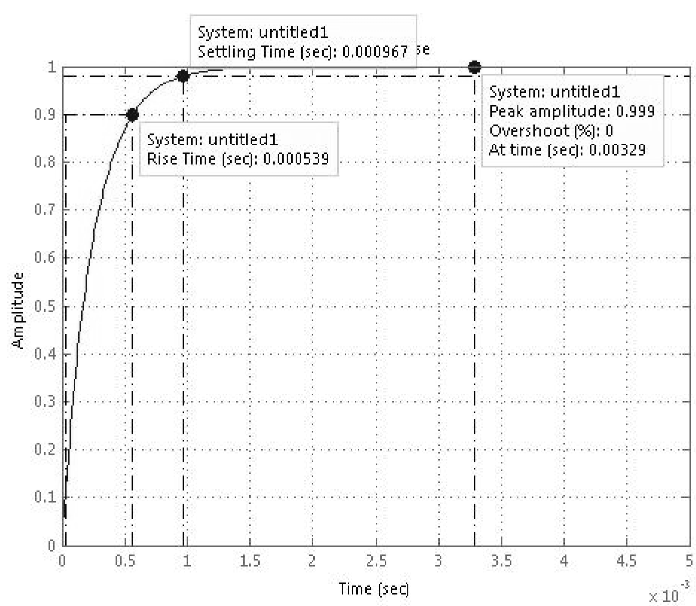
بازه پاسخها) در مورد اتنخاب بازه پاسخها برای کنترل کننده PID باید به این نکته توجه شود که انتخاب کوچک این رنج باعث می شود فضای جستجو به طور کامل پوشش نیابد و پاسخهای بهینه در خارج از این محدوده قرار داشته باشند. انتخاب فضای بسیار بزرگ نیز باعث می شوند مقادیر به دست آمده از مقادیر واقعی در حالت تجربی فاصله داشته باشند. پس مناسبتر است این بازه به صورت اصولی (از نظر تجربی و تیوری) انتخاب شود. مناسب ترین بازه برای این منظور بازه [۰,۱۰۰] می باشد.

انتخاب) مناسب ترین روش برای انتخاب ژنها در این بخش، همانند بخش قبل روش رولت گردان است که یک روش کامل و مناسب با نتایج مناسب برای مساله می باشد.

بازترکیب) پس از بررسی روشهای مختلف برای اعمال این عملگر به الگوریتم ژنتیکی و بررسی نتایج حاصله مناسب ترین روش بازترکیب دو نقطه ای با ضریب ۰,۷ به دست آمد.

جهش) در بخشهای قبل تاثیر عملگر جهش در شناسايي پارامترهای موتور DC مشاهده شد. حال به بررسي تاثیر تغییرات نرخ این عملگر الگوریتم ژنتیکی در طراحی کنترل کننده PID پرداخته می شود. برای این منظور با تغییر نرخ عملگر جهش، به مشاهده و مقایسه پاسخها پرداخته می شود. در نهایت مناسب ترین مقدار برای این عملگر ۰,۱۸ به دست می آید.

با اعمال این عملگرها و در نظر گرفتن معیارهای ذکر شده در بخش قبل، بهترین نتیجه بدست آمده در شکل ۶-۲۶ مشاهده می شود:



شکل ٦-٢٦ مناسب ترین پاسخ بدست آمده برای سیستم

از این شکل مشاهده می شوند که مقادیر خطای حالت دایم، فراجهش، زمان صعود و زمان نشست به قرار زیر هستند:

$E_{ss} = 0$

$O.S. = 0$

$t_r = 0.0005$

$t_s = 0.0009$

K_p	۷
K_i	۱۱,۷
K_d	۱۵,۶

جدول ٦-١٣آخرین نسل پاسخها به ازای معیارهای خطا و فراجهش و زمانهای صعود و نشست

۶-۵- مقایسه روش الگوریتم ژنتیکی با روش های دیگر

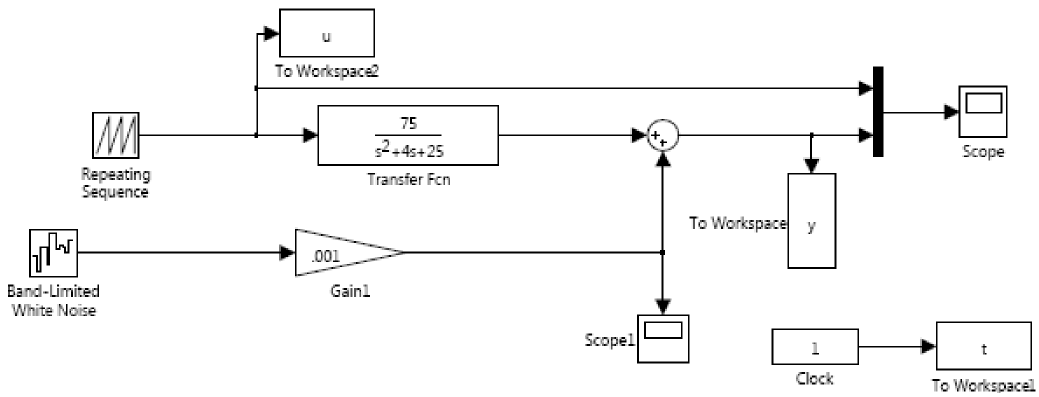
۶-۵-۱- شناسايي پارامترهای دینامیکی موتور **DC** با روش **RLS**[1]

