



طراحی کنترل کننده تناسبی-انتگرالی - مشتقی بهینه با الگوریتم بهینه سازی بهبود یافته اجتماع پرندگان

مجید مرادی زیرکوهی^۱

^۱ استادیار کنترل، دانشکده فنی و مهندسی، دانشگاه صنعتی خاتم الانبیاء بهبهان، بهبهان، ایران

تاریخ دریافت: ۱۳۹۳/۰۷/۱۸؛ تاریخ بازنگری: ۱۳۹۳/۱۱/۲۶؛ تاریخ پذیرش: ۱۳۹۴/۰۱/۳۰

چکیده

در این مقاله، طراحی کنترل کننده تناسبی-انتگرالی-مشتقی (تام) بهینه بر مبنای "الگوریتم بهینه سازی بهبود یافته اجتماع پرندگان" (ePSO) ارائه می شود. مزیت این روش جدید نسبت به روش‌های متداول در طراحی کنترل کننده این است که ارائه آن محدود به کلاس خاصی از سیستم‌ها نیست. در طراحی کنترل کننده تام بهینه، مجموع زمان صعود، زمان نشست، فرجهش و انتگرال قدر مطلق خطا کمینه می‌شوند. سه نوع الگوریتم بهینه سازی اجتماع پرندگان (sPSO, mPSO, ePSO) با روش‌های دیگر بهینه سازی از جمله، جامعه مورچه‌ها (Ant colony) مقایسه می‌گردند. تحلیل آماری نتایج، برتری الگوریتم جدید "بهینه سازی اجتماع پرندگان با برون یابی" (ePSO) را در تنظیم ضرایب کنترل کننده تام نشان می‌دهد. الگوریتم ePSO با الگوریتم‌های دیگر بهینه سازی اجتماع ذرات از آن جهت فرق می‌کند که نیازی به معادله سرعت ندارد و با به روز کردن موقعیت ذرات توسط برون یابی با همگرایی سریع‌تر و تعداد گام کمتر به جواب بهینه می‌رسد.

کلمات کلیدی: الگوریتم بهینه سازی اجتماع پرندگان؛ جامعه مورچه‌ها؛ کنترل کننده تام بهینه؛ برون یابی.

Designing Optimal PID Controller Using Modified Particle Swarm Optimization

M. Moradi Zirkohi¹

¹ Assist. Prof., Elect. Eng, Behbahan Khatam Alanbia University of Technology, Behbahan, Iran

Abstract

In this paper, designing optimal PID controller using modified particle swarm optimization is presented. The advantage of this new method compared to conventional methods of controller design is that it is not limited to a certain class of systems. In designing phase, sum of rising time, settling time, overshoot and integral of squared-error are minimized. There kind of particle swarm optimization algorithms such as ePSO, mPSO and sPSO are compared with other methods of optimization including Ant Colony. The results clearly show how superior the new proposed method is to the other methods. The proposed method differs from the other optimization algorithms in such a way that, the proposed algorithm does not need a velocity equation. The position of the particle is updated directly by extrapolating the current particle position with the global best particle position obtained so far.

Keywords: Particle Swarm Optimization; Ant Colony; Optimal PID Controller; Extrapolating.

۱- مقدمه

علیرغم فراوانی روش‌های موجود در منابع در مورد طراحی کنترل کننده های تام، تلاش برای ارائه روش‌های جدید همچنان ادامه دارد. دلیل آن، کاربرد گسترده این کنترل کننده توانمند در صنعت است و همچنین پارامترهای کنترل کننده در اغلب سیستم‌ها، به سادگی تنظیم نمی‌شوند [۱]. در حال حاضر، بیشتر روش‌های تنظیم پارامترهای تام برای سیستم مرتبه اول، مرتبه دوم و در صورت لزوم با تاخیر زمانی ارائه شده اند. از این رو، در طراحی این کنترل کننده برای سایر سیستم‌ها، ابتدا آنها را به ساختار مذکور تقریب میزنند. چنانچه خطای تقریب بزرگ باشد، طراحی کنترل کننده با روش‌های مذکور بی‌فایده خواهد بود.

