

یک تحلیل مقایسه‌ای از الگوریتم‌های هوش جمعی کلونی زنبور مصنوعی و بهینه‌سازی گروهی ذرات در طراحی یک کنترل کننده PID فازی کسری و پیاده‌سازی آن بر روی موتور DC

روح‌الله مقصودی^۱، یعقوب حیدری^{۲*}، بهزاد مشیری^۳

اطلاعات مقاله	چکیده
<p>واژگان کلیدی: کنترل کننده PID فازی کسری، کلونی زنبور مصنوعی، بهینه‌سازی گروهی ذرات، قوام.</p>	<p>پیاده‌سازی یک کنترل کننده PID فازی کسری بر روی یک موتور DC در این مقاله مورد مطالعه و تحقیق قرار گرفته است. یک کنترل کننده PID فازی مرتبه کسری یک کنترل کننده PID است که دارای ضرایب مشتقی و انتگرالی غیرصحیح است که برای مواجهه با عدم قطعیت‌ها به صورت فازی هم عمل می‌کند. استراتژی طراحی این کنترل کننده شامل تعیین پنج ضریب می‌باشد. این مقاله از الگوریتم‌های هوش جمعی کلونی زنبور مصنوعی و بهینه‌سازی گروهی ذرات برای طراحی ضرایب کنترل کننده مذکور استفاده می‌نماید. الگوریتم کلونی زنبور مصنوعی در طراحی کنترل کننده فازی کسری ارائه شده در مقایسه با روش طراحی براساس بهینه‌سازی گروهی ذرات و همچنین PID معمولی در مواجهه با عدم قطعیت‌ها قوام سیستم را به میزان قابل توجهی بهبود بخشیده است.</p>

۱- مقدمه

تئوری فازی). هر یک از این روش‌ها مزایا و معایب خاص خود را دارند. از معایب روش‌های کلاسیک می‌توان به عدم پاسخگو بودن آن نسبت به سیستم‌هایی که پارامترهای تحت کنترل متغیر دارند را نام برد. با به کار بردن روش‌های هوشمند، مزایای عمده‌ای همچون پاسخ دینامیکی سریع، عدم حساسیت به پارامترهای سیستم و پایداری مناسب سیستم در حالت ماندگار را می‌توان در نظر داشت. یکی دیگر از نقاط قوت کنترل کننده‌های هوشمند، پیاده‌سازی آن‌ها در فرآیندهایی است که مدل ریاضی آن‌ها کاملاً شناخته شده نیست. محاسبات کسری نقش اساسی در تئوری کنترل ایفا می‌کند و امکانات بهتری در اختیار قرار می‌دهد. در مرجع [۱] مفاهیم اساسی که در استراتژی‌های کنترلی استفاده می‌شوند، توسط نویسندگان بیان می‌گردند. مقالات [۱ و ۲] کاربرد

دامنه وسیع سرعت و سهولت کنترل موتورهای DC باعث کاربرد گسترده آن در صنعت شده است. کنترل این نوع از موتورها عمدتاً با کنترل جریان آرمیچر و یا کنترل میدان انجام می‌پذیرد. روش‌های ارائه شده برای کنترل موقعیت موتورهای DC به طور کلی به سه دسته تقسیم می‌شوند: (۱) روش‌های کلاسیک مثل استفاده از کنترل کننده‌های PI و PID. (۲) روش‌های مدرن (تطبیقی و بهینه). (۳) روش‌های هوشمند (نظیر الگوریتم‌های هوش جمعی و

* پست الکترونیک نویسنده مسئول: masudheidari@yahoo.com

۱. مربی گروه کامپیوتر، دانشگاه آزاد اسلامی واحد محمودآباد، محمودآباد،

ایران

۲. مربی گروه برق، دانشگاه آزاد اسلامی واحد محمودآباد، محمودآباد، ایران

۳. استاد، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، عضو ارشد

بررسی و لزوم به‌کارگیری الگوریتم‌های PSO و ABC پرداخته می‌شود. طراحی کنترل‌کننده فازی کسری به‌کمک الگوریتم‌های مذکور برای یک موتور DC در بخش ۴ تشریح شده است. بخش ۵، شامل نتایج حاصل از پیاده‌سازی عملی کنترل‌کننده و مقایسه آن با سایر کنترل‌کننده‌های اعمال شده است. بخش ۶ هم نتیجه‌گیری مقاله خواهد بود.

۲- محاسبات کسری

محاسبات مرتبه کسری توسط لیبنیز^۳ در سال ۱۶۹۵ معرفی شد. مفهوم FOC^۴ یا کنترل‌کننده مرتبه کسری به معنی سیستم‌ها یا کنترل‌کننده‌هایی است که توسط معادلات مشتقی مرتبه کسری توصیف می‌شوند. انتگرال و مشتق مرتبه کسری موضوع جدیدی نیست. لیبنیز حدود ۳۰۰ سال پیش آن را به L.Hospital ذکر کرده بود. در سال‌های اخیر، محققان نشان دادند که معادلات مرتبه کسری می‌توانند پدیده‌های گوناگون را مناسب‌تر از مرتبه صحیح آن مدل کنند و یک ابزار عالی برای توضیح ساختارهای با دینامیک پیچیده فراهم کنند. یکی از دلایل عمده استفاده از کنترل‌کننده‌های مرتبه کسری در مهندسی کنترل مقاوم بودن بیشتر آن‌ها نسبت به کنترل‌کننده‌های مرتبه صحیح می‌باشد. به طوری که کسری بودن دینامیک کنترل‌کننده، باعث افزایش درجه آزادی در انتخاب پارامترهای طراحی می‌شود و این امکان را به طراح می‌دهد که با توجه به کسری بودن دینامیک کنترل‌کننده به پاسخ‌هایی با رفتار مناسب‌تر و نرم‌تر دست یابد. به عبارتی با استفاده از کنترل‌کننده‌های مرتبه کسری افزایش کارایی و عملکرد کنترل‌کننده محتمل‌تر خواهد بود [۸]. حسابان کسری دارای گستره‌ی بیشتری نسبت به نوع صحیح آن است. اگر به‌جای مرتبه‌ی صحیح مشتق یا انتگرال، از مرتبه کسری آن استفاده شود باید از حسابان کسری برای حل مشتق و انتگرال کسری استفاده

محاسبات کسری در سیستم‌های کنترلی و مفاهیم کنترل‌کننده‌ی PID کسری را بیان می‌کنند. برای طراحی ضرایب PID روش‌های کلاسیکی مانند زیگلر-نیکولز و تابو وجود دارد [۳]، اما برای تنظیم ضرایب PID کسری الگوریتم‌های مشخصی وجود ندارد. در مرجع [۴] برای طراحی کنترل‌کننده‌ی PID کسری که برای کنترل یک تنظیم‌کننده‌ی ولتاژ اتوماتیک (AVR) به‌کار می‌رفت، از الگوریتم تکاملی^۱ PSO استفاده شده است. برای کنترل این تنظیم‌کننده از الگوریتم زنبور مصنوعی استفاده شده و نتایج با الگوریتم‌های هوشمند دیگر نظیر PSO و DE^۲ مقایسه شده است [۵]. از دیگر موارد کاربرد الگوریتم‌های تکاملی، استفاده از الگوریتم بهینه‌سازی ذرات است که در طراحی یک PID کسری برای کنترل مسیر یک مدل پروازی (موشک) با ۶ درجه آزادی استفاده می‌شد [۶]. به‌طور کل دلیل استفاده از کنترل اتوماتیک را در پیچیدگی پلنت و خروجی‌هایی با کیفیت دقیق دانست. الگوریتم تدریجی دیگری که برای بهینه کردن ضرایب کنترل‌کننده PID کسری استفاده می‌شود، روش بهینه‌سازی گروهی مورچه آشوبی (CAS) است. این الگوریتم، روشی جدید براساس رفتار آشوبی یک فرد و خودسازمانده‌ی جمعی افراد می‌باشد که در [۷] برای کنترل AVR استفاده شده است.

