مجله مهندسی مکانیک، شماره پیاپی ۹۲، جلد ۵۰، شماره ۳، پاییز ۱۳۹۹، صفحه ۸۷-۹۶ – پژوهشی DOI: 10.22034/jmeut.2020.10438

بررسی تحلیلی روش‌های بهینه‌سازی هوشمند مشاهده‌گر اغتشاش تطبیقی نروفازی برای شبیه‌ساز کنترل وضعیت ماهواره

سید محمد رضا رضوی دانشجوی کارشناسی ارشد، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران

حامد شهبازی* استادیار، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران

مریم ملک زاده استادیار، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران

علیرضا آریایی استادیار، گروه مهندسی مکانیک، دانشگاه اصفهان، اصفهان، ایران

چکیده

در این مقاله اثر استفاده از الگوریتم‌های هوشمند مختلف برای بهینه‌سازی مشاهده‌گر اغتشاش تطبیقی نرو فازی موردبررسی قرارگرفته است. ابتدا یک کنترلگر تطبیقی مدل مرجع برای شبیه‌ساز زیرسیستم کنترل وضعیت ماهواره طراحی‌شده است. سپس برای تضعیف اثر اغتشاش از مشاهده‌گرهای اغتشاش تطبیقی نرو فازی استفاده‌شده است. در این مقاله سیستم فازی مربوطه با استفاده از الگوریتم‌های هوشمند ژنتیک، ازدحام ذرات، رقابت استعماری، زنبورعسل، مورچگان و به‌خصوص الگوریتم گرادیان ازدحام ذرات که موجب افزایش سرعت و بهینه‌تر شدن پاسخ می‌گردد، بهینه‌شده است. الگوریتم گرادیان ازدحام ذرات با ترکیب ایده‌هایی از یادگیری تقویتی گرادیان سیاست و روش ازدحام ذرات یک روش ترکیبی بهینه‌سازی برای کنترل یک سیستم پیچیده غیرخطی است که کاربردهای فراوانی در جهان واقعی دارد. در این روش با گرفتن ایده از روش‌های یادگیری تقویتی، گرادیان سیاست برای یک سیستم نرو فازی محاسبه می‌شود و در روابط بهینه‌سازی ازدحام ذرات وارد می‌گردد تا بهینه‌سازی علاوه بر فاکتورهای لحاظ شده در روش‌های ازدحامی در جهت گرادیان سیاست نیز انجام شود. برای بهینه‌سازی پارامترهای سیستم نرو فازی و داده‌های ورودی و خروجی در تابع هزینه در نظر گرفته‌شده است. در انتها نیز سیستم‌های نرو فازی بهینه‌شده توسط الگوریتم‌های مذکور با یکدیگر مقایسه می‌شوند و نشان داده می‌شود که الگوریتم گرادیان ازدحام ذرات نسبت به الگوریتم ازدحام ذرات عملکرد بهتری دارد.

واژه‌های کلیدی: شبیه‌ساز کنترل وضعیت ماهواره، بهینه‌سازی، مشاهده‌گر اغتشاش مودلغزشی تطبیقی نرو فازی، الگوریتم‌های هوشمند، الگوریتم ازدحام ذرات، الگوریتم گرادیان ازدحام ذرات.

Analytical Investigation of Intelligent Optimization Algorithms for Adaptive Neuro-Fuzzy Disturbance Observer for Spacecraft Attitude Control Simulator

S. M. R. Razavi Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

H. Shahbazi Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

M. Malekzadeh Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

A. R. Ariaee Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

Abstract

In this paper, the effect of using various intelligent algorithms to optimize the adaptive neuro-fuzzy disturbance observer has been investigated .First, a model reference adaptive control is designed for the spacecraft simulator. Then, in order to reduce the disturbance effect, an adaptive neuro-fuzzy disturbance observer is used. In this paper, the fuzzy system is optimized using Intelligent Genetic Algorithm, Particle Swarm Optimization, Imperialist Competitive Algorithm, Bee Colony, Ant Colony Optimization, and especially Policy Gradient Particle Swarm Algorithm, which speeds up and optimizes the response. The Policy Gradient Particle Swarm algorithm is a combination of gradient policy reinforcement learning and particle swarming ideas and is a hybrid optimization method to control a nonlinear complex system with many applications in the real world. In this method, influenced by reinforcement idea, the policy gradient for a non-fossilized system is calculated, and in the optimization of particle swarm relations, optimization is performed in addition to the factors included in the congestion methods in the direction of the policy gradient. It is intended to optimize the fuzzy neuro system parameters and input and output data in the cost function. Finally, the neuro-fuzzy systems optimized by these algorithms are compared and it is shown that the gradient particle swarm algorithm performs better than the particle swarm algorithm.

Keywords: Satellite simulation simulator, Optimization, Adaptive neuro fuzzy disturbance observer, Intelligent algorithms, Particle Swarm Algorithm, Gradient Particle Swarm Algorithm.

۱- مقدمه

بهینه‌سازی اهمیت زیادی در بسیاری از شاخه‌های علوم دارد. به‌عنوان‌مثال فیزیکدان‌ها، شیمیدان‌ها و مهندسان علاقه دارند تا یک طرح بهینه برای طراحی یک فرآیند شیمیایی را به کار ببرند و محصول تولیدشده را با داشتن شرایطی مثل هزینه و آلودگی کم، بیشینه کنند. همچنین در برازش غیرخطی مدل و منحنی نیز، به‌نوعی به بهینه‌سازی، نیاز داریم. اقتصاددانان و تحقیق‌کنندگان در عملیات نیز باید جایابی بهینه منابع در جامعه و صنعت را پیدا کنند. روش‌های مطرح‌شده برای بهینه‌سازی می‌توانند در دو دسته عمده طبقه‌بندی

* نویسنده مکاتبه کننده، آدرس پست الکترونیکی: hamedshahbazi@gmail.com

تاریخ دریافت: ۹۶/۰۷/۲۸
تاریخ پذیرش: ۹۸/۱۲/۰۵

شوند؛ بهینه‌سازی محلی و بهینه‌سازی فراگیر یا عام. برای بهینه‌سازی عام، اغلب از روش‌های تکاملی استفاده می‌شود. در سی سال گذشته، نوع جدیدی از الگوریتم‌های تقریب ظهور یافته‌اند که اساساً هدف از آن‌ها ترکیب روش‌های ابتکاری در چارچوب‌های کلان‌تر به‌منظور کاوش کارا و اثربخش فضای جستجو می‌باشد. امروزه از این روش‌ها با عنوان روش‌های فرا ابتکاری[1] نامبرده می‌شود.

در عمل ماهواره‌های در حال چرخش تحت اغتشاشات خارجی مختلف قرار می‌گیرند و معادلات حاکم بر آن‌ها بسیار غیرخطی می‌شوند. روش‌های کنترل مقاوم غیرخطی نسبت به روش‌های خطی دارای عملکرد بهتری است. یکی از مهم‌ترین این روش‌ها روش مود لغزشی هست که به دلیل پیاده‌سازی ساده، عدم حساسیت به عدم قطعیت‌ها و اغتشاشات خارجی یکی از موفق‌ترین روش‌های کنترلی غیرخطی در سیستم‌های همراه با عدم قطعیت هست.

در مرجع [۱] با استفاده از قانون مدل داخلی یک مشاهده‌گر اغتشاش مودلغزشی طراحی‌شده است که دیگر مشکل غیر صفر شدن خطای ماندگار تخمین اغتشاش را برای اغتشاشاتی که با زمان تغییر می‌کنند را نداشت و دیگر نیازی نبود که بهره‌ی سوئیچ زنی از مقدار اغتشاش بیش‌تر باشد بلکه کافی بود تا بهره سوئیچ زنی از اختلاف بین اغتشاش تخمین زده‌شده و اغتشاش وارده بر سیستم یا همان خطای مشاهده‌گر اغتشاش کمتر باشد که این خود باعث کوچک شدن بهره و درنتیجه کاهش پدیده نوسانات ناخواسته شد. وی در این طراحی پایداری سیستم کنترلگر –مشاهده‌گر اغتشاش را به‌وسیله نظریه لیاپانوف اثبات کرد.

یکی دیگر از راه‌های غلبه بر این مشکل طراحی کنترلگر تطبیقی سیستم‌های غیرخطی با استفاده از سیستم‌های فازی است که در سال‌های اخیر بسیار موردتوجه قرارگرفته است. طراحی کنترلگرهای تطبیقی مقاوم با استفاده از سیستم‌های تخمین‌گر فازی برای یک گروه از سیستم‌های غیرخطی تک ورودی_تک خروجی با عدم قطعیت پارامتری و عدم قطعیت‌های غالب ناشی از جملات غیرخطی در [۲] بیان گردیده است. در [۳] یک روش کنترلگر مقاوم فازی برای کنترل مدل‌های فازی تاگاکی سوگنو گسسته زمانی یا پیوسته زمانی با عدم قطعیت پارامتری توسعه داده‌شده است. این روش برای کنترل سیستم‌های دارای عدم قطعیت سیستم‌های غیرخطی و پیچیده و همین‌طور برای سیستم‌هایی که دقیق مدل‌سازی نشده‌اند، به کار گرفته‌شده است.

در سال ۲۰۰۷ [۴] آتش پز گرگری یک الگوریتم به نام الگوریتم رقابت استعماری برای بهینه‌سازی مسائل با الهام گرفتن از رقابت استعماری معرفی نمود. وی سپس در سال توانایی این روش را نسبت به روش‌های دیگری چون کلونی مورچگان را نشان داد. وی سپس تحقیقات و مسائل بهینه‌سازی بسیاری را با استفاده از این الگوریتم به انجام رسانید که از آن جمله می‌توان به طراحی کنترلگر PID بهینه، تنظیم پارامترهای بهینه توابع عضویت سیستم کنترلی فازی یک وسیله نقلیه در سال ۲۰۰۸ به انجام رسانید. وی در همین سال الگوریتم مذکور را برای بهینه‌سازی یک سیستم فازی روبات نیز به کاربرد.