هدف این مقاله، ارائه روش واحدی در طراحی کنترل کننده تام بهینه برای سیستم‌های مختلف است. برای رسیدن به این هدف، از روش‌های هوشمند نظیر، الگوریتم‌های اجتماع پرندگان و مورچه‌ها استفاده می‌شود.

الگوریتم‌های بهینه سازی به عنوان ابزاری قدرتمند در بسیاری از کاربردهای کنترلی مانند، شناسایی پارامترها و طراحی سازه‌های کنترلی شناخته شده است [۲]. الگوریتم‌های بهینه سازی، همچنین استفاده گسترده ای در بهینه کردن کنترل کننده‌ها به خصوص با روش منطق فازی [۳] و شبکه عصبی [۴] پیدا کرده است. این مقاله، به تنظیم بهینه ضرایب کنترلر یک سیستم با استفاده از سه روش بهینه سازی اصلاح شده اجتماع پرندگان متمرکز شده است و سپس نتایج با الگوریتم جامعه مورچه‌ها مقایسه می‌شود. در ادامه، ابتدا به بررسی الگوریتم‌های بهینه سازی اصلاح شده اجتماع پرندگان می پردازیم و سپس با استفاده از این الگوریتم‌ها، کنترلر کننده تام بهینه طراحی می‌شود.

۲- الگوریتم بهینه سازی اجتماع ذرات برای

طراحی کنترلر

الگوریتم بهینه سازی اجتماع ذرات (PSO)، الگوریتم بهینه‌سازی تکاملی با جستجوی موازی در فضای جواب است. در این الگوریتم، جامعه‌ای از ذرات که هرکدام یک جواب ممکن برای مسأله بهینه سازی است در نظر گرفته می‌شود. این ذرات در ابتدا بصورت تصادفی در فضا قرار می‌گیرند. در هر گام، ارزش هر ذره به کمک تابع ارزش (معیار بهینه

سازی) محاسبه شده، با در نظر گرفتن بهترین جواب ذرات در جامعه، به سمت و اندازه معینی حرکت خواهد کرد؛ بنابراین، هر یک از ذرات جامعه در هر گام، موقعیت و سرعت خاصی دارند. موقعیت و سرعت ذره i ام در گام t ام به ترتیب، با بردارهای $x_i(t)$ و $v_i(t)$ نمایش داده می‌شوند. در ابتدای شروع الگوریتم، موقعیت و سرعت اعضای جامعه با بردارهای تصادفی $x_i(0)$ و $v_i(0)$ مقدار دهی می‌شوند.

بهترین مکانی را که ذره i ام در طول تکرار الگوریتم تا لحظه t ام داشته است با $pbest_i(t)$ نام گذاری می‌کنیم. همچنین بهترین مکانی که ذرات تا لحظه t ام داشته اند را با بردار $gbest$ نشان می‌دهیم. همانطور که اشاره شد، این برتری را تابع ارزش در بهینه سازی تعیین می‌کند. برای جلوگیری از همگرایی زودرس و همچنین به دام افتادن ذرات در کمینه های محلی فنونی ارائه شده است. به عنوان مثال، وقتی ذرات بیش از حد به بهترین ذره جامعه نزدیک می‌شوند، برای جستجوی بیشتر، فضا آنها را دوباره بصورت تصادفی پراکنده می‌کنیم. این روش، الگوریتم را در مقابل خطر به دام افتادن در کمینه‌های محلی نیز مقاوم می‌کند [۵].

سرعت ذره i ام در گام $(t+1)$ ام از الگوریتم ترکیبی از سه مولفه سرعت قبلی ذره، سمت حرکت به بهترین موقعیت ذره در گذشته و سمت حرکت به بهترین موقعیت جامعه در گذشته تشکیل می‌شود که با استفاده از رابطه (۱) بدست می‌آید [۵]:

$$v_i(t+1) = \zeta[\omega \cdot v_i(t) + c_1 \cdot \Phi_1 \cdot (pbest_i(t) - x_i(t)) + c_2 \cdot \Phi_2 \cdot (gbest(t) - x_i(t))] \quad (1)$$

