مجله مهندسی مکانیک مدرس، دی 1394، دوره 15، شماره 10، ص‌ص 259-268

ماهنامه علمی پژوهشی

مهندسی مکانیک مدرس

mme.modares.ac.ir

کنترل یادگیر تکرارشونده مرتبه‌کسری بهینه برای کنترل بازوی ربات تک-لینک

ایمان قاسمی[1]، ابوالفضل رنجبر نوعی[2]، سید جلیل ساداتی رستمی[3*]

1- کارشناسی ارشد، دانشکده برق و کامپیوتر، دانشگاه صنعتی نوشیروانی، بابل
2- دانشیار، مهندسی برق گرایش کنترل، دانشگاه صنعتی نوشیروانی، بابل
3- استادیار، مهندسی برق گرایش کنترل، دانشگاه صنعتی نوشیروانی، بابل
* بابل، صندوق پستی: 47148-71167 j.sadati@nit.ac.ir

اطلاعات مقاله

مقاله پژوهشی کامل
دریافت: 23 مرداد 1394
پذیرش: 10 شهریور 1394
ارائه در سایت: 23 شهریور 1394

کلید واژگان:
سیستم‌های کنترل یادگیر تکرار شونده
قانون بروزرسانی کنترل یادگیر تکرارشونده
مرتبه‌کسری نوع D^α و PD^α
بازوی ربات تک-لینک
بهینه‌سازی مبتنی بر جغرافیای زیستی

چکیده

در این مقاله، نوع جدیدی از سیستم‌های کنترل یادگیر تکرارشونده مرتبه‌کسری تحت عنوان کنترل یادگیر تکرارشونده مشتقی مرتبه‌کسری و کنترل یادگیر تکرارشونده تناسبی-مشتقی مرتبه‌کسری برای سیستم خطی‌سازی شده بازوی ربات تک-لینک ارائه می‌شود. در قانون بروزرسانی کنترل یادگیر تکرارشونده مشتقی، آریمتو کلاسیک از مشتق مرتبه اول (با تابع تبدیل s) خطای ردیابی استفاده می‌شود. روش ارائه‌شده در این مقاله برای بروزرسانی قانون کنترل یادگیری تکرارشونده از مشتق مرتبه‌کسری (با تابع تبدیل s^α، $\alpha\in(0\ 2]$) خطای ردیابی استفاده می‌کند. برای اولین بار، ابتدا سیستم ربات غیرخطی، با اعمال روش خطی‌سازی فیدبک ورودی- حالت خطی گشته، سپس به آنالیز و تحلیل همگرایی قانون کنترل یادگیر تکرارشونده نوع PD^α برای سیستم‌های خطی پرداخته می‌شود. در ادامه، یک معیار برای انتخاب بهینه ضرایب کنترل‌کننده با استفاده از الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی ارائه می‌گردد. در قسمت اول شبیه‌سازی، هر دو قانون بروزرسانی کنترل یادگیر تکرارشونده مرتبه‌کسری (نوع D^α و نوع PD^α) بر روی بازوی ربات تک-لینک خطی شده، پیاده‌سازی می‌شود، و عملکرد این دو کنترل‌کننده به ازای مقادیر متفاوت α، نمایش داده می‌شود. در ادامه و جهت بهبود عملکرد سیستم کنترلی حلقه بسته، ضرایب کنترل یادگیر تکرارشونده مرتبه‌کسری (ضریب تناسبی k_P و ضریب مشتقی k_D و α) با استفاده از الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی، بهینه‌سازی می‌شوند. نهایتاً کنترل یادگیر تکرارشونده پیشنهادی با نوع متداول آن مورد مقایسه قرار می‌گیرد.

Optimal Fractional Order Iterative Learning Control for single-link robot control

Iman Ghasemi[1], Abolfazl Ranjbar Noei[2], Sayed Jalil Sadati Rostami[3*]

1- Faculty of Electrical and Computer Engineering, Babol University of Technology, Babol, Iran
2, 3- Department of Electrical Engineering, Babol University of Technology, Babol, Iran
* P.O.B. 71167-47148, Babol, Iran, j.sadati@nit.ac.ir

ARTICLE INFORMATION

Original Research Paper
Received 14 August 2015
Accepted 01 September 2015
Available Online 14 September 2015

Keywords:
Iterative learning control systems
Updating law of iterative learning control
fractional order of type D^α and PD^α
single-link robot arm
Biogeography-based optimization (BBO)

ABSTRACT

In this paper, a new type of iterative learning control systems with fractional order known as iterative learning control with fractional order derivative and iterative learning control with fractional proportional–derivative for linearized systems of single-link robot arm is introduced. First order derivative of classic Arimoto is used for tracking error in updating law of derivative iterative learning control. The suggested method in this paper implements tracking error for updating control law of iterative learning of fractional order. For the first time, nonlinear robot system is linearized by input feedback linearization. Then, convergence analysis of iterative learning control law of type PD^α is studied. In the next step, we define a criteria for parameters optimization of proposed controller by using Biogeography-based optimization algorithm. Both updating laws of fractional order iterative learning control (D^α-type ILC and PD^α-type ILC) are applied on linearized robot arm and performance of both controllers for different value of α is presented. For improving the performance of closed loop system, coefficient of fractional order iterative learning control (proportional k_P and derivative k_D coefficients and α) is optimized by BBO algorithm. Proposed iterative learning control is compared with common type of system.

1- مقدمه

اکثر موجودات زنده (بخصوص انسان) با هر بار تکرار یک عملی، با تجربه‌ای که از آن عمل بدست می‌آورند تلاش می‌کنند، رفتار بعدی خود را بهبود بخشند، بنابراین محققین به فکر ابداع برنامه‌ای افتادند تا قدرت یادگیری را به سیستم‌های مهندسی نیز بسط دهند. نیاز به روشی برای بهبود فرآیندهای تکرارشونده، باعث ایجاد روش‌های جدیدی برای این فرآیندها شد [1]. الگوریتم کنترل یادگیر تکرارشونده یکی از روش‌های مهم در زمینه سیستم‌های کنترل یادگیر است که برای اولین بار در سال 1984 توسط

Please cite this article using:

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

I. Ghasemi, A. Ranjbar Noei, S. J. Sadati Rostami, Optimal Fractional Order Iterative Learning Control for single-link robot control, *Modares Mechanical Engineering*, Vol. 15, No. 10, pp. 259-268, 2015 (In Persian)

آریموتو و همکارانش مطرح شده است [2]. سیستم‌های کنترل یادگیر تکرارشونده به شاخه کنترل هوشمند تعلق دارد و شامل تکنیک‌های جدیدی برای کنترل فرآیندهای تکرار شونده در یک فاصله زمانی مشخص است، بطوریکه در آنها کنترل‌کننده از تجربیات تکرارهای گذشته می‌آموزد تا خود را چنان اصلاح نماید که عملکرد سیستم حلقه بسته با افزایش تعداد تکرارها بهبود یابد. بازوهای ربات از آن دسته از فرآیندهایی هستند که باید وظیفهٔ مشخصی را بطور متناوب در یک طول زمانی ثابت و با دقت بالا، تکرار کنند [3,2]. در واقع، در این فرآیندها قرار است یک وظیفهٔ تکراری نظیر جوشکاری، رنگ‌پاشی، برش‌کاری و رسیدن به مسیر مطلوب با دقت بالا، در یک مسیر هندسی از قبل تعیین‌شده در هر تکرار انجام شود. در چنین فرآیندهایی روش‌های کنترل کلاسیک بدلیل ثابت بودن پارامترهای کنترل‌کننده، عملکرد یکسانی از خود نشان می‌دهند. چنانچه عملکرد آن راضی‌کننده نباشد (مثلاً خطای حالت ماندگار وجود داشته باشد)، در این صورت این مسئله در تمامی تکرارهای پروسه خود را نشان می‌دهد. لذا با توجه به عدم قطعیت مدل‌سازی و اغتشاشات، کنترل بازخورد نمی‌تواند به تنهایی به عملکرد ردیابی با دقت بالا دست پیدا کند [4].

به صورت کلی یک قانون بروزرسانی کنترل یادگیر تکرار شونده به صورت زیر است:

$$u_{k+1}(t) = u_k(t) + \Delta u_k(t) \tag{1}$$

در رابطه فوق، $u_{k+1}(t)$ ورودی فرآیند تحت کنترل در تکرار $k+1$ام و $\Delta u_k(t)$ یک عبارت اصلاحی بوده که نحوه بدست آوردن آن نوع قانون کنترل یادگیر تکرار شونده را مشخص می‌کند. تفاوت روش‌های حل ارائه‌شده برای مسئله کنترل یادگیر تکرار شونده در نحوه‌ی تعیین $\Delta u_k(t)$ است. شرح کاملی از قانون‌های کنترل یادگیر تکرارشونده مرسوم خطی و غیرخطی در [5] وجود دارد.

محاسبات مرتبه‌کسری توسط لایبنیتز[1] در سال 1695 معرفی شد [6]. مشتقات مرتبه‌کسری ابزار بسیار مناسبی برای توصیف خواصی همچون حافظه‌داری و ارث‌پذیری بسیاری از مواد و فرآیندها هستند. همین مسئله سبب شده در سال‌های اخیر تحقیقات گسترده‌ای در زمینه محاسبات مرتبه کسری، به ویژه کنترل‌کننده‌های مرتبه‌کسری صورت بگیرد. یکی از دلایل عمده استفاده از کنترل‌کننده‌های مرتبه‌کسری انعطاف‌پذیری بیشتر آنها نسبت به کنترل‌کننده‌های مرتبه‌صحیح و همچنین امکان کنترل سیستم‌های پیچیده است. به‌طوریکه کسری بودن دینامیک کنترل‌کننده باعث افزایش درجه آزادی در انتخاب پارامترهای طراحی می‌شود و این امکان را به طراح می‌دهد که به پاسخ‌های با رفتار مناسب‌تر و نرم‌تر دست یابد. به عبارتی با استفاده از کنترل‌کننده‌های مرتبه‌کسری افزایش کارایی و عملکرد کنترل‌کننده مانند کاهش زمان همگرایی و کاهش خطای حالت ماندگار، محتمل‌تر خواهد بود [6]. پروفسور توستین استفاده از کنترل‌کننده مشتقی مرتبه‌کسری برای کنترل موقعیت اشیا بزرگ را در سال 1958 مورد بحث قرار داد که اشباع محرک‌ها نیاز به حد فاز کافی در محدوده و زیر نقطه بحرانی داشت [7]. در سال 1988 با پیشرفت محاسبات مرتبه‌کسری در مهندسی کنترل یک کنترل‌کننده مقاوم با نام کرون[2] توسط اوستالوپ[3] معرفی شده است [8]. بعدها، این کنترل‌کننده در مقالات [10,9] بیشتر مورد مطالعه قرارگرفته و توسعه داده شد. به این ترتیب، نخستین کنترل‌کننده‌های مرتبه‌کسری معرفی شدند و کاربرد محاسبات مرتبه‌کسری در کنترل مطرح شد. در سال 1999 کنترل‌کننده‌های PID مرتبه‌کسری توسط پدلبنی معرفی شدند [11]. روش‌های متنوعی برای طراحی این نوع کنترل‌کننده‌ها معرفی شده و نشان داده شده است که این کنترل‌کننده‌ها در مقایسه با کنترل‌کننده‌های PID کلاسیک مقاوم‌تر هستند.