در مقاله حاضر یک کنترل‌کننده PID فازی کسری که پارامترهای آن براساس الگوریتم کلونی زنبور مصنوعی به‌دست آمده‌اند، برای کنترل موقعیت یک موتور DC طراحی شده است. کیفیت مطلوب این کنترل‌کننده به‌صورت تجربی در آزمایشگاه در مقایسه با کنترل‌کننده PID کلاسیک، PID کسری و هم‌چنین PID فازی کسری که پارامترهای آن براساس سایر الگوریتم‌های هوش جمعی نظیر ABC طراحی شده‌اند، بررسی شده است. این مقاله به‌صورت زیر دسته‌بندی شده است. بخش ۲ مروری بر محاسبات کسری خواهد بود. در بخش ۳ به

³ - Leibniz

⁴ - Fractional Order Controller

¹ - Particle Swarm Optimization

² - Differential Equation

روش‌ها به دو دسته الگوریتم‌های تکاملی و یا هوش جمعی تقسیم‌بندی می‌شوند که شامل الگوریتم‌های تکاملی، ژنتیک، الگوریتم بهینه‌سازی ذرات، الگوریتم بهینه‌سازی کلونی مورچه و اخیراً کلونی زنبور مصنوعی می‌باشند. در پژوهش حاضر از الگوریتم‌های PSO و ABC استفاده شده است که به‌طور اجمال به توضیح آن‌ها می‌پردازیم.

۳-۱- الگوریتم بهینه‌سازی گروهی ذرات

PSO الگوریتمی ساده، مقاوم^۴ و آسان جهت پیاده‌سازی است. در PSO یک مجموعه از ذرات در گروه تعریف می‌شوند. در PSO، یک جمعیتی از ذرات در فضای جستجو شروع به حرکت می‌کند. با تعقیب ذرات بهینه جاری و تغییر موقعیت‌های ذرات به‌منظور کشف نقطه بهینه سراسری، هر ذره به‌وسیله موقعیت^۵ و سرعتش^۶ توصیف می‌شود و یک پاسخ بالقوه در فضای جستجو تشکیل می‌دهد. موقعیت یک ذره در واقع یک پاسخ ممکن از تابعی است که قرار است بهینه شود. ارزیابی یک تابع به‌وسیله موقعیت ذره، شایستگی یک پاسخ را فراهم می‌آورد. تعداد پارامترهایی که باید بهینه شوند، ابعاد^۷ مسئله را تعیین می‌کنند. به‌عنوان مثال در مورد بررسی ما هر الگوی مورد جستجو هشت بعد دارد. هر ذره براساس تابع تطبیقی که قرار است بهینه شود، ارزیابی می‌شود. در ابتدا، همه ذرات در یک فضای جستجوی D-بعدی به‌صورت تصادفی در یک بازه از پیش تعریف شده مقداردهی اولیه می‌شوند.

گروه در PSO با تخصیص یک موقعیت تصادفی و غیرآگاهانه به هر ذره در فضای جستجو مقداردهی اولیه می‌شود. موقعیت و سرعت ذره \mathbf{p}_i (i=1,2,...,NP) در بعد Dام به ترتیب به‌صورت $\mathbf{X}_i=(x_{i1},x_{i2},\dots,x_{iD})$ و $\mathbf{V}_i=(v_{i1},v_{i2},\dots,v_{iD})$ نشان داده می‌شود. هر ذره موقعیت و سرعت خود را براساس بهترین موقعیت خودش

شود. اپراتور مشتق‌گیر-انتگرال‌گیر به‌وسیله ${}_a D_t^\alpha$ نشان داده می‌شود. این عملگر نمادی است که برای گرفتن مشتق و انتگرال کسری به‌کار می‌رود. برای مقادیر مثبت α نماد مشتق و برای مقادیر منفی α نماد انتگرال است. تعاریفی که معمولاً برای مشتق کسری به‌کار می‌رود گرانوالد-لتنیکوف^۱، ریمان-لیویل^۲ و کاپوتو^۳ می‌باشند و به‌صورت زیر بیان می‌گردند [۲].

تعریف گرانوالد-لتنیکوف به‌صورت زیر می‌باشد.

(۱)

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{[t-\alpha/h]} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (1)$$

تعریف دوم، تعریف ریمان-لیویل است که به‌عنوان ساده‌ترین و راحت‌ترین تعریف مورد استفاده قرار می‌گیرد و به‌صورت زیر تعریف می‌شود:

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d}{dt} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (2)$$

برای شرایط اولیه صفر تبدیل لاپلاس آن به فرم زیر می‌باشد:

$$L\{{}_0 D_t^\alpha f(t)\} = s^\alpha F(s) \quad (3)$$

۳- الگوریتم‌های هوش جمعی

برای اعمال کنترل‌کننده PID و PID کسری، باید ضرایب آن‌ها را براساس یکی از الگوریتم‌های بهینه‌سازی محاسبه نماییم. از آنجاکه روش‌های کلاسیک تنظیم ضرایب کنترل‌کننده همانند زیگلر-نیکولز برای نوع کسری آن مناسب نمی‌باشد، به سراغ روش‌های هوشمند در این زمینه می‌رویم. در این تحقیق، از روش تخمین بهینه‌ی هوشمند برای محاسبه‌ی ضرایب استفاده شده است. این

⁴ - Robust

⁵ - Position

⁶ - Velocity

⁷ - Dimension

³ - Grunwald-Letnikov

⁴ - Riemann-Liouville

³ - Caputo

۱. ایجاد و مقداردهی جمعیت اولیه.
۲. آغاز حلقه.
۳. محاسبه میزان شایستگی ذرات.
۴. محاسبه بهترین ذرات در گروه.
۵. انتخاب بهترین ذره.
۶. محاسبه بردار سرعت ذرات.
۷. به روز کردن بردار موقعیت ذرات.
۸. تا زمانی که شرط توقف ارضا نشد به گام ۲ برو.

۳-۲- الگوریتم کلونی زنبور مصنوعی

زنبورهای کاوش‌گر غذا به سه دسته تقسیم می‌شوند: زنبورهای کارگر^۱، زنبورهای ناظر^۲ و زنبورهای پیش‌آهنگ^۳. تمامی زنبورهایی که در حال حاضر مشغول استخراج از یک منبع غذایی هستند، کارگر نام دارند. زنبورهای کارگر منبع غذایی را کشف می‌کنند و اطلاعات را در مورد آن منبع غذایی به کندو حمل می‌کنند و با زنبورهای ناظر به اشتراک می‌گذارند. زنبورهای ناظر در کندو منتظر اطلاعاتی می‌مانند که توسط زنبورهای کارگر در مورد یک منبع غذایی کشف شده به اشتراک گذاشته می‌شود و زنبورهای پیش‌آهنگ مشغول جستجو برای منابع غذایی جدید نزدیک کندو خواهند بود. زنبورهای کارگر اطلاعات در مورد منابع غذایی را با رقصیدن در منطقه معینی درون کندو تحت عنوان منطقه رقص تسهیم می‌کنند. مدت زمان این رقص چرخشی^۴ متناسب با محتوای شهد یک منبع غذایی است که در حال حاضر توسط زنبورهای رقص استخراج می‌شود. زنبورهای ناظر رقص را تماشا می‌کنند و یک منبع غذایی بر طبق تناسب احتمالی با کیفیت آن منبع غذایی انتخاب می‌کنند. هرگاه یک زنبور ناظر و یا پیش‌آهنگ منبع غذایی را انتخاب کرد، تبدیل به یک زنبور کارگر می‌شود. در نتیجه منابع غذایی خوب زنبورهای ناظر بیشتری را در مقایسه با منبع بد به خود جذب می‌کنند. هرگاه یک منبع غذا به طور کامل استخراج شد، همه زنبورهای

(Pbest) یعنی بهترین پاسخ ذره جاری که تا لحظه فعلی به دست آمده و بهترین موقعیت کل گروه (gbest) به روز می‌کند.

