

به نام خدا

طراحی کنترلر PID با استفاده از الگوریتم ژنتیکی پیوسته و الگوریتم ملکه زنبور عسل

گردآوری

مینا سلیم، محمد تقی و کبل باغمیشه

تبریز-دانشگاه تبریز-دانشکده مهندسی برق و کامپیوتر-آز تحقیقاتی سیستم های هوشمند
mvakil@tabrizu.ac.ir mina_salim158@yahoo.com

کلمات کلیدی

کنترلر PID - الگوریتم ملکه زنبور عسل - الگوریتم ژنتیکی پیوسته

چکیده

در این مقاله کنترلر PID با استفاده از دو روش الگوریتم ژنتیکی پیوسته و الگوریتم ملکه زنبور عسل طراحی و نتایج حاصل با هم مقایسه شده اند. در این طراحی سعی شده است که پاسخ حالت گذرا با استفاده از مینیمم کردن ماکزیمم فراجش (M_p) و زمان نشست (t_s) و زمان صعود (t_r)، انتگرال قدرمطلق خطای پاسخ پله (E_{ss}) به حداقل برسند. برای این منظور یک تابع هزینه با ترکیب مشخصات بالا در نظر گرفته شده است و این تابع در الگوریتم های ذکر شده مینیمم می گردد. پارامترهای کنترلر، ژن های کروموزوم ها را در جمعیت تشکیل می دهند. در این مقاله، چند تابع انتقال استاندارد در نظر گرفته شده و کنترلر PID با استفاده از هر دوروش بالا برای چند تابع تبدیل استاندارد طراحی شده است و نتایج حاصله از شبیه سازی با هم مقایسه شده است.



۱. مقدمه

کنترلر PID به دلیل ساده بودن و کارایی بالای آن به طور وسیع مورد استفاده قرار می گیرد. برای طراحی کنترلر PID سه پارامتر باید تعیین مقدار شوند. این سه پارامتر عبارتند از: بهره تناسبی: k_p ، بهره مشتقی: k_d و بهره انتگرالی k_i . روش سنتی طراحی با استفاده از روش سعی و خطا است. از معایب این روش، پیچیدگی و زمان بر بودن آن می باشد. روش های جدید برای طراحی کنترلر PID ، روشهای تکاملی هستند که بعضی از آنها عبارتند از: الگوریتم ژنتیکی پیوسته، الگوریتم ژنتیکی باینری، الگوریتم مورچه گسسته و پیوسته، الگوریتم پرندگان گسسته و پیوسته، الگوریتم استعماری و الگوریتم ملکه زنبور عسل.

در این تحقیق، ما مسئله طراحی کنترلر PID را تبدیل به یک مسئله بهینه سازی نموده و از الگوریتم ژنتیکی پیوسته و نبالگوریتم ملکه زنبور عسل برای مینیمم سازی یک تابع هدف استفاده کرده ایم. این تابع هدف همزمان چهار پارامتر ماکزیمم فراجش (M_p) ، زمان نشست (t_s) ، زمان صعود (t_r) و انتگرال قدر مطلق خطای پله (E_{ss}) را مینیمم می کند.

این مقاله شامل بخشهای زیر است: در بخش دوم الگوریتم ملکه زنبور عسل را به اختصار مطرح کرده و تفاوت آن را با الگوریتم ژنتیکی بررسی می کنیم. در بخش سوم تابع هزینه معرفی و ویژگی های آن شرح داده می شود. در بخش چهارم نتایج حاصل از دو الگوریتم ارائه و با هم مقایسه می شوند. بخش پنجم اختصاص به نتیجه گیری دارد.

۲. الگوریتم ملکه زنبور عسل

الگوریتم های تکاملی (EA) به عنوان الگوریتم های بهینه سازی که توانایی پیدا کردن راه حل های نزدیک به بهینه برای مسائل عددی و $real-valued$ test ها (که در عمل راه حل های بهینه با زمان محاسباتی معقول برای آنها وجود ندارد) شناخته شده اند [1,2]. هوش دسته جمعی یک شاخه جدید الگوریتم های تکاملی بوده و ملهم از رفتار دسته جمعی کولونی حشرات اجتماعی یا حیوانات اجتماعی هوش دسته جمعی نامیده می شود. الگوریتم ژنتیکی پیوسته قدیمی ترین نوع الگوریتم های تکاملی می باشند که به طور وسیع برای حل مسائل بهینه سازی مورد استفاده قرار گرفته شده است. این الگوریتم برای اولین بار توسط جان هولند در سال 1975 میلادی مطرح شد. اما الگوریتم زنبور عسل بسیار جدید بوده و برای اولین بار در سال 2001 میلادی مطرح شده است. قابلیت استفاده از این