سیستم‌های مبتنی بر تکرار، قابلیت یادگیری و تنظیم ورودی مناسب برای انجام کار تکراری را دارند. لیکن فرآیند تکرار، زمان‌بر و هزینه‌بر است. لذا ارائه الگوریتم‌های کنترل که از تکرار لازم برای تکمیل پروسه یادگیری بکاهند؛ مورد توجه محققین است [12]. اولین مقاله ILC مرتبه‌کسری در [12]، مورد بررسی قرار می‌گیرد که در آن یک قانون یادگیری مرتبه کسری نوع D^{α} پیشنهاد شده است و شرایط همگرایی در حوزه فرکانسی که روش مهمی در FOILC[4] است را مورد تجزیه و تحلیل قرار داده است.

بهینه‌سازی مبتنی بر زیست جغرافیا، توسط دَن سایمون[5] در سال 2008 ارائه شده است [13]. تکنیک BBO یک الگوریتم بهینه‌سازی جهانی جدید بر پایه تئوری زیست جغرافیا است که به مطالعه توزیع جغرافیایی ارگانیزم‌های زیستی می‌پردازد. مدل‌های ریاضی زیست جغرافیا، چگونگی مهاجرت گونه‌ها از یک جزیره به جزیره‌ی دیگر، چگونگی ظهور یک گونه‌ی جدید، و نحوه‌ی انقراض یک گونه را تشریح می‌کند. سایمون در [14]، توسط نسخه‌ای جدید به بررسی بیشتر و ساده‌تری از الگوریتم BBO با استفاده از تئوری احتمالات پرداخت. از دیگر کاربردهای بهینه‌سازی مبتنی بر زیست جغرافیا می‌توان به کاهش زمان تخمین حرکت در فیلم [15]، گسترش دینامیکی شبکه‌های حسگر بی‌سیم [16]، در سیستم‌های قدرت برای حل مسائل توزیع اقتصادی بار [17] و پخش بار بهینه [18] اشاره کرد.

در این مقاله، ابتدا سیستم غیرخطی بازوی ربات تک-لینک با روش خطی‌سازی فیدبکی، خطی شده، و سپس تجزیه و تحلیل همگرایی یک قانون بروزرسانی کنترل یادگیر تکرارشونده نوع PD^{α} برای سیستم‌های خطی ارائه می‌گردد. در نهایت، برای بهبود عملکرد سیستم کنترل یادگیر ارائه شده ضرایب کنترل‌کننده با استفاده از الگوریتم بهینه‌سازی مبتنی بر زیست جغرافیا، برای هر دو قانون کنترل یادگیر تکرارشونده نوع D^{α} و نوع PD^{α}، ارائه شده بهینه‌سازی می‌شود و در نتیجه با قانون یادگیری تکرارشونده مرتبه‌کسری غیربهینه مورد مقایسه قرار می‌گیرد.

2- قانون بروزرسانی کنترل یادگیر تکرارشونده مرتبه صحیح و کسری

ایده اصلی کنترل یادگیر تکرارشونده در شکل 1 نشان داده شده است. فرض می‌شود تمام سیگنال‌های نشان داده شده در فاصلهٔ زمانی $t \in [0, t_f]$ تعریف شده‌اند. اندیس k نشان‌دهنده تعداد آزمایش و یا تکرار است. اساس سیستم‌های کنترل یادگیر تکرارشونده این است که، در حین تکرار kام، بعضی از اطلاعات نظیر سیگنال ورودی $u_k(t)$، خروجی واقعی $y_k(t)$، و سیگنال خطای ردیابی $e_k(t)$ در حافظه ذخیره می‌شوند. این اطلاعات به منظور بهبود ورودی کنترل، و کاهش خطای بدست آمده بین خروجی واقعی و خروجی مطلوب سیستم، و افزایش عملکرد سیستم حلقه بسته، برای به هنگام کردن قانون کنترل یادگیر تکرارشونده در تکرار $k+1$ام بکار گرفته می‌شود. ورودی جدید به گونه‌ای باید طراحی شده باشد که باعث شود خطا نسبت به تکرار قبل کمتر شود.

1- Leibniz
2- CRONE: (Commande Robuste d'Ordre Non-Entier)
3- Oustaloup
4- Fractional Order Iterative Learning Control
5- Dan Simon

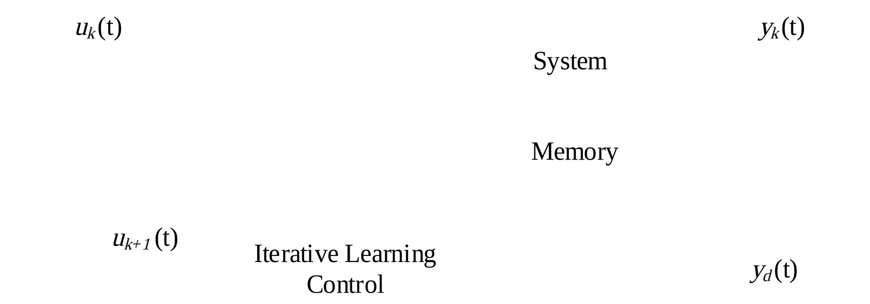

Fig. 1 Basic idea of ILC

شکل 1 ایده اساسی کنترل یادگیر تکرارشونده

از معروف‌ترین قوانین کنترل یادگیر تکرارشونده می‌توان به قانون یادگیری تناسبی و قانون یادگیری مشتقی اشاره کرد. برای قانون بروزرسانی کنترل یادگیر تکرارشونده مشتقی $\Delta u_k(t)$ به صورت زیر است [19].

$$\Delta u_k(t) = \Gamma \frac{d}{dt} e_k(t) \qquad t = 0,1,\dots,t_f, \quad k = 0,1,\dots \tag{2}$$

با قرار دادن رابطه (2) در (1) داریم:

$$u_{k+1}(t) = u_k(t) + \Gamma \frac{d}{dt} e_k(t) \tag{3}$$

در قانون بروزرسانی کنترل یادگیرتکرار شونده تناسبی $\Delta u_k(t)$ و $u_{k+1}(t)$ به صورت زیر می‌باشد:

$$\Delta u_k(t) = \Gamma\, e_k(t), \qquad t = 0,1,\dots,t_f, \quad k = 0,1,\dots \tag{4}$$

$$u_{k+1}(t) = u_k(t) + \Gamma\, e_k(t) \tag{5}$$

در روابط فوق $e_k(t)$ سیگنال خطای ردیابی بین مسیر خروجی واقعی $y_k(t)$ و مسیرمطلوب $y_d(t)$ در تکرار kام است و به صورت زیر بدست می‌آید:

$$e_k(t) \triangleq y_d(t) - y_k(t) \tag{6}$$

همانطور که در روابط فوق مشاهده می‌شود نحوه‌ی بدست آوردن $\Delta u_k(t)$ نوع قانون کنترل یادگیر تکرار شونده را مشخص می‌کند. در رابطه بالا Γ بهره‌ی یادگیری است که بر اساس دانش قبلی از فرآیند تحت کنترل طراحی می‌شود. $u_k(t)$ سیگنال کنترلی ورودی در تکرا رkام، و k نشان‌دهنده‌ی تعداد تکرار، که متغیری صحیح (گسسته) است و $t\in[0,t_f]$ نشان‌دهنده‌ی متغیر زمان که ممکن است متغیری صحیح (گسسته) و یا پیوسته باشد و t_f طول هر تکرار بوده و معلوم است.

با استفاده از روابط (2) تا (6) قانون بروزرسانی کنترل یادگیر تکرارشونده در حوزه‌ی فرکانس به صورت زیر است.

$$u_{k+1}(s) = u_k(s) + \Gamma s^{\alpha} E_k(s) \tag{7}$$

$$E_k(s) = y_d(s) - y_k(s) \tag{8}$$

با توجه به رابطه (7)، زمانی که $\alpha = 0$ باشد یک قانون یادگیری تکرار شونده تناسبی بدست می‌آید و همچنین زمانی که $\alpha = 1$ باشد یک قانون یادگیری تکرار شونده مشتقی خواهیم داشت. قانون بروزرسانی کنترل یادگیر تکرار شونده مرتبه‌کسری نوع D^{α} و PD^{α} را می‌توان به ترتیب با استفاده از رابطه (9) و (10) محاسبه کرد:

$$u_{k+1}(s) = u_k(s) + k_{\mathrm{D}} s^{\alpha} E_k(s) \tag{9}$$

$$u_{k+1}(s) = u_k(s) + k_{\mathrm{P}}\, E_k(s) + k_{\mathrm{D}} s^{\alpha} E_k(s) \tag{10}$$

در روابط بالا ضریب تناسبی k_{P} و ضریب مشتقی k_{D}، ضرایب حقیقی ثابت هستند و بهرۀ یادگیری نامیده می‌شوند که باید بطور مناسب انتخاب شوند. در این مقاله بررسی می‌کنیم چنانچه α یک عدد حقیقی بین $\alpha \in (0,2]$ باشد چه تأثیری بر نحوه‌ی همگرایی خروجی واقعی بر خروجی مطلوب خواهد داشت.

3- معرفی دینامیک و ساختار حرکتی بازوی رباتیک

از آنجا که کاربردهای بازوهای رباتیک معمولاً شامل حرکات تکرار شونده است طبیعی است که برای اصلاح رفتار و افزایش بهره‌وری و دقت آنها از کنترل یادگیر تکرارشونده استفاده کنیم. لذا در این قسمت یک مثال روباتیکی آورده می‌شود که این مثال از مرجع [20] انتخاب شده است. برای مطالعه و شبیه‌سازی از یک مدل هدایت مفصل بازوی ربات تک-لینک استفاده شده است که دینامیک آن در رابطه زیر نشان داده شده است:

$$\ddot{\theta}(t) = \frac{1}{J}\left(u(t) - F(t)\right) + \frac{1}{J}\left(\frac{1}{2}m + M\right) gl \times \sin\theta(t) \tag{11}$$

که در آن، $\theta(t)$ موقعیت بازوی ربات، $u(t)$ گشتاور مفصل اعمال شده، $F(t)$ گشتاور اصطکاک، m و l به ترتیب جرم و طول بازوی ربات، M جرم بار نوک، g شتاب گرانش زمین و J اینرسی لحظه‌ای مفصل هستند که مقدار اینرسی مفصل و گشتاور اصطکاک به صورت زیر داده شده است:

$$J = Ml^2 + ml^2/3 \tag{12}$$

$$F(t) = \begin{cases} f^{+} + B^{+}\dot{\theta}(t) \\ f^{-} + B^{-}\dot{\theta}(t) \end{cases} \tag{13}$$

در رابطه فوق، f^{+} و f^{-} نشان‌دهنده ضریب اصطکاک کولونی، B^{+} و B^{-} ضریب اصطکاک چسبناک است. بردار حالت سیستم غیرخطی بازوی ربات را به شکل زیر در نظر می‌گیریم:

$$x = [x_1 \quad x_2]^{\mathrm{T}} = [\theta \quad \dot{\theta}]^{\mathrm{T}} \tag{14}$$

که با نوشتن معادلات دینامیکی سیستم بر حسب بردارهای حالت داریم:

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x) \tag{15}$$

که در آن:

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{1}{J}\,(f + Bx_2) + \frac{1}{J}\left(\frac{1}{2}m + M\right) gl\, \sin x_1 \end{bmatrix},$$
$$g(x) = \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix}, \quad h(x) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{16}$$

تا اینجا معادلات فضای حالت سیستم بازوی ربات بدست آمد در بخش بعد به طراحی یک کنترل‌کننده بر پایۀ خطی‌سازی کامل معادلات حالت می‌پردازیم.

