کنترل بهینه ربات های صنعتی مبتنی بر رویتگر با استفاده از الگوریتم رقابت استعماری

امیر محمد بختیاروند

دانشجوی کارشناسی ارشد، دانشکده ی مهندسی برق، دانشگاه آزاد اسلامی واحد نجف آباد، ایران

Amirbakhtiar71@gmail.com

عباس چترایی

استادیار، دانشکده ی مهندسی برق، دانشگاه آزاد اسلامی واحد نجف آباد، ایران

Abbas.chatraei@gmail.com

چکیده

در این مفاله، یک کنترل کننده خطی ساز فیدبک بهینه مبتنی بر رویتگر برای کنترل یک ربات صنعتی طراحی شده است. هدف اصلی این پژوهش، ردیابی مسیر یک ربات صنعتی بدون استفاده از سنسور سرعت همراه با یک انرژی مناسب می باشد. ابتدا یک رویتگر بهره بالا جهت تخمین زاویه ها و سرعت ای زاویه ای مفصل های بازوی ربات ارائه می شود، سپس با استفاده از تخمین متغیر های حالت خطی سازی فیدبک به منظور ردیابی مسیر انجام می شود. بهره های کنترلی توسط الگوریتم رقابت استعماری به نحوی تعیین می گردند تا انرژی ربات را در طی مسیر کاهش دهند. پایداری سیستم حلقه بسته توسط تئوری لیاپانوف بررسی می شود و همچنین شبیه سازی های انجام شده توسط نرم افزار متلب، کارآیی طراحی ها را نمایش خواهند داد.

واژگان کلیدی: ربات های صنعتی، رویتگر بهره بالا، خطی سازی فیدبک، الگوریتم رقابت استعماری

۱ـ مقدمه

ربات های صنعتی امروزه در اکثر کارخانه ها و خطوط مونتاژ فعال می باشند، زیرا آن ها قادر هستند فرآیند های سخت و دشوار را با دقت و امنیت بالایی انجام دهند. کنترل مسیر موضوع مهمی است که بسیار مورد توجه محققان و پژوهشگران قرار گرفته است، به این علت که اکثر ربات ها در یک مسیر مشخص فعالیت می کنند و کنترل مسیر آن ها بر نحوه ی عملکرد شان تاثیر زیادی دارد. اندازه گیری زاویه ها و سرعت های زاویه ای مفصل های ربات برای کنترل مسیر آن ها بسیار مهم می باشد که این کار توسط سنسور های مربوطه انجام می شود. برخی از سنسورهای پیشرفته ی سرعت و شتاب گران و نادر هستند و همچنین استفاده از سنسور باعث ایجاد نویز در سیستم حلقه بسته خواد شد، به همین خاطر بهتر است برای غلبه بر این مشکل از رویتگر های حالت استفاده شود. نکته ی مهم دیگری که بهتر است به آن توجه شود، بهینه سازی انرژی است که سبب می شود تا یک بازده بالا با حداقل انرژی حاصل شود و همچنین این موضوع باعث می شود تا استهلاک ربات کاهش و طول عمر آن افزایش یابد.

معادله ی دینامیکی بازوهای ربات غیرخطی می باشد و برای کنترل آن ها می بایستی از روش های کنترل غیرخطی مثل مد لغزشی و خطی سازی فیدبک استفاده نمود. از آن جایی که معادله ی دینامیکی همه ی ربات ها از حل معادله ی اویلر ـ لاگرانژ حاصل می شود، لذا روش های کنترلی آن ها به یکدیگر قابل اعمال می باشد. در (Alam et al, 2018) برای ردیابی تراجکتوری های یک بازوی رباتیکی انعطاف پذیر، یک کنترل کننده مد لغزشی طراحی گردید. روش خطی سازی فیدبک مبتنی بر حذف ترم های غیرخطی می باشد که که این روش نیز جهت ردیابی تراجکتوری ها و یا به دلیل پیچیده بودن دینامیک ربات به کار برده می شود که در (Cox and Hurmuuzlu, 2018) و (Bannwarth et al, 2015) به همین دلیل استفاده شده است. در پژوهشی دیگر (Chandour et al, 2014) این روش در دو حلقه داخلی و خارجی جهت کنترل مسیر یک کوادراتور در صورت بروز عیب و در حالت عادی مورد استفاده قرار گرفت. همچنین در (Chatraei and Zada, 2013) یک کنترل کننده خطی ساز فیدبک به منظور ردیابی تراجکتوری های بهینه ی یک بازوی ربات طراحی گردید.

بهینه سازی ابزاری است ریاضی که هدف آن یافتن نقاط مینیمم یا ماکزیمم در توابع می باشد. بهینه سازی توسط دانشمندی به نام بلمن وارد حوزه مهندسی کنترل گردید که با استفاده از آن می توان ورودی های کنترلی ایجاد نمود که برخی معیارهای مورد نظر در سیستم بهینه شوند. روش های مختلفی برای حل مسائل کنترل بهینه وجود دارد که در برخی پژوهش های انجام شده (Korayem and Tourajizadeh, 2011)، (Tourajizadeh et al, 2016)، (Tchilian et al, 2017)، (Korayem et al, 2014)، (Korayem et al, 2015) و (Kamal et al, 2016) روش LQR برای حل این مسائل به کاربرده شده است. در یک تحقیق دیگر (Chatraie and Zada, 2011) یک کنترل کننده بهینه ترکیبی برای یک بازوی ربات که وظایف تکراری انجام می دهد طراحی شده است که برای مدل خطی شده ی سیستم، یک ورودی کنترلی بهینه محاسبه می شود و راه حل بهینه در حافظه ی سیستم ذخیره می شود تا راه حل بهینه ی بعدی محاسبه شود. در (Prsic et al, 2017) بهره های کنترل کننده ربات بازو موازی با استفاده از الگوریتم کرم شب تاب به صورت بهینه تنظیم شده اند و همچنین در (Jalali et al, 2013) کنترل کننده مد لغزشی با استفاده از الگوریتم رقابت استعماری بهینه شده است.

رویتگر حالت با استفاده از ورودی و خروجی یک سیستم می تواند تخمینی از متغیر های حالت آن را ارئه دهد. برای کنترل یک سیستم در اختیار داشتن متغیر های حالت آن ضروری می باشد، اما گاهی ممکن است که همه ی متغیر های حالت در دسترس نباشند، به همین خاطر می توان از رویتگر حالت استفاده نمود. در (Liang et al, 2015) برای مجموعه ای از بازوهای رباتیکی که به صورت شبکه ای فعالیت می کنند، یک کنترل کننده تطبیقی شراکتی همراه با یک رویتگر سرعت طراحی گردید. در یک تحقیق دیگر یک کنترل کننده گام به عقب غیر خطی با یک رویتگر سرعت ارائه شد (Chen and Lin, 2005). کنترل کننده فیدبک خروجی زمان محدود، بدون در نظر گرفتن سنسور سرعت طراحی شد و همچنین یک رویتگر غیر خطی نیز جهت جبران کمبود سنسور سرعت ارائه گردید (Hao et al, 2018). در (Ansareshlaghi and Eberhard, 2018) کنترل کننده خطی ساز

فیدبک مبتنی بر رویتگر غیرخطی، جهت کنترل مسیر ربات لامبدا به کار برده شد و همچنین در (Heredia and Yu, 2000) و (Merabet and Gu, 2008) از رویتگر بهره بالا جهت تخمین متغیر های حالت استفاده شده است.