الگوریتم در حل مسائل خاص، کاربردهای آن را بسیارمتنوع نموده است که از آن جمله می توان به بهینه سازی توابع چند متغیره [3] و بهینه سازی توابع با قیود غیر خطی [4] و خوشه بندی (clustering) [5] و آموزش شبکه های عصبی LVQ [6] استفاده شده است. تا کنون دو نوع از الگوریتم زنبور عسل معرفی شده که عبارتند از

Bee Colony Optimization (BCO) و *Queen Bee Evolution* در این مقاله برای طراحی کنترلر *PID* از الگوریتم ملکه زنبور عسل استفاده شده است. از آنجایی که الگوریتم ژنتیکی پیوسته به طور وسیع مورد استفاده قرار گرفته است ما از توضیح در مورد آن صرفنظر می کنیم و بحث خود را بر روی الگوریتم ملکه زنبور عسل متمرکز می کنیم. الگوریتم ملکه زنبور عسل دارای مفاهیم مشترک با الگوریتم ژنتیکی مانند ژن ها، کروموزم، جمعیت، اپراتور *crossover* و اپراتور جهش (*mutation*) می باشد [7]. این الگوریتم دو تفاوت عمده با الگوریتم ژنتیکی دارد: اول اینکه در الگوریتم ژنتیکی معمولی ابتدا یک تابع هزینه برای هر یک از افراد جمعیت اولیه محاسبه می شود و افراد جمعیت بر اساس افزایش تابع هزینه مرتب می شوند. سپس تعدادی از بدترین افراد جمعیت دور ریخته شده و بقیه که دارای هزینه کمتری هستند برای تولید فرزندان انتخاب می شوند. یعنی تعدادی مادر و به همان تعداد پدر انتخاب شده و با استفاده از اپراتور *crossover* فرزندان تولید می شوند که تعداد آنها برابر با تعدادی از اعضای جمعیت هستند که قبلاً دور ریخته شده بودند. به این ترتیب فرزندان جایگزین بدترین افراد جامعه می گردند. در الگوریتم ملکه زنبور عسل مراحل انتخاب جمعیت اولیه و مرتب کردن آنها بر اساس تابع هزینه و دور ریختن بدترین اعضا مشابه الگوریتم ژنتیکی است. اما در این الگوریتم برای تولید فرزندان فقط یک مادر که همان ملکه زنبورها است، انتخاب می گردد و ملکه با تعدادی از افراد جمعیت که پدرها هستند با استفاده از اپراتور *crossover* فرزندان مختلف را تولید می کند [8]. در نتیجه تعداد ازدواج ها در الگوریتم ملکه زنبور عسل بسیار کمتر از تعداد ازدواج ها در الگوریتم ژنتیکی معمولی است که این باعث می گردد که سرعت این الگوریتم در مقایسه با الگوریتم ژنتیکی معمولی بسیار زیاد باشد. اما سرعت زیاد همگرایی منجر به پدیده همگرایی زودرس می شود. در پدیده همگرایی زودرس، الگوریتم به جای پیدا کردن جواب اپتیمم، به یک مینیمم محلی همگرا خواهد شد. برای حل این مشکل دو نرخ جهش برای الگوریتم ملکه زنبور عسل در نظر گرفته می شود. یکی نرخ جهش نرمال و دیگری نرخ جهش قوی. برخی از ژن ها با نرخ جهش نرمال با احتمال P_m و بقیه با نرخ جهش قوی با احتمال P'_m جهش می کنند که به طور معمول P'_m بزرگتر از P_m

است. در نتیجه تنوع به وجود آمده در فرزندان بیشتر شده و از همگرایی زودرس جلوگیری می شود. نسبت بین این دو احتمال برابر پارامتر k تعریف می شود.