4- کنترل کننده خطی‌سازی با فیدبک ورودی-حالت

نمودار شکل 2 روش خطی‌سازی ورودی-حالت، را نشان می‌دهد. همانطور که در آن قابل مشاهده است این روش مسائل غیرخطی را در دو مرحله حل می‌کند. اول، یک تبدیل حالت و یک تبدیل ورودی چنان پیدا می‌کنیم که دینامیک غیرخطی تبدیل به یک دینامیک خطی نامتغیر با زمان معادل به شکل آشنای $\dot{z} = Az + Bv$ شود. دوم، از روش‌های استاندارد خطی (مانند جایگذاری قطب) در طراحی v استفاده می‌کنیم.

برای استفاده از روش خطی‌سازی ورودی-حالت ابتدا باید بررسی شود که آیا سیستم قابلیت خطی‌سازی کامل حالت‌ها را دارد یا خیر؟ یک سیستم قابلیت خطی‌سازی کامل حالت‌ها را دارد، اگر و فقط اگر دو شرط زیر به صورت همزمان برقرار باشند [21]:

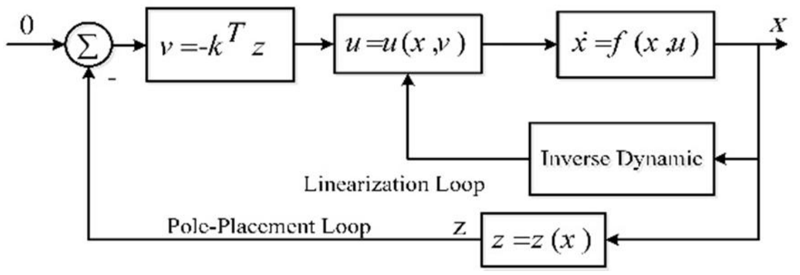

Fig. 2 An input-state linearization

شکل 2 خطی‌سازی ورودی-حالت

1. بردارهای ماتریس $G(x) = [g(x),\ ad_f g(x),\ \dots,\ ad_f^{n-1} g(x)]$ نسبت به هم مستقل باشند و یا ماتریس $G(x)$ دارای رتبه کامل باشد.
2. فضای گستره‌ی $\Delta = \mathrm{span}[g(x),\ ad_f g(x),\ \dots,\ ad_f^{n-2} g(x)]$ غیر پیچشی باشد.

در رابطه بالا $ad_f g(x)$، مشتقات کروشهٔ لی است که به صورت رابطه (17) تعریف می‌شود. n مرتبه نسبی سیستم است که در سیستم بازوی ربات $n = 2$ است. با داشتن یک میدان برداری $f(x)$ و حالات x، ژاکوبی f به صورت ∇f نمایش داده می‌شود، و رابطه آن به صورت (18) است:

$$ad_f g(x) = \nabla g . f - \nabla f . g \tag{17}$$

$$\nabla f = \frac{\partial f}{\partial \mathrm{x}}, \nabla g = \frac{\partial g}{\partial \mathrm{x}} \tag{18}$$

این ژاکوبی توسط ماتریس $n \times n$ از مؤلفه‌های $(\nabla f)_{ij} = \partial f_i / \partial xj$ تشکیل شده است. برای اینکه نشان دهیم سیستم بازوی ربات به طور کامل قابلیت خطی‌سازی را دارد، دو شرط بالا را مورد بررسی قرار می‌دهیم. با محاسبه میدان‌های برداری $ad_f g(x)$ در رابطه (19)، ماتریس $G(x)$ را می‌توان به صورت (20) بدست آورد.

$$ad_f g(x) = \nabla g\, f - \nabla f\, g = \begin{bmatrix} -\frac{1}{J} \\ \frac{1}{J^2} B \end{bmatrix} \tag{19}$$

$$G(x) = [g(x), ad_f g(x)] = \begin{bmatrix} 0 & -\frac{1}{J} \\ \frac{1}{J} & \frac{1}{J^2} B \end{bmatrix} \tag{20}$$

ماتریس $G(x)$ دارای رتبه کامل $(\mathrm{rank}(G(x)) = 2)$ است. توزیع $\Delta = \mathrm{span}\{g\}$ نیز گسترده است. از آنجا که شرایط کنترل‌پذیری و غیرپیچشی برقرار است، حال باید تبدیل حالت $z = z(x)$ و تبدیل ورودی $u = \sigma(x) + \beta(x)v$ را طوری پیدا کنیم که خطی‌سازی ورودی-حالت بدست آید. $z(x)$ باید طوری بدست آید که معادلات زیر را ارضا کند:

$$\begin{cases} \nabla z_1 ad_f^i g = 0 \quad i = 0, \dots, n-2 \\ \nabla z_1 ad_f^{n-1} g \neq 0 \end{cases} \tag{21}$$

پس اولین مؤلفهٔ z_1 از بردار حالت جدید z بایستی رابطهٔ (22) را برقرار سازد:

$$\frac{\partial z_1}{\partial \mathrm{x}_2} = 0, \qquad \frac{\partial z_1}{\partial \mathrm{x}_1} \neq 0 \tag{22}$$

لذا z_1 بایستی تنها تابعی از x_1 باشد. ساده‌ترین حل برای معادلات (22) عبارت است از:

$$z_1 = x_1 \tag{23}$$

حالت‌های دیگر را می‌توان از روی z_1 بدست آورد:

$$z_2 = \nabla \mathrm{z}_2 f = x_2 \tag{24}$$

بنابراین، تبدیل ورودی عبارت است از:

$$u = \sigma(x) + \beta(x) v \tag{25}$$

$$\sigma(x) = -\frac{L_f^n z_1}{L_g L_f^{n-1} z_1} = -\frac{L_f^2 z_1}{L_g L_f z_1} = f + B x_2 - \left(\frac{1}{2} m + M\right) g l \sin x_1 \tag{26}$$

$$\beta(x) = -\frac{1}{L_g L_f^{n-1} z_1} = J \tag{27}$$

در نتیجهٔ تبدیلات حالت و ورودی فوق، به معادلات خطی زیر می‌رسیم:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = v \\ y = z_1 \end{cases} \tag{28}$$

فرم خطی‌سازی شدهٔ سیستم غیرخطی بازوی ربات با استفاده از روابط (21) تا (28) عبارت است از:

$$\begin{cases} \dot{z} = Az + Bv \\ y = Cz \end{cases} \tag{29}$$

که در آن، مقادیر ماتریس‌های A، B و C برابر است با:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \tag{30}$$

$$y = [1 \quad 0] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{31}$$

بنابراین، با تبدیل حالت و تبدیل ورودی (22) تا (28)، مسئله پایدارسازی دینامیک‌های غیرخطی (11) با استفاده از کنترل ورودی اصلی $u_k(t)$ به مسئله پایدارسازی دینامیک‌های جدید (29) با استفاده از ورودی جدید v تبدیل شده است. چون دینامیک جدید خطی و کنترل‌پذیر است، لذا از روش جایگذاری قطب در طراحی v به صورت زیر استفاده می‌شود:

$$v = -K_1 z_1 - K_2 z_2 \tag{32}$$

کاملاً شناخته شده است که قانون کنترل پسخوردی حالت خطی قادر است که قطب‌ها را با انتخاب مناسب بهره‌های پسخوردی در هر جا قرار دهد.

5- آنالیز همگرایی قانون یادگیری مرتبه کسری ارائه شده

در این بخش، تجزیه و تحلیل همگرایی برای کنترل یادگیر تکرارشونده مرتبه کسری نوع PD^α (پیشنهاد شده در بخش 2)، برای سیستم‌های خطی ارائه می‌گردد. برای آنالیز همگرایی قانون یادگیری رابطه (10)، تابع انتقال فرآیند خطی‌سازی شده $G_C(s)$ در نظر گرفته می‌شود. و برای این تابع تبدیل ورودی $u(t)$، و خروجی $y(t)$ است. فرض می‌شود که $G_C(s)$ پایدار داخلی و BIBO است. با توجه به مسیر خروجی مطلوب $y_d(t)$ که در آن $t \in [0, t_f]$ ، فرض می‌شود که یک ورودی مطلوب منحصر به فرد $u_d(t)$ می‌توانیم پیدا کرد، که معادله زیر را برآورده سازد:

$$y_d(t) = G_c u_d(t) \tag{33}$$

با استفاده از روابط (8) و (10) برای قانون بروزرسانی PD^α-type ILC داریم:

$$\begin{aligned} u_{k+1}(s) &= u_k(s) + k_\mathrm{P}\left(y_d(s) - y_k(s)\right) + k_\mathrm{D}\mathrm{s}^\alpha\left(y_d(s) - y_k(s)\right) \\ &= u_k(s) + k_\mathrm{P}\, G_C(\mathrm{s})\left(u_d(s) - u_k(s)\right) + k_\mathrm{D}\mathrm{s}^\alpha\, G_C(\mathrm{s})\left(u_d(s) - u_k(s)\right) \\ &= \left(1 - k_\mathrm{P} G_C(\mathrm{s}) - k_\mathrm{D}\mathrm{s}^\alpha G_C(\mathrm{s})\right) u_k(s) + (k_\mathrm{P} + k_\mathrm{D}\mathrm{s}^\alpha) G_C(\mathrm{s}) u_d(s) \end{aligned} \tag{34}$$

از معادله فوق، ρ را به صورت زیر تعریف می‌کنیم:

$$\rho = 1 - k_\mathrm{P} G_C(\mathrm{s}) - k_\mathrm{D}\mathrm{s}^\alpha G_C(\mathrm{s}) \tag{35}$$

با استفاده از روش بازگشتی داریم:

$$\begin{aligned} u_1(s) &= u_0(s)\rho + (k_\mathrm{P} + k_\mathrm{D}\mathrm{s}^\alpha) G_C(\mathrm{s}) u_d(s) \\ u_2(s) &= u_0(s)\rho^2 + \frac{1-\rho^2}{1-\rho}(k_\mathrm{P} + k_\mathrm{D}\mathrm{s}^\alpha) \times G_C(\mathrm{s}) u_d(s) \\ u_3(s) &= u_0(s)\rho^3 + \frac{1-\rho^3}{1-\rho}(k_\mathrm{P} + k_\mathrm{D}\mathrm{s}^\alpha) \times G_C(\mathrm{s}) u_d(s) \\ &\vdots \\ u_{k+1}(s) &= u_0(s)\rho^{k+1} + \frac{1-\rho^{k+1}}{1-\rho}(k_\mathrm{P} + k_\mathrm{D}\mathrm{s}^\alpha) G_C(\mathrm{s}) u_d(s) \end{aligned} \tag{36}$$

در معادله فوق، اگر $|\rho| < 1$ باشد. آنگاه داریم:

$$\lim_{k\to\infty}\left(u_{k+1}(s)\right) = \lim_{k\to\infty}\left(u_0(s)\rho^{k+1} + (1-\rho^{k+1}) u_d(s)\right) = u_d(s) \tag{37}$$

پس همگرایی قانون یادگیری ارائه شده به صورت زیر است:

$$|1 - k_{\mathrm{P}} G_C(j\omega) - k_{\mathrm{D}} (j\omega)^{\alpha} G_C(j\omega)| < 1, \quad \forall \omega \qquad (38)$$

اگر ضرایب k_{P}،α و k_{D} طوری انتخاب شوند که در معادله (38) صدق کنند، آنگاه قانون PD^{α}-type ILC مورد نظر همگرا بوده و رابطه (39) تحقق می‌یابد:

$$\lim_{k\to\infty}\left(y_d(s) - y_k(s)\right) = 0 \qquad (39)$$

6- بهینه‌سازی عملکرد کنترل یادگیر تکرارشونده مرتبه‌کسری با استفاده از الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی

در قسمت قبل بیان شده است که اگر ضرایب α، k_{P} و k_{D} طوری انتخاب شوند که در معادلهٔ (38) صدق کنند، آنگاه قانون بروزرسانی کنترل یادگیر تکرارشونده نوع PD^{α} مورد نظر همگرا خواهد بود. در این قسمت یک معیار برای انتخاب بهینه این ضرایب با استفاده از الگوریتم بهینه‌سازی مبتنی بر زیست جغرافیا [13] ارائه می‌گردد.