در این مقاله، ما یک کنترل کننده خطی ساز فیدبک بهینه مبتنی بر رویتگر را ارائه می دهیم تا بازوی ربات بتواند مسیر مشخص شده را بدون استفاده از سنسور سرعت و همراه با یک انرژی مناسب ردیابی کند. در بخش دوم معادله دینامیکی بازوی ربات معرفی خواهد شد و سپس در قسمت سوم کنترل کننده خطی ساز فیدبک بهینه ارائه می شود. در قسمت چهارم رویتگر نیز به طراحی ها اضافه خواهد شد و کنترل کننده مبتنی بر آن طراحی می گردد. سپس در آخر نیز شبیه سازی های انجام شده توسط نرم افزار متلب ارائه می شوند.

۲ـ معادله ی دینامیکی بازوی ربات

معادله ی دینامیکی یک بازوی ربات با n مفصل به صورت زیر می باشد (Dawson et al, 2004):

$$M(q)\ddot{q}+H(q,\dot{q})+G(q)=T \qquad (۱)$$

که q و $\dot{q}$ به ترتیب بردار های موقعیت زاویه ای و سرعت زاویه ای مفصل های بازوی ربات می باشند. $M(q)\in R^{n\times n}$ ماتریسی اینرسی ربات است که متقارن و مثبت معین نیز می باشد. $H(q,\dot{q})\in R^{n}$ بردار کوریولیس و نیرو های گریز از مرکز می باشد و $G(q)\in R^{n}$ نیز بردار گرانشی است. $T\in R^{n}$ بردار گشتاور و یا ورودی های کنترلی می باشد. حال می توان (۱) را به صورت ساده تری باز نویسی نمود:

$$M(q)\ddot{q}+N(q,\dot{q})=T \qquad (۲)$$

به طوریکه

$$N(q,\dot{q})=H(q,\dot{q})+G(q) \qquad (۳)$$

$N(q,\dot{q})\in R^{n}$ از حاصل جمع برداری کوریولیس و گرانشی به دست می آید.

۳ـ کنترل کننده خطی ساز فیدبک بهینه

خطی سازی فیدبک روشی رایج است که برای کنترل سیستم های غیرخطی به کاربرده می شود. در صورتی که دینامیک دقیقی از سیستم و همچنین اندازه متغیرهای حالت را در اختیار داشته باشیم به راحتی می توان سیستم مورد نظر را طوری کنترل نمود که یک ورودی مرجع را ردیابی کند. این روش قادر است دینامیک غیر خطی سیستم حلقه بسته را خطی کند که در این صورت می توان با استفاده از روش های کنترل خطی سیستم را کنترل نمود.

کنترل بهینه بسطی از روش های ریاضی برای دست یافتن به یک قانون کنترلی مناسب می باشد. روش ها و الگوریتم های مختلفی در کنترل بهینه وجود دارد که الگوریتم رقابت استعماری یکی از این روش ها می باشد. مقادیر زیادی از بهره های کنترلی را می توان یافت که با در نظر گرفتن آن ها دینامیک خطای ردیابی پایدار است، اما همه ی آن ها بهینه نیستند که با استفاده از این الگوریتم می توان بهره های کنترلی را طوری پیدا نمود تا تابع هزینه ی مورد نظر کاهش یابد.

برای آن که بخواهیم بازوی ربات مسیر مورد نظر را ردیابی کند بایستی از خطی سازی فیدبک استفاده کنیم، همچنین الگوریتم رقابت استعماری می تواند بهره های کنترلی را به گونه ای تعیین کند که انرژی ربات در طی مسیر کاهش یابد.

۳ـ۱ کنترل کننده خطی سازی فیدبک

برای استفاده از این روش، بایستی بعد معادله ی دینامیکی ربات را در نظر گرفت:

$$\ddot{q} = M(q)^{-1}\left(-N(q,\dot{q}) + T\right) \quad (۴)$$

حال بایستی ورودی کنترلی T را طوری تعیین کنیم تا ترم های غیر خطی حذف گردند:

$$T = M(q)\left(v + M(q)^{-1} N(q,\dot{q})\right) \quad (۵)$$

با جایگذاری رابطه (۵) در (۴) نتیجه ی زیر حاصل خواهد شد:

$$\ddot{q} = v \quad (۶)$$

که v ورودی کنترلی جدید می باشد و بایستی یک قانون کنترلی مناسب را برای آن در نظر گرفت:

$$v = \ddot{q}_d + k_2(\dot{q}_d - \dot{q}) + k_1(q_d - q) \quad (۷)$$

که q_d و $\dot{q}_d$ به ترتیب بردار زاویه ها و سرعت های زاویه ای مطلوب می باشند. همچنین k_1 و k_2 دو ماتریس قطری به صورت زیر می باشند:

$$\begin{aligned} k_1 &= diag(k_{1i}) \\ k_2 &= diag(k_{2i}), i = 1,\dots n \end{aligned} \quad (۸)$$

با جایگذاری قانون کنترلی (۷) در رابطه (۶) به دینامیک خطای سیستم حلقه بسته دست یافت:

$$\ddot{e} + k_2\dot{e} + k_1 e = 0 \quad (۹)$$

که در رابطه بالا بهره های k_1 و k_2 بایستی به گونه ای انتخاب شوند تا دینامیک خطا پایدار (هورویتز) باشد.

۳ـ۲ بهینه سازی کنترل کننده خطی ساز فیدبک

این قسمت می خواهیم با استفاده از الگوریتم رقابت استعماری بهره های کنترلی را به گونه ای تنظیم کنیم تا انرژی ربات کاهش یابد. مراحل الگوریتم به شیوه ی (Atashpaz-Gargari and Lucas, 2007) می باشد. ابتدا بایستی بهره های k_1 و k_2 را در یک بازه ای که دینامیک خطا در آن پایدار است در نظر گرفت و بعد از بایستی کشور ها را تعیین نمود:

$$country(k_1, k_2) \quad (۱۰)$$

سپس هزینه ی هر کشور با توجه به تابع هزینه ی انرژی محاسبه خواهد شد:

$$J = \int_0^{t_f} T^T W T dt \quad (۱۱)$$

با توجه به رابطه (۱۱) کشور هایی که هزینه ی کمتری دارند را استعمارگر imp و آن هایی را که هزینه ی بیشتری دارند مستعمره col در نظر گرفته می شوند. این مستعره ها و استعمارگر ها با یکدیگر امپراطوری ها را تشکیل می دهند. با حرکت مستعمره ها به سوی استعمارگر ها، سیاست جذب اتفاق می افتد که مدل سازی آن به صورت زیر می باشد:

$$x^{new}_{col} = x^{old}_{col} + \beta r \otimes \left(x_{imp} - x^{old}_{col}\right) \tag{۱۲}$$