در سال ۲۰۱۲ شهبازی [۵] با استفاده از روش گرادیان سیاست یک روش آموزش برای پیاده‌روی دوطرفه روبات انسان‌نما رو یک‌خم معرفی نمود. او این کار را با ارائه یک تولیدکننده الگوی مرکزی قابل‌برنامه‌ریزی که با استفاده از آموزش گرادیان سیاست آموزش‌دیده است، انجام داد. در سال ۲۰۰۹ چروبینی[2] [۶] دو روش یادگیری برای راه رفتن روبات را بر اساس الگوریتم یادگیری گرادیان سیاست را باهم مقایسه کرد. وی نشان داد که گسترش یک الگوریتم گرادیان سیاست خطی که با توجه به پارامتر حساب می‌گردد، زمانی که تنها چند آزمایش انجام می‌شود، برای راه‌حل‌های بهتر، امکان‌پذیر است. وی همچنین به‌طور تجربی نشان داد، اثربخشی روش یادگیری خطی و میزان همگرایی این روش هنگامی‌که ارتباط پارامترها در موقع یادگیری در نظر گرفته می‌شود، بیش‌تر می‌باشد.

ساهو یک کنترلگر PID فازی برای یک سیستم قدرت چند منبعی توسط الگوریتم تکامل تفاضلی را بهینه کرد. وی در انتها نشان داد، حتی زمانی که شرایط بار دچار تغییرات زیادی گردد نیاز به تغییر پارامترهای کنترلگر بهینه نیست [۷].

در سال ۲۰۱۶، کومر پال از الگوریتم دینامیک ازدحام ذرات برای بهینه‌سازی تابع عضویت سیستم فازی استفاده کرد [۸]. این الگوریتم توانایی فاصله گرفتن از نقاط بهینه محلی را دارد و می‌تواند مجموعه‌ای از پارامترها را برای ایجاد یک سیستم کنترلی ایجاد کند. وی عملکرد بهتر این سیستم نسبت به سیستمی که توابع عضویت آن به‌صورت اکتشافی به‌دست می‌آیند را برای پارک کردن یک خودرو در گاراژ موردبررسی قرارداد.

در سال ۲۰۰۸ کارابوگا الگوریتم زنبورعسل را با الگوریتم‌های تفاضل محدود و ازدحام ذرات مقایسه کرد و کارایی آن را نشان داد [۹]. در مرجع [۱۰] یک الگوریتم بهینه‌سازی ABC (ABCISB) برای آموزش خودکار پارامترهای شبکه عصبی مصنوعی پسخور (یک شبکه معمولی عصبی) پیشنهادشده است. در الگوریتم ABC پیشنهادی، از اطلاعات همسایگان با عملکرد بهتر برای تسریع همگرایی زنبورهای شاغل و مشرف استفاده می‌شود. علاوه بر این، یک استراتژی جدید انتخاب باهدف ارتقاء قابلیت جستجو و تعادل در اکتشاف و بهره‌برداری به‌طور جداگانه معرفی می‌شوند.

جوانگ در با استفاده از الگوریتم مورچگان در فضای پیوسته تعداد قوانین فازی تعیین و تمام پارامترهای آزاد در هر قاعده‌ی فازی را بهینه کرد. وی با مقایسه نتایج این الگوریتم با الگوریتم‌های ژنتیک، ازدحام ذرات و ترکیبی آن‌ها عملکرد بهتر الگوریتم مورچگان را نسبت به الگوریتم‌های دیگر نشان داد [۱۱].

در سال ۲۰۰۷ چگینی، یک روش جدید برای بهینه‌سازی سیستم کنترلی فازی که هم تابع عضویت و هم قواعد را در نظر می‌گرفت بر اساس الگوریتم ژنتیک ارائه داد و نشان داد که عدم تقارن توابع عضویت و قواعد فازی منجر به بهینه‌سازی بهتر سیستم کنترلی فازی می‌گردد. وی تأثیر بهینه‌سازی سیستم کنترلی فازی را برای حالت‌های مختلف سیستم را تجزیه‌وتحلیل کرد و نشان داد که قابلیت اطمینان کنترل‌کننده درحالی‌که حالت‌های بیشتری از سیستم را در بهینه‌سازی

[1] Metaheuristic

[2] Cherubini

بهبود می‌بخشد به‌طور قابل‌توجهی بهبود می‌یابد[۱۲]. در سال ۲۰۱۳ دایاز[1] یک الگوریتم ژنتیک ساده برای بهینه‌سازی یک مجموعه قواعد کنترلگر فازی را به کاربرد، سپس آن را بر روی یک سروو سیستم جریان مستقیم به کاربرد [۱۳].

سافیشکومار در سال ۲۰۱۷ برای موتور القایی سه فاز کنترل‌کننده‌های شبکه عصبی و نرو فازی را طراحی کرد. وی نشان داد که کنترل‌کننده نروفازی نسبت به کنترل‌کننده شبکه عصبی در شرایط کاری و همچنین اغتشاشات محیط به‌خوبی کار می‌کند [۱۴]. در [۱۵] توابع عضویت سیستم فازی بهینه‌شده است که یکی از قسمت‌های مهم طراحی سیستم‌های فازی بهینه می‌باشد. وی ابتدا از یک الگوریتم تکاملی چندهدفه ژنتیک که با یک مکانیزم تطبیقی تنظیم اندازه جمعیت و احتمال همگذری ترکیب‌شده توابع عضویت مذکور را بهینه کرد. سپس از آن برای کنترل الاستیک ابر استفاده کرد. وی نشان داد عملکرد سیستم فازی مذکور بهتر از عملکرد سیستم‌های فازی موجود هستند.

در مرجع [۱۶] مفیو با ترکیب گرادیان منفی که ضعف قرار گرفتن در مینیمم‌های محلی را داشت و الگوریتم بهینه‌سازی ازدحام ذرات که تلاش‌های محاسباتی فراوانی را می‌طلبید، الگوریتم جدیدی را به‌دست آورد که نه‌تنها با سرعت و همین‌طور شتاب بالا به پاسخ نهایی می‌رسید، بلکه در نقاط بهینه محلی نیز قرار نمی‌گرفت.

ملک‌زاده در سال ۲۰۱۷ با استفاده از نظریه مودلغزشی مرتبه بالا، برای تخمین سرعت زاویه‌ای از یک مشاهده‌گر حالت فوق پیچشی استفاده کرد. این روش برای کاهش پدیده‌ی نوسانات ناخواسته بدون افزایش تلاش کنترلی با توجه به مومنتوم و نرخ آن بسیار مفید می‌باشد. مقاومت و اثربخشی طرح پیشنهادی خود را در شرایط عدم قطعیت و اغتشاش با آزمایش بر روی شبیه‌ساز بررسی کرد [۱۷]. وی همچنین در سال ۲۰۱۸ دو نوع کنترل مقاوم سنتز و فوق پیچشی تطبیقی را روی سیستم شبیه‌ساز ماهواره پیاده‌سازی کرد و نشان داد الگوریتم فوق پیچشی تطبیقی باعث کاهش پدیده‌ی نوسانات ناخواسته بدون افزایش تلاش کنترلی می‌گردد [۱۸]. وی همچنین در سال ۲۰۱۹ با در نظر گرفتن مومنتوم‌های زاویه‌ای به‌عنوان متغیرهای داخلی حالت، لزوم کار کردن موتورها در زمان رسیدن و قرار گرفتن سیستم در وضعیت نهایی را از بین برد [۱۹].

در این مقاله، با استفاده از الگوریتم‌های تکاملی متنوع و بخصوص الگوریتم ترکیبی گرادیان ازدحام ذرات یک سیستم نرو فازی موجود در مشاهده‌گر اغتشاش تطبیقی نرو فازی بهینه‌شده و نتایج آن‌ها با یکدیگر مقایسه شده است. درواقع نوآوری مقاله در این است که با ترکیب الگوریتم ازدحام ذرات با گرادیان سیاست، الگوریتمی ترکیبی که دارای عملکرد بهتری نسبت به الگوریتم ازدحام ذرات معمولی می‌باشد، معرفی می‌کند و همچنین کارایی الگوریتم‌های مختلف را با مقایسه پاسخ سیستم نروفازی بهینه‌شده توسط آن‌ها با یکدیگر مقایسه می‌کند.

در ادامه مقاله، بخش دوم معادلات سینماتیکی و دینامیکی سیستم ماهواره استخراج می‌گردد. سپس در بخش سوم کنترلگر تطبیقی مدل مرجع با استفاده از معیار پایداری لیاپانوف بیان می‌گردد. در بخش چهارم یک مشاهده‌گر اغتشاش تطبیقی نرو فازی طراحی می‌گردد. در بخش پنجم به معرفی مختصر در مورد الگوریتم‌های تکاملی و گرادیان سیاست پرداخته می‌شود. بخش ششم نیز به شبیه‌سازی کنترلگر و مشاهده‌گرهای اغتشاش تطبیقی نروفازی برای سیستم شبیه‌ساز ماهواره پرداخته می‌شود و با استفاده از الگوریتم‌های هوشمند ژنتیک، ازدحام ذرات، رقابت استعماری، زنبورعسل، مورچگان و به‌خصوص الگوریتم گرادیان ازدحام ذرات سیستم نرو فازی بهینه‌سازی می‌گردد. در بخش هفتم به مقایسه و نتیجه‌گیری در مورد سیستم‌های بهینه‌شده توسط الگوریتم‌های تکاملی مذکور پرداخته می‌شود.