الگوریتم BBO یک تکنیک تکاملی بر پایهٔ جمعیت است که از پدیدهٔ مهاجرت حیوانات و پرندگان به جزایر الهام گرفته شده است. این روش ویژگی‌های مشترکی با دیگر روش‌های بهینه‌سازی‌های مبتنی بر زیست‌شناسی مانند ژنتیک و اجتماع ذرات معلق دارد. مناطق جغرافیایی که مکان مناسبی برای سکونت گونه‌های زیستی هستند، دارای شاخص شایستگی بالا یا HSI[1] هستند. ویژگی‌هایی که تعیین‌کنندهٔ HSI هستند، شامل فاکتورهایی مانند: میزان بارندگی، تنوع پوشش گیاهی، مساحت، دما و خاک هستند. متغیرهایی که این شایستگی را شکل می‌دهند، متغیرهای شاخص شایستگی (SIVs[2]) خوانده می‌شوند. SIVها می‌توانند متغیرهای مستقل از زیستگاه و HSI متغیر وابسته به زیستگاه در نظر گرفته می‌شوند. تعداد زیادی از گونه‌ها در زیستگاه‌های با HSI بالا قرار دارند، و در نتیجه در آنها نرخ مهاجرگریزی (مهاجرت از[3]) گونه به زیستگاه همسایه زیاد و نرخ مهاجرپذیری (مهاجرت به[4]) کمی دارند. زیستگاه‌های با HSI پایین تعداد گونه کمی دارند و لذا نرخ مهاجرپذیری به آنها زیاد و نرخ مهاجرگریزی کمی دارند. شکل 3 یک نمونه از فرآوانی گونه‌ها در یک زیستگاه را نشان می‌دهد، نرخ مهاجرگریزی μ و نرخ مهاجرپذیری λ تابعی از تعداد گونه‌ها در زیستگاه می‌باشند.

ماکزیمم نرخ مهاجرگریزی برابر است با E و زمانی اتفاق می‌افتد که زیستگاه شامل بیشترین تعداد گونه‌ای که می‌تواند تأمین کند باشد، ماکزیمم نرخ مهاجرپذیری به زیستگاه I است و وقتی رخ می‌دهد که در زیستگاه هیچ گونه‌ای نباشد. تعداد تعادل گونه‌ها در نقطه‌ای است که نرخ‌های مهاجرگریزی و مهاجرپذیری با هم برابر هستند، و برابر با S_0 است. نرخ مهاجرگریزی و نرخ مهاجرپذیری تابع تعداد گونه‌ها در زیستگاه است. آنها را می‌توان به شرح زیر محاسبه کرد:

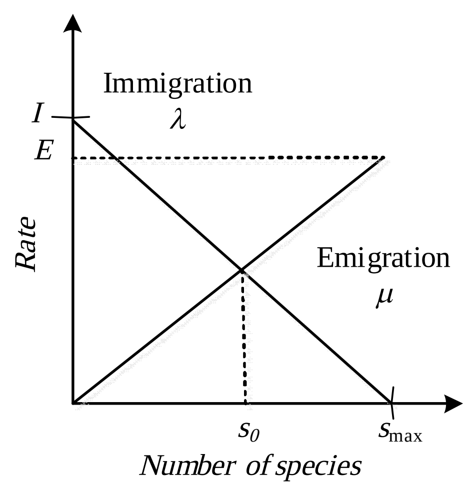

Fig. 3 Species model of a single habitat

شکل 3 مدل گونه‌ها یک زیستگاه تنها.

1- Habitat Suitability Index
2- Suitability Index Variables
3- Emigration
4- Immigration

$$\lambda_S = I\left\{\left(1 - \frac{S}{S_{\max}}\right)\right\} \qquad (40)$$

$$\mu_S = E\left(\frac{S}{S_{\max}}\right) \qquad (41)$$

در حالت خاص هنگامی که $E = I$ باشد نرخ‌های مهاجرپذیری و مهاجرگریزی به صورت زیر است.

$$\mu_S + \lambda_S = E \qquad (42)$$

در روابط (40) و (41)، I حداکثر نرخ ممکن برای مهاجرپذیری است، E حداکثر نرخ ممکن برای مهاجرگریزی است، S تعداد گونه‌های فردی Sام و $S_{\max}$ حداکثر تعداد گونه‌ها در زیستگاه است، که در آن نقطه نرخ مهاجرپذیری صفر و حداکثر نرخ ممکن مهاجرگریزی E، رخ می‌دهد. مفهوم BBO بر اساس مهاجرت و جهش است، که بصورت زیر بحث شده است.

6-1- استراتژی مهاجرت

عمل مهاجرت در جغرافیای زیستی شبیه عمل باز ترکیب عمومی در ژنتیک و استراتژی تکاملی بوده و برای اصلاح پاسخ‌های غیر نخبه انجام می‌شود. مهاجرت را می‌توان به صورت $H_i(\mathrm{SIV}) \leftarrow H_j(\mathrm{SIV})$ بیان کرد [22]. فرض می‌کنیم N زیستگاه وجود دارد که H_i یکی از آنها است که نرخ مهاجرپذیری آن λ_i است. و H_j زیستگاه بعدی است که، نرخ مهاجرگریزی آن μ_j است. یک اپراتور مهاجرت تعمیم داده شده از اپراتور مهاجرت BBO استاندارد، به نام مهاجرت مخلوط[5] به صورت زیر تعریف می‌شود [22].

$$H_i(\mathrm{SIV}) \leftarrow \gamma H_i(\mathrm{SIV}) + (1-\gamma) H_j(\mathrm{SIV}) \qquad (43)$$

در رابطه بالا γ یک عدد حقیقی بین 0 تا 1 است که می‌تواند بصورت اتفاقی و یا قطعی انتخاب شود.

6-2- استراتژی جهش

اتفاقات ناگهانی باعث می‌شوند که تعداد گونه‌ها از مقدار تعادل خود منحرف شوند و HSI زیستگاه به طور ناگهانی تغییر کند. در BBO این اتفاق با جهش SIV نشان داده می‌شود. شاید همیشه تمام جواب‌هایی که توسط الگوریتم BBO بدست آمده است بهینه نباشد، یا ممکن است ما را از هدف اصلی که رسیدن به جواب بهینه است دور کند بدین منظور در این الگوریتم بعد از عمل مهاجرت باید عملگر جهش روی راه حل‌ها اتفاق بیافتد، از جهش برای تغییر راه حل ها استفاده خواهیم کرد، هدف کلی جهش ایجاد تنوع در راه‌حل‌ها یا افزایش زیستگاه‌ها در میان جمعیت است [23،13]. احتمال تعداد گونه‌ها P_{S}، بیانگر آن است که زیستگاه شامل دقیقاً S گونه باشد. P_{S} از زمان t تا زمان $(t + \Delta t)$ با استفاده از λ_s و μ_s، به صورت معادله (44) بروزرسانی می‌شود که برای مشخص کردن نرخ جهش استفاده می‌شود. و فرض کنید یک زیستگاه با گونه S برای اجرای عملیات جهش مشخص شده است، یک متغیر انتخاب شده (SIV) بر اساس احتمال وجود آن P_{S} به طور تصادفی تغییر دهید. از احتمال تعداد گونه‌های موجود در زیستگاه P_{S} نرخ جهش $m(s)$ بصورت زیر محاسبه می‌شود:

$$P_s(t + \Delta t) = P_s(t) + \left(1 - \lambda_s \Delta t - \mu_s \Delta t\right) + P_{s-1}\lambda_{s-1}\Delta t + P_{s+1}\mu_{s+1}\Delta t \qquad (44)$$

5- Blended Migration

$$m(s) = m_{\max}(1 - \frac{P_s}{P_{\max}}) \quad (45)$$

که در آن $m_{\max}$ (بیشترین مقدار نرخ جهش) یک پارامتر تعریف شده توسط کاربر است، و $P_{\max}$ حداکثر احتمال تعداد گونه‌ها است، درحالی‌که P_S احتمال وجود گونه S در زیستگاه است.

6-3- فلوچارت و تابع هزینه الگوریتم

برای انتخاب بهینه ضرایب FOILC با استفاده از الگوریتم بهینه‌سازی BBO، هر SIVs (متغیرهای شاخص شایستگی) برای کنترل‌کنندهٔ ILC نوع PD^{α} از سه پارامتر α، k_{P} و k_{D} تشکیل شده است و هر SIVs برای کنترل‌کننده ILC نوع D^{α} دارای دو پارامتر α و k_{D} است. برای محاسبه HSI یا تابع هزینه به ازای هر SIV الگوریتم کنترل یادگیر تکرار شونده با مقدار اولیهٔ آن SIV شروع شده و تا تعداد تکرار از پیش تعیین‌شده ادامه می‌یابد. در انتهای تکرار آخر الگوریتم ILC یک استاندارد زمان ضرب‌در توان دوم خطا[1] را به عنوان HSI یا تابع هدف در نظر می‌گیریم. تابع هدف بصورت زیر تعریف می‌شود:

$$J_{\mathrm{ITSE}} = \int_0^{t_{\mathrm{sim}}} t.\,e^2(t)\,dt = \int_0^{t_{\mathrm{sim}}} t.\,(y_d(t) - y_k(t))^2\,dt \quad (46)$$

در معادلات بالا، t_{sim} محدوده زمانی از شبیه‌سازی که در آن بهینه سازی انجام شده است. بصورت کلی نمودار الگوریتم BBO تعمیم داده شده در شکل 4 قابل مشاهده است.

7- شبیه‌سازی

در این بخش، ابتدا به منظور ارزیابی عملکرد و بررسی مزایای استفاده از محاسبات مرتبه‌کسری، از قوانین بروزرسانی کنترل یادگیر تکرارشوندهٔ مرتبه کسری پیشنهادی (نوع D^{α} و نوع PD^{α})، برای کنترل بازوی ربات تک-لینک که در بخش 3 معرفی شده است، استفاده می‌شود و سپس با استفاده از الگوریتم BBO ضرایب کنترل‌کننده ILC مرتبه‌کسری به صورت بهینه محاسبه می‌شود. کلیهٔ شبیه‌سازی‌های این مقاله توسط نرم‌افزار متلب انجام شده است.