۳. فرمول بندی مسئله طراحی کنترلر (PID)

تابع انتقال کنترلر PID به صورت زیر است:

$$H(s) = k_p + k_d s + \frac{k_i}{s} \quad (1)$$

که در آن، k_p بهره تناسبی، k_d بهره مشتقی و k_i بهره انتگرالی می

باشند. در حوزه زمان یک معیار کارایی کنترلر میتواند مشخصات پاسخ پله مانند ماکزیمم فراجش و زمان صعود و زمان نشست و انتگرال قدر مطلق خطای پاسخ پله می باشد. ما یک تابع که از ویژگی های فوق به عنوان پارامتر استفاده می کند به عنوان تابع هزینه در نظر می گیریم [9].

$$f(k) = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(t_s - t_r) \quad (2)$$

که در آن k یک بردار است که درایه های آن را بهره های تناسبی و مشتقی و انتگرالی تشکیل می دهند:

$$k = [k_d \quad k_p \quad k_i] \quad (3)$$

و β یک فاکتور وزنی است که مقدار کمتر از 0.7 برای آن باعث کاهش t_s, t_r می گردد و مقدار بیشتر از 0.7 باعث کاهش M_p و E_{ss} می گردد. به ازای $\beta = 0.7$ تأثیر هر چهار پارامتر یکسان خواهد بود. در هر دو الگوریتم ژنتیکی پیوسته و الگوریتم ملکه زنبور عسل $k = [k_d \quad k_p \quad k_i]$ یک کروموزوم از جمعیت را تشکیل می دهد. در هر حلقه از اجرای الگوریتم اعضای کروموزوم ها در تابع کنترلر $(H(s))$ قرار داده می شوند و پاسخ پله سیستم محاسبه می گردد و با کمک آن t_s, t_r, M_p, E_{ss} محاسبه و در تابع هزینه جاگذاری می شوند. هدف الگوریتم کمینه کردن تابع هزینه می باشد. برای k_d, k_p, k_i یک حد

$$G(s) = \frac{1}{(s+1)^m} \quad (4)$$

بالاتر 10 و یک حد پایین برابر 0 در نظر می گیریم. تابع تبدیل سیستم برابر

در نظر گرفته شده است که یک تابع تبدیل استاندارد برای طراحی کنترلر PID می باشد [10]. برای این سیستم کنترلر PID به ازای مقادیر $m=1,4$ و برای سه مقدار $\beta=0.5, 0.7, 1.3$ طراحی خواهیم کرد.

۴. نتایج شبیه سازی شده

۴-۱. مقادیر پارامترهای الگوریتم ها

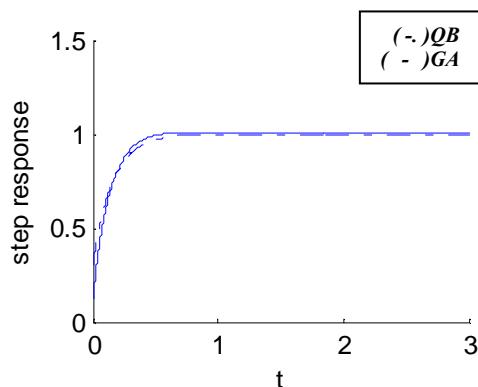
در الگوریتم ژنتیکی پیوسته:

سایز جمعیت: 50، نرخ انتخاب: 0.5، نرخ جهش اولیه: 0.1، این نرخ متغیر است و در هر تکرار از الگوریتم طبق رابطه (5) $murate = murate \times e^{-j}$ کاهش می یابد. که در آن $murate$ ، نرخ جهش و j شماره حلقه می باشد.

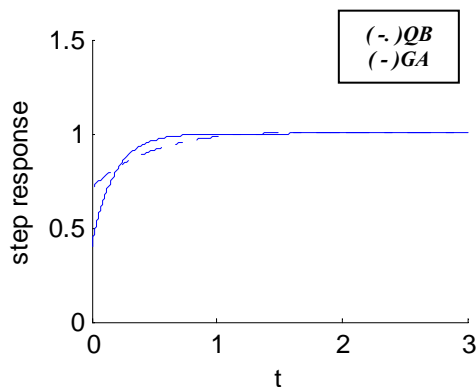
در الگوریتم ملکه زنبور عسل: سایز جمعیت: 40، نرخ انتخاب: 0.5، نرخ جهش قوی: 0.2، نرخ جهش نرمال: 0.01، $\zeta = 0.7$.

۴-۲. نتایج بدست آمده

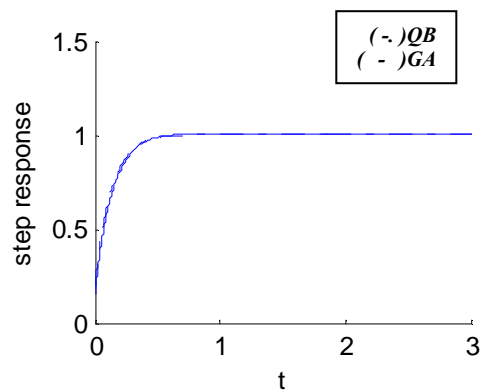
برای شبیه سازی از محیط برنامه نویسی *MATLAB* استفاده شده است. اشکال 1 تا 6 پاسخ پله سیستم با کنترلر را برای مقادیر مختلف β, m نشان می دهند.