۲- معادلات سینماتیکی و دینامیکی شبیه‌ساز

شکل۱ شبیه‌ساز زیرسیستم کنترل وضعیت ماهواره را نشان می‌دهد که شامل بخش‌هایی چون پلتفرم (عملگرها، مکانیزم برقراری تعادل، حسگر تعیین وضعیت و سایر تجهیزات الکتریکی و الکترونیکی از قبیل رایانه، باتری در قسمت‌های مختلف آن قرار گرفته است)، پایه‌هایی برای نگه‌داری میز در حالت بی‌باری، یاتاقان هوایی نیم‌کروی و همچنین پایه نگه‌دارنده مجموعه می‌باشد. در ادامه به تشریح معادلات سینماتیکی و دینامیکی شبیه‌ساز ماهواره پرداخته می‌شود.

شکل ۱- شبیه‌ساز زیرسیستم کنترل وضعیت ماهواره [۱۹]

با استفاده از معادله زیر مومنتوم زاویه‌ای کل شبیه‌ساز ماهواره به دست می‌آید:

$$\vec{H} = I\,\vec{\omega} + \vec{h}_{\omega} \qquad (۱)$$

در این رابطه I ممان اینرسی، ω سرعت زاویه‌ای شبیه‌ساز و h_w اندازه حرکت چرخ‌های عکس‌العملی است. اگر این رابطه در معادله اویلر در مختصات بدنی جایگذاری گردد، می‌توان نوشت [۲۰]:

$$T = \dot{\vec{H}} + \vec{\omega} \times \vec{H} \qquad (۲)$$

T گشتاورهای اعمالی بر شبیه‌ساز ماهواره می‌باشد. درنهایت با توجه به مرجع [۱۹] معادلات دینامیکی شبیه‌ساز ماهواره به‌صورت زیر به دست می‌آید:

$$\dot{\vec{\omega}} = I^{-1}\vec{T} - I^{-1}(\vec{\omega} \times I\vec{\omega}) + I^{-1}(mgr_s) \times K \qquad (۳)$$

۳- طراحی کنترلگر تطبیقی مدل مرجع

در این مقاله برای نشان دادن و مقایسه عملکرد مشاهده‌گرهای

[1] Diaz

اغتشاش کلاسیک و فازی از یک کنترلگر تطبیقی مدل مرجع استفاده‌شده است. در ابتدا و در حین طراحی کنترلگر، اغتشاش وارد بر ورودی سیستم چه برای مدل مرجع و چه برای سیستم شبیه‌ساز ماهواره صرف‌نظر شده است؛ بنابراین معادلات دینامیکی شبیه‌ساز ماهواره در فرم همراه در ابتدا به‌صورت زیر در نظر گرفته می‌شود:

$$\ddot{y} = f + bu \tag{۴}$$

y معادل موقعیت‌های زاویه‌ای شبیه‌ساز زیرسیستم کنترل وضعیت ماهواره در نظر گرفته‌شده است. همچنین f و b با توجه به رابطه (۳) به‌صورت زیر به دست می‌آیند:

$$\begin{cases} f = \dot{R}(\theta)\omega - R(\theta)I^{-1}(\omega \times I\omega) \\ b = R(\theta)I^{-1} \end{cases} \tag{۵}$$

ورودی کنترلی به‌صورت زیر در نظر گرفته می‌شود:

$$u_n = \frac{1}{\hat{b}}(-\hat{f} + \ddot{y}_d - \lambda(t)\dot{\tilde{y}} - K(t)\mathrm{sgn}(\sigma_c)) \tag{۶}$$

که در آن $\tilde{y} = y - y_d$ هست. y_d مقدار مطلوب زوایای اویلر و $\hat{f}$ تخمینی از f و $\hat{b}$ تخمینی از b و σ_c *سطح لغزش* می‌باشد. با جایگذاری رابطه (۶) در (۴) می‌توان نوشت:

$$\ddot{y} = \hat{f} + \hat{b}\left(\frac{1}{\hat{b}}(-\hat{f} + \ddot{y}_d - \lambda(t)\dot{\tilde{y}} - K(t)\mathrm{sgn}(\sigma_c))\right) \tag{۷}$$

برای مدل مرجع در ابتدا (هنگام طراحی کنترلگر)، معادلات همراه به‌صورت زیر به دست خواهند آمد:

$$\ddot{y}_m = f_m + b_m u_c \tag{۸}$$

که در آن u_c به‌صورت زیر در نظر گرفته‌شده است:

$$u_c = \frac{1}{\hat{b}_m}(-\hat{f}_m + \ddot{y}_d - \lambda_m\dot{\tilde{y}} - K_m\mathrm{sgn}(\sigma_c)) \tag{۹}$$

در این مقاله $\hat{b}_m = \hat{b}$ و $\hat{f}_m = \hat{f} - (mgr_s) \times K$ می‌باشد، همچنین خروجی y_m معادل موقعیت‌های زاویه‌ای در نظر گرفته‌شده است، بنابراین خطای پاسخ سیستم از رابطه زیر به دست می‌آید:

$$e = y - y_m \tag{۱۰}$$

که در آن e خطای تنظیم هست. با ساده‌سازی رابطه (۷) رابطه‌ی زیر نتیجه می‌شود:

$$\ddot{y} = \ddot{y}_d - \lambda(t)(\dot{y} - \dot{y}_d) - K(t)\mathrm{sgn}(\sigma_c) \tag{۱۱}$$

با استفاده از معیار و روش پایداری لیاپانوف برای کنترل تطبیقی مدل مرجع [۲۳] مراحل زیر طی می‌شود:

$$\begin{aligned} \ddot{e} = \ddot{y} - \ddot{y}_m &= (\lambda_m - \lambda(t))\dot{y} - \dot{e}\lambda_m \\ &+ (\lambda(t) - \lambda_m)\dot{y}_d + (K - K(t))\mathrm{sgn}(\sigma_c) \end{aligned} \tag{۱۲}$$

که در این رابطه λ_m مقدار مطلوب λ و K نیز مقدار مطلوب $K(t)$ می‌باشد. هدف از کنترل در سیستم ماهواره، کنترل وضعیت ماهواره می‌باشد به‌گونه‌ای که حالات سیستم و درواقع زوایای اویلر به مقادیر مطلوب خود میل کنند. در این صورت فرض بر آن است که تمام حالات سیستم قابل‌اندازه‌گیری هستند. معادله دینامیکی خطا به‌صورت زیر نوشته می‌شود:

$$\ddot{e} + \dot{e}\lambda_m = \Delta\lambda\dot{u}_c - \Delta\lambda\dot{y} + \Delta K\mathrm{sgn}(\sigma_c) \tag{۱۳}$$

که در آن $\Delta K = K - K(t)$ و $\Delta\lambda = \lambda_m - \lambda(t)$ می‌باشد. تابع لیاپانوف به‌صورت زیر انتخاب می‌شود:

$$V_c = \frac{1}{2}\dot{e}^2 + \frac{1}{2\gamma_1}\Delta\lambda^2 + \frac{1}{2\gamma_2}\Delta K^2 \tag{۱۴}$$

که در آن γ_2 و γ_1 بزرگ‌تر از صفر هستند. با مشتق گرفتن از رابطه (۱۴) نسبت به زمان، پس از انجام محاسبات درنهایت می‌توان نوشت:

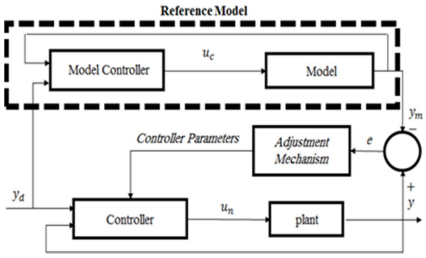
$$\begin{aligned} \dot{V}_c &= \dot{e}\Delta\lambda\dot{u}_c - \Delta\lambda\dot{e}\dot{y} + \dot{e}\Delta K\mathrm{sgn}(\sigma_c) - \dot{e}^2\lambda_m \\ &+ \left(\dot{e}\dot{u}_c - \dot{e}\dot{y} + \frac{1}{\gamma_1}\Delta\dot{\lambda}\right)\Delta\lambda + (\dot{e}.\mathrm{sgn}(\sigma_c) + \frac{1}{\gamma_2}\Delta\dot{K})\Delta K \end{aligned} \tag{۱۵}$$

با توجه به اینکه λ_m بزرگ‌تر از صفر است، بنابراین جمله‌ی $-\dot{e}^2\lambda_m$ همواره کوچک‌تر از صفر است، لذا برای منفی معین شدن مشتق تابع لیاپانوف باید بقیه‌ی جملات برابر با صفر گردند:

$$\begin{aligned} &\dot{e}\dot{y}_d - \dot{e}\dot{y} + \frac{1}{\gamma_1}\Delta\dot{\lambda} = 0 \\ &\rightarrow \Delta\dot{\lambda} = (\dot{e}\dot{y}_d - \dot{e}\dot{y})\gamma_1 \\ &\rightarrow \lambda(t) = \gamma_1\int(\dot{e}\dot{y}_d - \dot{e}\dot{y})dt + \lambda_0 \end{aligned} \tag{۱۶}$$

$$\begin{aligned} &\dot{e}.\mathrm{sgn}(\sigma_c) + \frac{1}{\gamma_2}\Delta\dot{K} = 0 \\ &\rightarrow \Delta\dot{K} = -\gamma_2\dot{e}.\mathrm{sgn}(\sigma_c) \\ &\rightarrow K(t) = -\gamma_2\int\dot{e}.\mathrm{sgn}(\sigma_c)\,dt + K_0 \end{aligned} \tag{۱۷}$$

با توجه به روابط می‌توان مکانیزم تطبیقی موردنیاز برای تعیین پارامترها را در هرلحظه از زمان استخراج کرد. شکل۲ بلوک دیاگرام کنترلگر تطبیقی مدل مرجع را (بدون در نظر گرفتن اغتشاش وارده) نشان می‌دهد.