هدف این بخش بررسی مزایا و معایب روش الگوریتم ژنتیکی در مقایسه با سایر روشها است. بدین منظور به شناسايي پارامترهای دینامیکی موتور DC به کمک روش RLS پرداخته می شود.

توضیحات مربوط به روش RLS و نحوه اجرای این روش در ضمیمه ب آمده است.

[1] Recursive Least Square

برای اجرای روش RLS در ابتدا لازم است اطلاعات ورودی و خروجی جمع آوری شوند. سپس با استفاده از این اطلاعات به شناسايي سيستم پرداخته شود. برای این منظور بلوک دیاگرامی همانند شکل ۶-۲۷ طراحی می شود:



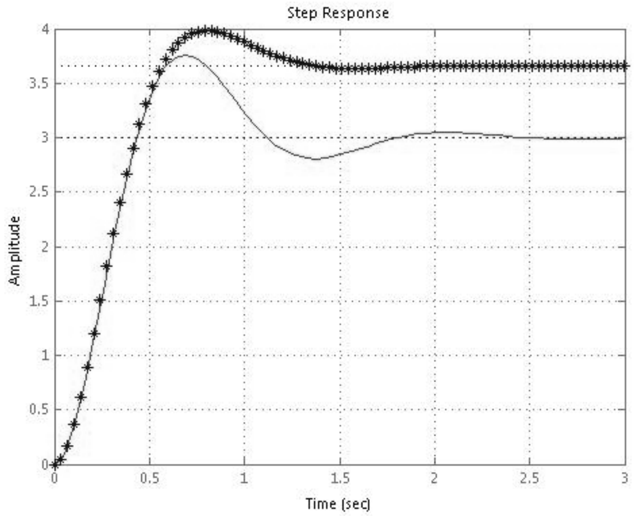
شکل ۶-۲۷بلوک دیاگرام مربوط به شناسايي سيستم به کمک روش RLS

در شکل۶-۲۷، تابع تبدیل سیستم واقعی قرار داده شده است:

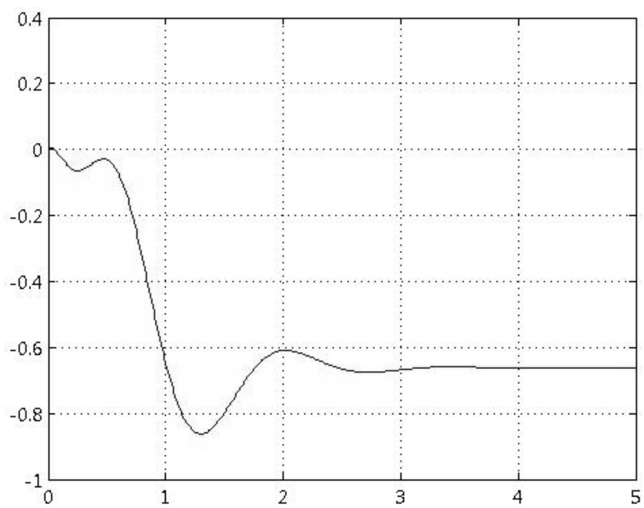
$$G(s)=\frac{75}{S^2+4S+25} \quad (۶-۷)$$

با اعمال ورودی مناسب برای این سیستم خروجی های مورد نظر جمع آوری می شوند. در ادامه با نوشتن الگوریتم برنامه به شناسايي سيستم پرداخته می شود.