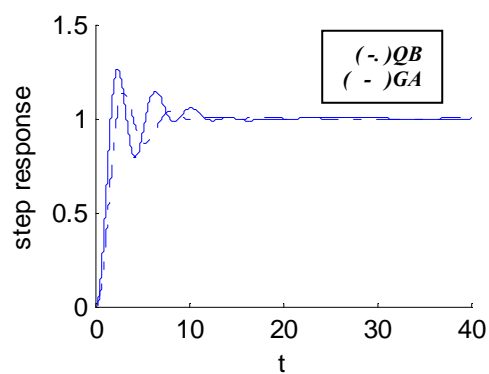
شکل 1- $m = 1, \beta = 0.5$



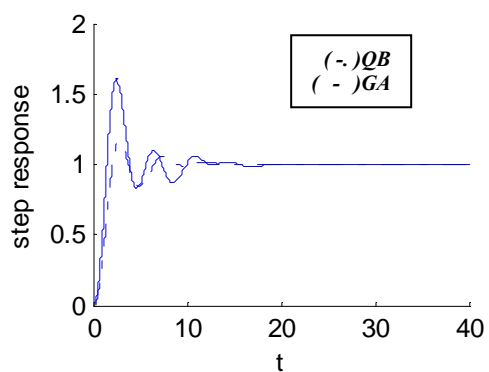
شکل 2- $m = 1, \beta = 0.7$



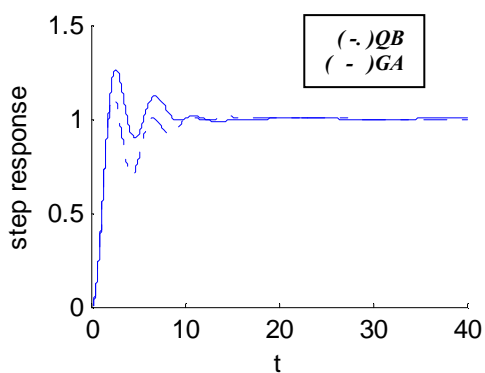
شکل 3- $m = 1, \beta = 1.3$



شکل 4- $m = 4, \beta = 0.5$



شکل 5- $m = 4, \beta = 0.7$



شکل 6- $m = 4, \beta = 1.3$

در جدولهای ۱ تا ۶ مقادیر مختلف پارامترهای سیستم های طراحی شده ارائه شده است:

جدول ۱- پارامترهای سیستم طراحی شده به ازای $m = 1, \beta = 0.5$

								<i>min</i>	
<i>algorithm</i>	<i>kd</i>	<i>kp</i>	<i>ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>Ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	0.13	8.52	9.86	0.00	2.45	0.42	8.84	6.06	124.37
<i>QBO</i>	0.44	9.40	8.85	0.01	2.90	0.60	12	7.53	90.06

جدول ۲- پارامترهای سیستم طراحی شده به ازای $m = 1, \beta = 0.7$

								<i>min</i>	
<i>algorithm</i>	<i>kd</i>	<i>Kp</i>	<i>ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	0.67	9.93	9.28	0	2.16	0.61	12.26	6.87	116.01
<i>QBO</i>	2.43	9.81	9.58	0	2.55	1.26	25.32	10.45	109.79

جدول ۳- پارامترهای سیستم طراحی شده به ازای $m = 1, \beta = 1.3$

								<i>min</i>	
<i>algorithm</i>	<i>kd</i>	<i>kp</i>	<i>Ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	0.18	8.69	9.6	0	2.39	0.46	9.3	3.95	122.64
<i>QBO</i>	0.24	9.75	9.87	0	2.47	0.57	11.48	4.15	100.05

جدول ۴- پارامترهای سیستم طراحی شده به ازای $m = 4, \beta = 0.5$

								<i>min</i>	
<i>algorithm</i>	<i>kd</i>	<i>kp</i>	<i>Ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	5.94	2.88	1.27	0.26	4.04	10.88	22.8	8.92	129.62
<i>QBO</i>	3.32	2.4	0.69	0.14	3.76	8.68	18.34	7.33	101.35

جدول ۵- پارامترهای سیستم طراحی شده به ازای $m = 4, \beta = 0.7$

								<i>min</i>	
<i>algorithm</i>	<i>kd</i>	<i>kp</i>	<i>Ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	6.68	3.89	3.52	0.6	5.14	11.44	23.94	9.1	117.64
<i>QBO</i>	4.05	2.59	0.8	0.15	3.76	11.53	24.46	6.55	93.33