شکل۲- بلوک دیاگرام کنترلگر تطبیقی مدل مرجع

۴- شبیه‌سازی مشاهده‌گر اغتشاش نرو فازی تطبیقی

قانون زیر را برای تخمین اغتشاش در نظر گرفته می‌شود [۱]:

$$\begin{cases} \dot{\hat{\xi}} = A\hat{\xi} + g\varphi_d\mathrm{sgn}(\sigma_d) + P^{-1}C^T b\sigma_c \\ \hat{d} = C\hat{\xi} \end{cases} \tag{۱۸}$$

که در آن $\hat{\xi}$ تخمین بردار حالت تخمین‌گر اغتشاش و g بردار بهره و P ماتریس بهره می‌باشد که باید مثبت معین متقارن باشد و $\varphi_d\mathrm{sgn}(\sigma_d)$ معادل خطای مشاهده‌گر اغتشاش می‌باشد. هرگاه خطای اغتشاش غیر صفر باشد، سیگنال سوئیچ زنی مدل داخلی را برای تخمین مجانبی اغتشاش تحریک می‌کند. لازم به ذکر است که ترم $P^{-1}C^T b\sigma_c$ برای پایداری کلی سیستم کنترلگر مشاهده‌گر اغتشاش ضروری است. باید توجه نمود که باید (A, C) مشاهده‌پذیر باشد. در این رابطه g باید به‌گونه‌ای طراحی گردد که $(A - gC)$ هورویتز باشد. سپس برای هر ماتریس مثبت معین متقارن داده‌شده Q، ماتریس مثبت معین متقارن P یافت می‌شود که در شرایط زیر صدق کند.

$$(A - gc)^T P + P(A - gc) = -Q \tag{۱۹}$$

با یافتن ماتریس‌های Q و P و بهره سوئیچ زنی φ می‌توان طبق رابطه (۱۸) اغتشاش وارده بر سیستم را تخمین زد. اثبات پایداری سیستم کنترلگر-مشاهده‌گر اغتشاش همراه با جزئیات در مرجع [۱] وجود دارد.

برای تخمین اغتشاش در مشاهده‌گر اغتشاش از مکانیزمی برای تنظیم بهره‌ی سوئیچ زنی استفاده‌شده است که بر اساس اصول فازی

می‌باشد. این مکانیزم بر اساس خطا و مشتق زمانی خطای مشاهده‌گر اغتشاش و ورودی کنترلی، بهره‌ی مناسب موردنیاز مشاهده‌گر اغتشاش را در هر لحظه تعیین می‌نماید. با توجه به این‌که افزایش φ سرعت همگرایی، میزان نوسانات ناخواسته و همین‌طور تلاش کنترلی را افزایش می‌دهد، در این مقاله با استفاده از مکانیزم فازی تطبیقی سعی در تنظیم φ به‌صورت تابعی متغیر با زمان شده به‌طوری‌که مصالحه‌ای بین محدودیت تلاش کنترلی و افزایش سرعت پاسخ و کاهش پدیده‌ی نوسانات ناخواسته ایجاد گردد. با توجه به اینکه ایجاد قواعد فازی دارای پیچیدگی‌هایی هست، برای رسیدن به این هدف از روش سی مینز و سیستم استنتاج فازی تطبیقی استفاده‌شده است. در روش سی مینز به داده‌ها انباشتگی‌هایی نسبت داده می‌شود که تعداد این انباشتگی j در نظر گرفته می‌شود. داده‌های مربوط به هریک از این انباشتگی‌ها ازنظر ویژگی شبیه به یکدیگر هستند درحالی‌که داده‌های انباشتگی‌های مختلف ازنظر ویژگی در حد امکان هیچ شباهتی به یکدیگر ندارند. در روش سی مینز تابع هدف باید کمینه گردد [۲۴]:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{J} u_{ij}^{m} d_{ij} \qquad (۲۰)$$

که در این رابطه N تعداد داده‌ها، J تعداد انباشتگی‌ها و u_{ij} درجه عضویت y_i در انباشتگی J ام و m وزن هر تابع عضویت فازی u_{ij} و d_{ij} فاصله بین نقطه y_i و مرکز انباشتگی هست که تابع فاصله نامیده می‌شود. فاصله مجذور اقلیدسی در روش سیمینز معمول به‌صورت زیر هست [۲۴]:

$$d_{ij} = \|y_i - \mu_j\|^2 \qquad (۲۱)$$

μ_j نمونه اولیه مرکز انباشتگی j هست. با استفاده از این تابع، الگوریتم سی مینز تحت شرایط لازم برای کمینه کردن تابع تکرار می‌گردد.

$$\mu_j = \sum_{i=1}^{N} u_{ij}^{m} y_i \Big/ \sum_{i=1}^{N} u_{ij}^{m} \qquad (۲۲)$$

$$u_{ij} = (d_{ij})^{1/(1-m)} \Big/ \sum_{h=1}^{J} (d_{ih})^{1/(1-m)} \qquad (۲۳)$$

که در آن محدودیت $\sum_{j=1}^{J} u_{ij} = 1$ اعمال گردیده است.

انفیس یک سیستم استنتاج فازی است که در چارچوب شبکه‌های تطبیقی اجرا می‌شود. انفیس یک سیستم یکپارچه از هر دو شبکه عصبی مصنوعی و سیستم استنتاج فازی است [۲۴]. شکل۳ ساختار یک شبکه نرو فازی را نشان می‌دهد. برای استفاده از استنتاج‌های فازی در سیستم‌های کنترلی باید از سیستم‌های فازی‌ساز و غیر فازی‌ساز نیز در مجموعه‌های ورودی، خروجی استفاده کرد. یک سیستم فازی از ۵ بلوک اصلی تشکیل‌شده است که شامل بلوک‌های تصمیم‌گیری که عملیات استنتاج قواعد را شکل می‌دهد، بلوک قواعد، داده‌ها و بلوک-های فازی‌ساز و نافازی‌ساز می‌باشد. قواعد به‌وسیله یک متخصص تعیین می‌گردند؛ در مواردی که این امکان وجود ندارد، انباشتگی‌ها این وظیفه را انجام می‌دهند.

همان‌گونه که شکل۳ نشان می‌دهد، سیستم نرو فازی شامل ۵ لایه است. هر گره در لایه اول یک گره سازگار است که با یک تابع عضویت فازی خاص که مشخص‌کننده درجه‌هایی که ورودی‌ها آن را برآورده می‌کنند، می‌باشد. در لایه دوم گره‌ها در سیگنال ورودی ضرب شده و نتیجه را به لایه سوم می‌دهد. در لایه سوم وزن نرمالیزه شده گره i ام به‌صورت زیر محاسبه می‌کند:

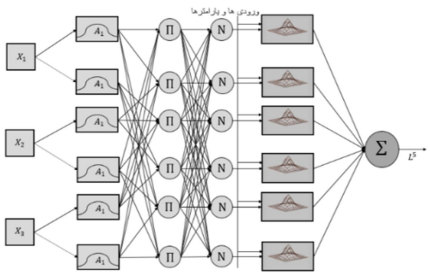
$$L_i^3 = \overline{w_i} = \frac{w_i}{\sum_{i=1}^{n} w_i} \qquad (۲۴)$$

که در آن $\overline{w_i}$ به‌عنوان وزن نرمالیزه شده می‌باشد. لایه چهارم یک گره سازگار می‌باشد که به‌وسیله یک تابع به‌صورت زیر نشان داده می‌شود:

$$L_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \qquad (۲۵)$$

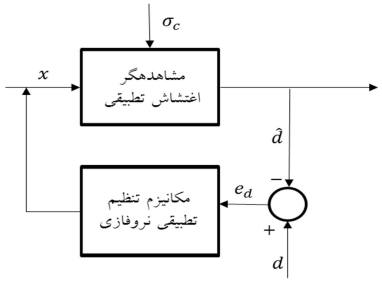
که در آن x ورودی، y خروجی و p_i، r_i، q_i مجموعه پارامترهای سیستم استنتاجی تاگاکی سوگنو می‌باشد. لایه پنجم یک گره تکی ثابت است که خروجی کلی را به‌صورت زیر محاسبه می‌کند.

$$L^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (۲۶)$$



شکل۳- ساختار یک شبکه نرو فازی

در این مقاله تابع $\varphi(t)$ که با زمان تغییر می‌کند، به روش سعی و خطا، تعریف‌شده است. سپس با توجه به تلاش کنترلی و همچنین خطای مشاهده‌گر اغتشاش و میزان تغییرات آن، به‌عنوان مجموعه‌های ورودی و تابع $\varphi(t)$ به‌عنوان مجموعه مطلوب در نظر گرفته می‌شود. استفاده از این روش باعث هوشمندی و همچنین آزادی عمل بیشتری نسبت به روش‌های کلاسیک می‌شود. در این روش خروجی تابع ابتدا توسط یک ناظر بر سیستم برای مشاهده‌گر اغتشاش کلاسیک فرستاده می‌شود. بنابراین دارای آزادی عمل بیشتری برای افزایش خروجی تابع و درنتیجه افزایش سرعت پاسخ و همگرایی خطای مشاهده‌گر اغتشاش نسبت به مشاهده‌گرهای اغتشاش کلاسیک دارد. همچنین $\varphi(t)$ به‌گونه‌ای توسط ناظر مجازی انتخاب می‌شود که در شرایط پایداری تعیین‌شده در روش کلاسیک نیز صدق کند، بنابراین از پایداری سیستم نیز اطمینان حاصل‌شده است. شکل۴ بلوک دیاگرام مربوط به سیستم کنترلی بر پایه مشاهده‌گر اغتشاش نرو فازی تطبیقی را نشان می‌دهد.