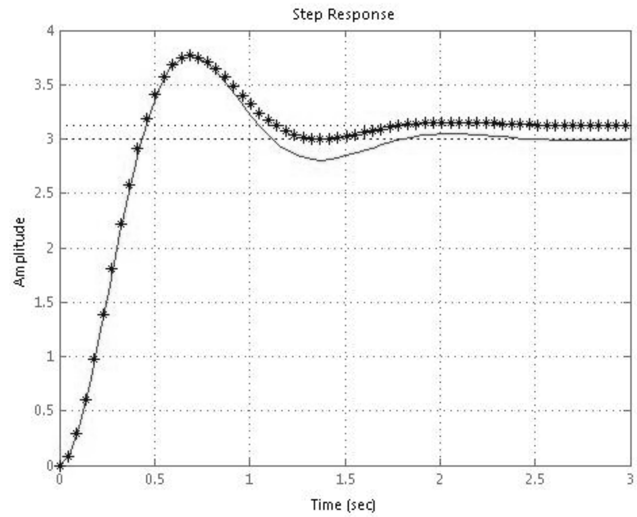
نکته بسیار مم در اینجا این است که برای مقایسه صحیح دو روش باید یک معیار مشترک در نظر گرفته شود که بهترین معیار زمان نمونه برداری (Sampling Time) می باشد. حال به مقایسه پاسخهای دو روش و خطای حاصله، به ازای زمان های نمونه برداری مختلف پرداخته می شود.



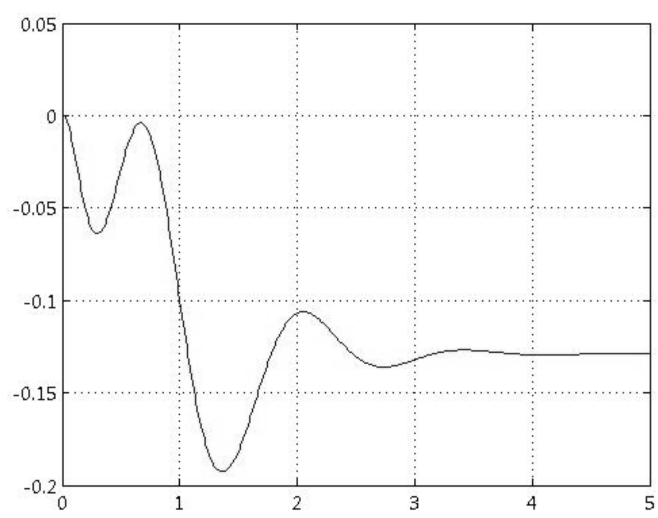
شکل ۶-۲۸ پاسخ سیستم واقعی و سیستم شناسایي شده به کمک روش RLS به ازای N=۵۰
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسایي شده)



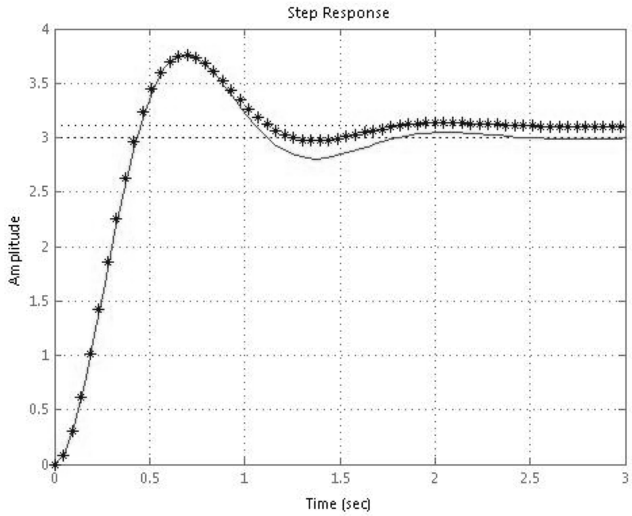
شکل ۶-۲۹ خطای سیستم شناسایي شده به کمک روش RLS به ازای N=۵۰



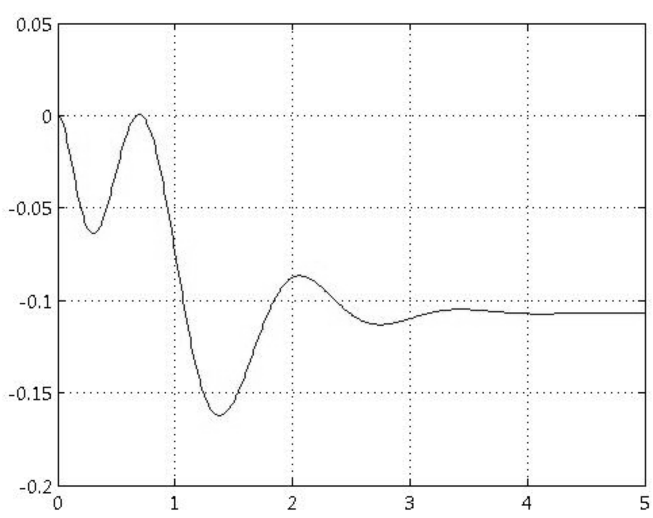
شکل ۶-۳۰ پاسخ سیستم واقعی و سیستم شناسايي شده به کمک روش GA به ازای N=۵۰
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسايي شده)