که x^{new}_{col} موقعیت جدید مستعمره و x^{old}_{col} موقعیت قبلی مستعمره می باشند. x_{imp} نیز بیان گر موقعیت استعمارگر است. همچنین r ماتریسی است که درایه های آن اعداد تصادفی در $[0,1]$ می باشد و β نیز ضریب جذب است. با توجه به سیاست جذب ممکن است، جای مستمره و استعمارگر با هم جابه جا شود. در مرحله بعد بایستی قدرت هر امپراطوری محاسبه شود:

$$P_n = c\left(imp_n\right) + \xi mean\left(c\left(col.imp_n\right)\right) \tag{۱۳}$$

که $c\left(imp\right)$ هزینه ی استعمارگر و $mean\left(c\left(col.imp_n\right)\right)$ میانگین هزینه ی مستعمرات هر امپراطوری می باشند. همچنین ثابت ξ عددی در $[0,1]$ می باشد. در مرحله بعد رقابت استعماری شروع می شود و مستعمره ها توسط امپراطوری هایی تصرف می شوند که احتمال تصاحب بیشتری داشته باشند. قدرت نرمالیزه هر امپراطوری به صورت زیر محاسبه می شود:

$$P.N_n = P_n - max\left(P_i\right) \tag{۱۴}$$

حال با دانستن قدرت نرمالیزه می توان احتمال تصاحب هر امپراطوری را محاسبه نمود:

$$P.P_n = \left| \frac{P.N_n}{\sum_{n=1}^{imp} P.N_n} \right| \tag{۱۵}$$

سپس بردار P تشکیل می شود:

$$P = \left[P.P_1, \ldots, P.P_{Nimp}\right] \tag{۱۶}$$

مجددا بردار دیگری که درایه های آن اعداد تصادفی $[0,1]$ می باشند و با بردار P هم بعد است ایجاد می شود:

$$L = \left[L_1, \ldots, L_{Nimp}\right] \tag{۱۷}$$

با تفاضل دو بردار نتیجه ی زیر حاصل می شود:

$$C = P - L = \left[C_1, \ldots, C_{Nimp}\right] \tag{۱۸}$$

از (۱۸) مشخص می شود که کدام امپراطوری، مستعمره ی امپراطوری ضعیف را تصاحب می کند. این عمل آن قدر تکرار می شود تا k^*_1 و k^*_2 حاصل شوند. با توجه به بهره های کنترلی بهینه، قانون کنترلی (۷) به صورت زیر بازنویسی می شود:

$$v = \ddot{q}_d + k^*_2\left(\dot{q}_d - \dot{q}\right) + k^*_1\left(q_d - q\right) \tag{۱۹}$$

همچنین دینامیک خطای سیستم حلقه بسته نیز به صورت زیر تغیر می کند:

$$\ddot{e} + k^*_2\dot{e} + k^*_1 e = 0 \tag{۲۰}$$

۴ـ کنترل کننده خطی ساز فیدبک بهینه مبتنی بر رویتگر بهره بالا

در برخی موارد فیدبک گرفتن از متغیر های حالت امکان پذیر نمی باشد، لذا می توان با استفاده از یک رویتگر این کار را آسان نمود. رویتگر می تواند با استفاده از خروجی قابل اندازه گیری سیستم و ورودی آن متغیر های حالت را تخمین بزند. در این مقاله ازیک رویتگر بهره بالا جهت تخمین متغیر های حالت استفاده شده است.

در طراحی های قبلی، فرض بر این بود که اندازه ی حالت ها به صورت کامل در دسترس می باشد، اما در این قسمت تنها اندازه زاویه های واقعی مفصل های ربات در اختیار داریم و از هیچ سنسور سرعتی جهت اندازه گیری سرعت های زاویه ای مفصل ها استفاده نشده است. در این بخش، ابتدا یک رویتگر بهره بالا جهت تخمین متغیر های حالت ارائه می شود و سپس در مراحل بعدی کنترل کننده مبتنی بر رویتگر طراحی خواهد شد.

۴ـ۱ رویتگر بهره بالا

با توجه به (۴) می توان نمایش فضای حالت بازوی ربات را به صورت زیر ارائه نمود:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + g(x_1)T \\ y = x_1 \end{cases} \qquad (۲۱)$$

که $x_1 = q$، $x_2 = \dot{q}$ و همچنین y خروجی قابل اندازه گیری می باشد. $f(x_1, x_2)$ و $g(x_1)$ به شکل زیر تعین می شوند:

$$\begin{aligned} f(x_1, x_2) &= -M(x_1)^{-1} N(x_1, x_2) \\ g(x_1) &= -M(x_1)^{-1} \end{aligned} \qquad (۲۲)$$

رویتگر بهره بالا تخمینی از مدل سیستم را ارائه می دهد و بر مبنای اختلاف میان خروجی قابل اندازه گیری و زاویه های تخمین زده شده عمل می کند که این رویتگر به صورت زیر طراحی می گردد:

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + H_1(y - \hat{x}_1) \\ \dot{\hat{x}}_2 = f(\hat{x}_1, \hat{x}_2) + g(\hat{x}_1)T + H_2(y - \hat{x}_1) \end{cases} \qquad (۲۳)$$

به طوریکه

$$\begin{aligned} f(\hat{x}_1, \hat{x}_2) &= -M(\hat{x}_1) N(\hat{x}_1, \hat{x}_2) \\ g(\hat{x}_1) &= -M(\hat{x}_1) \end{aligned} \qquad (۲۴)$$

همچنین $\hat{x}_1$ و $\hat{x}_2$ به ترتیب تخمینی از زاویه ها و سرعت های زاویه ای می باشند. H_1 و H_2 نیز بهره های رویتگر هستند. با تفاضل میان (۲۱) و (۲۳) دینامیک خطای رویتگر به دست می آید:

$$\begin{cases} \dot{\tilde{e}}_1 = -H_1 \tilde{e}_1 + \tilde{e}_2 \\ \dot{\tilde{e}}_2 = -H_2 \tilde{e}_1 + s(x, \hat{x}, t) \end{cases} \qquad (۲۵)$$

به صورتی که

$$\tilde{e} = \begin{bmatrix} \tilde{e}_1 = x_1 - \hat{x}_1 \\ \tilde{e}_2 = x_2 - \hat{x}_2 \end{bmatrix} \qquad (۲۶)$$

همچنین s نیز به صورت زیر حاصل می شود:

$$s(.) = f(.) - f(\hat{.}) + \left(g(.) - g(\hat{.})\right)T \qquad (۲۷)$$

حال می توان نمایش فضای حالت خطای رویتگر را ارائه داد:

$$\dot{\hat{e}}(t)=H\tilde{e}(t)+Ws(t) \tag{۲۸}$$

به طوریکه

$$\tilde{e}=\begin{bmatrix}\tilde{e}\\ \dot{\tilde{e}}\end{bmatrix},H=\begin{bmatrix}-H_1 & I_{n\times n}\\ -H_2 & 0\end{bmatrix},W=\begin{bmatrix}0_{n\times n} & I_{n\times n}\end{bmatrix}^T \tag{۲۹}$$

دقت شود ماتریس H به صورتی بایستی انتخاب گردد که دینامیک خطای رویتگر بهره بالا هورویتز باشد.