در حالی که در فرآیند جستجو، هر ذره بهترین پاسخ قبلی‌اش را به خاطر می‌آورد، و اگر پاسخ فعلی از بهترین پاسخ قبلی بهتر باشد، موقعیت مرتبط با پاسخ فعلی Pbest می‌شود و در غیراین صورت بدون تغییر باقی می‌ماند. موقعیت ذره‌ای که به یک موقعیت جدید نقل مکان داده می‌شود، توسط سرعتی که در هر گام زمانی t به روز می‌شود، محاسبه می‌گردد. این موقعیت جدید به‌عنوان حاصل جمعی از موقعیت مکانی قبلی ذره و سرعت لحظه حاضر او می‌باشد.

$$X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^{t+1} \quad (4)$$

بهترین در میان همه پاسخ‌های Pbest یک پاسخ gbest است و موقعیت مرتبط با gbest پاسخ‌های بالقوه مسئله خواهند بود. سرعت هر ذره هم براساس رابطه (۵) به روز می‌شود:

(۵)

$$V_{i,d}^{t+1} = w V_{i,d}^t + c_1 \text{rand}_1 \times (pbest_{i,d}^t - x_{i,d}^t) + c_2 \text{rand}_2 \times (gbest_d^t - x_{i,d}^t)$$

که w بیان‌گر وزن اینرسی بوده و مقدار قبلی $V(t)$ در محاسبه سرعت جدید را کنترل می‌کند، c_1 و c_2 هم فاکتورهای یادگیری هستند که به ترتیب تاثیر نسبی مولفه‌های شناختی و اجتماعی را مشخص می‌کنند. در این مقاله w ، c_1 و c_2 ثابت و به ترتیب 0.5 ، 2 و 2 در نظر گرفته شده‌اند. Rand_1 و Rand_2 اعداد تصادفی مستقلی هستند که به‌طور یکنواخت در بازه $[0,1]$ توزیع شده‌اند. $V_{i,d}^t$ ، $X_{i,d}^t$ و $pbest_{i,d}^t$ به ترتیب سرعت، موقعیت و بهترین تجربه شخصی ذره نام در بعد d از تکرار t ام است. $gbest_d^t$ بهترین ذره بعد d ام در گروه برای تکرار t ام می‌باشد [۹]. برای الگوریتم PSO می‌توان شبه کدی به‌صورت زیر در نظر گرفت:

¹ - Employed

² - Onlooker

³ - Scout

⁴ - Waggle Dance

هنگامی که تمامی زنبورهای کارگر این فرآیند را تمام کردند، اطلاعات شایستگی‌شان را با زنبورهای ناظر به اشتراک می‌گذارند، هر کدام از ناظرها بر اساس معادله احتمالی که در بالا به آن اشاره شد یک منبع غذایی را انتخاب می‌کنند. با این رویه، منابع غذایی خوب، ناظرهای بیشتری را به نسبت منابع بد به خود جذب می‌کنند. پس از این‌که همه زنبورهای ناظر یک منبع غذایی برای خود پیدا کردند، هر یک به تعداد معینی سیکل در همسایگی‌اش به جستجو خواهد پرداخت (limit) و یک منبع غذایی پیدا کرده و میزان شایستگی آن را می‌سنجد. بهترین منبع غذایی مابین تمامی منابع غذایی که در همسایگی منبع آم قرار داشتند و خود منبع آم، به‌عنوان محل جدید منبع آم محسوب خواهد شد و اگر مقدار شایستگی بهبود نیافت آن‌گاه تبدیل به یک زنبور پیش‌آهنگ می‌شود و به‌طور کامل تصادفی به دنبال منبع غذایی جدیدی می‌گردد. این کار معادل با تخصیص منابع (پاسخ‌ها) جدید تصادفی به زنبورهای پیش‌آهنگ و تبدیل آن‌ها به کارگر خواهد بود. پس از این‌که محل جدید هر منبع غذایی تعیین شد، تکرار^۳ دیگر الگوریتم ABC آغاز می‌شود. تمامی این فرآیند تکرار و تکرار می‌شود تا شرط خاتمه ارضا گردد [۱۰-۱۳]. منبع غذایی در همسایگی یک منبع خاص با تغییر مقدار یکی از پارامتر (بعد‌های پاسخ فعلی و بدون تغییر گذاشتن پارامترهای دیگر حاصل می‌شود. فرض کنید هر پاسخ شامل d پارامتر باشد، $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$ برای تعیین یک پاسخ در نزدیکی X_i ، مثلاً X_i' ، یکی از پارامترهای پاسخ X_i به‌عنوان مثال $X_k = (X_{k1}, X_{k2}, \dots, X_{kd})$ به‌صورت تصادفی انتخاب می‌شوند. به‌جز پارامتر انتخاب شده X_k ، تمامی پارامترهای دیگر X_i' مشابه X_i هستند، $X_i' = (X_{i1}, X_{i2}, \dots, X_{i(j-1)}, X_{ij}, X_{i(j+1)}, \dots, X_{id})$ مقدار پارامتر X_k با استفاده از رابطه (۷) محاسبه می‌گردد.

$$X_{ij}' = X_{ij} + U(X_{ij} - X_{kj}) \quad (7)$$

که u یک متغیر تصادفی یکنواخت در بازه $[-1, 1]$ می‌باشد. اگر مقدار نتیجه از بازه قابل قبول برای پارامتر j فراتر رود، با حداکثر مقدار بازه تنظیم می‌شود [۱۳]. در شکل زیر مراحل الگوریتم کلونی زنبور مصنوعی به‌صورت فلوجارتی آمده است.

کارگر مرتبط با آن منبع، آن را ترک می‌کنند و ممکن است تبدیل به زنبورهای پیش‌آهنگ و یا ناظر شوند. زنبورهای پیش‌آهنگ می‌توانند به‌عنوان عاملین فرآیند اکتشاف^۱ تصور شوند، درحالی‌که زنبورهای ناظر و کارگر به‌عنوان عاملین فرآیند استخراج^۲ محسوب می‌شوند [۱۰-۱۲]. در الگوریتم ABC (Karaboga, Basturk, 2008) هر منبع غذایی یک پاسخ ممکن برای مسئله مورد نظر می‌باشد و مقدار شاهد موجود در هر منبع غذایی نشان‌دهنده کیفیت پاسخی است که به‌وسیله مقدار شایستگی نمایانده می‌شود. در ابتدا نیمی از زنبورها کارگر و نیمی هم ناظر در نظر گرفته می‌شوند. تعداد منابع غذایی مشابه تعداد زنبورهای کارگر است و دقیقاً به‌ازای هر منبع یک زنبور کارگر وجود دارد. زنبورهای کارگر ترک-کننده یک منبع غذایی تبدیل به پیش‌آهنگ و به محض یافتن یک منبع غذایی مجدداً کارگر می‌شوند. این الگوریتم با تخصیص منابع غذایی (پاسخ) (که به‌صورت تصادفی تولید شده‌اند) به زنبورهای کارگر، شروع می‌کند. در هر تکرار، هر زنبور کارگر یک منبع غذایی که در نزدیکی منبع جاری‌اش باشد تعیین کرده و مقدار شاهد (شایستگی) آن را ارزیابی می‌کند. موقعیت منبع غذایی آم به‌صورت $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$ نمایانده می‌شود. $F(X_i)$ اشاره به مقدار شاهد منبع غذایی که در X_i واقع گردیده، دارد. پس از مشاهده رقص زنبورهای کارگر، یک زنبور ناظر با احتمال P_i به منطقه منبع غذایی واقع در X_i می‌رود.

$$p_i = \frac{F(X_i)}{\sum_{k=1}^S F(X_k)} \quad (6)$$

که S تعداد منابع غذا یا تعداد زنبورهای کارگر یا تعداد پاسخ‌های کاندید است.