مسیر خروجی مطلوب که بازوی ربات با افزایش تعداد تکرار باید آنرا ردیابی کند در شکل 5 نشان داده شده است و معادلهٔ آن چنین است:

$$\theta_d(t) = \theta_b + (\theta_b - \theta_f)(15r^4 - 6r^5 - 10r^3) \quad (47)$$

$$r = \frac{t}{t_f - t_0} \quad (48)$$

برای تعیین ورودی فرآیند از روش ارائه شده در این مقاله استفاده می‌کنیم. بدین منظور ابتدا سیستم غیرخطی بازوی ربات را مطابق با شکل 2 خطی‌سازی نموده و سپس با استفاده از استراتژی فیدبک حالت و انتخاب بردار بهرهٔ فیدبک $K = [K_1\ K_2]$، قطب‌های حلقه بسته را در مکان مطلوب قرار می‌دهیم. پس از خطی‌سازی و اعمال استراتژی فیدبک حالت با استفاده قوانین بروزرسانی کنترل یادگیر تکرار شونده با معادلهٔ (9) و (10)، مطابق با شکل 1 ورودی فرآیند را محاسبه می‌کنیم.

در شبیه‌سازی با فرض قطب‌های مطلوب حلقه بسته $\{-1, -2\}$ آنگاه بردار بهرهٔ فیدبک حالت $K = [2, 3]$ بدست می‌آید. حالت‌های اولیه در ILC در هر تکرار برروی صفر تنظیم می‌شود. موقعیت زاویه‌ای بازوی ربات $\theta_b = 0$ و $\theta_f = 90$، و مقادیر $t_0 = 0$ و $t_f = 1$ ثانیه است. مقادیر پارامترهای

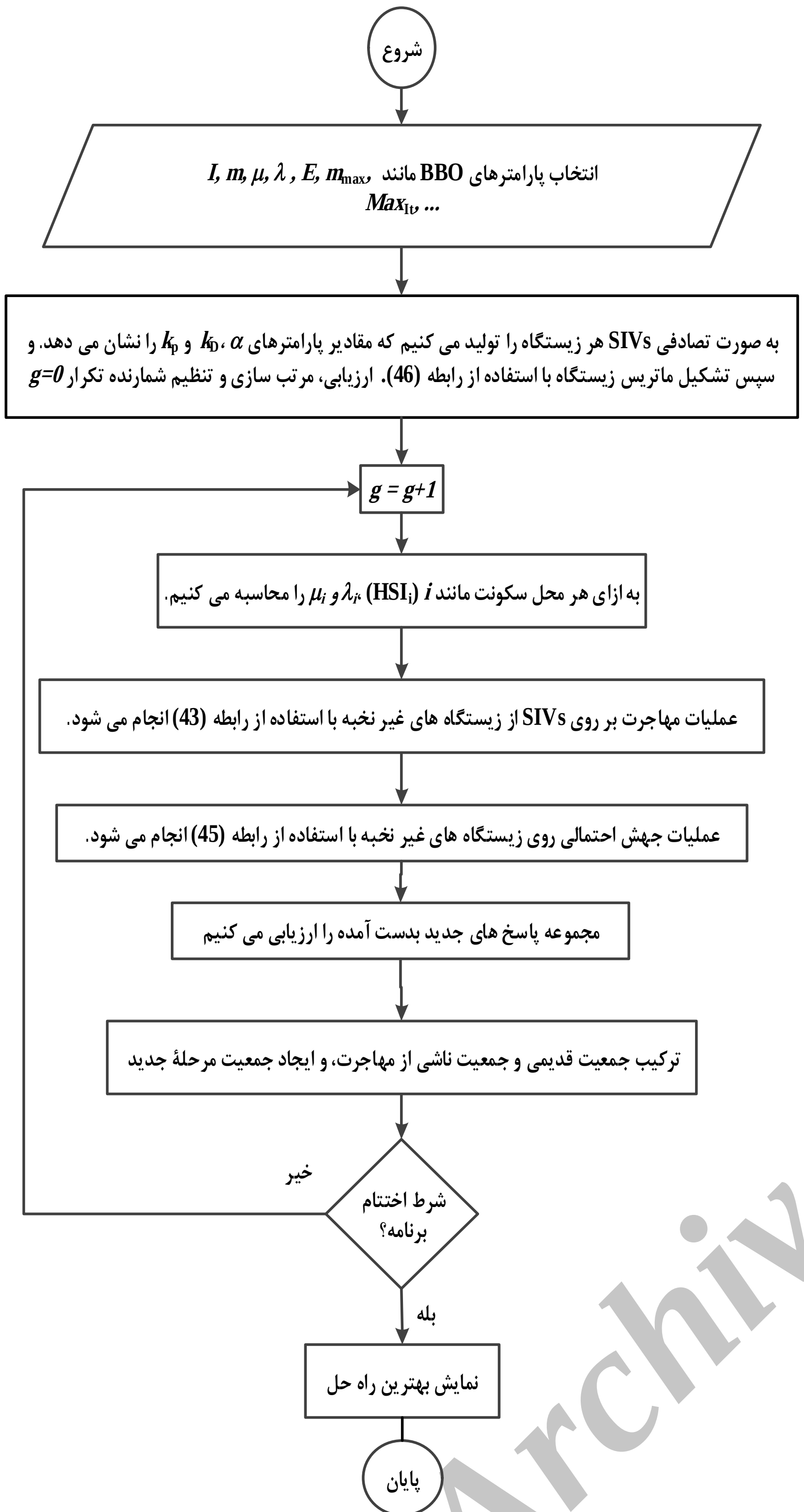

Fig. 4 The flowchart of extended BBO algorithm for iterative learning control problem.

شکل 4 نمودار الگوریتم BBO توسعه‌یافته برای مسئله کنترل یادگیر تکرارشونده.

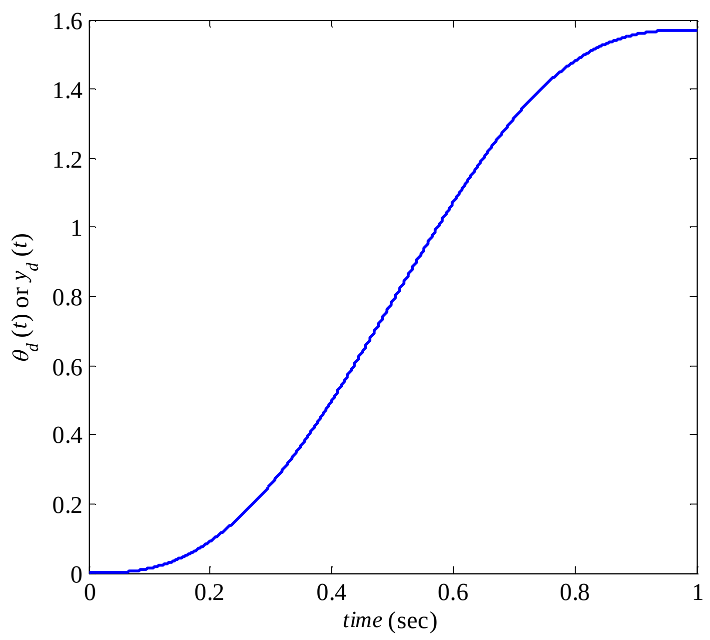

Fig. 5 Desired output trajectory $\theta_d(t) = y_d(t)$. [12]

شکل 5 مسیر خروجی مطلوب $\theta_d(t) = y_d(t)$ [12].

1- Integral of Time multiplied square value of the Error (ITSE)

سیستم بصورت جدول 1 انتخاب شده است [20]. در اینجا ما فرض می‌کنیم که سرعت زاویه‌ای در دسترس است. سپس برای $\alpha = 1$ در رابطه (9) بهترین انتخاب Γ برابر J است که توسط رابطۀ (12) محاسبه می‌شود [24].

7-1-کنترل یادگیر تکرار شونده مرتبه کسری

در این بخش، قوانین بروزرسانی کنترل یادگیر تکرارشونده مرتبه‌کسری پیشنهادی بر روی بازوی ربات تک-لینک خطی‌سازی شده پیاده‌سازی می‌شود. در شبیه‌سازی از نینتیجر[1] که یک جعبه‌ابزار برای کمک به پیاده‌سازی کنترل‌کننده‌های مرتبه‌کسری در متلب است، استفاده شده است [25]. عملکرد ردیابی توسط رابطه زیر ارزیابی می‌شود:

$$J_k(e_{m1}) = \sup_{t\in[0,1]}(\theta_d(t) - \theta_k(t))^2 \quad (49)$$

که در آن $J_k(e_{m1})$ مجذور خطای موقعیت بر حسب درجه (deg) است. نتایج بدست آمده از توان دوم حداکثر خطای ردیابی $J_k(e_{m1})$، بر حسب 40 تکرار در شکل 6 نشان داده شده است. سیگنال کنترلی ورودی بدست آمده پس از 40 تکرار در شکل 7 نشان داده شده است و خروجی بدست آمده از این ورودی برای قانون بروزرسانی ILC مرتبه‌کسری (نوع D^α) به ازای مقادیر مختلف α در شکل 8 نشان داده شده است که در آن از قانون بروزرسانی ILC نوع D^α رابطه (9)، برای تعیین ورودی فرآیند در تکرار $k + 1$ام استفاده شده است. با مشاهده شکل 6 می‌توان عملکرد کنترل

جدول 1 پارامترهای سیستم بازوی ربات تک-لینک

Table 1 Parameters of a single-link robot manipulator

نوع پارامتر	مقدار
f^+	8.43 Nm
f^-	-8.26 Nm
B^+	$4.94\,\frac{\text{Nm}}{\text{rad}}/\text{sec}$
B^-	$3.45\,\frac{\text{Nm}}{\text{rad}}/\text{sec}$
M	4 kg
m	2 kg
l	0.5 m
g	$9.8\ {}^{\text{m}}/_{\text{s}^2}$

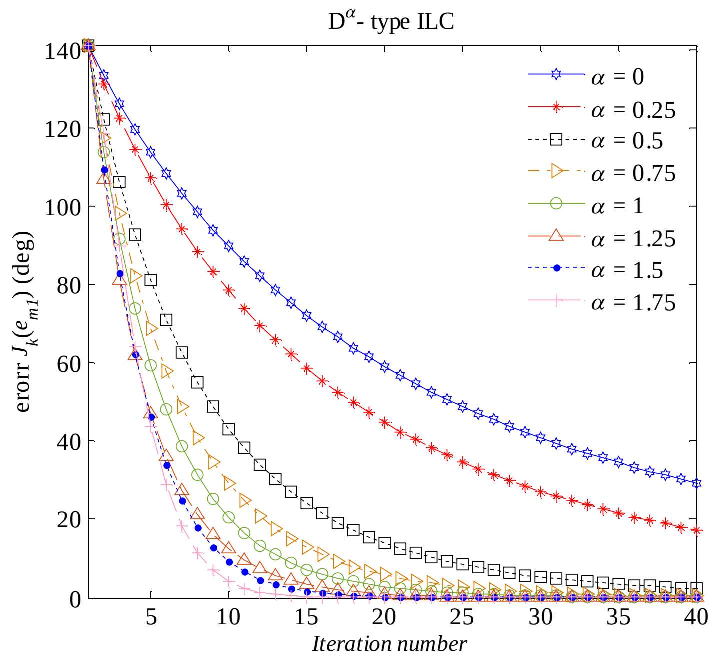

Fig. 6 Maximum square error (MSE) of the tracking versus the iteration number, Comparison of D^α-type ILC with different α.

شکل 6 نمودار توان دوم حداکثر خطای ردیابی $J_k(e_{m1})$ بر حسب تعداد تکرار، با استفاده از قانون بروزرسانی D^α-type ILC برای مقادیر مختلف α.