۴-۲ کنترل کننده خطی ساز فیدبک مبتنی بر رویتگر بهره بالا

در این بخش با استفاده از تخمین متغیر های حالت خطی سازی فیدبک را جهت ردیابی مسیر به صورت زیر انجام می دهیم:

$$T=M(\hat{x}_1)\left(\hat{v}+M(\hat{x}_1)^{-1}N(\hat{x}_1,\hat{x}_2)\right) \tag{۳۰}$$

با جایگذاری (۳۰) در (۲۳) نتیجه ی زیر حاصل خواهد شد:

$$\begin{cases}\dot{\hat{x}}_1=\hat{x}_2+H_1(y-\hat{x}_1)\\ \dot{\hat{x}}_2=\hat{v}+H_2(y-\hat{x}_1)\end{cases} \tag{۳۱}$$

با در نظر گرفتن قانون کنترلی زیر

$$\hat{v}=\dot{x}_{2d}-k_2(\hat{x}_2-x_{2d})-k_1(\hat{x}_1-x_{1d}) \tag{۳۲}$$

می توان به دینامیک خطای ردیابی مسیر دست یافت:

$$\ddot{e}(t)+k_2\dot{\hat{e}}(t)+k_1\hat{e}(t)=H_a\tilde{e}(t) \tag{۳۳}$$

که در رابطه (۳۳) H_a به صورت زیر تعیین می شود:

$$H_a=\begin{bmatrix}H_2 & 0_{n\times n}\end{bmatrix} \tag{۳۴}$$

حال می توان نمایش فضای حالت خطای ردیابی مسیر را به صورت زیر نوشت:

$$\dot{\hat{e}}=A\hat{e}(t)+B\tilde{e}(t) \tag{۳۵}$$

به طوریکه

$$\hat{e}=\begin{bmatrix}\hat{e}\\ \dot{\hat{e}}\end{bmatrix},A=\begin{bmatrix}0_{n\times n} & I_{n\times n}\\ -k_1 & -k_2\end{bmatrix},B=\begin{bmatrix}0_{n\times n} & 0_{n\times n}\\ H_2 & 0_{n\times n}\end{bmatrix} \tag{۳۶}$$

۴-۳ بهینه سازی کنترل کننده خطی ساز فیدبک مبتنی بر رویتگر بهره بالا

در این بخش الگوریتم رقابت استعماری به همان روش که در بخش های قبلی به صورت مفصل شرح داده شد بهره های کنترلی را با توجه به تابع هزینه ی انرژی به صورت بهینه تعیین می کند. و قانون کنترلی (۳۲) به صورت زیر بازنویسی می شود:

$$\hat{v}=\dot{x}_{2d}-k^*_2(\hat{x}_2-x_{2d})-k^*_1(\hat{x}_1-x_{1d}) \tag{۳۷}$$

در واقع این قانون کنترلی باعث می شود تا انرژی ربات در طی مسیر کاهش یابد..

۴ـ۴ آنالیز پایداری سیستم حلقه بسته

با در نظر گرفتن قانون کنترلی بهینه (۳۷) دینامیک خطای ردیابی به صورت زیر بیان می شود:

$$\ddot{\hat{e}}(t)+k^*_2\dot{\hat{e}}(t)+k^*_1\hat{e}(t)=H_a\tilde{e}(t) \quad (۳۸)$$

همچنین نمایش فضای حالت آن نیز به شکل زیر می باشد:

$$\dot{\hat{e}}=A\hat{e}(t)+B\tilde{e}(t) \quad (۳۹)$$

به طوریکه

$$\hat{e}=\begin{bmatrix}\hat{e}\\ \dot{\hat{e}}\end{bmatrix}, A=\begin{bmatrix}0_{n\times n} & I_{n\times n}\\ -k^*_1 & -k^*_2\end{bmatrix}, B=\begin{bmatrix}0_{n\times n} & 0_{n\times n}\\ H_2 & 0_{n\times n}\end{bmatrix} \quad (۴۰)$$

با توجه به نمایش فضای حالت خطای رویتگر (۲۸) و نمایش فضای حالت خطای ردیابی مسیر (۳۹) می توان نمایش فضای حالت خطای سیستم حلقه بسته (ربات + رویتگر + کنترل کننده خطی ساز فیدبک بهینه) را به صورت زیر ارائه داد:

$$\dot{e}(t)=Ce(t)+Ds(t) \quad (۴۱)$$

به صورتی که

$$e=\begin{bmatrix}\hat{e}\\ \tilde{e}\end{bmatrix}, C=\begin{bmatrix}A & B\\ 0_{2n\times 2n} & H\end{bmatrix}, D=\begin{bmatrix}0_{n\times 2n} & W\end{bmatrix}^T \quad (۴۲)$$

حال بایستی جهت بررسی پایداری سیستم حلقه بسته یک تابع لیاپانوف را در نظر گرفت:

$$V=e^T Le \quad (۴۳)$$

همچنین از آن جایی که ماتریس C هورویتز می باشد، می توان معادله ی لیاپانوفی زیر را در نظر گرفت:

$$C^T L+LC=-R \quad (۴۴)$$

که R و L هر دو ماتریس های مثبت معین و متقارن می باشند. با مشتق گیری از (۴۳) و انجام یک سری عملیات های ریاضی به نتیجه ی زیر خواهیم رسید:

$$\dot{V}=-e^T Re+2e^T LDs \quad (۴۵)$$

حال بایستی به رابطه زیر توجه نمود:

$$\begin{aligned}\lambda_{min}(L)\|e\|^2 &\le e^T Le\le \lambda_{max}(L)\|e\|^2\\ \lambda_{min}(R)\|e\|^2 &\le e^T Re\le \lambda_{max}(R)\|e\|^2\end{aligned} \quad (۴۶)$$