شکل۴- ساختار یک کنترل‌کننده همراه با مشاهده‌گر اغتشاش مودلغزشی تطبیقی نرو فازی

۵- الگوریتم‌های تکاملی

الگوریتم رقابت استعماری از شبیه‌سازی فرایند سیاسی اجتماعی

استعمار به‌دست‌آمده است. در ابتدا یک سری کشور داریم که هریک ویژگی‌های مخصوص به خود را دارند که درواقع همان ویژگی‌هایی هستند که در مسئله بهینه‌سازی به دنبال بهینه کردن آن ویژگی‌ها هستیم. از بین این کشورها برخی از آن‌ها بهترند و به همین دلیل به‌عنوان استعمارگر شناخته‌شده و سعی در جذب مستعمره به سمت خود را دارند. هر استعمارگر به همراه مستعمراتی که در اطرافش وجود دارد تشکیل یک امپراتوری را می‌دهد که هدفشان درنهایت بهینه کردن وضعیت کل امپراتوری می‌باشد. برای ایجاد تنوع در پاسخ‌ها، مستعمره‌ها شروع به تغییر جهت ناگهانی می‌کنند که به آن پدیده‌ی انقلاب می‌گویند. افرادی که در داخل یک امپراتوری خاص هستند، دارند با یکدیگر همکاری می‌کنند تا این‌که وضعیت این امپراتوری بهینه شود و درعین‌حال با یکدیگر طبق رابطه موجود رقابت نیز می‌کنند. در این رقابت، امپراتوری که ضعیف‌تر از بقیه است، یک مستعمره را از دست می‌دهد و این مستعمره به‌تصادف از طریق استعمارگران دیگر تصاحب می‌شود به‌گونه‌ای که استعمارگری که امتیازش بیش‌تر است شانس تصاحبش بیش‌تر خواهد بود. این امر از طریق یکی از توابع احتمالی موجود در مرجع [۴] صورت می‌گیرد. در ادامه اگر امپراتوری دارای هیچ مستعمره‌ای نبود، استعمارگر مربوط به این امپراتوری به‌عنوان مستعمره به امپراتوری‌های دیگر وارد می‌گردد و این روند آن‌قدر تکرار می‌گردد تا این‌که تنها یک امپراتوری که دارای بهترین پاسخ که همان استعمارگر مربوط به این امپراتوری است باقی بماند. برخی از روابط این الگوریتم به‌صورت زیر است:

$$T.C._n = Cost(imperialist_n) + \zeta mean\{Cost(colonies\ of\ empire_n)\} \quad (۲۷)$$

$$x \sim U(0, \beta \times d), \theta \sim (-\gamma, \gamma) \quad (۲۸)$$

که در آن‌ها γ پارامتر تنظیم انحراف از مسیر اصلی و θ یک عدد تصادفی با توزیع یکنواخت، d فاصله بین مستعمره و امپراطور و β یک عدد بزرگ‌تر از یک و ζ یک عدد کوچک‌تر از یک می‌باشد. θ هرچه بالاتر باشد تنوع پاسخ‌ها بیشتر و یا انحراف بیش‌تر می‌گردد. هرچند β و θ و ζ مقادیری اختیاری هستند، اما به‌صورت تجربی در مرجع [۴] نشان داده‌شده که $\beta = 2$ و $\theta = \frac{\pi}{4}$ و $\zeta = 0.1$ باعث همگرایی کشورها (پاسخ‌ها) به مینیمم مطلق می‌شود. ضریب فشار نیز عددی مثبت است که میزان سختگیری در انتخاب را نشان می‌دهد و هرچه بزرگ‌تر باشد دقت پاسخ بالاتر اما سرعت الگوریتم پایین‌تر خواهد بود. احتمال انقلاب نیز مشخص می‌کند که در چه کسری از کشورها باید انقلاب روی دهد و ضریب تأثیر انقلاب نشان می‌دهد که چقدر یک کشور خاص متحول می‌گردد.

در الگوریتم زنبورعسل زنبورهای کارگر و ملکه به دنبال منابع غذایی می‌گردند. در الگوریتم زنبورعسل یک مستعمره از زنبورهای عسل برای رسیدن به منابع غذایی مصنوعی (که همانا حل‌های مناسب برای مسئله داده‌شده است) جستجو می‌کند. با به کار بردن این الگوریتم، مسئله بهینه‌سازی ابتدا به یک مسئله یافتن بردار پارامترهایی که یک تابع هزینه را کمینه می‌کند تبدیل می‌گردد. سپس زنبورهای مصنوعی به‌صورت تصادفی یک جمعیتی از بردارهای پاسخ و سپس توسعه آن‌ها به‌وسیله به‌کارگیری روش‌هایی چون حرکت به سمت پاسخ‌های بهتر به‌وسیله یک مکانیزم جستجوی محلی درحالی‌که دیگر پاسخ‌های ضعیف در نظر گرفته نمی‌شوند. روابط مربوط به الگوریتم زنبورعسل در مرجع [۱۰] وجود دارد.

$$F_i = \frac{1}{C_i}, \bar{F} = {}^{1}/_{n} \sum_1^n F_i, d_i = \frac{F_i}{\bar{F}}, n_{Bee}^i = [d_i \times n_0] \quad (۲۹)$$

که در آن‌ها C_i مقدار هزینه به ازای زنبورهای سایت i ام و $\bar{F}$ میانگین عکس مقادیر هزینه و n_0 تعداد زنبورهای اختصاص‌یافته در ابتدا و d_i با توجه به مراجع یک نرمال‌ساز می‌باشد. در این مقاله تابع احتمال جستجوی تصادفی برای یافتن پاسخ‌های بهتر به‌صورت متغیر و به‌صورت سعی و خطا به‌صورت زیر به‌دست‌آمده است.

$$\pi_i = \begin{cases} 0.6 & d_i < 0.9 \\ 0.2 & 0.9 \le d_i < 0.95 \\ 0.05 & 0.95 \le d_i < 1.15 \\ 0 & d_i > 1.15 \end{cases} \quad (۳۰)$$

n_{Bee}^i تعداد زنبورهای اختصاص‌یافته به سایت i ام می‌باشد. در این الگوریتم n_0 معمولاً ۰٫۳ تعداد اسکات‌ها در نظر گرفته می‌شود [۱۰].

در الگوریتم مورچگان ابتدا یک جمعیت اولیه از راه‌حل‌ها به‌صورت تصادفی انتخاب می‌شوند. سپس یک آرشیو از راه‌حل‌ها ایجادشده و تعداد k راه‌حل از راه‌حل‌های بهتر موجود در جمعیت اولیه برحسب توابع برازندگی‌شان در آرشیو جای می‌گیرند. برای تعیین راه‌حل‌های جدید ابتدا یکی از راه‌حل‌های موجود در الگوریتم با یک رابطه احتمالاتی به‌عنوان مورچه راهنما انتخاب می‌شود. جهت این انتخاب احتمالاتی به هر سطر j برحسب تابع برازندگی آن، یک ضریب وزنی به‌صورت زیر نسبت می‌دهیم:

$$\omega_{ij} = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(j-1)^2}{2q^2k^2}} \quad (۳۱)$$

احتمال انتخاب شدن سطر j ام به‌صورت زیر به دست می‌آید:

$$P_j = \frac{\omega_i}{\sum_{r=1}^{k} \omega_r} \quad (۳۲)$$

متغیر q پارامتر الگوریتم است که در آن مقادیر کم q نشان می‌دهد که ترجیحاً بهترین راه‌حل‌ها انتخاب‌شده‌اند و باعث کمترین خطای ممکن می‌گردد و از طرف دیگر سرعت رسیدن به پاسخ نهایی را کاهش می‌دهد [۱۱].

در الگوریتم DE همه جواب‌ها دارای شانس مساوی جهت انتخاب شدن می‌باشند؛ یعنی شانس انتخاب شدن آن‌ها وابسته به مقدار شایستگی آن‌ها نیست، پس‌ازاین که یک جواب جدید با استفاده از یک اپراتور جهش خود-تنظیم و اپراتور ادغام تولید شد، جواب جدید با مقدار قبلی مقایسه می‌شود و در صورت بهتر بودن جایگزین می‌گردد. در این الگوریتم ابتدا عملگر جهش اعمال‌شده و سپس عملگر تقاطع اعمال می‌شود تا بدین‌وسیله نسل جدید ایجاد گردد. برای اعمال عملگر جهش از توزیع خاصی استفاده نمی‌شود. بلکه طول گام جهش برابر با مقدار از فاصله میان اعضای فعلی تعیین می‌شود. رابطه اصلی این الگوریتم به‌صورت زیر می‌باشد:

$$y = a + \beta(b - c) \quad (۳۳)$$

که در آن a و b و c سه پاسخ از جمعیت قبلی می‌باشند و β پارامتر مقیاس می‌باشد. مقدار این پارامتر نباید تغییر علامت بدهد، زیرا در این صورت از حالت میانگین صفر بودن خارج می‌گردد و باعث ناپایداری الگوریتم می‌شود. همچنین با کاهش پارامتر مقیاس، طول گام‌های جهش کمتر می‌شود و لذا زمان بیش‌تری برای همگرایی الگوریتم نیاز است، ولی دقت الگوریتم کاهش می‌یابد. همچنین میانگین خطا در تکرار ثابت افزایش می‌یابد. همچنین با توجه به مرجع

[۷] مفهوم احتمال تقاطع نیز در این الگوریتم وجود دارد که بیش‌تر شدن این احتمال باعث افزایش سهم پاسخ‌های موقت در ایجاد پاسخ-های جدید خواهد شد.