1- Ninteger

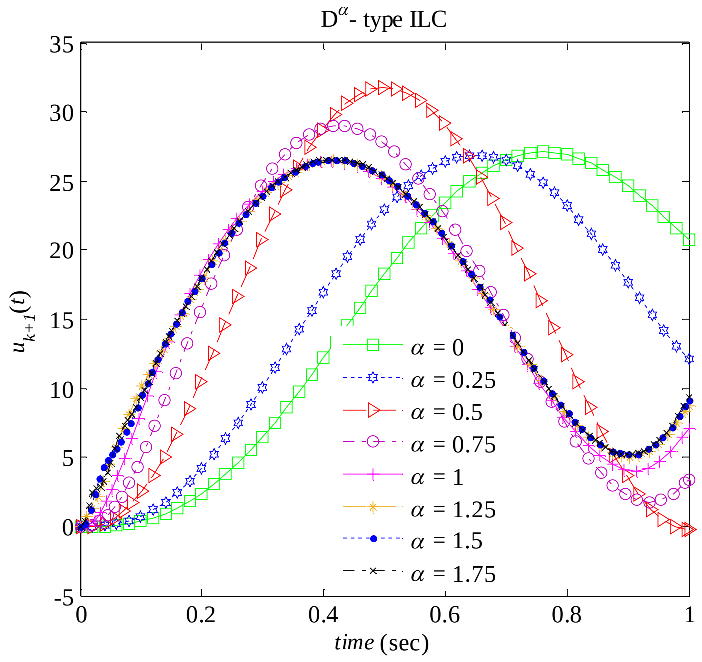

Fig. 7 Comparison The control effort vs. by using updating law D^α-type ILC.

شکل 7 نمودار سیگنال کنترلی ورودی با استفاده از قانون بروزرسانی D^α-type ILC.

یادگیر تکرار شوندۀ مرتبه صحیح و کسری را مورد ارزیابی و مقایسه قرار داد. در واقع، اگر $\alpha = 0$ باشد فقط متغیرهای حالت برای بروزرسانی ILC استفاده می‌شوند.

که در این حالت ما یک قانون بروزرسانی ILC تناسبی (نوع P) داریم. هنگامی که $\alpha = 1$ باشد آنگاه مشتقی از متغیرهای حالت که شامل شتاب زاویه‌ای[2] است مورد استفاده قرار می‌گیرد و در نتیجه ما یک قانون بروزرسانی ILC مشتقی (نوع D) خواهیم داشت.

همانطور که در شکل 8 نشان داده شده است بعد از 40 تکرار به ازای $\alpha = \{0.75, 1, 1.25, 1.5, 1.75\}$ خروجی واقعی به خروجی مطلوب به صورت کامل همگرا شده است و برای $\alpha = \{0, 0.25, 0.5\}$ نیاز داریم که تعداد تکرار افزایش پیدا کند. چنانچه شرط پایان تکرار ILC را $J_k(e_{m1}) < 1^\circ$ در نظر بگیریم، نتایج بدست آمده در شکل 9 نشان می‌دهد که به ازای $\alpha = 1.75$ در تکرار 13 شرط $J_k(e_{m1}) < 1^\circ$ ارضا می‌شود.

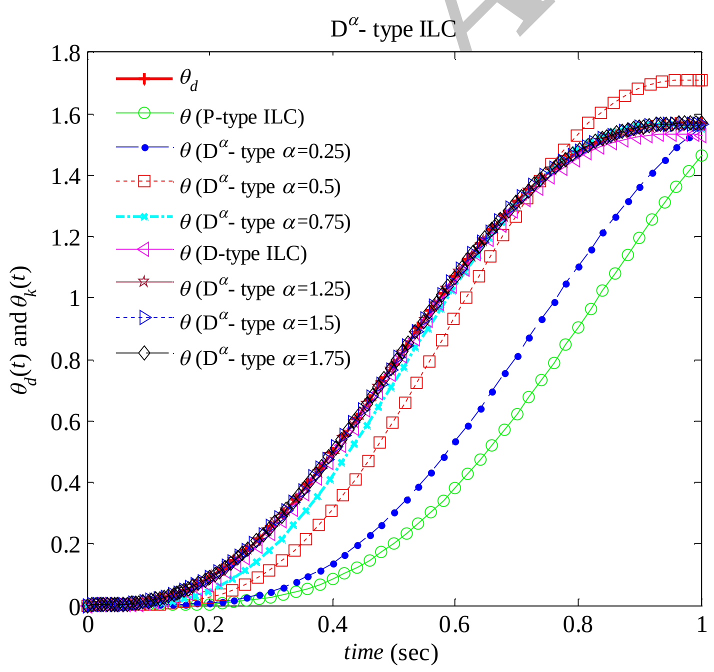

Fig. 8 Desired and the output response of the D^α-type ILC update law, for different values of α and $k = 40$.

شکل 8 نمودار موقعیت مطلوب و موقعیت واقعی با استفاده از قانون بروزرسانی D^α-type ILC برای αهای مختلف و $k = 40$.

1- Angular Acceleration

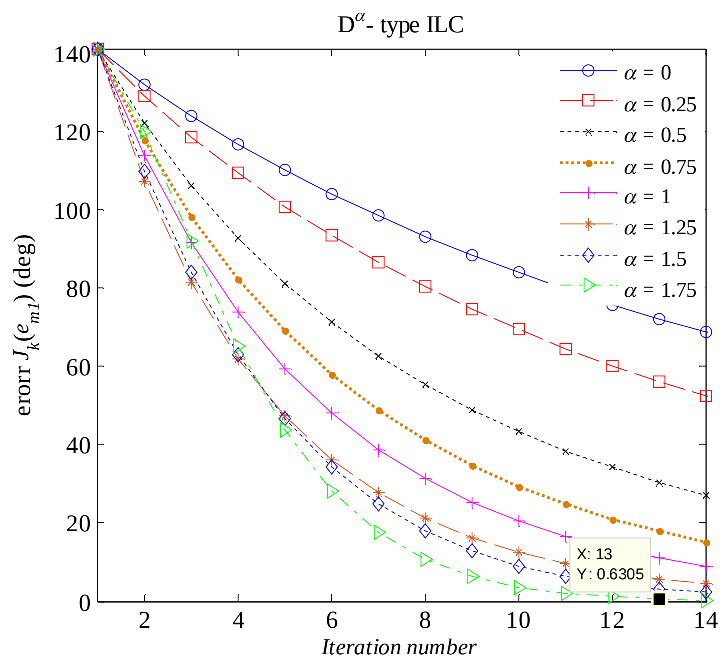

Fig. 9 Maximum absolute angular tracking errors $J_k(e_{m1})$, versus the iteration number, Comparison of D^{α}-type ILC with different α. With respect to the iteration number 14.

شکل 9 نمودار توان دوم حداکثر خطای ردیابی $J_k(e_{m1})$ بر حسب تعداد تکرار، با استفاده از قانون بروزرسانی D^{α}-type ILC برای مقادیر مختلف α و $k = 1, \dots, 14$.

خروجی واقعی بدست آمده در تکرار 15ام به ازای αهای مختلف برای قانون بروزرسانی ILC نوع D^{α} در شکل 10 نشان می‌دهد که در $\alpha = 1.75$ بیشترین سرعت همگرایی را نسبت به دیگر αها خواهیم داشت.

برای نمایش نحوۀ عملکرد قانون بروزرسانی ILC نوع PD^{α} از معادلۀ (10) برای تعیین ورودی فرآیند در تکرار 1 + kام استفاده شده است. برای شبیه‌سازی مقادیر بهره‌های یادگیری $k_P = J/2$ و $k_D = J$ است و بقیۀ تنظیمات همانند قانون بروزرسانی ILC نوع D^{α} است. نتایج شبیه‌سازی توان دوم حداکثر خطای ردیابی بر حسب تعداد تکرار، برای مقادیر مختلف α در شکل 11 نشان داده شده است.

نحوۀ همگرایی خروجی واقعی $\theta(t)$ به خروجی مطلوب $\theta_d(t)$، با استفاده از قانون بروزرسانی PD^{α}-type ILC برای تکرار $k = 40$ در شکل 12 نشان داده شده است. همانطور که در این شکل مشاهده می‌شود به ازای برخی از مقادیر α قانون بروزرسانی PD^{α}-type ILC بخوبی توانسته است

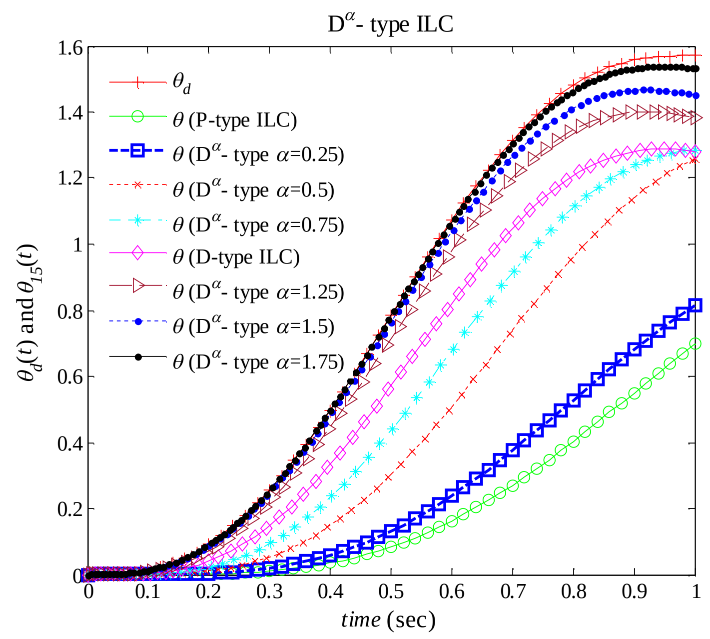

Fig. 10 The control effort for different values of α when $e_m < 0.01^{\circ}$, by using updating law D^{α}-type ILC and k=15.

شکل 10 نمودار خروجی مطلوب و خروجی واقعی با استفاده از قانون بروزرسانی D^{α}-type ILC برای αهای مختلف و $k = 15$.

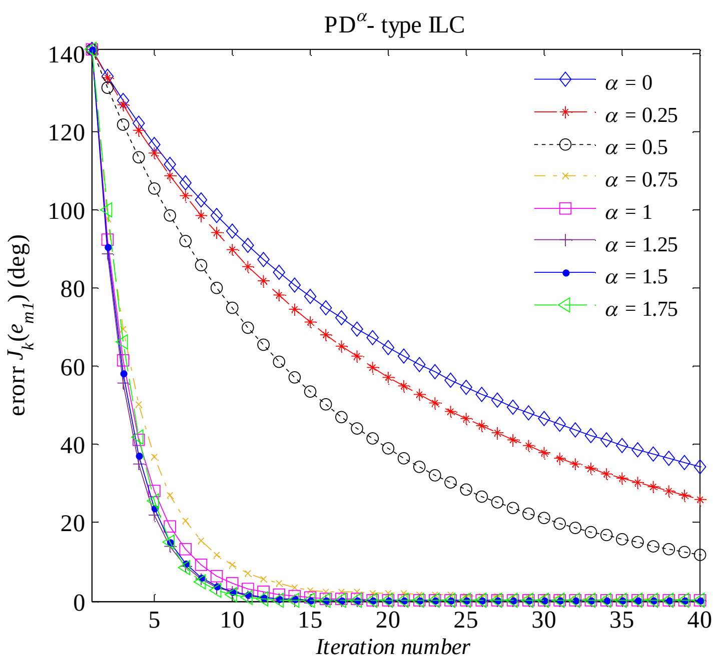

Fig. 11 Maximum absolute angular tracking errors versus the iteration number, Comparison of PD^{α}-type ILC with different α and $k = 1, \dots, 40$.