که ضریب انقباض الگوریتم نامیده می‌شود. ω ضریب اینرسی است که میزان وابستگی سرعت ذره در هر گام را نسبت به گام قبلی تعیین می‌کند. اعداد ثابت c_1 و c_2 را به ترتیب ضرایب اعتماد به خود و اعتماد به گروه می‌نامند. اعداد تصادفی Φ_1 و Φ_2 ، میزان وابستگی سرعت ذره را به $pbest_i(t)$ و $gbest(t)$ مشخص می‌کند. پارامترهای بالا در سرعت همگرایی و امکان رسیدن مسئله به جواب بهینه نقش دارند. در این خصوص، از سوی محققان پیشنهادهای متفاوتی در انتخاب این پارامترها شده است. در الگوریتم بهینه سازی اجتماع ذرات رایج (CPSO) ضرایب ω ، c_1 ،

بهترین موقعیت ذرات (gbest) و موقعیت کنونی هر ذره به روز می‌شود. این الگوریتم، دارای دو ضریب e_1, e_2 است که بصورت غیرخطی با پیش رفتن الگوریتم کاهش می‌یابند. موقعیت به روز شده هر ذره بصورت رابطه (۶) است:

$$x_i(t) = gbest + [e_1 \times r \times gbest] + [e_1 \times (gbest - x_i(t-1) \times \exp(e_2 \times ((f(gbet) - f(x_i(t-1)))) / f(gbest))))] \quad (6)$$

که در آن r یک عدد تصادفی، e_1, e_2 ضرایب درون یابی، $f(gbest)$ مقدار ارزش متناظر با $gbest$ و $f(x_i(t))$ مقدار ارزش موقعیت کنونی ذره هستند. e_1 و e_2 بصورت رابطه (۷) تعریف می‌شوند:

$$e_i = \frac{0.8 - 0.2 e^{-(t/t_{max})^\alpha}}{1 + e^{-(t/t_{max})^\alpha}}, i = 1, 2 \quad (7)$$

توجه شود α که یک عدد است و با توجه به معادله (۷) مثلا با فرض $\alpha = 2$ داریم $0.3 \leq e_i < 0.68$. این پارامتر، نقش مهمی را در همگرایی الگوریتم ایفا می‌کند. اولین جزء معادله (۶)، عملگر اصلی برای درون یابی است. جزء دوم، تضمین کننده حرکت ذرات بسوی بهترین موقعیت و جزء سوم، اندازه گام برای جابجایی بمنظور بروز کردن موقعیت ذره است. در ابتدا اختلاف بین $f(x_i(t))$ و $f(gbest)$ بسیار زیاد است، از این رو ذره جابجایی بیشتری را بسمت بهترین موقعیت دارد و این فاصله در مراحل بعدی کاهش می‌یابد تا اینکه در نهایت، ذره به بهترین موقعیت نزدیک شود. همانطور که گفته شد، موقعیت ذره با استفاده از برون یابی بروز می‌شود. وقتی فاصله بین $gbest$ و $x_i(t)$ زیاد است، فاصله بین $gbest$ و موقعیت‌های برون یابی شده نیز همچنین نسبتا زیاد است. بخاطر تصادفی بودن پارامتر r ، موقعیت اصلاح شده ذره در در شعاعی متناظر با اندازه گام در مجاورت $gbest$ قرار می‌گیرد.

در ابتدا اندازه گام بزرگ است و از این رو، موقعیت ذره همانطور که در شکل ۱ نشان داده شده، در شعاعی بزرگ قرار می‌گیرد.

در مراحل آخر که ارزش ذره کنونی نزدیک به ارزش ذره دارای بهترین موقعیت است، اندازه گام کاهش می‌یابد و در

Φ_1 ، Φ_2 ثابت انتخاب می‌شوند. در این مقاله، این ضرایب متغیر بوده، با اصلاح آنها، دو روش بهینه سازی اصلاح شده اجتماع پرندگان را مورد بررسی قرار دهیم.