اگر مقدار شایستگی منبع جدید بهتر از مقدار بهترین شایستگی به‌دست آمده تاکنون (یعنی بهتر از شایستگی منبع قبلی) باشد، آن‌گاه زنبور به‌طرف آن منبع غذایی جدید حرکت می‌کند و قبلی را رها می‌کند، در غیراین‌صورت در منبع غذایی قبلی‌اش باقی می‌ماند.

¹ - Exploring

² - Exploit

³ - Iteration

موقعیت موتورهای DC را می‌توان به سه دسته تقسیم بندی کرد:

۱. روش‌های کلاسیک مثل استفاده از

کنترل‌کننده‌های PI و PID

۲. روش‌های مدرن مثل تطبیقی و بهینه

۳. روش‌های هوشمند مثل کاربرد تئوری فازی و

شبکه‌های عصبی (محاسبات نرم)

در کنترل موتور DC باید دنبال روشی باشیم که تا حد امکان به پارامترهای سیستم حساس نبوده و نسبت به تغییرات ممان اینرسی (J) و اصطکاک (B) پاسخ مقاومی داشته باشد. تلاش فراوانی برای کنترل محرک‌های DC موجود در صنعت انجام می‌شود ولی از آن‌جا که پارامترهای این سیستم در حین کار تغییر می‌کند از کنترل‌کننده PID هم در حالت کسری و هم در حالت فازی استفاده می‌کنیم تا با به خدمت گرفتن منطق فازی و کسری کردن کنترل‌کننده پاسخ بهینه جهت کنترل موقعیت دقیق و مقاوم موتور DC حاصل گردد.

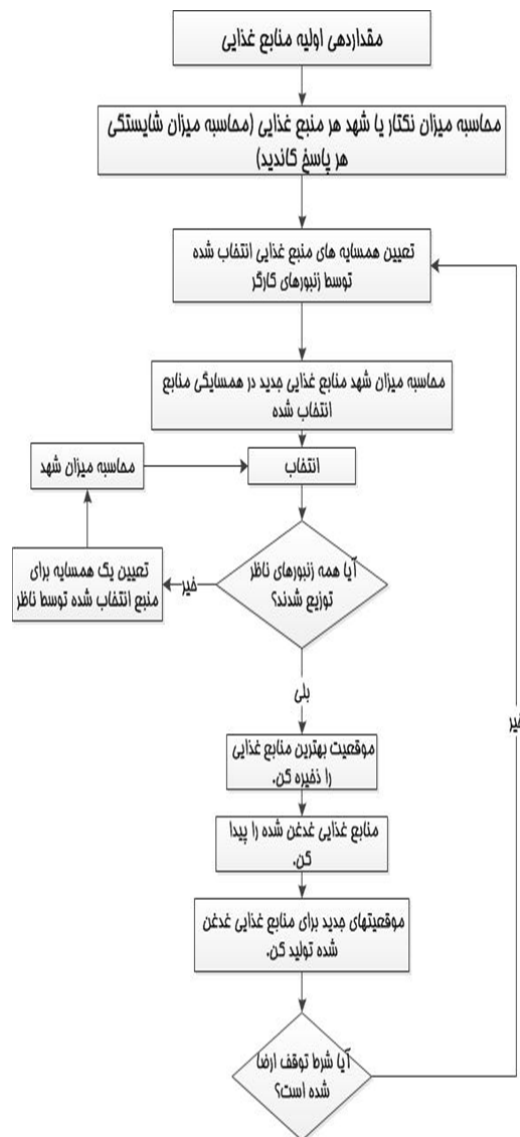
۴-۲- شناسایی مدل موتور DC

روش‌های کلاسیک مختلفی برای شناسایی پارامترهای موتور DC وجود دارد [۱۷]. هدف این تحقیق پیاده‌سازی عملی کنترل‌کننده‌های مرتبه کسری می‌باشد. بنابراین برای طراحی کنترل‌کننده یک سیستم سرو نیاز به یک مدل ریاضی از سیستم خواهد بود. در این قسمت برای اندازه‌گیری موقعیت موتور از یک تاکومتر که در محور آن قرار دارد استفاده شده است. مدل ریاضی شناسایی شده تابع انتقال موقعیت موتور به صورت یک معادله درجه اول:

$$G(s) = \frac{\Omega(s)}{u_r(s)} = \frac{k_m}{sT_m + 1} \quad (8)$$

می‌باشد. این معادله براساس روشی مبتنی بر پاسخ زمانی یک فرآیند شناسایی شده است که در آن $\Omega(s)$ موقعیت موتور، $u_r(s)$ ولتاژ سیم پیچ روتور موتور DC، k_m بهره و T_m ثابت زمانی موتور می‌باشد. k_m و T_m پارامترهای ناشناخته موتور بوده که هدف ما شناسایی این پارامترها می‌باشد.

روش ذکر شده به‌طور وسیعی در شناسایی رفتار دینامیکی سیستم‌ها چه در حالت بی باری و چه در حالت بار کامل مورد استفاده قرار می‌گیرد. بعضی از خصوصیات



شکل ۱- دیاگرام الگوریتم ABC [۱۷]

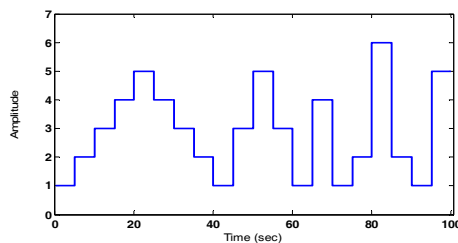
۴- طراحی کنترل‌کننده PID فازی کسری

۴-۱- مدل موتور DC

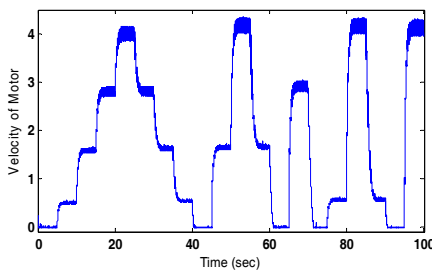
دامنه وسیع موقعیت و سهولت کنترل موتورهای DC باعث کاربرد فراوان آن‌ها در صنعت شده است. کنترل موتورهای DC عمدتاً با کنترل جریان آرمیچر و یا کنترل میدان صورت می‌گیرد. آسانی کنترل و ارزانی مدار درایو موتورهای DC نسبت به مدار درایو موتورهای AC موجب شده است که در بسیاری از کاربردهای تنظیم دقیق موقعیت، موتور DC انتخاب مصرف‌کنندگان و صنعت‌گران باشد. روش‌های ارائه شده برای کنترل

کارت اتوماسیون آدونتک HG1710L دارای ۱۶ ورودی آنالوگ $AI_0, AI_1, \dots, AI_{15}$ می‌باشد که زمین همه آن‌ها $A_{0GND}, A_{15GND}, \dots$ با همدیگر مشترک است و دارای دو خروجی DA_{0out} و DA_{1out} با زمین‌های مشترک A_{0GND} است. برای برقراری ارتباط بین موتور و کامپیوتر از خروجی کارت به ورودی درایور موتور وصل می‌کنیم و یکی از ورودی‌های کارت هم باید به خروجی تاکومتر متصل گردد تا فیدبک موقعیت برقرار شود.