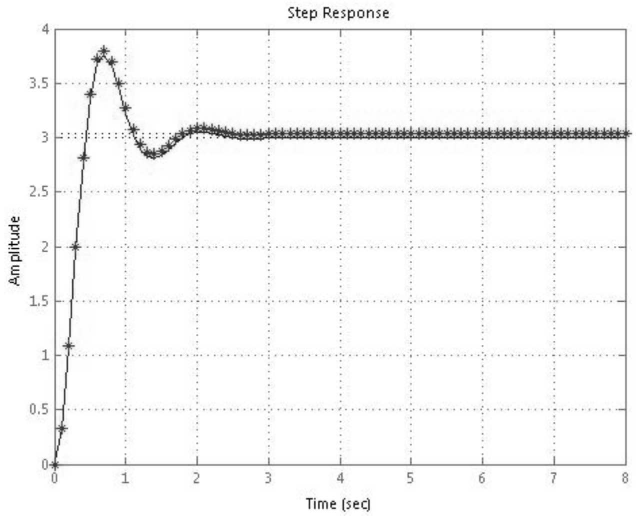
شکل ۶-۳۱ خطای سیستم شناسايي شده به کمک روش GA به ازای N=۵۰



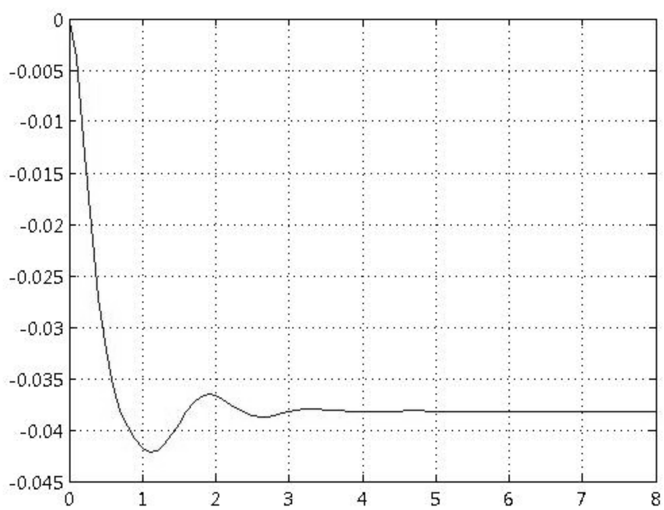
شکل ۶-۳۲ پاسخ سیستم واقعی و سیستم شناسايي شده به کمک روش RLS به ازای N=۷۵
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسايي شده)



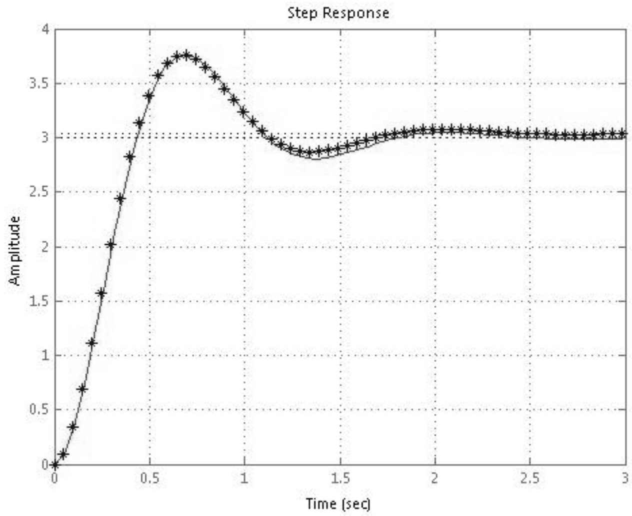
شکل ۶-۳۳ خطای سیستم شناسايي شده به کمک روش RLS به ازای N=۷۵



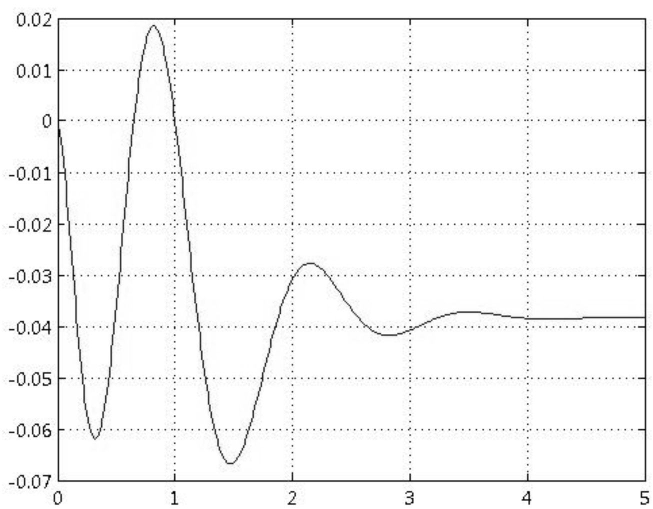
شکل ۶-۳۴ پاسخ سیستم واقعی و سیستم شناسايي شده به کمک روش GA به ازای N=۷۵
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسايي شده)