الگوریتم‌های ژنتیک یکی از الگوریتم‌های جستجوی تصادفی است و ایده آن برگرفته از طبیعت می‌باشد. الگوریتم‌های ژنتیک برای روش‌های بهینه‌سازی کلاسیک در حل مسائل خطی، محدب و برخی مشکلات مشابه بسیار موفق بوده‌اند ولی الگوریتم‌های ژنتیک برای حل مسائل گسسته و غیرخطی بسیار کاراتر می‌باشند. در طبیعت از ترکیب کروموزوم‌های بهتر، نسل‌های بهتری پدید می‌آیند. گاهی اوقات جهش‌هایی نیز در کروموزوم‌ها روی می‌دهد که ممکن است باعث بهتر شدن نسل بعدی شوند. الگوریتم ژنتیک نیز با استفاده از این ایده اقدام به حل مسائل می‌کند. روابط مربوط به الگوریتم ژنتیک در [۱۳] وجود دارد. البته الگوریتم ژنتیک یک جواب خوب پیدا می‌کند ولی ممکن است جواب بهینه را پیدا نکند. پشتوانه ریاضی ضعیفی دارد و ممکن است در دو بار اجرای مختلف، جواب‌های متفاوتی دریافت کرد.

الگوریتم ازدحام ذرات جزو الگوریتم‌های فراابتکاری هست که با کمترین اطلاعات ممکن مسئله را حل می‌کند (جزو الگوریتم‌های تکاملی) به دلیل این‌که یک مکانیزم تکامل دهنده دارد تکرار می‌شود. این الگوریتم در شاخه‌ی اصلی هوش ازدحامی جای می‌گیرد. اصل آن به اشتراک‌گذاری اطلاعات و تجربیات می‌باشد (برخلاف الگوریتم ژنتیک که در آن چیزی به‌عنوان همکاری یا جریان اطلاعاتی وجود ندارد. هرکدام از والدین از وضعیت والد دیگر خبر ندارد) برای اینکه که اعضای جمعیت با همدیگر همکاری کنند نیاز به ارتباط وجود دارد. هر ذره در فضا دارای یک موقعیت است، چون دارای موقعیت است. این موقعیت به صورت‌های مختلفی نظیر بردار یا عدد قابل توصیف می‌باشد، متناظر با این موقعیت می‌توان معیار مربوط به این موقعیت نیز حساب شود و متناظر با این موقعیت، مقدار تابع هدف را نیز داریم که هرکدام یک‌جهت حرکت (بردار) نیز دارند که در حالت کلی آن را به اسم سرعت می‌شناسیم. بهترین موقعیت تجربه‌شده و به‌تبع آن مقدار تابع هدف متناظر با آن موقعیت را نیز باید داشته باشیم. باید توجه داشت که گام حرکت هرچقدر کوچک‌تر باشد نتایج دقیق‌تر، ولی مدت‌زمان بیش‌تری طول می‌کشد. روابط مربوط به الگوریتم ازدحام ذرات در منبع [۸] آمده است.

روش‌های گرادیان سیاست، نوعی از فن‌های یادگیری تقویتی هستند که سعی می‌کنند یک سیاست پارامتری شده را بر اساس امید بازگشت (پاداش انباشته‌شده در درازمدت) با استفاده از گرادیان کاهنده، بهینه‌سازی کنند. مزایای روش گرادیان سیاست برای کاربردهای واقعی زیاد است. مهم‌ترین آن‌ها این است که نمایش سیاست می‌تواند به‌گونه‌ای انتخاب شود که عمل تحت یادگیری را به‌خوبی تعریف نماید و از دانش مرتبط با عمل تحت یادگیری به‌خوبی بهره گیرد، به‌نحوی‌که حتی تعداد پارامترهای مورداستفاده نسبت به روش‌های مبتنی بر یادگیری تابع مقدار، کمتر باشد. همچنین روش‌های گرادیان سیاست، به دلیل خاصیت عمومی بودنشان، می‌توانند هم به‌صورت مبتنی بر مدل و هم به‌صورت آزاد از مدل، مورداستفاده قرار گیرند. همچنین روش‌های سیاستی برخط هستند. درنهایت روش‌های گرادیان سیاست همیشه جهت تعریف پارامترها و نرخ یادگیری که هر دو، سرعت همگرایی را مشخص می‌کنند که این مزایا روش‌های جدیدی را برای بیشینه کردن امید بازگشت به وجود آورده است. بنا به دلایل بالا و مزایایی که برای روش‌های گرادیان سیاست ذکر شد، این روش‌ها به شکل خاص جهت به‌کارگیری در مسائل مختلف بسیار موردتوجه قرارگرفته‌اند. برای حل مسئله یادگیری پارامترهای سیستم نرو فازی در ابتدا رفتار سیستم نرو فازی را در قالب یک مسئله یادگیری تقویتی (عامل-نقاد) تعریف می‌کنیم [۲۵].

روش‌های یادگیری تابع سیاست یک دسته مهم از روش‌های تغییر تابع سیاست محسوب می‌شوند که به‌طور مستقیم یک سیاست بهینه را در فضای سیاست جستجو می‌کنند در ابتدا از یک سیاست اولیه π_0 شروع به کار می‌کند و با آزمایش و تولید عمل‌ها در هرلحظه یک دنباله‌ای از پاداش‌های R_k را به‌دست می‌آورد که مبنای تغییر سیاست قرار می‌گیرند. با تغییر سیاست در هر دوره سعی می‌شود تا به سمت سیاست بهینه π^* که مجموع پاداش‌های (کاهش‌یافته در زمان) آن بیشینه است حرکت شود. تابع مجموع پاداش‌های (کاهش‌یافته در زمان) به شکل زیر تعریف می‌شود [۲۵]:

$$V^{\pi}(x) = E(\sum_{k=1}^{\mu} \rho^k R_k) \qquad (۳۴)$$

در اینجا $V^{\pi}(x)$ مقدار مورد انتظار مجموع کل پاداش‌های به‌دست‌آمده با اعمال سیاست π در طول آزمایش π مقدار نمونه‌برداری شده و $\rho \in [0,1]$ اعمال ضریب کاهش هست. از نماد E برای نمایش مقدار مورد انتظار استفاده می‌شود که معمولاً از طریق آزمایش چندگانه و میانگین‌گیری پیاده‌سازی می‌گردد. درروش مورداستفاده که برگرفته از یادگیری گرادیان سیاست است، تابع سیاست با یک بردار پارامتر θ پارامتری می‌شود و آن را به شکل π_θ نمایش می‌دهیم. هدف اصلی بهینه‌سازی سیاست π_θ بر اساس بهینه‌سازی بردار پارامتر θ است. با یافتن بهترین بردار پارامتر θ^* سیاستی را پیدا خواهیم کرد که بیشترین مقدار به دست آید. به دلیل وابستگی این مقدار به بردار پارامتر θ در ادبیات روش‌های گرادیان سیاست $V^{\pi_\theta}(x)$ را به شکل مختصر با $J(\theta)$ نمایش می‌دهند. $J(\theta)$ مقدار مجموع پاداش‌های کلی به‌دست‌آمده با استفاده از بردار پارامتر θ در سیاست π_θ را مشخص می‌کند. به‌طور مختصر این تابع ارزش بردار θ می‌نامیم. روش‌های گرادیان سیاست، گرادیان این تابع نسبت به θ (∇J) را محاسبه می‌کنند و θ را در جهت این گرادیان تغییر می‌دهند تا به بردار پارامتر بهینه θ^* برسند. در هر دوره گرادیان سیاست در یک نرخ یادگیری α_h ضرب شده و با بردار پارامتر قبلی θ_h جمع می‌شود. تا بردار پارامتر دوره بعدی θ_{h+1} به دست آید. این محاسبه در معادله زیر آمده است [۲۵]:

$$\theta_{h+1} = \theta_h + \alpha_h . \nabla J \qquad (۳۵)$$

در حالت کلی همه‌ی روش‌های یادگیری گرادیان سیاست مرتباً گرادیان را در دوره محاسبه کرده و بردار پارامتر θ را بروز رسانی می‌کنند. الگوریتم آن‌قدر ادامه می‌یابد تا به یک بردار پارامتر بهینه θ^* همگرا گردد. تفاوت روش‌های مختلف گرادیان سیاست در شیوه برآورد بردار گرادیان ∇J است. تخمین زدن ∇J معمولاً نیازمند انجام یک دوره آزمایش و انجام محاسبات پیچیده است. در روش ارائه‌شده از ترکیب یادگیری تقویتی گرادیان سیاست با الگوریتم تکاملی ازدحام ذرات برای رسیدن به یک بهینه‌ساز بهتر استفاده می‌شود که از رابطه‌ی زیر به

دست می‌آید:

$$V_{ij}(t+1) = wV_{ij}(t) + r_1C_1\left(p_{ij}(t) - x_{ij}(t)\right) + r_2C_2\left(g_{ij}(t) - x_{ij}(t)\right) + r_3C_3(K_{ij}(t) - x_{ij}(t)) \quad (۳۶)$$

که در آن r_3 عدد تصادفی با توزیع یکنواخت و C_3 پارامتر یادگیری می‌باشد. در این رابطه اگر $C_3 = r_3 = 0$ باشد همان رابطه الگوریتم ازدحام ذرات به‌دست می‌آید.