در سیستم کنترل حلقه بسته موقعیت موتور، از موقعیت موتور به‌عنوان سیگنال فیدبک استفاده می‌شود. به این منظور از تاکومتر یا شفت انکودر که در شفت موتور نصب می‌شود، جهت دریافت سیگنال فیدبک موقعیت موتور استفاده می‌شود. موقعیت موتور DC توسط کنترل‌کننده‌های پیشنهادی و با استفاده از خطای بین فرمان ورودی موقعیت و سیگنال فیدبک موقعیت تنظیم خواهد شد. در ابتدا با توجه به پاسخ حلقه باز سیستم به‌ازای مقادیر مختلف از یک ورودی خاص، تقریبی از تابع تبدیل سیستم را به‌دست می‌آوریم. تقریب سیستم یک مدل خطی و درجه اول با دو پارامتر عدم قطعیت می‌باشد. مناسب‌ترین نوع سیگنال ورودی ولتاژ آرمیچر، سیگنالی است که از یک سطح به سطح دیگر بدون هیچ‌گونه ثابت زمانی سوییچ شود (شکل ۳). به‌همین منظور یک سری ورودی پله PRBS به‌عنوان سیگنال ورودی در نظر گرفته شده‌اند تا رفتار سیستم نیز مشاهده شود (شکل ۴).



شکل ۳- سیگنال ورودی (ولتاژ آرمیچر)

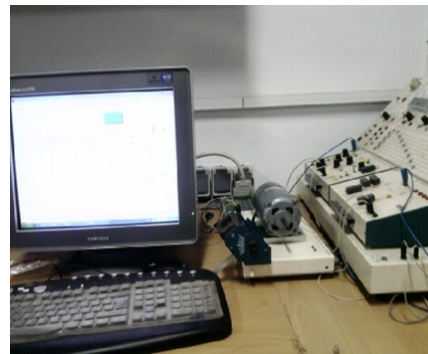


شکل ۴- سیگنال خروجی (موقعیت موتور)

در شکل ۴ سیگنال خروجی که همان پاسخ (موقعیت سیستم) به ورودی پله مانند (شکل ۳) می‌باشد؛ نشان داده

دینامیکی همچون زمان صعود، زمان نشست، ثابت زمانی، زمان تأخیر و مقدار حالت پایدار را می‌توان با استفاده از این روش به‌دست آورد. برای شناسایی سیستم مورد نظر سیگنال‌های ورودی-ولتاژ تغذیه- و خروجی سیستم -که موقعیت شفت موتور بودند، به‌طور تجربی اندازه‌گیری شده‌اند. به‌عبارت دیگر از این روش تجربی می‌توان جهت تکمیل فرآیند کنترل سخت‌افزار در حلقه یا HWIL سود جست.

شبیه‌سازی سخت‌افزار در حلقه یکی از ابزارهایی است که به‌طور گسترده در طراحی، تست و اعتبارسنجی سیستم‌های مهندسی استفاده می‌شود. در شبیه‌سازی HWIL بخشی از یک سیستم (و یا محیط آن) با یک مدل عددی^۱ جایگزین شده و از طریق حس‌گرها، عمل‌گرها و یا نرم‌افزارهای کنترلی با سخت‌افزارهای باقی‌مانده ارتباط پیدا می‌کند. هدف از شبیه‌سازی HWIL، مصالحه‌ای بین انعطاف‌پذیری، حداقل هزینه و تقریب شبیه‌سازی عددی با سخت‌افزاری کامل از لحاظ دقت و اعتبار می‌باشد. نوعاً، سخت‌افزار کامل دارای عدم قطعیت و یا نقاط بحرانی طراحی می‌باشد که مدل‌سازی آن دشوار می‌باشد [۱۸]. برای برپایی این سیستم تجربی، از یک مجموعه‌ای شامل موتور و تاکومتر استفاده می‌شود. به‌طوری‌که خروجی به یک کارت اخذ داده مدل آدونتک HG-1710L متصل شده است. کارت آدونتک از طریق برد ترمینال PCLD-8710 به یک کامپیوتر متصل است.



شکل ۲- مجموعه‌ای در نظر گرفته شده برای شناسایی و کنترل موتور DC

داد. توسعه مشتق و انتگرال از مرتبه صحیح به کسری، مقاومتی بیشتری در روش‌های تنظیم این کنترل‌کننده و دستیابی آسانی را در تعیین پارامترهای کنترل‌کننده PID کسری در مقایسه با PID معمولی فراهم کرده است. در این کنترل‌کننده ۵ پارامتر قابل تنظیم وجود دارد یعنی در مقایسه با PID معمولی ۲ پارامتر λ و μ بیشتر دارد [۲]. شکل کنترل‌کننده اصلی کسری به صورت زیر می باشد:

$$K(s) = K_p + \frac{K_I}{S^\lambda} + K_D S^\mu \quad (۱۳)$$

۴-۴- طراحی به کمک PSO و ABC

الگوریتم‌های ABC و PSO برای طراحی پارامترهای k_i ، k_p ، k_d ، λ و μ کنترل‌کننده به خدمت گرفته شدند

به‌گونه‌ای که سیستم کنترل‌شده به‌وسیله معیار کارآیی پیشنهادی دارای قابلیت قوام و دفع اغتشاش بالایی می‌باشد. پنج پارامتر کنترلی یک منبع غذایی (در ABC) یا یک ذره (در PSO) را تشکیل می‌دهند، $FS = (k_i, k_p, k_d, \lambda, \mu)$ جمعیت (گروه) وجود داشته باشد، بدیهی است که بعد جمعیت $n * 5$ خواهد شد. مراحل طراحی به‌صورت خلاصه در زیر آمده است:

- ۱- مقداردهی اولیه منابع غذایی جمعیت بر اساس بازه تعریف شده برای هر یک از پنج پارامتر کنترلی.
- ۲- برای هر یک از منابع غذایی یا FS میزان شایستگی‌شان با استفاده از معیار سنجش زیر محاسبه می‌گردد.

$$Fitness(Fs) = (|Fs(1) - K_i| + |Fs(2) - K_p| + |Fs(3) - K_d| + |Fs(4) - \lambda| + |Fs(5) - \mu|) \quad (۱۴)$$

- ۳- حال به‌کمک دیاگرام مطرح شده از ABC در شکل ۱ پاسخ‌های کاندید بهینه به‌دست می‌آیند.
 - ۴- اگر تعداد تکرارهای کل الگوریتم به حداکثر رسید، آن‌گاه به مرحله ۵ برو.
 - ۵- بهترین منبع غذایی به‌دست‌آمده، پارامترهای کنترل‌کننده بهینه ما خواهند بود.
- پارامترهای مورد استفاده در اجرای الگوریتم ABC به‌صورت زیر بوده‌اند:

شده است. همان‌طور که از شکل پیداست به‌طور کلی پاسخ سیستم همانند دینامیک یک سیستم مرتبه‌ی اول می‌باشد که به‌وسیله‌ی معادله‌ی (۸) توصیف شده است. بنابراین خصوصیات دینامیکی دستگاه مورد نظر، k_m ضریب بهره و T_m ثابت زمانی به آسانی با توجه به پاسخ سیستم قابل محاسبه می‌باشند. همان‌طور که می‌دانیم T_m ثابت زمانی سیستم، همان فاصله‌ی زمانی است که خروجی سیستم به ۶۳ درصد مقدار نهایی خود می‌رسد. از آن‌جایی که باید محاسبه‌ی T_m را برای کلیه‌ی سطوح ورودی‌ها انجام داد، محدوده‌ی T_m به‌صورت زیر به‌دست می‌آید:

$$0.36 < T_m < 0.46 \quad (۹)$$

همچنین بهره‌ی DC، k_m با توجه به رابطه‌ی زیر به‌دست می‌آید:

دامنه‌ی سیگنال ورودی / مقدار نهایی خروجی $k_m =$

باتوجه به عدم قطعیت موجود در این پارامتر نیز محدوده‌ی k_m به‌صورت زیر می‌باشد:

$$0.4 < K_m < 1.4 \quad (۱۰)$$

برای به‌دست آوردن مدل نامی می‌توان از میانگین تخمین پارامترها استفاده نمود. مدل نامی موتور به قرار زیر می‌باشد:

$$G(s) = \frac{1.238}{0.38s + 1} \quad (۱۱)$$

۴-۳- PI و PID کسری

کنترل‌کننده PID به خاطر کاربرد آسان و عملکرد مقاوم آن، به‌طور وسیعی در فرآیندهای صنعتی مورد استفاده قرار می‌گیرد. دینامیک کلی کنترل‌کننده PID به‌صورت زیر است:

$$K(s) = K_p + \frac{K_I}{S} + K_D S \quad (۱۲)$$

در سال‌های اخیر پودلابنی یک تحقق از کنترل‌کننده PID را پیشنهاد نمود. این کنترل‌کننده جدید $PI^\lambda D^\mu$ نام دارد که شامل یک انتگرال‌گیر مرتبه λ و یک مشتق‌گیر مرتبه μ می‌باشد. او همچنین پاسخ خوب این کنترل‌کننده را در مقایسه با کنترل‌کننده PID کلاسیک وقتی که به یک سیستم اعمال می‌شدند را نشان

حال به ازای k_p, k_i, k_d و λ, μ متفاوت، بهترین جواب را به دست می آوریم، بهترین جواب را می توان از چند معیار به دست آورد:

- ۱- از لحاظ تعقیب موقعیت مطلوب در نظر گرفته شده
- ۲- براساس یک معیار شاخص خطا که شاخص دوم در نظر گرفته شده میانگین مربعات خطا $(MSE)^1$ که به صورت زیر می باشد.

$$MSE = \frac{\sum e^2}{N} \quad (14)$$

که در آن N تعداد نمونه ها می باشد. نتایج حاصل از پیاده سازی عملی نیز در بخش ۵ آورده شده است.

۴-۵- فازی نمودن کنترل کننده و جدول قواعد

در این بخش، برای بهبود پاسخ سیستم در برابر اغتشاش و عدم قطعیت های موجود در سیستم کنترل کننده PID کسری را فازی سازی کردیم. حال این کنترل کننده دارای دو ورودی و یک خروجی است. ورودی ها $e(t)$ ، $\dot{e}(t)$ و خروجی هم $U(t)$ می باشد که $e(t)$ خطا بین موقعیت مطلوب و موقعیت موتور و $\dot{e}(t)$ هم تغییرات خطاست. $U(t)$ هم سیگنال کنترلی است. کنترل کننده فازی مذکور شامل سه مرحله می باشد: فازی سازی، موتور استنتاج و غیرفازی سازی [۱۵ و ۱۶].

توابع تعلق مجموعه های فازی ورودی در شکل های ۵ و ۶ نشان داده شده است. سطوح متغیرهای زبانی به صورت زیر تخصیص داده می شوند: NB : negative big; Ns : negative small; ZE : zero; PS : positive small; PB : positive big این سطوح از خصوصیات و ویژگی های پلنت مورد نظر انتخاب می شود. بازه ای این متغیرها از -0.1 تا 0.1 است. که براساس قدر مطلق مقدار خطا و تغییرات آن به دست می آید.

جدول ۱- پارامترهای مورد استفاده در اجرای الگوریتم ABC

NP	40	اندازه کلونی (زنبورهای کارگر + زنبورهای ناظر)
Food Number	NP/2	تعداد منابع غذایی که برابر با نصف اندازه کلونی یا اندازه جمعیت است.
limit	100	یک منبع غذایی که نمی تواند از طریق تعداد تلاشهایی که با پارامتری تحت عنوان 'limit' مشخص می گردد، بهبود یابد توسط زنبور کارگرش ترک خواهد شد.
Max Cycle	100	تعداد حلقه های تکرار که در واقع یک معیار توقف محسوب خواهد شد.
R	0	فاکتور احتمالی
MR	0.4	فاکتور احتمالی
D	5	تعداد ابعاد
domain	[8 12]	دامنه

ضرایب کنترل کننده PID معمولی را براساس روش های کلاسیک (زیگلر-نیکولز) به ترتیب $k_p = 0.1251$ و $k_i = 9.2270$ $k_d = 0.1238$ به دست آمده اند. ضرایب کنترل کننده PID کسری را هم بر اساس روش های بهینه سازی ABC و PSO به شکل زیر محاسبه نمودیم. البته چندین اجرا از هر الگوریتم را ذکر کردیم تا قابلیت پایداری و ثبات ABC و برتری آن بر PSO به وضوح مشاهده گردد.

جدول ۲- ضرایب به دست آمده توسط الگوریتم ABC

دفعات اجرا	μ	λ	k_d	k_p	k_i
۱	۰/۱۹۶۲	۰/۸۹۹۸	۰/۰۹۸۹	۱/۰۰۰۱	۱۰/۰۷۷
۲	۰/۲۰۰۱	۰/۹۰۱۰	۰/۱۰۰۲	۰/۹۹۹۴	۹/۹۹۹۳
۳	۰/۱۹۶۶	۰/۸۹۷۹	۰/۰۹۹۷	۰/۹۹۹۸	۹/۹۹۹۶
۴	۰/۲۰۸۷	۰/۸۹۹۷	۰/۰۹۹۹	۰/۹۹۹۵	۹/۹۹۲۹
۵	۰/۱۹۵۴	۰/۹۰۰۵	۰/۱۰۰۰	۱/۰۰۰۲	۹/۹۹۹۷

جدول ۳- ضرایب به دست آمده توسط الگوریتم PSO

دفعات اجرا	μ	λ	k_d	k_p	k_i
۱	۰/۲۶۱۸	۰/۹۵۶۶	۰/۱۲۳۸	۱/۱۲۵۱	۹/۲۲۷۰
۲	۰/۲۱۰۸	۰/۸۸۴۹	۰/۱۰۷۰	۰/۹۲۴۰	۹/۲۷۶۵
۳	۰/۹۶۴۹	۰/۹۳۶۰	۰/۱۲۰۰	۱/۰۰۶۷	۹/۰۵۹۹
۴	۰/۱۸۲۶	۰/۷۰۲۴	۰/۱۴۴۷	۰/۶۸۰۳	۹/۳۰۰۶
۵	۰/۴۱۴۶	۱/۰۸۲۴	۰/۱۴۲۳	۰/۹۶۱۸	۹/۰۳۶۹