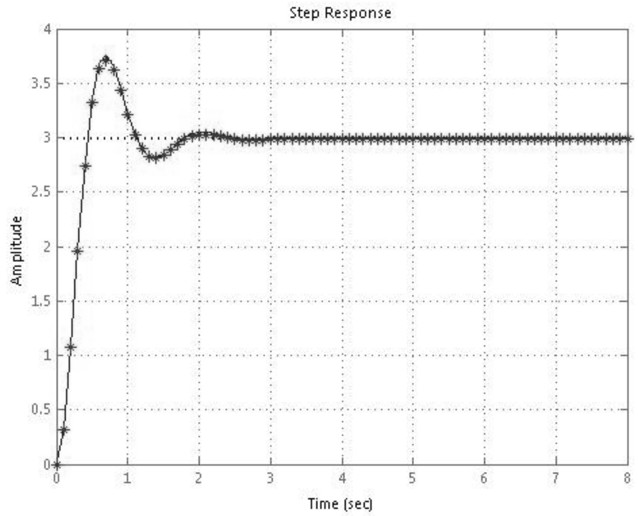
شکل ۶-۳۵ خطای سیستم شناسايي شده به کمک روش GA به ازای N=۷۵



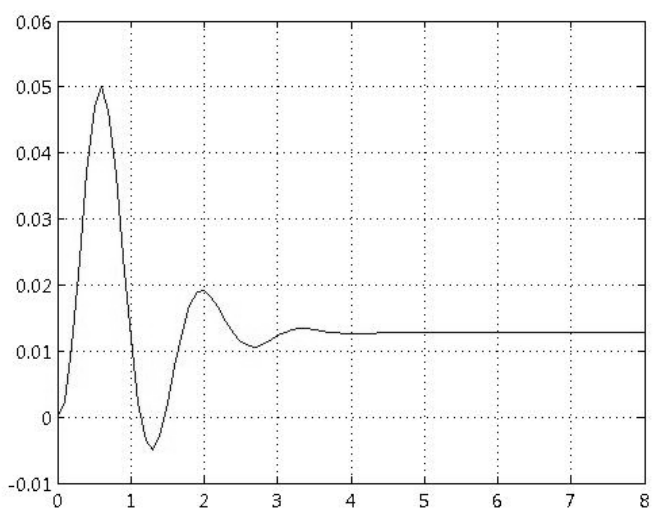
شکل ۶-۳۶ پاسخ سیستم واقعی و سیستم شناسايي شده به کمک روش RLS به ازای N=۱۰۰
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسايي شده)



شکل ۶-۳۷ خطای سیستم شناسايي شده به کمک روش RLS به ازای N=۱۰۰



شکل ۶-۳۸ پاسخ سیستم واقعی و سیستم شناسايي شده به کمک روش GA به ازای N=۱۰۰
(- : پاسخ سیستم واقعی ; * : پاسخ سیستم شناسايي شده)



شکل ۶-۳۹ خطای سیستم شناسايي شده به کمک روش GA به ازای N=۱۰۰

پارامترهای شناسايي شده			تعداد نقاط نمونه برداری	روش شناسايي
ω_n	η	k		
۴,۸۱	۰,۴۵۱	۳,۷۳	۵۰	RLS
۵,۱۰	۳,۹۱	۴,۱۶		GA
۴,۸۹	۰,۴۲۱	۴,۱۳	۷۵	RLS
۴,۹۶	۰,۴۰۵	۴,۰۸		GA
۵,۰۷	۰,۴۳۱	۴,۰۹	۱۰۰	RLS
۵,۰۱	۰,۳۹۹	۴,۰۰۲		GA

جدول ۶-۱۴ مقایسه پارامترهای شناسايي شده به دو روش GA و RLS

خطای محاسبه شده در این اشکال به صورت زیر می باشد:

e = (پاسخ پله سیستم واقعی) - (پاسخ پله سیستم شناسايي شده) (۶-۸)

از مقایسه اشکال ۶-۲۸ تا ۶-۳۹ مشخص است که روش الگوریتم ژنتیکی دارای پاسخ های بهتری نسبت به روش RLS است.

این تفاوت زمانی محسوس تر است که تعداد نقاط نمونه برداری کم باشد. در این صورت پاسخ روش RLS دارای خطای زیادی است در صورتیکه روش GA خطای کمتر و قابل قبولی دارد. یعنی در یک آزمایش عملی، هنگامیکه اطلاعات جمع آوری شده کمی در اختیار باشد، روش الگوریتمهای ژنتیکی

پاسخهای بسیار مناسب تری را نسبت به روش RLS بدست خواهد آورد. مخصوصا شناسايي پارامتر η که تاثیر زیادی روی پاسخ سیستم دارد، توسط روش GA با دقت بیشتری انجام شده است.