که $\lambda_{min}(R)$ ،$\lambda_{min}(L)$ و $\lambda_{max}(R)$ ،$\lambda_{max}(L)$ به ترتیب کم ترین و بیشترین مقدار ویژه ماتریس های R و L می باشند. حال می توان نتیجه گرفت:

$$\dot{V} \le -\lambda_{min}(R)\|e\|^2 + 2\|s\|\|D\|\|L\|\|e\| \tag{۴۷}$$

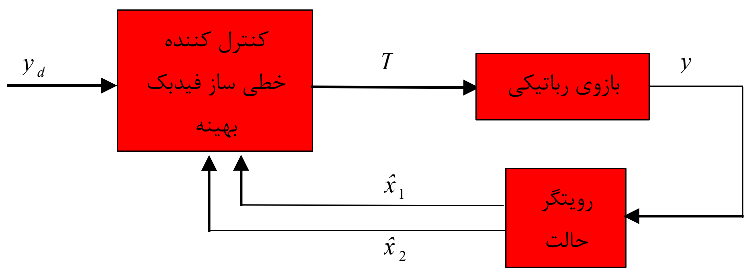
همچنین مقدار s از خطای رویتگر کوچکتر می باشد (Merabet and Gu, 2008):

$$\|s\| \le \alpha\|\tilde{e}\| \tag{۴۸}$$

با توجه به نکات ذکر شده به یک ناتساوی به صورت زیر دست خواهیم یافت:

$$\dot{V} \le -(\lambda_{min}(R) - 2\alpha\lambda_{max}(L))\|e\|^2 \tag{۴۹}$$

در صورتی که $\alpha \prec \frac{\lambda_{min}(R)}{2\lambda_{max}(L)}$ آن گاه $\dot{V}$ منفی معین خواهد شد. با استفاده از تئوری لاسال معکوس می توان گفت که سیستم به صورت مجانبی پایدار است (Merabet and Gu, 2008). شکل (۱) بلوک دیاگرام سیستم حلقه بسته را نشان می دهد.



شکل (۱) سیستم حلقه بسته

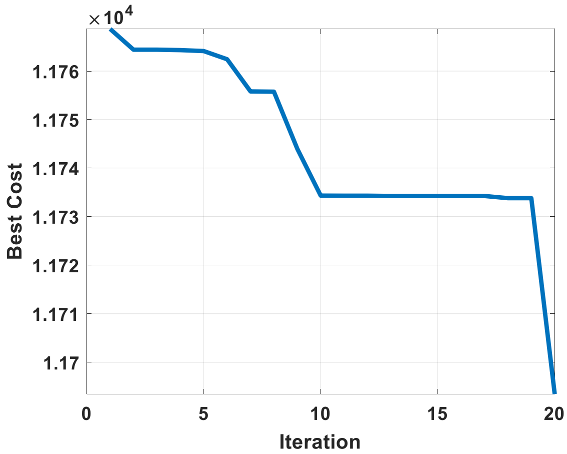
۵ـ نتایج شبیه سازی

بازوی رباتی که در این مقاله روش کنترلی بر روی آن اجرا شده است یک ربات ABBIRB140 با سه درجه آزادی می باشد که معادله ی دینامیکی آن از (Chatraie and Zada, 2011) اخذ شده است. در جدول (۱) مقادیر پارامترهای الگوریتم رقابت استعماری بیان شده است.

جدول (۱) مقادیر پارامترهای الگوریتم رقابت استعماری

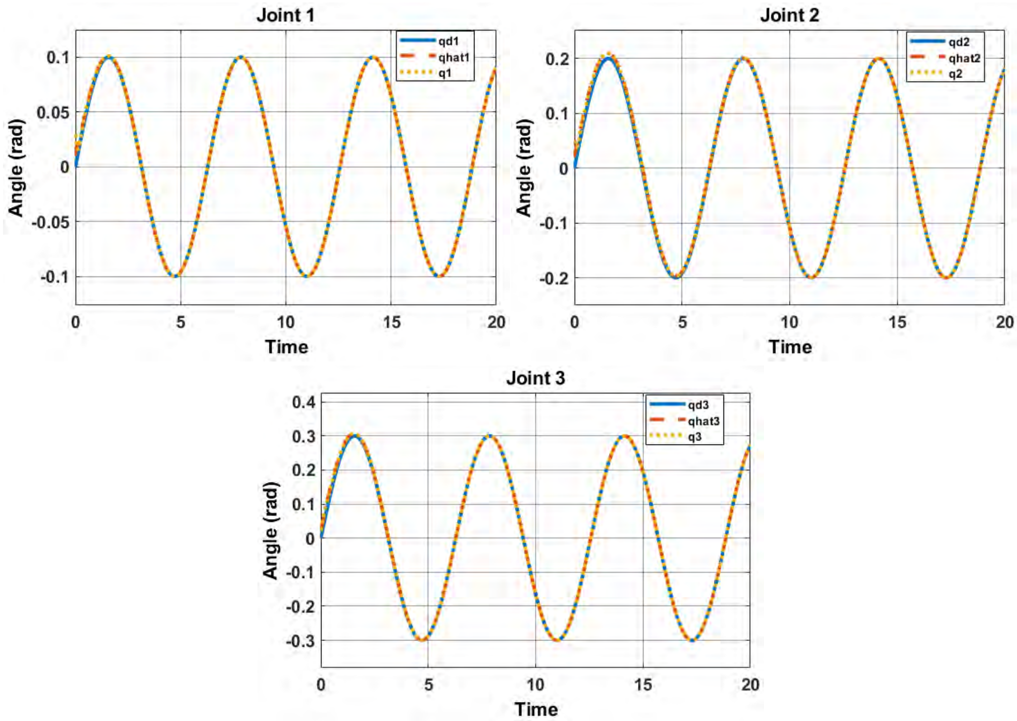
نماد	توصیف	مقدار
N	تعداد کشور ها	100
N_{imp}	تعداد امپراطوری ها	10
n	تعداد تکرار الگوریتم	20
β	ضریب جذب	2
ξ	ضریب میانگین هزینه مستعمره ها	$0/2$
W	ماتریس وزنی ورودی کنترلی	$diag(0/1)$

در شبیه سازی مقادیر اولیه ی زاویه های واقعی مفصل ها $q(0)=[0,0,0]^T$ و سرعت های زاویه ای اولیه $\dot{q}(0)=[0,0,0]^T$ می است. مسیر مورد نظر جهت ردیابی به صورت $q_d(t)=[0/1sin(t), 0/2sin(t), 0/3sin(t)]^T$ تعین شده است. همچنین بهره های رویتگر به صورت $H_1=diag(20)$ و $H_2=diag(200)$ انتخاب شده اند. بهره های کنترلی نیز در بازه $[diag(25), diag(75)]$ در نظر گرفته شده اند. شکل (۲) نشان می دهد که بعد از ۲۰ بار تکرار الگوریتم تابع هزینه (انرژی) کاهش یافته است و بهره های کنترلی بهینه با توجه به کاهش انرژی به صورت $k^*_1=diag(46/8192, 25/7095, 28/3459)$ و $k^*_2=diag(45/4854, 71/536, 34/9562)$ ارائه گردیدند.