۳- الگوریتم اجتماع ذرات اصلاح شده (mPSO)

برای همگرایی سریع تر و عملکرد بهتر می‌توان ضرایب معادله سرعت را براساس مقادیر پیشنهادی در مرجع [۵] و [۶] تنظیم کرد. ζ ضریب انقباض به صورت زیر برحسب پارامتر $\varphi = c_1 + c_2$ تعریف می‌شود:

$$\zeta = \frac{2}{|4 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (2)$$

پارامترهای c_1 و c_2 در گام t ام از رابطه (۳-۴) محاسبه می‌شوند:

$$c_1 = (c_{1f} - c_{1i}) \frac{t}{t_{max}} + c_{1i} \quad (3)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{t}{t_{max}} + c_{2i} \quad (4)$$

که در آن t_{max} حداکثر تعداد گام‌های الگوریتم و $c_{2i} = c_{1f} = 0.5$ و $c_{1i} = c_{2f} = 2.5$ می‌باشند.

این ضرایب در اندازه و جهت سرعت اعضا و در نتیجه در چگونگی جستجو در فضای جواب حائز اهمیت هستند. بدین ترتیب موقعیت ذره در گام $(t+1)$ ام از رابطه (۵) برحسب موقعیت قبلی و سرعت ذره قابل محاسبه است:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

پس از مقدار دهی به این ضرایب، الگوریتم در هر گام سرعت و موقعیت خود را تصحیح می‌کند و گام به گام به جواب بهینه نزدیکتر می‌شود. میزان مطلوب بودن جواب‌ها بوسیله تابع ارزش تعیین می‌شود که طراحی مناسب آن، نقش کلیدی در همگرایی الگوریتم دارد.

۴- بهینه سازی اجتماع ذرات با برویابی (ePSO)

یکی از روش های جدید برای بهینه سازی، استفاده از برون یابی است [۶]. در این الگوریتم، موقعیت کنونی هر ذره با درون یابی بهترین موقعیت ذرات بمنظور اصلاح جستجو بسمت نقطه کمینه مطلق به روز می‌شود. موقعیت هر ذره بدون استفاده از معادله سرعت (۴) و تنها با استفاده از

تابع تبدیل $C(s)$ برای PID بصورت رابطه (۸) تعریف می‌شود:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (8)$$

که در آن K_p ، K_i و K_d بترتیب، ضرایب تناسبی، انتگرالی و مشتقی هستند. منظور از طراحی کنترل کننده تام، تعیین این ضرایب است. تابع هزینه ای بصورت رابطه (۹) برای طراحی کنترل کننده در نظر می‌گیریم:

$$f_{Total} = \alpha_1 f_{MO} + \alpha_2 f_{RT} + \alpha_3 f_{ST} + \alpha_4 f_{IAE} \quad (9)$$

در این تابع هزینه f_{Total} که آن را شاخص عملکرد می‌نامیم، f_{MO} ، f_{RT} ، f_{ST} و f_{IAE} به ترتیب، فراجش، زمان صعود، زمان نشست و انتگرال قدرمطلق خطا هستند. ضرایب α_i ($i=1, \dots, 4$) با توجه به شرایط مسئله قابل انتخاب است. کمینه کردن تابع هزینه فوق، بطور کلی سبب می‌شود که خروجی سیستم، هم از لحاظ پاسخ گذرا و هم از لحاظ پاسخ ماندگار، مطلوب باشد. با کم بودن فراجش، زمان صعود و زمان نشست پاسخ سریع بوده، با نوسانات کوچک است. کم بودن انتگرال قدر مطلق خطا نیز موجب می‌شود که خروجی سیستم، تفاوت کمی با خروجی مطلوب داشته باشد. فرض کنید که تابع تبدیل سیستم بصورت رابطه (۱۰) است:

$$G(s) = \frac{4.228}{s^3 + 2.14s^2 + 9.27s + 4.228} \quad (10)$$

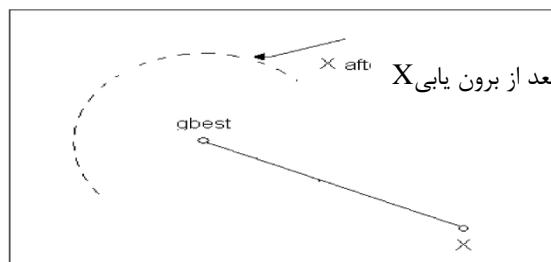
۶- شبیه سازی

در الگوریتم sPSO پارامترها بصورت $c1=2, c2=2$ و $w=1$ و در الگوریتم ePSO طبق روابط (۲) تا (۷) تعریف شده اند. علاوه بر این $\alpha=2$ انتخاب می‌شود. تعداد ذرات جامعه ۷۵ و ابعاد فضای جواب یعنی ابعاد هر ذره، برابر با تعداد ضرایب مجهول در طراحی کنترل کننده یعنی، ۳ انتخاب شده است. تمام الگوریتم‌ها در این مقاله برای ۲۰ بار شبیه سازی شده و برای هر روش مقدار میانگین، کمینه، حداکثر و انحراف معیار خطا محاسبه شده است. برای مقایسه نتایج، مسئله با استفاده از روش‌های زیگلر نیکلز و الگوریتم جامعه مورچه ها نیز حل شده است.

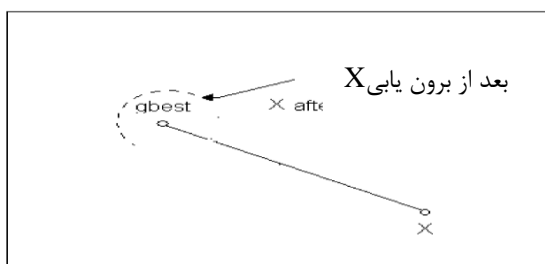
ضرایب کنترل کننده تام بهینه در الگوریتم های مختلف در جدول ۱ آمده است. همان‌طور که می‌بینیم، مقدار

نتیجه همانطور که در شکل ۲ نشان داده شده، شعاع بروز شدن موقعیت ذره کم می‌شود.

الگوریتم ePSO با الگوریتم های sPSO و mPSO از آن جهت فرق می‌کند که نیازی به معادله سرعت ندارد و موقعیت ذره با استفاده از برون یابی به روز می‌شود [۶].



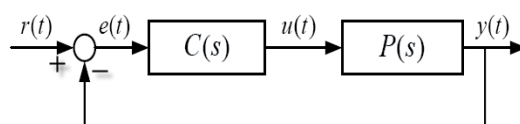
شکل ۱- مراحل اولیه که $f(gbest)$ بسیار کمتر از $f(x_i(t))$ است.



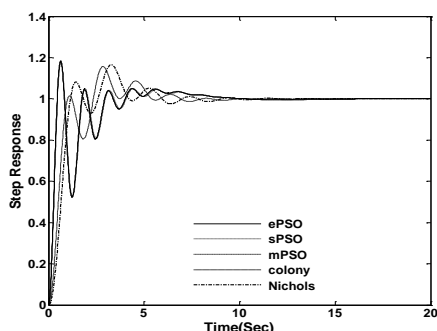
شکل ۲- مراحل نهایی که $f(gbest)$ تقریباً برابر $f(x_i(t))$ است.

۵- تحلیل سیستم و کنترل کننده

کنترل کننده تام بطور وسیعی در کنترل فرایندهای صنعتی استفاده می‌شود. در کنار قابلیت‌های فراوان آن، این کنترل کننده بسادگی می‌تواند برای اغلب فرایندهای صنعتی پیاده سازی شود. در سیستم شکل ۳، $C(s)$ کنترل کننده و $G(s)$ سیستمی است که باید کنترل شود. هدف، طراحی کنترل کننده‌ای است که در آن، خروجی $y(t)$ دارای خصوصیات مطلوب باشد.