فصل هفتم: نتیجه گیری

۷-۱- نتیجه گیری

همان طور که مشاهده شد، در این پروژه ابتدا به شناسايي پارامترهای دینامیکی و الکتریکی موتور پرداخته شد و سپس یک کنترل کننده PID برای کنترل سرعت موتور طراحی شد.

با توجه به توانايي بالای الگوریتم ژنتیکی در یافتن نقاط بهینه مطلق توابع، این الگوریتم یک راه بسیار مناسب برای شناسایی پارامترها یک سیستم می باشد. در دامه به مقایسه این روش با روش RLS پرداخته شد. همان طور که ملاحظه شد روش الگوریتم ژنتیکی دارای پاسخهای مناسبتر وخطای کمتری در مقایسه با روش RLS بود.این تفوات هنگامی که تعداد اطلاعات (یا زمان نمونه برداری) کمتر بود، با وضوح بیشتری مشخص بود. از مزایای دیگر روش الگوریتم ژنتیکی در مقایسه با روش RLS این است که الگوریتم ژنتیکی می تواند برای سیستم های غیر خطی نیز مورد استفاده قرار بگیرد و در یافتن پاسخ های بهینه موفق باشد.

۷-۲- پیشنهادات و فعالیتهای آینده

در مورد فعالیتهايي که در ادامه موضوع این پروژه انجام داد، می توان به موارد زیر اشاره نمود:

- اجرای سیستم به صورت عملی

یکی از فعالیتهايي که می توان در ادامه راه این پروژه انجام داد پیاده سازی این طراحی به صورت عملی است. برای این امر در ابتدا باید به جمع آوری اطلاعات مورد نیاز ، یعنی ورودی و خروجی

موتور DC پرداخت. پس از جمع آوری اطلاعات و انتقال آن به رایانه، در نرم افزار MATLAB به پردازش اطلاعات پرداخته می شود و با توجه به اطلاعات موجود، شناسايي پارامترها انجام می شود.

- طراحی با روشهای هوشمند دیگر و مقایسه با این روش

همانطور که ئر این پایان نامه ملاحظه شد، این طراحی به کمک روش الگوریتم ژنتیکی انجام شد. در ادامه موضوع این پروژه می توان این طراحی را به کمک روش های هوشمند دیگری همچون شبکه های عصبی[1]، الگوریتم بهینه سازی اجتماع پرندگان[2] و همچنین الگوریتم بهینه سازی کلونی مورچگان[3] انجام داد و نتیج حاصله را با روش الگوریتم ژنتیکی مقایسه نمود و مزایا و معایب هر روش را ذکر کرد.

- شناسايي پارامترهای موتور **AC**

مطلب دیگری که می توان در ادامه این پایان نامه ذکر نمود، پیاده سازی این سیستم روی موتور AC می باشد.یعنی در ابتدا به شناسايي پارامترهای موتور AC پرداخته شود و سپس یک کنترلر برای آن طراحی شود.

---

[1] Neural networks
[2] Particle swarm optimization (PSO)
[3] Ant colony optimization

پیوست

پیوست الف) الگوریتم ژنتیکی هیبرید

اصول کار الگوریتم ژنتیکی هیبرید بر پایه شیوه مقداردهی اولیه نسل ها و تولید فرزندان بهتر به کمک پارامتر بازترکیب می باشد.

روش مقداردهی اولیه آن بدین صورت است که بخشی از مقادیر اولیه برای کروموزومها را انتخاب می کند و باقی ژنها را به صورت تصادفی تولید می کند. در ادامه تابع معیار برای این مقادیر اولیه محاسبه می شود. سپس به صورت تصادفی دو عدد از والدین انتخاب شده و از بازترکیب بین آنها فرزندان تولید می شوند. مجددا تابع معیار برای فرزندان به وجود آمده نیز محاسبه می شود. اگر تابع هزینه محاسبه شده توسط فرزندان کمتر از تابع هزینه والدین باشد، والدین ضعیفتر (با تابه هزینه بیشتر) جای خود را به فرزندان قویتر داده و الگوریتم تکرار می شود. این حلقه تا زمان رسیدن به جواب بهینه ادامه خواهد داشت.