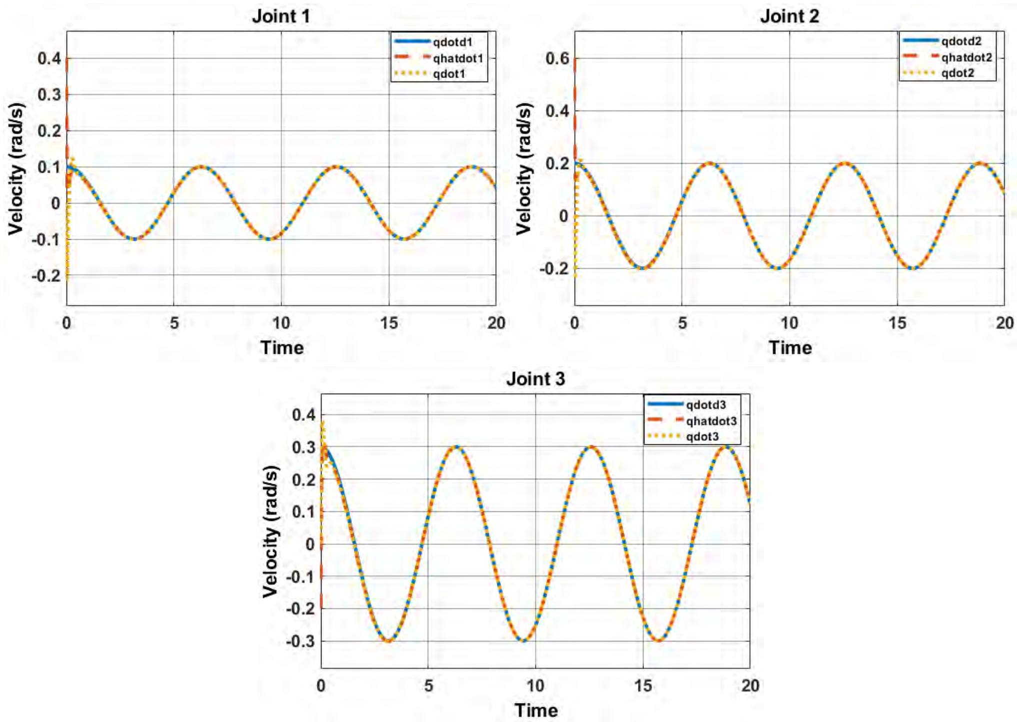


شکل (۲) مقدار تابع هزینه ی بهینه در هر تکرار

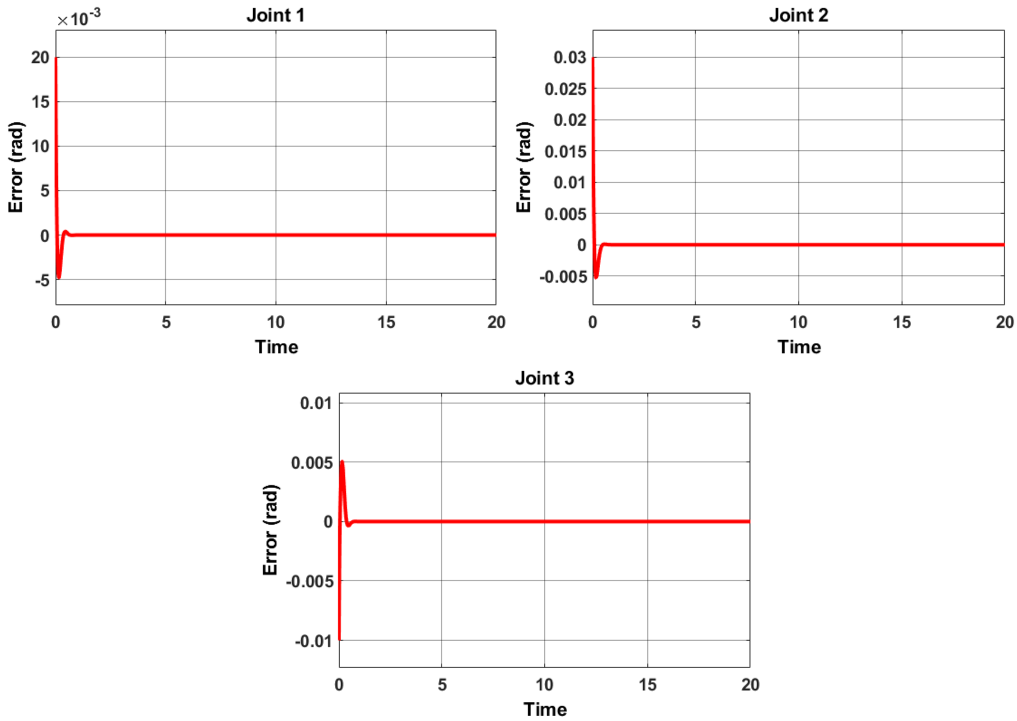
شکل (۳) زاویه های مطلوب، تخمین زده شده و واقعی مفصل های ربات را نشان می دهد. با مشاهده ی این شکل متوجه می شویم که گراف های مربوطه پس از چند ثانیه بر روی هم منطبق شده اند. شکل (۴) نشان دهنده ی سرعت های زاویه ای مطلوب تخمین زده شده و واقعی هر مفصل بازوی ربات می باشد که در بعد از مدت کوتاهی این سرعت های زاویه ای با یکدیگر برابر شده اند. شکل (۵) خطای تخمین زاویه ها را نشان می دهد، با توجه به این شکل می توان درک نمود که زاویه ها با چه دقت مناسبی تخمین زده شدند. شکل (۶) نشان دهنده ی خطای تخمین سرعت های زاویه ای است و این شکل نیز کارآیی مناسب رویتگر بهره بالا را نشان می دهد. شکل (۷) خطای ردیابی مسیر مورد نظر را نمایش می دهد و با مشاهده ی این شکل می توان درک نمود که هر مفصل توانسته مسیر مربوط به خود را به خوبی ردیابی کند زیرا که گراف ها پس چندی به صفر همگرا شده اند. شکل (۸) گشتاور های اعمال شده به مفصل های ربات را با در نظر گرفتن بهره های کنترلی بهینه نشان می دهد. با توجه به این شکل ها می توان فهید که ربات مسیر مورد نظر را با دقت مناسبی توانسته ردیابی کند، همچنین رویتگر بهره بالا به خوبی توانسته متغیر های حالت را جهت کنترل مسیر تخمین بزند و الگوریتم رقابت استعماری بعد از ۲۰ بار تکرار توانسته بهره های کنترلی را طوری تنظیم کند که انرژی ربات در طی مسیر کاهش یابد. در واقع شبیه سازی نشان می دهد که ما توانسته ایم به هدف مورد نظر که ردیابی مسیر از پیش تعیین شده، بدون استفاده از سنسور سرعت و همراه با یک انرژی مناسب است، دست یابیم.