شکل ۳- بلوک دیاگرام یک سیستم کنترل فیدبک دار



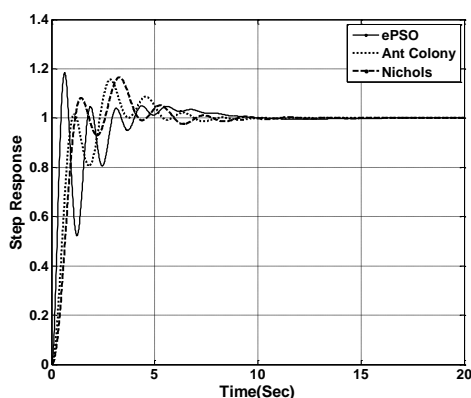
شکل ۴- پاسخ پله الگوریتم‌ها

جدول ۲- مقایسه معیارهای عملکرد در الگوریتم‌های

مختلف					
الگوریتم	حداکثر	زمان	زمان	معیار	مجموع
	فراجش	صعود	نشست	خطا	خطا
	f_{MO}	f_{RT}	f_{ST}	f_{IAE}	f_{Total}
sPSO	۰.۱۷۹۳	۰.۳۱	۲.۸۶	۰.۸۵۹۱	۴.۲۰۸۵
ePSO	۰.۱۸۴۷	۰.۳	۲.۸۳	۰.۸۶۵۲	۴.۱۷۹۹
mPSO	۰.۱۸۳۴	۰.۳۱	۲.۸۳	۰.۸۶۸۲	۴.۱۸۱۶
Ant colony	۰.۱۵۶۹	۰.۶۴	۴.۹۹	۰.۸۸۶	۶.۶۵۲۵
Zeigler Nichols	۰.۱۶۴۷	۰.۷۳	۵.۳۸	۰.۹۶۴	۷.۲۲۴۲

برای مقایسه بهتر پاسخ پله روش‌های الگوریتم پرندگان و برون یابی، جامعه مورچه‌ها و زیگلر نیکلز جداگانه در شکل ۵ آورده شده است.

معیارهای حداقل خطا، حداکثر خطا، میانگین خطا و انحراف معیار برای الگوریتم‌ها نیز در جدول ۳ آمده است که ePSO بهترین نتایج را داده است.



شکل ۵- پاسخ پله الگوریتم‌های ePSO، Ant Colony، Nichols

انحراف معیار در روش ePSO بسیار کمتر از دو روش دیگر است که این نشان‌دهنده پایداری این روش در پیدا کردن جواب بهینه است. کلیه الگوریتم‌ها نیز برحسب شاخص عملکرد f_{Total} مقایسه شده اند. با توجه به شاخص عملکرد می بینیم که سه الگوریتم اجتماع پرندگان، عملکرد بهتری نسبت الگوریتم جامعه مورچه‌ها و روش چارلز نیکولز دارند. در این میان، الگوریتم ePSO بهترین عملکرد را نشان داده است. این موضوع با مقایسه الگوریتم‌های مختلف برحسب معیارهای عملکرد فراجش، زمان صعود، زمان نشست، انتگرال قدرمطلق خطا و شاخص عملکرد در جدول ۲ نیز، مجدداً تایید می‌شود. نتایج دو روش آخر از مرجع [۷] استخراج شده است. نتایج نشان می‌دهند که الگوریتم اجتماع پرندگان با روش برون یابی (ePSO)، بهترین نتیجه را داده است. الگوریتم جامعه مورچه‌ها دارای کمترین فراجش و "الگوریتم اجتماع پرندگان با برون یابی" در مقایسه با الگوریتم پرندگان اصلاح شده دارای زمان نشست یکسان و زمان صعود کمتر است.

شکل ۴، تابع هدف (هزینه) هر سه الگوریتم ارائه شده را در ۳۰ تکرار نشان می‌دهد. همانطور که می بینیم، نرخ همگرایی در الگوریتم پرندگان با استفاده از فن برون یابی کمتر از دو روش دیگر است. یعنی این الگوریتم در تعداد گام‌های کمتری به جواب بهینه می‌رسد.