جدول ۶- پارامترهای سیستم طراحی شده به ازای $m = 4, \beta = 1.3$

								<i>min</i>	
<i>algorithm</i>	<i>Kd</i>	<i>kp</i>	<i>Ki</i>	<i>Mp</i>	<i>Ess</i>	<i>tr</i>	<i>ts</i>	<i>cost</i>	<i>t(sec)</i>
<i>GA</i>	4.91	2.77	1.34	0.26	3.87	7.88	16.96	5.48	117.77
<i>QBO</i>	5.28	2.44	0.58	0.11	4.08	9.26	18.48	5.40	93.79

۵. نتیجه گیری

با استفاده از هردوروش الگوریتم ژنتیکی پیوسته والگوریتم ملکه زنبور عسل می توان کنترلر مناسب برای توابع انتقال متفاوت طراحی کرد. زمان لازم برای رسیدن به جواب در الگوریتم ملکه زنبور عسل بسیار کمتر از زمان لازم برای الگوریتم ژنتیکی پیوسته است. به ازای $m=1$ الگوریتم ژنتیکی پیوسته، تابع هدف را بیشتر کاهش می دهد و پارامترهای سیستم طراحی شده، M_p و t_r و t_s و Ess مقادیر کمتری خواهند داشت. به ازای $m=4$ الگوریتم ملکه زنبور عسل، تابع هدف را بیشتر کاهش می دهد و پارامترهای سیستم طراحی شده، M_p و t_r و t_s و Ess کاهش می یابند. بنابراین الگوریتم ژنتیکی پیوسته برای سیستم های مرتبه پایین والگوریتم ملکه زنبور عسل برای سیستم های مرتبه بالا مناسب تر است.

۶. سپاسگزاری

مؤلفین برخورد لازم می دانند که از مسئولین محترم مرکز تحقیقات مخابرات ایران به دلیل انجام حمایت مالی از این پروژه تحقیقاتی تشکروقدردانی نمایند.

۷. مراجع

- [1]K. Benatchba, L. Admane, M. Koudil, *Using Bees to Solve a Data_Mining Problem Expressed as a Max-Sat One, , Evolutionary Computation, pp. 212-220, Vol. 3562, 2005.*
- [2]D. Karoboga, B. Basturk, *On the Performance of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, pp:687-697, Vol. 8, Issue 1, 2008.*
- [3]D. Karaboga, B. Basturk, *A powerful and efficient algorithm for numerical function optimization:artificial bee colony, Journal of Global Optimization, Volume 39, Issue 3, PP. 459 – 471, 2007.*
- [4]L.D. Qin, Q.Y. Jiang, Z.Y. Zou, and Y.J. Cao *A Queen –Bee Evolution Based on Genetic Algorithm for Economic Power Dispatch, 39th International Universities Power Engineering Conference, UPEC 2004. Vol. 1, PP. 453 – 456, 2004.*
- [5]M. Fathian, B. Amiri, A. Maroosi, *Application of honey-bee mating optimization algorithm on clustering, Applied Mathematics and Computation, no.190, pp. 1502-1503, 2007.*
- E. Koc, *Application of Bees Algorithm to the Ghanbarzadeh, [6]D. T. Pham, S. Orti, A. Training of Learning Vector Quantization Networks for Control Chart Pattern Recognition, Inf. and Comm. Technologies, Vol. 1, pp. 1624- 1629, 2006.*
- [7]M. Koudil, K. Benatchba, A. Tarabet, EI. Batoul Shahraoui, *Using Artificial Bees to Solve Partitioning and Scheduling Problems in Codesign, Applied Mathematics and Computation, pp. 1710-1722, Vol. 186, Issue 2, 2007.*

[8]M. F. Azeem, A. M. Saad, Modified queen bee evolution based genetic algorithm for tuning of scaling factors of fuzzy knowledge base controller, Proc. of the IEEE INDICON.

Vol. 1, pp. 299 – 303, 2004.

[9]L. Gaing, A Particle Swarm Optimization Approach For Optimum Design of PID Controller in AVR System, IEEE International Conference on Energy conversion, pp. 384-391, Vol.19,

Issue2, 2004.

*at: [10]K.J. Åström, T. Hägglund, Benchmark Systems for PID Control, accessed
<http://www.esaiterrassa.upc.es/pid00/text.pdf>*