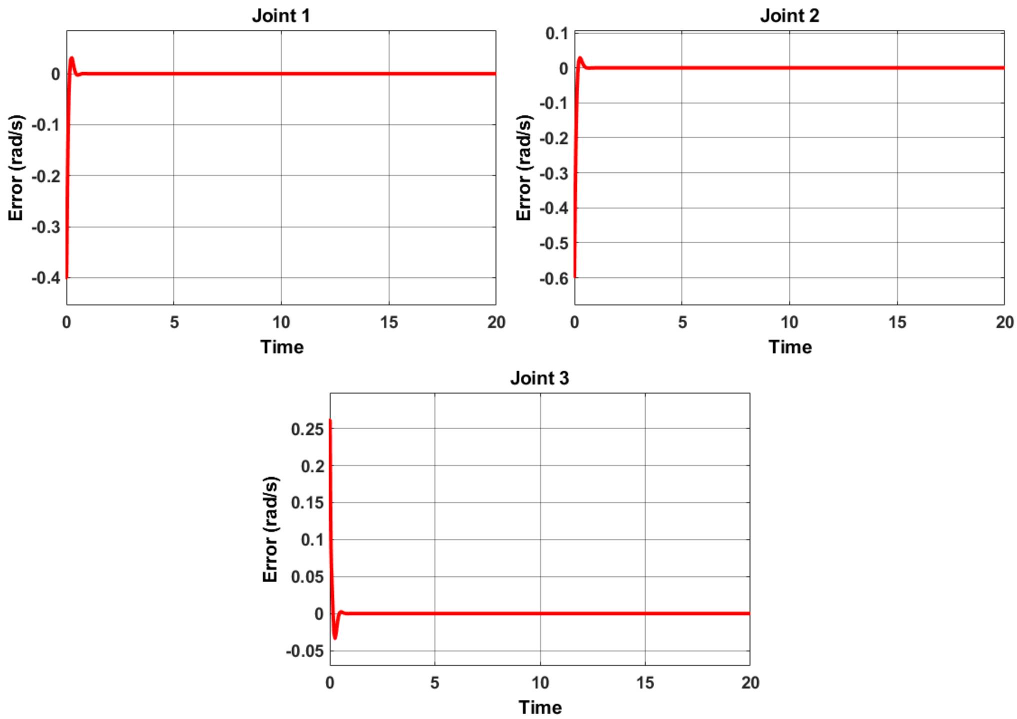
شکل (۳) زاویه های مطلوب (گراف های آبی رنگ پیوسته)، زاویه های تخمین زده شده (گراف های قرمز رنگ ناپیوسته) و زاویه های واقعی (گراف های زرد رنگ نقطه چین)



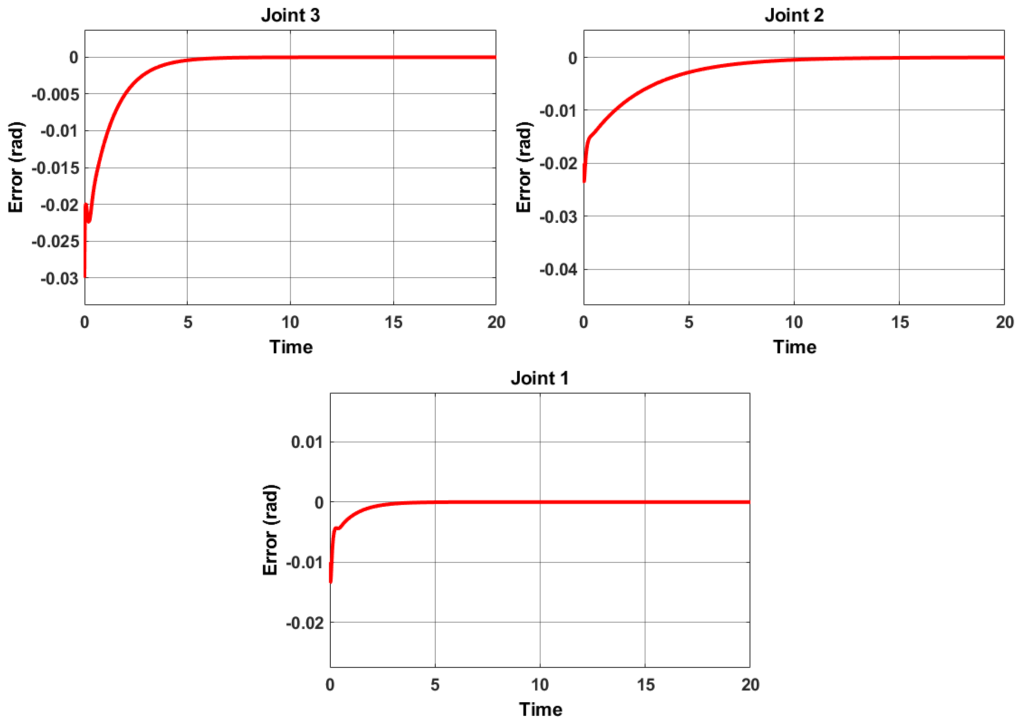
شکل (۴) سرعت های زاویه ای مطلوب (گراف های آبی رنگ پیوسته)، سرعت های تخمین زده شده (گراف های قرمز رنگ ناپیوسته) و سرعت های زاویه ای (گراف های زرد زنگ نقطه چین)