شکل 11 نمودار توان دوم حداکثر خطای ردیابی $J_k(e_{m1})$ بر حسب تعداد تکرار، با استفاده از قانون بروزرسانی PD^{α}-type ILC برای مقادیر مختلف α و $k = 1, \dots, 40$.

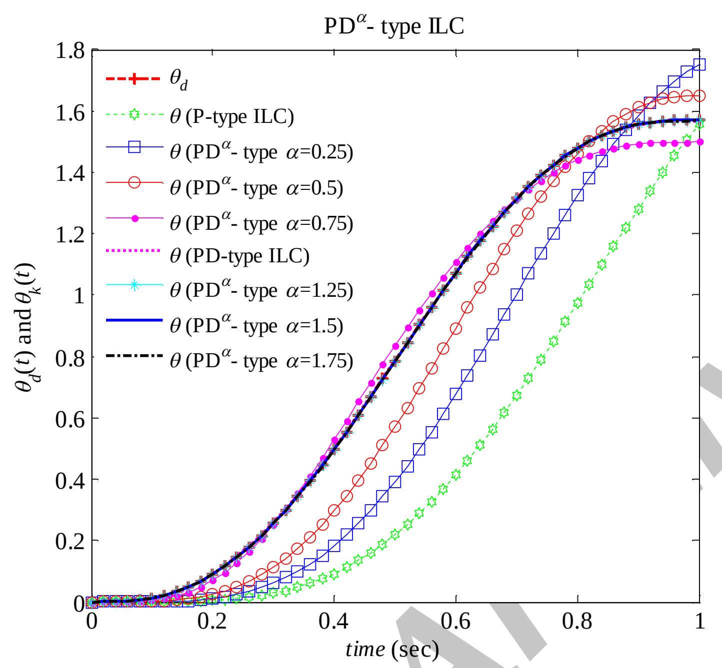

Fig. 12 Desired and the output diagram by using updating law PD^{α}-type ILC for different values of α and k=40.

شکل 12 نمودار خروجی مطلوب و خروجی واقعی بدست آمده با استفاده از قانون بروزرسانی PD^{α}-type ILC برای مقادیر مختلف α و $k = 40$.

همگرایی خروجی واقعی به خروجی مطلوب را تأمین کند

سیگنال کنترلی ورودی بدست آمده با استفاده از قانون بروزرسانی کنترل یادگیر تکرارشونده PD^{α} برای مقادیر مختلف α در تکرار $k = 40$، در شکل 13 نشان داده شده است. با توجه به شکل 11 مشاهده می‌شود قانون بروزرسانی PD^{α}-type ILC (برای $\alpha = 1.75$) با در نظر گرفتن شرط پایان تکرار ILC زمانی که $e_{m1} < 1^{\circ}$) توانسته است در تکرار 11ام به خروجی مطلوب همگرا شود.

با توجه به نتایج بدست آمده توسط قوانین بروزرسانی PD^{α}-type ILC و D^{α}-type ILC در این بخش نتیجه می‌گیریم که با انتخاب مناسب ضرایب α، k_D و k_P می‌توان سرعت همگرایی را افزایش داد.

به همین دلیل در بخش بعد، با استفاده از الگوریتم BBO که در بخش 6 معرفی شده است ضرایب FOILC را به صورت بهینه بدست می‌آوریم.

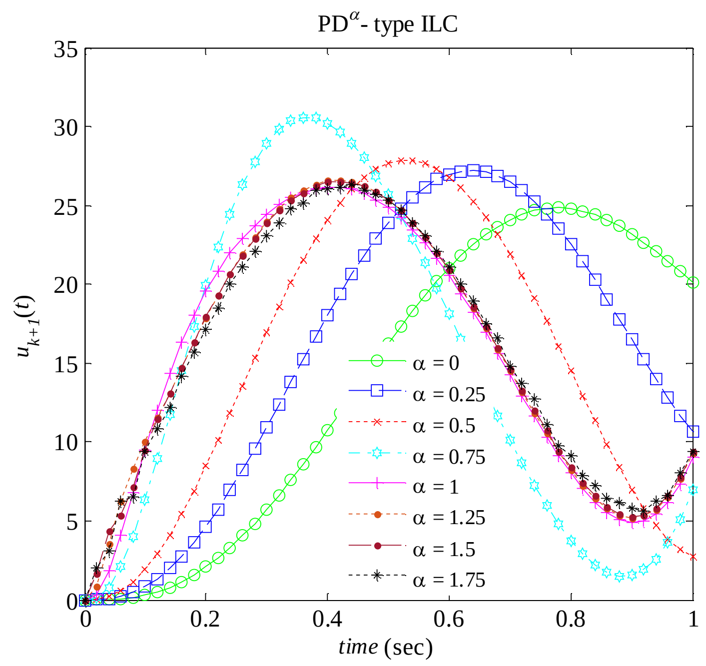

Fig. 13 The control effort for different values of α, by using updating law PD$^\alpha$-type ILC and k=40.

شکل 13 نمودار سیگنال کنترلی ورودی با استفاده از قانون بروزرسانی PD$^\alpha$-type ILC برای αهای مختلف و $k = 40$.

7-2- کنترل یادگیر تکرار شونده مرتبه کسری بهینه

برای تنظیم پارامترهای BBO، ماکسیمم تعداد تکرار $Max_{It} = 10$، تعداد محل‌های سکونت[1] $p = 50$، تعداد زیستگاه‌های نخبه $n_{keep} = 5$، بعد گونه‌های مسئله (SIV) $n_{Var} = 3$، حداکثر نرخ جهش $m_{max} = 0.05$، در معادلهٔ (43) مقدار $\gamma = 0.9$، حداکثر نرخ مهاجرت $I = 1$ و $E = 1$ می‌باشد.

مقادیر پارامترهای بدست آمده توسط الگوریتم BBO برای هر دو قانون بروزرسانی کنترل یادگیر تکرارشونده مرتبه‌کسری در جدول 2 نشان داده شده است. نحوهٔ همگرایی یکنواخت برای الگوریتم BBO بطوریکه تابع هزینه به تدریج کاهش می‌یابد در شکل 14 قابل‌مشاهده است. نحوه همگرایی خروجی واقعی به خروجی مطلوب برای هر دو قانون بروزرسانی کنترل یادگیر تکرار شوندهٔ PD$^\alpha$-type ILC و D$^\alpha$-type ILC در شکل 15 نشان داده شده است، که در این شکل پارامترهای هر دو کنترل‌کنندهٔ FOILC توسط الگوریتم BBO بدست آمده است.

همانطور که در شکل 15 قابل مشاهده است سرعت همگرایی برای زمانی که ضرایب FOILC با استفاده از الگوریتم BBO بدست آمده بهبود پیدا می‌کند. همانطور که در شکل 6 نشان داده شده است بیشترین سرعت همگرایی برای D$^\alpha$-type ILC در 13 تکرار اتفاق افتاده است اما با استفاده از BBO-D$^\alpha$-type ILC سرعت همگرایی بهبودیافته و مطابق با شکل 15، به 3 تکرار می‌رسد، با توجه به شکل 11 بیشترین سرعت همگرایی برای قانون بروزرسانی PD$^\alpha$-type ILC در تکرار 11ام اتفاق افتاده است، در صورتی که با استفاده از BBO-PD$^\alpha$ type ILC مطابق با شکل 15 سرعت همگرایی بهبودیافته و به 2 تکرار رسیده است. سیگنال کنترلی ورودی بدست آمده برای هر دو قانون یادگیری مرتبه‌کسری بهینه در شکل 16 نشان داده شده است. خروجی بدست آمده در تکرار 10ام در شکل 17 قابل مشاهده است.

8- نتیجه‌گیری

در این مقاله با ترکیب محاسبات مرتبه‌کسری و کنترل‌کنندهٔ ILC، دو نوع از قانون‌های بروزرسانی PD$^\alpha$-type ILC و D$^\alpha$-type ILC مورد بررسی قرارگرفته‌اند. تجزیه و تحلیل همگرایی PD$^\alpha$-type ILC پیشنهاد شده برای سیستم‌های خطی ارائه شده است. شبیه‌سازی بر روی سیستم بازوی ربات

جدول 2 پارامترهای بدست آمده توسط الگوریتم BBO برای مقایسه عملکرد PD$^\alpha$-type ILC و D$^\alpha$-type ILC.

Table 2 Parameter values obtained from BBO algorithm for performance Comparison between PD$^\alpha$-type ILC and D$^\alpha$-type ILC.

PD$^\alpha$ نوع ILC	D$^\alpha$ نوع ILC	نوع پارامتر خروجی الگوریتم BBO
3.179	-	k_P
1.808	3.3712	k_D
1.7302	1.5886	α
1.38×10^{-10}	2.01×10^{-9}	J_{ITSE}

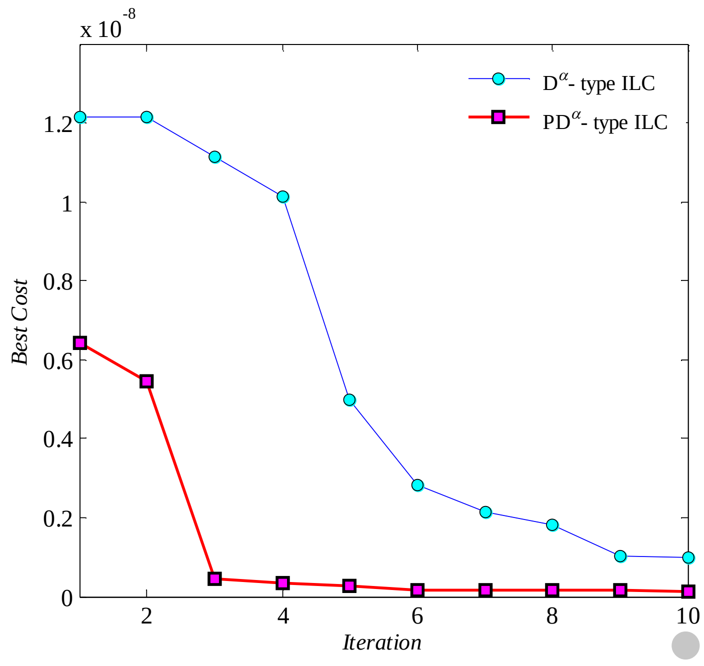

Fig. 14 Maximum absolute angular tracking errors versus the iteration number, for D$^\alpha$-type ILC and PD$^\alpha$-type ILC using the BBO algorithm.

شکل 14 نمودار همگرایی تابع هزینه الگوریتم BBO در محور تکرار، برای قوانین بروزرسانی PD$^\alpha$-type ILC و D$^\alpha$-type ILC.