جدول ۱- ضرایب کنترل کننده تام بهینه در الگوریتم‌های

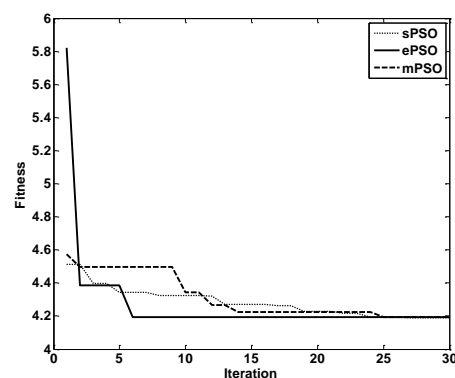
مختلف				
الگوریتم	K_D	K_I	K_P	f_{Total}
sPSO	۴.۱۶۷۹	۲.۱۵۹۸	۳.۴۵۶۵	۴.۲۰۸۵
ePSO	۴.۲۱۳۷	۲.۲۰۵۹	۳.۴۱۶۳	۴.۱۷۹۹
mPSO	۴.۳۲۲۶	۲.۱۹۸۷	۳.۳۸۲۹	۴.۱۸۱۶
Ant colony	۱.۱۵۱	۲.۲۱۹	۲.۵۱۷	۶.۶۵۲۵
Zeigler Nichols	۰.۵۶۵	۲.۱۲۶	۲.۱۹	۷.۲۲۴۲

جدول ۳- مقایسه الگوریتم‌ها بر اساس شاخص

الگوریتم	حداقل خطا	حداکثر خطا	میانگین خطا	انحراف معیار خطا
sPSO	۴.۲۰۸۵	۴.۷۵۱۱	۴.۳۳۹	۰.۱۸۱۴
mPSO	۴.۱۸۱۶	۴.۵۳۹	۴.۲۷۴۲	۰.۱۲۸۱
ePSO	۴.۱۷۹۹	۴.۴۰۵۲	۴.۲۶۲۲	۰.۰۷۴۶

۸- مراجع

- [1] Zhuang M, Atherton DP (1993) Automatic tuning of optimum PID controllers. IEE PROC-D 140(3): 216-224.
- [2] Cipperfield A, Flemming P, Fonscea C (1994) Genetic algorithms for control system engineering. Proc. Adapt. Comp. in Eng. Design and Control 128-133.
- [3] Linken DA, Nyongesa HO (1995) Genetic algorithm for fuzzy control. IEE P-Contr Theor Ap 142(3): 161-185.
- [4] Schaffer J, Eshelman L (1992) Combinations of genetic algorithms and neural networks: a survey of the state of the art. Proceedings of International workshop on Combination of Ga and NN, Baltimore, USA 1-37.
- [5] Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Dept. of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700032, India.
- [6] Senthil M, Rao MVC (2008) A novel and effective particle swarm optimization like algorithm with extrapolation technique. Elsevier journal Applied Soft Computing 1-13.
- [7] Hsiao Y and Chuang CL (2004) Ant colony optimization for designing of PID controllers. Proc. CACSD, Taiwan.



شکل ۶- تابع هزینه الگوریتم‌های sPSO، mPSO و ePSO

۶- نتیجه گیری

در این مقاله، سه نوع الگوریتم که از زندگی اجتماعی موجودات زنده و هوش جمعی آنها الهام گرفته است، برای طراحی کنترلر تام بهینه مورد استفاده قرار گرفت. کنترلر کننده تام، بگونه ای طراحی شده است که مجموع زمان صعود، زمان نشست، بیشترین فراجش و انتگرال قدر مطلق خطا را کمینه می‌کند. مقایسه سه روش الگوریتم پرندگان رایج، الگوریتم پرندگان اصلاح شده، الگوریتم پرندگان بهبود یافته با استفاده از فن برون یابی با روش‌های کلونی مورچه‌ها و زیگلر نیکلز نشان می‌دهد که الگوریتم پرندگان با استفاده از فن برون یابی با داشتن سرعت همگرایی خوب و تعداد گام لازم کمتر برای رسیدن به جواب بهینه روش مناسبی برای بهینه‌سازی، کاربردهای برخط و در حالت خاص برای طراحی کنترلر کننده PID است. الگوریتم ePSO، نیازی به معادله سرعت ندارد و موقعیت ذره مستقیماً با استفاده از فن برون یابی بروز می‌شود. این مسئله از لحاظ محاسباتی نیز ارزشمند است؛ در نتیجه این الگوریتم مناسب برای بهینه‌سازی و بطور خاص کاربردهای کنترلی است.