¹ Mean Square Error

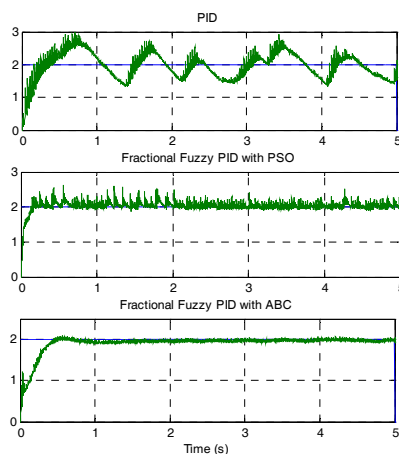
با توجه به جدول زیر، از آن جایی که برای هر متغیر ورودی ۵ مجموعه فازی تعریف شده است، ۵^۲ قاعده فازی قابل ترکیب است. هر دو قسمت فازی‌ساز و سیستم استنتاج به صورت تجربی تنظیم شده‌اند.

جدول ۴- پایگاه قواعد استنتاج فازی

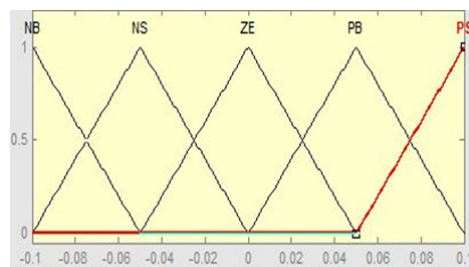
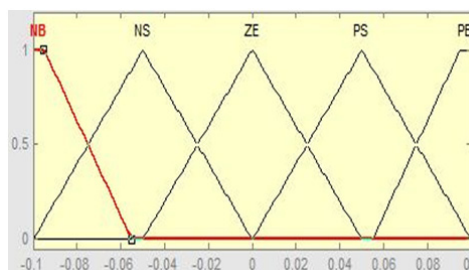
$\dot{e}(t) \mid e(t)$	NB	NS	ZE	PS	PB
NB	S	S	MS	MS	M
NS	S	MS	MS	M	MB
ZE	MS	MS	M	MB	MB
PS	MS	M	MB	MB	B
PB	M	MB	MB	B	B

۵- نتایج پیاده‌سازی

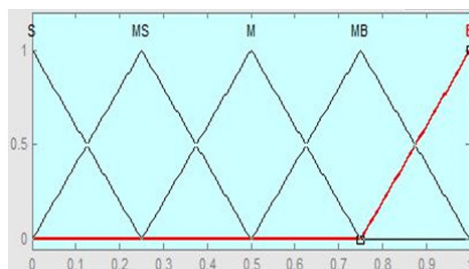
این پژوهش، پیاده‌سازی نیمه عملی و مقایسه کنترل‌کننده‌های مرتبه صحیح و کسری فازی با دخالت روش‌های هوشمند برای کنترل موقعیت موتور DC می‌باشد. در شکل‌های زیر موقعیت موتور درحالتی که ضرایب کنترل‌کننده‌های PID کسری فازی با استفاده از ABC و PSO به دست آمده‌اند و همچنین PID معمولی، نشان داده شده است. شکل‌های زیر براساس متوسط ضرایب ذکر شده در جداول ۲ و ۳ می‌باشند.



شکل ۸- خروجی‌های موتور (موقعیت) به ازای کنترل‌کننده‌های مذکور

شکل ۵- توابع تعلق متغیر ورودی $e(t)$ شکل ۶- توابع تعلق متغیر ورودی $\dot{e}(t)$

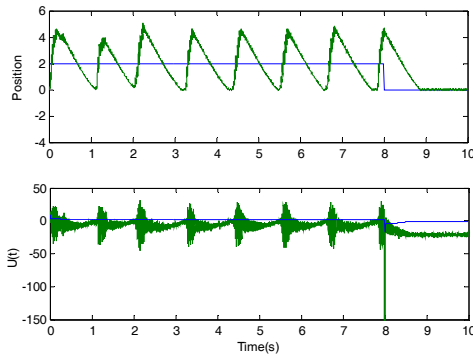
توابع تعلق متغیر خروجی $U(t)$ در شکل ۷ نشان داده شده است. سطوح زبانی این متغیر به صورت S : small; MS : medium small; M : medium; MB : medium big; B : big بازه تغییرات از صفر تا ۱ خواهد بود.

شکل ۷- توابع تعلق متغیر خروجی $U(t)$

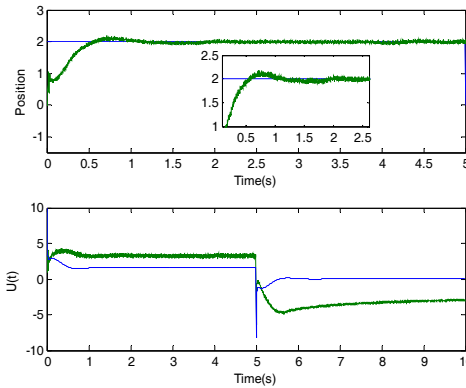
سپس، یک پایگاه قواعد 5×5 برای توسعه سیستم استنتاج، تعریف شده است (جدول ۴). عموماً قواعد فازی وابسته به پلنتی که باید کنترل شود، نوع کنترل‌کننده و تجربه عملی طراح سیستم خواهد بود. براساس مجموعه-های فازی ورودی نشان داده شده در شکل‌های ۵ و ۶ و متغیر خروجی، قواعد فازی جدول به صورت زیر ترکیب می‌شوند:

Rule 1: If $e(t)$ is NB and $\dot{e}(t)$ is NB then $U(t)$ is S .

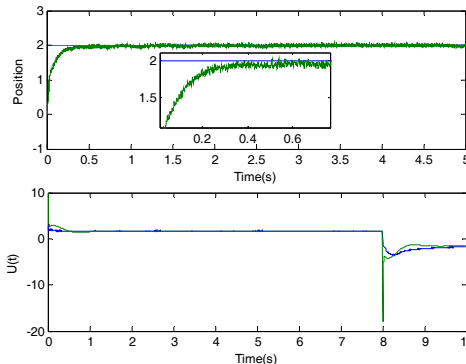
شکل‌های زیر موقعیت موتور را در حضور اغتشاش نشان می‌دهد.



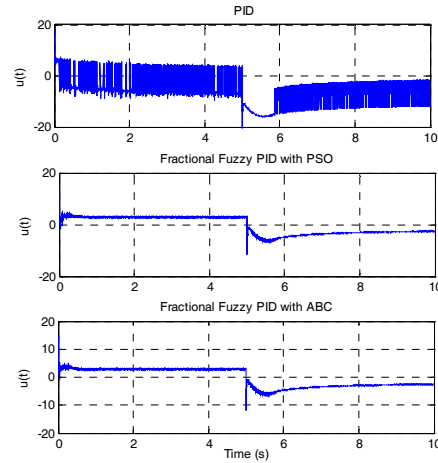
شکل ۱۰- خروجی موتور به ازای کنترل کننده PID در حضور اغتشاش



شکل ۱۱- خروجی موتور به ازای کنترل کننده PID فازی کسری تنظیم شده با الگوریتم PSO در حضور اغتشاش



شکل ۱۲- خروجی موتور به ازای کنترل کننده PID فازی کسری تنظیم شده با الگوریتم ABC در حضور اغتشاش



شکل ۹- تلاش کنترلی به ازای کنترل کننده‌های مذکور

همان‌گونه که از شکل‌های ۸ و ۹ مشخص است، PID معمولی نمی‌تواند کنترل کننده مناسبی برای موقعیت موتور باشد. درحالی‌که در اثر اعمال کنترل کننده PID کسری فازی، موقعیت موتور بعد از طی مدت زمان اندکی به موقعیت مطلوب می‌رسد. فراجاهش پاسخ کاهش یافته و پاسخ خیلی زودتر از پاسخ به دست آمده از PID معمولی میرا می‌شود، که سه معیار زمان نشست، فراجاهش و میرایی پاسخ و میزان خطا در تنظیم ضرایب با ABC به مراتب بهینه‌تر (کمتر) از PSO می‌باشد. مسئله دیگر تلاش کنترلی می‌باشد که در PID فازی کسری، این امر به میزان قابل توجهی کاهش یافته است. مطلب مذکور به صورت عددی در جدول زیر آمده است.