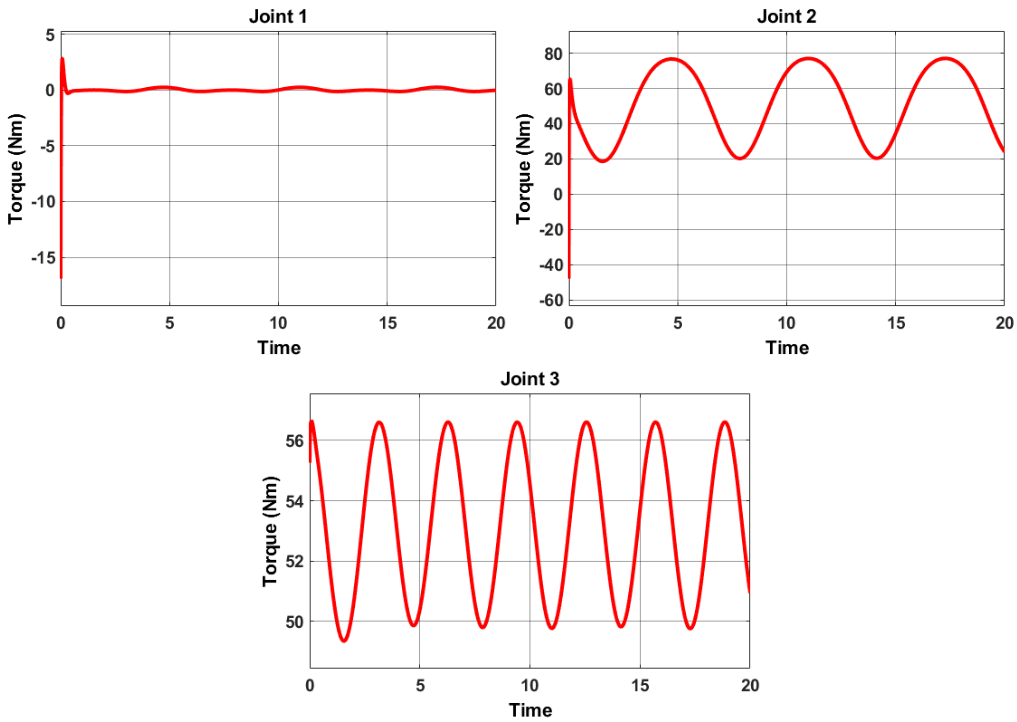
شکل (۵) خطای تخمین زاویه ها



شکل (۶) خطای تخمین سرعت های زاویه ای



شکل (۷) خطای ردیابی مسیر



شکل (۸) گشتاور های کنترلی بهینه

۶ـ نتیجه گیری

در این مقاله یک کنترل کننده خطی ساز فیدبک بهینه مبتنی بر رویتگر بهره بالا جهت کنترل مسیر یک بازوی ربات طراحی گردید. ابتدا رویتگر بهره بالا تخمینی از متغیر های حالت را ارائه داد و سپس کنترل کننده خطی ساز فیدبک با استفاده از تخمین حالت ها مسیر مشخص شده را ردیابی نمود، همچنین بهره های کنترلی توسط الگوریتم رقابت استعماری، با توجه به کاهش انرژی تعیین گردیند. شبیه سازی انجام شده نشان که ردیابی مسیر با دقت بالا و انرژی مناسب انجام شد و رویتگر بهره بالا توانست کمبود سنسور سرعت را جبران کند.

در این جا پیشنهاداتی ارائه می کنیم تا پژوهش های بعدی در این زمینه کامل تر انجام شود. برای آن که طراحی های انجام شده به صورت واقعی تری انجام شوند، بایستی اغتشاش های داخلی و خارجی را نیز در نظر گرفت و برای جبران آن ها از یک رویتگر اغتشاش استفاده نمود. همچنین اگر دینامیک عملگر ها را نیز در نظر بگیریم طراحی می تواند حقیقی تر باشد. از آنجایی که برخی پارامترها در معادله ی دینامیکی ربات به صورت دقیق سخت و دشوار می باشد و این موضوع می تواند در کنترل ربات مشکل ایجاد کند، برای غلبه بر این مشکل می توان از روش های کنترل تطبیقی استفاده نمود. همچنین علاوه بر انرژی، می توان زمان و خطای ردیابی را نیز بهینه نمود.

مراجع

Alam, Waqar. Mehmood, Adeel. Ali, Khurram. Javaid, Usman. Alharbi, Soltan. Iqbal, Jamshed. (2018). Nonlinear control of a flexible joint robotic manipulator with experimental validation. Strojniški vestnik-Journal of Mechanical Engineering. Vol. 64. No. 1. 47-55

Cox, Adam. Hurmuzlu, Yildirim. (2018). Feedback Linearization of Inertially Actuated Jumping Robots. Mechanical Engineering Research Theses and Dissertations

Bannwarth, JXJ. Munster, C. Stol, Karl A. (2015). Step ascension of a two-wheeled robot using feedback linearization. IEEE 2015 6th International Conference on Automation, Robotics and Applications (ICARA). 161-166

Ghandour, Jihad. Aberkane, Samir. Ponsart, Jean-Christophe. (2014). Feedback linearization approach for standard and fault tolerant control: Application to a quadrotor UAV testbed. Journal of Physics: Conference Series. Vol. 570. No. 8

Chatraei, Abbas. Záda, Václav. (2013). Global optimal feedback-linearizing control of robot manipulators. Asian Journal of Control. Vol. 15. No. 4. 1178-1187

Korayem, Moharam Habibnejad. Tourajizadeh, Hami. (2011). Maximum DLCC of spatial cable robot for a predefined trajectory within the workspace using closed loop optimal control approach. Journal of Intelligent & Robotic Systems. Vol. 63. No. 1. 75-99

Tourajizadeh, Hami. Yousefzadeh, Mahdi. Tajik, Ali. (2016). Closed loop optimal control of a Stewart platform using an optimal feedback linearization method. International Journal of Advanced Robotic Systems. Vol. 13. No. 3

Tchilian, Renan. Rafikova, Elvira. Gafurov, Salimzhan A. Rafikov, Marat. (2017). Optimal control of an underwater glider vehicle. Procedia Engineering. Vol. 176. 732-740

Korayem, MH. Tourajizadeh, Hami. Zehfroosh, A. Korayem, AH. (2015). Optimal regulation of a cable robot in presence of obstacle using optimal adaptive feedback linearization approach. Robotica. Vol. 33. No. 4. 933-952

Korayem, MH. Tourajizadeh, H. Zehfroosh, A. Korayem, AH. (2014). Optimal path planning of a cable-suspended robot with moving boundary using optimal feedback linearization approach. Nonlinear Dynamics. Vol. 78. No. 2. 1515-1543

Kamal, M, M. Shah, J, S. Khan, M, B. (2016). Stability Analysis and Trajectory Tracking Control of Robot for Industrial Application of Wielding. International Journal of Institutional & Industrial Research. Vol. 1. No. 2. 25-27

Chatraei, Abbas. Záda, Václav. (2011). Combined optimal control technique for robot manipulators1. Institute of Thermomechanics ASCR

Pršić, Dragan. Nedić, Novak. Stojanović, Vladimir. (2017). A nature inspired optimal control of pneumatic-driven parallel robot platform. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. Vol. 231. No. 1. 59-71

Jalali, Amin. Piltan, Farzin. Keshtgar, Maziyar. Jalali, Meysam. (2013). Colonial competitive optimization sliding mode controller with application to robot manipulator. International Journal of Intelligent Systems and Applications. Vol. 5. No. 7

Liang, Xinwu. Wang, Hesheng. Liu, Yun-Hui. Chen, Weidong. Hu, Guoqiang. Zhao, Jie. (2015). Adaptive task-space cooperative tracking control of networked robotic manipulators without task-space velocity measurements. IEEE transactions on cybernetics. Vol. 46. No. 10. 2386-2398

Chen, Fang-Shiung. Lin, Jung-Shan. (2005). Nonlinear control design of robotic manipulators with velocity observers. IFAC Proceedings Volumes. Vol. 38. No. 1. 193-198

Hao, Yong-Sheng. Su, Zhi-Gang. Wang, Xiangyu. (2018). Finite-Time Output Feedback Control for a Rigid Hydraulic Manipulator System. Mathematical Problems in Engineering

Ansarieshlaghi, Fatemeh. Eberhard, Peter. (2018). Trajectory Tracking Control of a Very Flexible Robot Using a Feedback Linearization Controller and a Nonlinear Observer. ROMANSY 22–Robot Design, Dynamics and Control Springer. 26-33

Heredia, Jose Antonio. Yu, Wen. (2000). A high-gain observer-based PD manipulator control for robot. Proc. of the proceedings of the American control conference. 2518-2522

Merabet, A. Gu, J. (2008). Estimated feedback linearization controller with disturbance compensator for robotic applications. The Mediterranean Journal of Measurement and Control. Vol. 4. No. 3. 101-110

Dawson, Darren M. Abdallah, Chaouki T. Lewis, Frank L. (2003). Robot manipulator control: theory and practice. CRC Press

Atashpaz-Gargari, Esmaeil. Lucas, Caro. (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. Evolutionary computation, 2007. CEC 2007. IEEE Congress on. 4661-4667