پیوست ب) شناسايي پارامترهای موتور **DC** به کمک روش **LSE**[1]

همانطور که ذکر شد، معادله ۵-۷ برای تمامی مقادیر n برقرار است. می توان با نمونه برداری و جمع آوری اطلاعات $v(t),\theta(t),i(t)$ ، مقادیر R_W, R_{w_y} را محاسبه نموده و در نهایت مقدار k را تخمین زد:

$$\hat{k} = R_W^{-1} R_{w_y} \quad \text{(ب-۱)}$$

مساله مهم این است که چگونه $\hat{k} = R_W^{-1} R_{w_y}$ رابطه $W(nT)K = y(nT)$ را برای تمامی مقادیر n برقرار می سازد؟ برای پاسخ به این مساله بدین صورت عمل می شود:

$$W(nT) \triangleq \begin{bmatrix} di(nT)/dt & i(nT) & \omega(nT) & 0 & 0 \\ 0 & 0 & -i(nT) & d\omega(nT)/dt & \omega(nT) \end{bmatrix} \in R^{2\times 5}$$

$$y(nT) \triangleq \begin{bmatrix} v(nT) \\ 0 \end{bmatrix} \in R^2$$

(ب-۲)

و خطا را بدین صورت تعریف می کنیم:

$$e(nT) \triangleq y(nT) - W(nT)K \in R^2 \quad \text{(ب-۳)}$$

در این حالت باید مقداری برای k یافته شود که خطا را به ازای تمامي مقادیر n به کمترین مقدار خود برساند. در نهایت هدف اصلی، یافتن مقداری برای k است که مربعات خطا را مینیمم کند:

[1] Least Square Error

$$E^2(K) \triangleq \sum_{n=1}^{N} (y(nT)-W(nT)K)^T (y(nT)-W(nT)K)$$

$$= \sum_{n=1}^{N} (y(nT)-\hat{y}(nT))^T (y(nT)-\hat{y}(nT))$$

$$= \sum_{n=1}^{N} (y_1(nT)-\hat{y}_1(nT))^2 + (y_2(nT)-\hat{y}_2(nT))^2 \qquad \text{(ب-۴)}$$

$$= \sum_{n=1}^{N} (e_1(nT)^2 + e_2(nT))^2$$

که در آن:

$$e_1(nT) \triangleq y_1(nT) - \hat{y}_1(nT) \quad , \quad e_2(nT) \triangleq y_2(nT) - \hat{y}_2(nT) \qquad \text{(ب-۵)}$$

در شناسايي سيستم ها، $y(nT)$ را خروجی سیستم می نامند هنگامی که مقدار خروجی تخمین زده شده ($\hat{y}(nT) = W(nT)K$) بر اساس پارامترهای تخمین زده شده k بدست آمده باشد. در نتیجه:

$$e(nT) \triangleq y(nT) - W(nT)K = y(nT) - \hat{y}(nT) \in R^2 \qquad \text{(ب-٦)}$$

مقدار خطا بوده و

$$E^2(K) \triangleq \sum_{n=1}^{N} (y(nT)-W(nT)K)^T (y(nT)-W(nT)K) \qquad \text{(ب-۷)}$$

مقدار کل مربع خطا می باشد.

حال نشان داده می شود که برای حل این مساله یک جواب وجود دارد و آن $R_w^{-1} R_{w_y}$ می باشد.

بدین منظور معادله ب-۷ بسط داده می شود: [به صورت $(AB)^T = B^T A^T$]

$$E^2(K) = \sum_{n=1}^{N} ({}_y T(nT) y(nT) - y^T(nT) W(nT) K - K^T W^T(nT) y(nT)$$

$$+ K^T W^T(nT) W(nT) K)$$

(ب-۸)

$$= \sum_{n=1}^{N} y^T(nT) y(nT) - \left(\sum_{n=1}^{N} y^T(nT) W(nT) \right) K$$

$$- K^T \left(\sum_{n=1}^{N} W^T(e_1(nT) \right) + K^T \left(\sum_{n=1}^{N} W^T(nT) W(nT) \right) K$$

با استفاده از معادلات ۵-۱۴ و ۵-۱۵ خواهیم داشت:

$$Ryw \triangleq \sum_{n=1}^{N} y^T(nT) W(nT) \in R^{1\times 5}$$

(ب-۹)

$$Ry \triangleq \sum_{n=1}^{N} y^T(nT) y(nT) \in R$$

که در آن رابطه $Ryw = R^T wy$ برقرار است. در نهایت مقدار $E^2(k)$ بدین صورت بازنویسی می شود:

$$E^2(K) = R_y - R_{yw} K - K^T R_{wy} + K^T R_w K$$

$$= Ry - R_{yw} R_w^{-1} R_{wy} + (K - {}_w^{-1} R_{wy})^T R_w (K - R_w^{-1} R_{wy}) \qquad \text{(ب-۱۰)}$$

برای مقادیر ثابت N، مقادیر R_y, R_{yw}, R_{wy}, R_w ثابت بوده و وابسته به k نمی باشند. [3]