۶- شبیه‌سازی

در این بخش به نتایج شبیه‌سازی کنترلگر تطبیقی مدل مرجع به همراه مشاهده‌گر اغتشاش کلاسیک و مشاهده‌گر اغتشاش تطبیقی نرو فازی به‌صورت جداگانه پرداخته می‌شود. لازم به ذکر است که از داده‌های مربوط به شبیه‌ساز کنترل وضعیت ماهواره که در مرجع [۱۸] آمده است، استفاده‌شده است که به‌صورت زیر می‌باشد:

$$I = \begin{bmatrix} 1.814 & -0.1185 & 0.02749 \\ -0.1185 & 1.735 & 0.01697 \\ 0.02749 & 0.01697 & 3.432 \end{bmatrix}, \quad m = 40\ \text{kg}, r_x = r_y \cong 0, r_z = 8.15 \times 10^{-4}\ \text{m} \quad (۳۷)$$

مقادیر مانور اولیه برحسب درجه به‌صورت زیر در نظر گرفته‌شده است: $\Theta = [30\ 40\ 70]^T$ همچنین مقادیر مطلوب به‌صورت $y_d = [0\ 0\ 0]$ درنظر گرفته‌شده است. مقدار ۱۰ درصد عدم قطعیت برای ماتریس اینرسی در نظر گرفته‌شده است. با توجه به پدیده‌ی اشباع در چرخ‌های عکس‌العملی در طراحی کنترلگر و مشاهده‌گرها محدودیت اندازه حرکت‌زاویه‌ای چرخ‌های عکس‌العملی و نرخ آن‌ها در نظر گرفته‌شده است؛ که مقادیر آن برای اندازه حرکت زاویه‌ای باید در بازه‌ی [-0.625,0.625] و نرخ آن در [-0.123,0.123] باشد. در انجام شبیه‌سازی‌ها فرض بر این است که اغتشاش وارده محدود، متناوب و نوع اغتشاش مشخص و به‌صورت زیر می‌باشد:

$$d(t) = 0.01 + 0.01\sin t \quad (۳۸)$$

با بردن رابطه (۵۱) به فرم (۲۷) ماتریس‌های A و C به‌صورت زیر به دست می‌آیند همچنین ماتریس نیز به‌صورت زیر در نظر گرفته می‌شود و با توجه به رابطه ماتریس به‌صورت زیر به دست می‌آید:

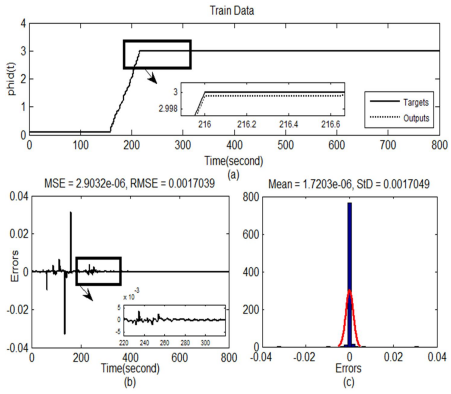
$$A = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$
$$Q = \begin{bmatrix} 8.5 & 0 & 0 \\ 0 & 8.5 & 0 \\ 0 & 0 & 8.5 \end{bmatrix} P = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (۳۹)$$

با استفاده از روش سعی و خطا ماتریس بهره نیز به‌صورت

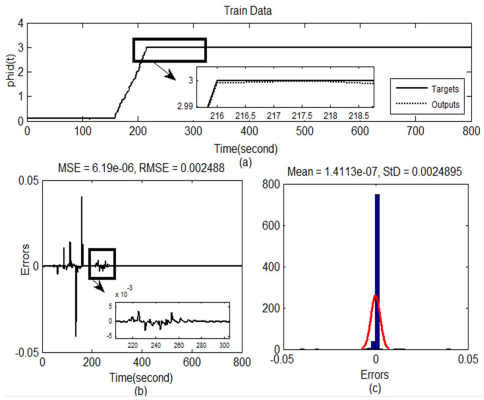
$$\varphi_d = diag[0.07, 0.07, 0.07] \quad (۴۰)$$

به‌دست می‌آید. برای طراحی مشاهده‌گر اغتشاش تطبیقی نرو فازی نیز مراحل بالا عیناً تکرار می‌گردد ولی در انتها به‌جای ماتریس φ_d ثابت تابع $\varphi_d(t)$ که نسبت به زمان تغییر می‌کند، استفاده می‌گردد. همان‌گونه که در قسمت قبل نیز بیان گردید برای پایداری سیستم کل باید $\varphi_d(t)$ به‌گونه‌ای تعیین گردد که در تمام لحظات این تابع در شرط (۳۳) صدق نماید تا از ناپایداری سیستم جلوگیری گردد. در این طراحی تعداد انباشتگی‌ها برابر ۷۰ و به‌صورت گوسی در نظر گرفته‌شده است و به دلیل مقایسه دقیق برای تمام الگوریتم‌ها بیشینه تعداد تکرار برابر ۵۰۰ و تعداد اپک‌ها برابر ۱۰۰۰ در نظر گرفته‌شده است. ۷۰ درصد داده‌ها برای آموزش و مابقی برای تست در نظر گرفته‌شده است.

در الگوریتم زنبورعسل تعداد زنبورهای پیشاهنگ برابر ۳۰ و نرخ شعاع همسایگی برابر ۰/۹۹ در نظر گرفته‌شده است. در الگوریتم مورچگان q= 0.5 و $\zeta = 0.7$، در الگوریتم ازدحام ذرات w=1، $C_1 = C_2 = 2$ نرخ وزنی اینرسی با توجه به [۸] برابر ۰/۹۹ در نظر گرفته‌شده است. در الگوریتم گرادیان ازدحام ذرات با سعی و خطا ضریب یادگیری برابر ۲ و مابقی مانند ضریب ازدحام ذرات می‌باشد. همچنین در الگوریتم ژنتیک ضریب ادغام برابر ۴۰ درصد، ضریب جهش برابر ۰/۵ و نرخ جهش برابر ۰/۱ در نظر گرفته‌شده است. در الگوریتم رقابت استعماری با توجه به مطالب ذکرشده در بخش قبل، تعداد امپراتوری‌ها برابر ۱۰، ضریب فشار برابر ۱، ضریب همبستگی برابر ۲، ضریب هزینه متوسط مستعمره برابر ۰/۱ و با سعی و خطا مقادیر احتمال انقلاب برابر ۰/۱و نرخ انقلاب برابر ۰/۰۵ ، در نظر گرفته‌شده است. در این مقاله، در الگوریتم تکاملی تفاضلی محدود برای یافتن پاسخ‌های بهتر از β متغیر در بازه [۰/۵-۱] استفاده‌شده است و احتمال تقاطع برابر ۰/۱ در نظر گرفته‌شده است که به‌صورت سعی و خطا به‌دست‌آمده است. شکل‌های۵-۱۱ نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم‌های مذکور را نشان می‌دهند. جدول۱ نتایج حاصل از مشاهده‌گر اغتشاش مود لغزشی و تطبیقی نروفازی بهینه‌شده با الگوریتم زنبورعسل را با یکدیگر مقایسه می‌کند. جدول۲ نتایج حاصل از بهینه‌سازی توسط الگوریتم‌های تکاملی مذکور را نشان می‌دهد.



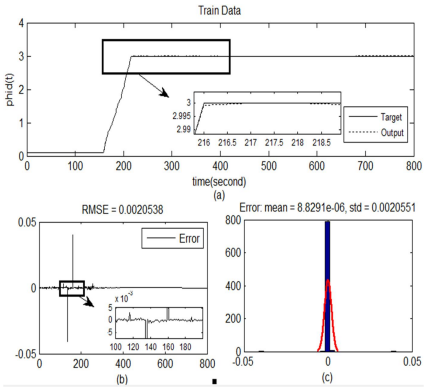
شکل ۵- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم BA. b) میانگین مربعات خطا



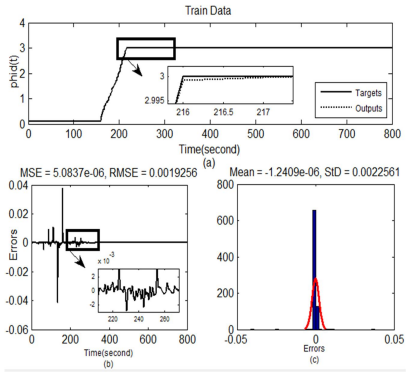
شکل ۶- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتمACOR. b) میانگین مربعات خطا

جدول۲- نتایج حاصل از شبیه‌سازی در دو مشاهده‌گر اغتشاش

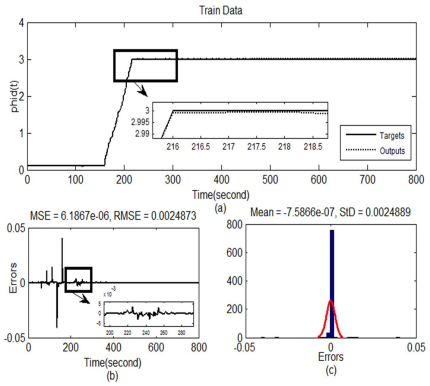
نوع مشاهده‌گر	SMDO	FSMDO
زمان نشست(t_s)	۵۶/۶	۵۸/۸
بیشینه اندازه تلاش کنترلی	-۰/۱۲۱	-۰/۱۲۰۶
میزان نوسانات ناخواسته	ناچیز	بسیارناچیز
خطای مشاهده‌گر اغتشاش	$۳/۵۶* ۱۰^{-۶}$	$۱/۳۱* ۱۰^{-۸}$
بیشینه فراجهش	۰	۰
خطای حالت ماندگار	$۵/۹۱* ۱۰^{-۶}$	$۱/۴* ۱۰^{-۹}$