جدول ۵- معیارهای مقایسه کنترل کننده‌های ارائه شده بدون

حضور اغتشاش

نوع کنترل کننده	میزان فراجاهش	زمان نشست	میان خطا (MSE)
PID	۸۶٪	۱۷/۸۶	۰/۱۵۶
Fractional Fuzzy PID with PSO	۵۳/۵٪	۱/۶	۰/۱۶۱
Fractional Fuzzy PID with ABC	۱/۷٪	۰/۵	۰/۰۸۷

۶- نتیجه‌گیری

در این مقاله برای طراحی ضرایب کنترل‌کننده PID از دو رویکرد حسابان کسری و روش‌های هوشمند استفاده شده است. سودمندی حسابان کسری و کارایی آن با استفاده از شبیه‌سازی و پیاده‌سازی عملی نشان داده شد. رویکرد هوشمند این پژوهش از دو مولفه سیستم‌های فازی و روش‌های بهینه‌سازی هوش جمعی بهره گرفته است. در رویکرد فازی با استفاده از گزاره‌های فازی که توصیف‌گر دانش خبره در طراحی کنترل می‌باشند، براساس شناخت از تاثیر ورودی بر خروجی سیستم جدول فازی مناسبی طراحی گردید. تعیین بهینه ضرایب کنترل‌کننده توسط روش‌های هوش گروهی ABC و PSO صورت گرفته است که به دلیل سرعت همگرایی و تنوع‌پذیری بالا، قادر به یافتن بهینه‌های مناسب‌تری در فضای جستجوی مسئله می‌باشند. کنترل‌کننده‌های تنظیم شده به‌وسیله ABC دارای قوام، پایداری و عملکرد بهتری نسبت به سایر کنترل‌کننده‌های استفاده شده می‌باشد. استفاده از این روش‌ها، مزیت عمده تصمیم‌سازی هوشمند کنترل‌کننده‌های کسری را در مقایسه با طراحی استاتیک آشکار می‌سازد.

با مقایسه شکل‌های ۱۰، ۱۱ و ۱۲ درمی‌یابیم، که با اعمال کنترل‌کننده PID فازی کسری، اغتشاش وارد شده به سیستم به میزان قابل‌توجهی کاهش می‌یابد. در صورتی که در اثر اعمال کنترل‌کننده PID نه تنها اثر اغتشاش تعدیل نمی‌شود بلکه شاهد تغییرات زیادی در موقعیت موتور می‌باشیم.

جدول ۶- خطای کنترل‌کننده‌های ارائه شده در حضور اغتشاش

میزان خطا (MSE)	نوع کنترل‌کننده
۲/۰۱۸۰	PID
۰/۰۹۸۲	Fractional Fuzzy PID with PSO
۰/۰۷۴۲	Fractional Fuzzy PID with ABC

۷. مراجع

- [1] Axtell, M., Bise, M., (1990). "Fractional calculus applications in control systems". Proceeding of the National Aerospace and Electronics Conference, New York, IEEE, pp. 563-566.
- [2] Vinagre, B.M., Podlubny, I., Petras, I., Chen, Y.Q. (2001). "Using fractional order adjustment rules and fractional order reference models in Model Reference Adaptive Control". Nonlinear Dynamics, Vol. 29, pp. 269-279.
- [3] Aytekin, B. (2011). "Tabu search algorithm based PID controller tuning for desired system specifications". Vol. 348, Issue 10, pp. 2795-2812.
- [4] Zamani, M., Karimi-Ghartemani, M., Sadati, N., Parniani, M. (2009). "Design of a fractional order PID controller for an AVR using particle swarm optimization". Control Engineering Practice, Vol. 17, Issue 12, pp. 1380-1387
- [5] Gozde, H., Taplamacioglu, M.C. (2011). "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system". Journal of the Franklin Institute, Vol. 348, Issue 8, pp. 1927-1946.
- [6] Aboeela, M.A.S., Ahmed, M.F., Dorrah, H.T. (2011). "Design of aerospace control systems using fractional PID controller". Journal of Advanced Research, Vol. 2, Issue 1, pp. 1-8.
- [7] Tang, Y., Cui, M., Hua, C., Li, L., Yang, Y. (2012). "Optimum design of fractional order PID controller for AVR system using chaotic ant swarm". Expert Systems with Applications, Vol. 39, Issue 8, pp. 6887-6896.
- [8] Chengbin, M., Hori, Y. (2007). "Fractional-order control: theory and applications in motion control". IEEE IES Magazine, pp. 6-16.

- [9] Kennedy, J., Eberhart, R.C. (1995). "Particle Swarm Optimization". Proceeding of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948.
- [10] Sabat, S.L., Udgata, S.K., Abraham, A. (2010). "Artificial bee colony algorithm for small signal model parameter extraction of MESFET". Engineering Application of Artificial Intelligence, Vol. 23, Issue 5, pp. 689- 694.
- [11] Karaboga, D., Akay, B. (2009). "A Comparative Study of Artificial Bee Colony Algorithm". Applied Mathematics and Computation, Vol. 214, Issue 1, pp. 108-132.
- [12] Sundar, S., Singh, A. (2010). "A swarm intelligence approach to the quadratic minimum spanning tree problem". Information sciences, Vol. 180, Issue 17, pp. 3182-3191.
- [13] Akay, B., Karaboga, D. (2012). "A modified artificial bee colony algorithm for real-parameter optimization". Information Sciences, Vol. 192, pp. 120–142.
- [14] Heidari, Y., Noee, A.R., Shayanfar, H.A., Salehi, S. (2010). "Robust control of DC motor using fuzzy sliding mode control with fractional PID compensator". The Journal of Mathematics and Computer Science, Vol. 1, Issue 4, pp. 238-246.
- [15] Zounemat-Kermani, M., Teshnelab, M. (2008). "Using adaptive neuro-fuzzy inference system for hydrological time series prediction". Applied soft computing, Vol. 8, Issue 2, pp. 928-936.
- [16] Jang, R. (1993). "ANFIS: Adaptive-network-based fuzzy inference systems". IEEE Trans. Syst. Man Cybern, Vol. 23, pp. 665-685.
- [17] Fallahi, M., Azadi, S. (2009). "Robust control of DC motor using fuzzy sliding mode control with PID compensator". Proceeding of the International Multi Conference of Engineers and Computer Scientists, Volume II.
- [18] Hosseinnia, S.H., Ghaderi, R., Ranjbar, A., Mahmoudian, M., Momani, S. (2010). "Sliding mode synchronization of uncertain fractional-order chaotic system". Computers and Mathematics with Applications, Vol. 59, pp. 1637-1643.