منابع و مراجع

[1] Randy L.Haupt,Sue Ellen Haupt, John Wiley and Sons, INC. publication, 2004, "*Practical genetic algorithms*"

[2] Melanie Mitchell*,* "*An Introduction to Genetic Algorithms*", Cambridge, Mass., MIT Press, ©1996

[3] John Chiasson, "*Modeling and high performance control of electric machines*", John Wiley and Sons, INC. publication

[4] Prof. P.S. BimBhra, "*Electrical machinery, Theory, Performance and Application*"

[5] Kirkpatrick, Beverlu L. Chapman, Stephan J, "*Electric machinery fundamental*", 3rd edition

[6] Benjamin C. Kuo, "*Automatic Control System*“, Prentice Hall, Sixth edition, 1994

[7] Katsuhiko Ogata, "*Control Engineering*“, Moderon control engineering, 4th edition

[8] Richard C. Dorf, Robert H Bishop, "*Modern Control Systems*“, Prentice Hall, 9th edition, 2001

[9] Lennart Ljung, Torsten Soderstrom, *"Theory and practice of recursive Identification"*, The MIT press Cambridge, Massachusetts, London, England

[10] Qi-Wen Yang Jing-Ping Jiang Guo Chen, Coll. of Electr. Eng., Zhejiang Univ. "*How to select optimal control parameters for genetic algorithms*"**,** Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000 IEEE International Symposium

[11] Zhang Zibo; Naghdy, F**,** "*Application of genetic algorithms to system identification*", Evolutionary Computation, 1995, IEEE International Conference

[12] Kristinsson, K., "*System identification and control using genetic algorithms*"; Dumont, G.A. Systems, Man and Cybernetics, IEEE Transactions

[13] Wassim A. Bedwani and Ossama M. Ismail, "*Genetic Optimization of Variable Structure PID control Systems*", AICCSA, Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications

[14] Krohling, R.A. *"Design of optimal disturbance rejection PID controllers using genetic algorithms*"; Rey, J.P. Evolutionary Computation, IEEE Transactions

[15] Qing Wang; Spronck, P**, "***An overview of genetic algorithms applied to control engineering problems*" ; Tracht, R.Machine Learning and Cybernetics, 2003 International Conference

[16] K.F. Man, K.S. Tang and S. Kwong, "*Genetic algorithms: concepts and applications,*" IEEE Transactions on Industrial Electronics, vol.43 no.5, pp.519-534, Oct 1996

[17] Songwu Lu Basar,T. "*Genetic algorithms-based identification"*, Decision & Control Lab., Illinois Univ., Urbana, IL, Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference

[18] Stender, J., "*Introduction to genetic algorithms*", Applications of Genetic Algorithms, IEE Colloquium

[19] Wang Minle Gao Xiaoguang, "*The optimization of genetic algorithm control parameters*", Second Artillery Eng. Inst., Xi'an, Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress

Abstract

Nowaday scientists use new technologies named "Intelligent systems" for designing industrial systems. These methods use intelligent patterns for their purpose. There are two different parts in this thesis.

In the first one considers the application of Genetic Algorithm (GA) Optimization to estimate the parameters of dynamical and electrical state of a DC Motor. The problem is organized in 2 sections. First problem is about dynamical treatment of a DC motor. After consideration the parametric model of a DC motor, 3 unknown parameters will be identified by assistance of collecting data and using proposed method. The second problem is about estimation of electrical parameters of a DC motor. There are only 2 equations describing the electrical treatment of a DC motor and it contains 5 unknown parameters. Least square estimation (LSE) is considered as a conventional method for parameter estimation, in comparison with GA method. In parameter identification using GA method, free noise system is considered and the necessity of inputs with persistent excitation has been omitted and parameters will be obtained with any input. Finally comparison between LSE and GA optimization is presented to indicate robustness, resolution, accuracy and quicker response of GA identification method in parameter estimation.

The second part presents GA for determining the optimal proportional-integral-derivative (PID) controller parameters, for speed control of a DC motor. The proposed approach has superior features, including easy implementation, stable convergence characteristic and good computational efficiency. The DC motor is modeled in Simulink and the GA algorithm is implemented in MATLAB. The GA method is efficient in improving the step response characteristics such as, reducing the steady-states error; rise time, settling time and maximum overshoot in speed control of a DC motor.

Key words: *Parameter identification, DC motor, PID controller, Genetic Algorithm*

SHAHROOD UNIVERSITY
OF TECHNOLOGY

Thesis submitted for
Degree of Master of Science in Control Engineering

Title

Parameter Identification of a DC Motor and Design a PID Controller Using Genetic Algorithm

Supervisor
Dr. M.H.Zarif

Student
Arash Maboud

Summer 2008