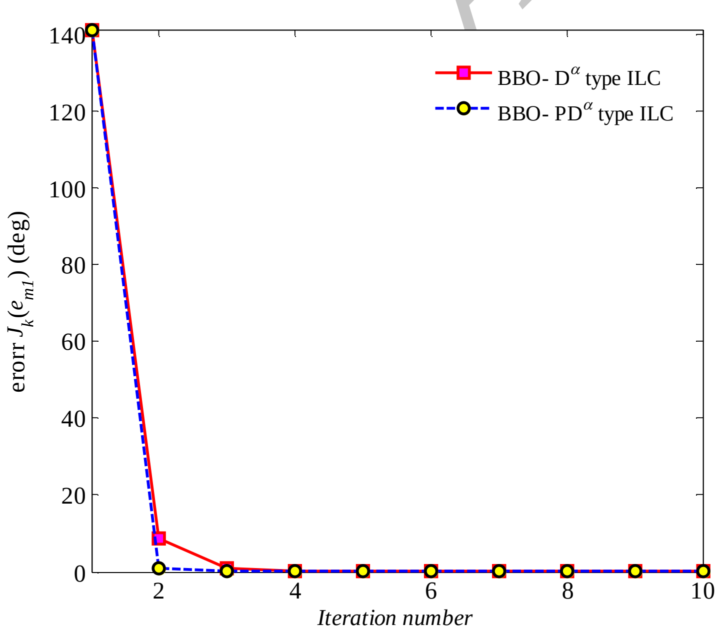

Fig. 15 Maximum absolute angular tracking errors versus the iteration number, for D$^\alpha$-type ILC and PD$^\alpha$-type ILC using the BBO algorithm.

شکل 15 نمودار توان دوم حداکثر خطای ردیابی $J_k(e_{m1})$ بر حسب تعداد تکرار، با استفاده از الگوریتم BBO برای قوانین بروزرسانی PD$^\alpha$-type ILC و D$^\alpha$-type ILC.

1- Population Size

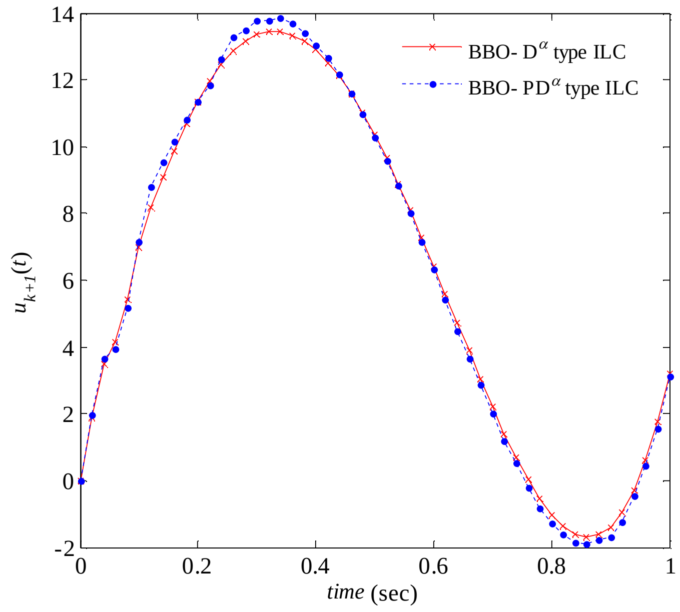

Fig. 16 The control signal for different values of α, by using updating laws BBO-D$^{\alpha}$-type ILC and BBO-PD$^{\alpha}$-type ILC for k=10.

شکل 16 نمودار سیگنال کنترلی ورودی برای مقادیر مختلف α، با استفاده از قانون بروزرسانی BBO-PD$^{\alpha}$-type ILC و BBO-D$^{\alpha}$-type ILC برای $k = 10$.

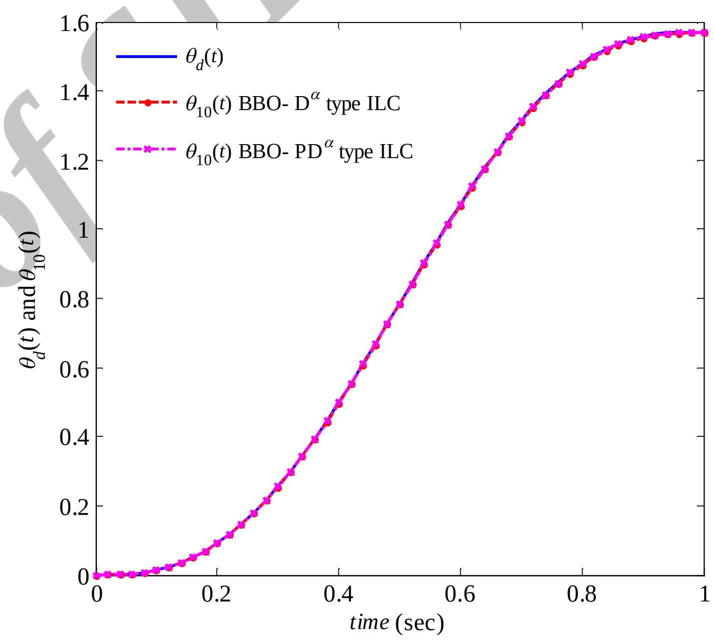

Fig. 17 Desired and the output trajectory of the robot arm of BBO-D$^{\alpha}$-type ILC and BBO-PD$^{\alpha}$-type ILC updating laws, for k=10 and different values of α.

شکل 17 نمودار خروجی مطلوب و خروجی واقعی بازوی ربات، با استفاده از قانون بروزرسانی BBO-PD$^{\alpha}$-type ILC و BBO-D$^{\alpha}$-type ILC برای $k = 10$ و مقادیر مختلف α.

خطی شده، پیاده‌سازی شده است و با تحلیل پاسخ‌های بدست آمده مشخص شده است که می‌توان با انتخاب بهینهٔ FOILC سرعت همگرایی را افزایش داد لذا برای بهبود عملکرد سیستم کنترلی از الگوریتم بهینه‌سازی BBO برای انتخاب بهینهٔ ضرایب FOILC استفاده شده است و نتایج بدست آمده از شبیه‌سازی بهبود سرعت همگرایی برای هر دو قانون یادگیری پیشنهاد شده در این مقاله را نشان می‌دهد. با توجه به نتایج بدست آمده در بخش شبیه‌سازی مشاهده می‌شود استفاده از کنترل‌کننده مرتبه‌کسری ILC پاسخ بهتری را نسبت به نوع متداول آن دارد، به‌طوریکه کسری بودن دینامیک کنترل‌کننده باعث افزایش درجه آزادی در انتخاب پارامترهای طراحی می‌شود و این امکان را به طراح می‌دهد که به پاسخ‌های با رفتار مناسب‌تر و نرم‌تر دست یابد و همچنین، می‌توان مشاهده نمود که قانون بروزرسانی PD$^{\alpha}$-type ILC نسبت به قانون بروزرسانی D$^{\alpha}$-type ILC دارای سرعت همگرایی بیشتری است که این عملکرد بهتر ناشی از بهرهٔ یادگیری اضافی $\boldsymbol{k_P}$ است.

9- مراجع

[1] Y. Wang, F. Gao, F. J. Doyle, Survey on iterative learning control, repetitive control, and run-to-run control, *Journal of Process Control*, Vol. 19, No. 10, pp. 1589-1600, 2009.
[2] S. Arimoto, S. Kawamura, F. Miyazaki, Bettering operation of robots by learning, *Journal of Robotic systems*, Vol. 1, No. 2, pp. 123-140, 1984.
[3] W. Chen, M. Tomizuka, Dual-Stage Iterative Learning Control for MIMO Mismatched System With Application to Robots With Joint Elasticity, *Control Systems Technology, IEEE Transactions on*, Vol. 22, No. 4, pp. 1350-1361, 2014.
[4] B. Zhang, D. Wang, Y. Ye, K. Zhou, Y. Wang, Cyclic pseudo-downsampled iterative learning control for high performance tracking, *Control Engineering Practice*, Vol. 17, No. 8, pp. 957-965, 2009.
[5] J.-X. Xu, Y. Tan, *Linear and nonlinear iterative learning control*: Springer Berlin, 2003.
[6] D. del Castillo Negrete, Fractional calculus: basic theory and applications (Part I), *Foro-Red-Mat: Revista electrónica de contenido matemático*, Vol. 16, No. 6, pp. 1, 2005.
[7] A. Tustin, J. Allanson, J. Layton, R. Jakeways, The design of systems for automatic control of the position of massive objects, *Proceedings of the IEE-Part C: Monographs*, Vol. 105, No. 1S, pp. 1-57, 1958.
[8] A. Oustaloup, *From fractality to non integer derivation: A fundamental idea for a new process control strategy*, in: *Analysis and optimization of systems*, Eds., pp. 53-64: Springer, 1988.
[9] A. Oustaloup, X. Moreau, M. Nouillant, The CRONE suspension, *Control Engineering Practice*, Vol. 4, No. 8, pp. 1101-1108, 1996.
[10] A. Oustaloup, J. Sabatier, P. Lanusse, From fractal robustness to CRONE control, *Fractional Calculus & Applied Analysis*, Vol. 2, No. 1, pp. 1-30, 1999.
[11] I. Polubny, Fractional-order systems and PIλDμ controller, *IEEE Transactions on Automatic Control*, Vol. 44, pp. 208-214, 1999.
[12] Y. Q. Chen, K. L. Moore, On D α-type iterative learning control, in *Proceeding of*, IEEE, pp. 4451-4456, 2001.
[13] D. Simon, Biogeography-based optimization, *Evolutionary Computation, IEEE Transactions on*, Vol. 12, No. 6, pp. 702-713, 2008.
[14] D. Simon, A probabilistic analysis of a simplified biogeography-based optimization algorithm, *Evolutionary computation*, Vol. 19, No. 2, pp. 167-188, 2011.
[15] N. Jerath, Implementation Of Biogeography Based Optimization on Image Restoration, *IJCAIT*, Vol. 1, No. 2, pp. 76-80, 2012.
[16] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm, *Journal of Sensor and Actuator Networks*, Vol. 1, No. 2, pp. 86-96, 2012.
[17] A. Bhattacharya, P. K. Chattopadhyay, Biogeography-based optimization for different economic load dispatch problems, *Power Systems, IEEE Transactions on*, Vol. 25, No. 2, pp. 1064-1077, 2010.
[18] A. Bhattacharya, P. Chattopadhyay, Application of biogeography-based optimisation to solve different optimal power flow problems, *IET generation, transmission & distribution*, Vol. 5, No. 1, pp. 70-80, 2011.
[19] J.-X. Xu, T. Heng Lee, H.-W. Zhang, Analysis and comparison of iterative learning control schemes, *Engineering Applications of Artificial Intelligence*, Vol. 17, No. 6, pp. 675-686, 2004.
[20] P. I. Corke, B. Armstrong-Hélouvry, A meta-study of PUMA 560 dynamics: A critical appraisal of literature data, *Robotica*, Vol. 13, No. 03, pp. 253-258, 1995.
[21] J.-J. E. Slotine, W. Li, *Applied nonlinear control*: Prentice-Hall Englewood Cliffs, NJ, 1991.
[22] H. Ma, D. Simon, Blended biogeography-based optimization for constrained optimization, *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 3, pp. 517-525, 2011.
[23] M. Mittal, Comparison between BBO and Genetic Algorithm, *International Journal of Science, Engineering and Technology Research*, Vol. 2, No. 2, pp. pp: 284-293, 2013.
[24] Y. Chen, C. Wen, *Iterative learning control: convergence, robustness and applications*: Springer-Verlag, 1999.
[25] D. Valério, J. S. da Costa, Ninteger: a non-integer control toolbox for MatLab, *Proceedings of Fractional Differentiation and its Applications, Bordeaux*, 2004.