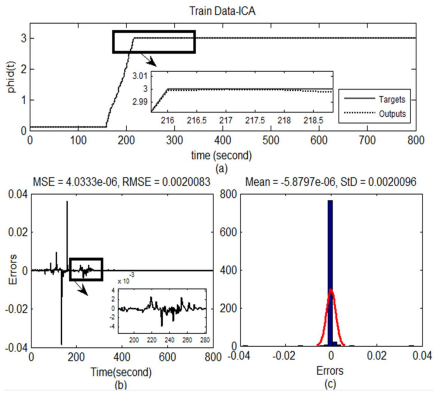
شکل ۷- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم PSO .b) میانگین مربعات خطا



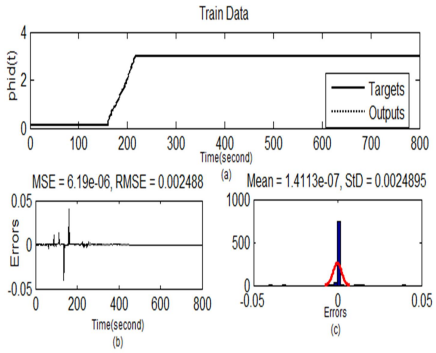
شکل ۸- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم GPSO.b) میانگین مربعات خطا



شکل ۹- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم GA .b) میانگین مربعات خطا



شکل ۱۰- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتم ICA .b) میانگین مربعات خطا



شکل

۱۱- نمودار تابع $\varphi_d(t)$ و تخمین آن توسط مشاهده‌گر اغتشاش نرو فازی بهینه‌شده توسط الگوریتمDE.b) میانگین مربعات خطا

جدول ۱- روش‌های هوشمند بهینه‌سازی FSMDO به‌وسیله الگوریتم‌های مختلف

نوع الگوریتم	مجذور مجموع مربعات خطا	میانگین خطا	سرعت الگوریتم (it/s)
الگوریتم رقابت استعماری	۰/۰۰۲۰۰۸۳	$۵/۸۷۹۷ * ۱۰^{-۶}$	≅ ۱/۹۶
الگوریتم زنبور عسل	۰/۰۰۱۷۰۳۹	$۱/۷۲۰۳ * ۱۰^{-۶}$	≅ ۲۹/۳۹
الگوریتم مورچگان	۰/۰۰۲۴۸۸	$۱/۴۱۱۳ * ۱۰^{-۷}$	≅ ۵/۴۴
الگوریتم ژنتیک	۰/۰۰۲۴۸۷۳	$۷/۵۸۶۶ * ۱۰^{-۷}$	≅ ۱/۶۵
الگوریتم تفاضل محدود	۰/۰۰۲۴۸۸	$۱/۴۱۱۳ * ۱۰^{-۷}$	≅ ۲/۱۳
الگوریتم ازدحام ذرات	۰/۰۰۲۰۵۳۸	$۸/۸۲۹۱ * ۱۰^{-۶}$	≅ ۱/۹۷
گرادیان ازدحام ذرات	۰/۰۰۱۹۲۵۶	$۱/۲۴۰۹ * ۱۰^{-۶}$	≅ ۳/۲۵

۷- بحث و نتیجه‌گیری

در این مقاله موضوع بهینه‌سازی مشاهده‌گر اغتشاش تطبیقی نروفازی با استفاده از الگوریتم‌های تکاملی موردبررسی قرارگرفته است. برای این منظور در ابتدا برای کاهش بهره و درنتیجه کاهش پدیده چترینگ و همچنین کاهش تلاش کنترلی از یک کنترلگر تطبیقی مدل مرجع لیاپانوف و همچنین برای تضعیف اغتشاشات و تخمین آن‌ها و کاهش هرچه بیش‌تر بهره از یک مشاهده‌گر اغتشاش نروفازی

استفاده‌شده است. سپس با استفاده از الگوریتم‌های تکاملی گوناگون سیستم نروفازی بهینه‌شده است و در انتها نتایج به‌دست‌آمده از بهینه‌سازی سیستم نرو فازی با الگوریتم‌های تکاملی (با تعداد تکرار یکسان) با یکدیگر مقایسه شده و همان‌گونه که در جدول (۱) نیز ملاحظه می‌شود، از بین الگوریتم‌های فوق الگوریتم زنبورعسل با مجذور مجموع مربعات خطا ۰/۰۰۱۷۳۹ دارای بهترین عمکرد و پس‌ازآن الگوریتم گرادیان ازدحام ذرات با مجذور مجموع مربعات خطا۰/۰۰۱۹بهترین عملکرد را دارد. الگوریتم ازدحام ذرات دارای مجذور مجموع مربعات خطا ۰/۰۰۲۰۵۳ می‌باشد و این نشان می‌دهد که الگوریتم گرادیان ازدحام ذرات نسبت به الگوریتم ازدحام ذرات دارای عملکرد بهتری می‌باشد.

۸- مراجع

[1] Lu Y.S., Sliding-Mode Disturbance Observer With Switching-Gain Adaptation and Its Application to Optical Disk Drives. IEEE Transaction on Industrial Electronics, Vol. 56, No.9, pp. 3743-3750, 2009.

[2] Hyeongcheol L., Masayoshi T., Robust adaptive control using a universal approximator for SISO nonlinear systems. IEEE Transactions on Fuzzy Systems, Vol. 8, No.10, pp. 95-106, 2000.

[3] Hao Jae L., Robust fuzzy control of nonlinear systems with parametric uncertainties. IEEE Transactions on Fuzzy Systems, Vol. 9, No.3, pp. 381-392, 2001.

[4] Atashpaz, E., Lucas C., Imperialism competitive algorithm: An algorithm for optimization inspired by Imperialistic competition. presented at the Congress on Evolutionary Computation, 2007.

[5] Shahbazi H., Jamshidi K., Monadjemi, A., Curvilinear bipedal walk learning in nao humanoid robot using a cpg based policy gradient method. Applied Mechanics and Materials, Vol. 110, pp. 5161-5166, 2012.

[6] Cherubini, A., Giannone F., Iocchi L., Lombardo, M., Oriolo, G., Policy gradient learning for a humanoid soccer robot. Robotics and Autonomous Systems, Vol. 57, No.2, pp. 808–818, 2009.

[7] Sahu R. K., Sekhar G. T., Panda S., DE optimized fuzzy PID controller with derivative filter for LFC of multisource power system in deregulated environment. Ain Shams Engineering Journal, Vol. 6, No.2, pp. 511-530, 2015.

[8] Paul K., Shill P.-C., Optimizing Fuzzy Membership Function using Dynamic Multi Swarm – PSO. In 5th International Conference on Informatics (ICIEV), Electronics and Vision, 2016.

[9] Karaboga D., Basturk B., Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In Foundations of Fuzzy Logic and Soft Computing. Verlag Berlin Heidelberg, 2007.

[10] Feiyi X., Chi-Man P., Haolun L., Yushu Z., Yurong S., Hao G., Training Feed-Forward Artificial Neural Networks with a modified artificial bee colony algorithm. Neurocomputing, Vol. 368, No. 4, pp. 478-491, 2019.

[11] Juang C., Chang P. H., Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization. IEEE TRANSACTIONS ON FUZZY SYSTEMS, Vol. 18, No.1, pp. 138-149, 2010.

[12] Chegeni Improved Genetic Algorithm-Based Optimization of Fuzzy Logic Controllers. seventh Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, 2007.

[13] Díaz N.P., Lagunas Jiménez R., González A., Tuning Fuzzy Control Rules via Genetic Algorithms: An Experimental Evaluation. Research Journal of Recent Sciences, Vol. 2, No.10, pp. 81-87, 2013.

[14] Sathishkumar H., Parthasarathy S.S., A novel neuro-fuzzy controller for vector controlled induction motor drive. International Conference on Alternative Energy in Developing Countries and Emerging Economies, 2017.

[15] Ullah A., Li S., Hussain A., Shen Y., Genetic optimization of fuzzy membership functions for cloud resource provisioning. IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No.8, pp. 2117- 2130, 2016.

[16] Mathew M., A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. Applied Soft Computing, Vol. 12, No. 2, pp. 353-359, 2012.

[17] Malekzadeh M., Rezayati M., Saboohi M., Hardware-in-the-loop attitude control via a high-order sliding mode controller/observer. Journal of Aerospace Engineering, Vol. 3, No. 2, pp. 1-17.

[18] Malekzadeh M., Robust Control of Spacecraft: Application to an Actuated Simulator. International Journal of Control, Automation and Systems, Vol. 16, No. 3, pp. 1-8.

[19] Malekzadeh m., Sadeghian H., Attitude control of spacecraft simulator without angular velocity measurement. Control Engineering Practice, Vol. 8, No. 4, pp. 72–81.

[20] Zipfel Peter H., Modeling and Simulation of Aerospace Vehicle Dynamics. Florida: University of Florida, 2007.

[21] Kim B., Velenis E., Kriengsiri P., Tsiotras P., Designing a Low Cost Spacecraft Simulator, IEEE Control Systems Magazine, Vol. 12, No.3, pp. 50-62, 2003.

[22] Regan F., Anandakrishnan S., Dynamics of Atmospheric Re-Entry. American Institute of Aeronautics and Astronautics, 1992.

[23] Astrom K. J., Bjorn W., Adaptive Control. Chicago: Courier Corporation, 2008.

[24] Yuhui Z., Byeungwoo J., Danhua X., Jonathan Wu, Q., Image segmentation by generalized hierarchical fuzzy C-means algorithm. Journal of Intelligent and Fuzzy Systems, Vol. 28, No.2, pp. 961-973, 2015.

[25] Peters J., Schaal, S., Policy Gradient Methods for Robotics. Presented at the Intelligent Robots and Systems, Beijing. China 2006.

[26] Storn R., PRICE K., Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Vol. 11, No.4, pp. 341–359, 1996.