

کنترل تطبیقی یک ربات دوبازویی با ترکیب خطی سازی پسخور و شبکه عصبی پیشنهادی

اردشیر محمدزاده

گروه مهندسی برق، دانشکده فنی و مهندسی، دانشگاه بناب، بناب، ایران a.mzadeh@bonabu.ac.ir

نشریه سالانه‌های غیرخطی در مهندسی برق

دوره ۵ - شماره ۲

پاییز و زمستان ۱۳۹۷

صفحات ۵۵ الی ۷۳

ISSN: 2322-3146

<http://jnsee.sut.ac.ir>

(تاریخ دریافت مقاله: ۱۳۹۶/۱۱/۱۶ تاریخ پذیرش مقاله: ۱۳۹۷/۰۹/۱۰)

چکیده

در این مقاله، با ترکیب شبکه عصبی پیشنهادی و کنترل کننده خطی سازی پسخور، یک روش کنترلی تطبیقی برای ردیابی و کنترل وضعیت یک ربات جابه جا کننده صنعتی ارائه می شود. ابتدا فرض می شود که دینامیک سیستم کاملاً معلوم است و سیگنال کنترلی بر اساس خطی سازی پسخور طراحی می شود. سپس برای حذف اثرات نامعینی و اغتشاش، پارامترهای شبکه عصبی بر اساس قضیه لیاپانوف چنان تنظیم می شوند که شرط لغزشی برقرار شود. نتایج شبیه سازی در شرایط مختلف از جمله حضور اغتشاش و تأخیر با کنترل کننده مد لغزشی معمولی مقایسه می شود. نتایج شبیه سازی نشان می دهد که کنترل کننده پیشنهادی عملکرد مناسبی دارد و می تواند در کاربردهای مختلف مورد استفاده قرار بگیرد.

واژه‌های کلیدی

کنترل کننده عصبی،
نرون‌های تحریک کننده و
ساکن کننده،
مد لغزشی، خطی سازی پسخور،
کنترل تطبیقی و ربات.



Sahand University
of Technology

Journal of Nonlinear
Systems in Electrical
Engineering

Vol.5, No.2,

Autumn and Winter 2018

ISSN: 2322 – 3146

<http://jnsee.sut.ac.ir>

Adaptive Control of Two-Link Robot Based on the Feedback Linearization Method and the Proposed Neural Network

Ardashir Mohammadzadeh

Assistant Prof, University of Bonab, Bonab, Iran

ABSTRACT

In this paper, an adaptive control method based on a proposed neural network and the feedback linearization technique is proposed to control industrial robot manipulators. First, it is assumed that the dynamics of the system is known and the controller is designed using feedback linearization. Then, to eliminate the effect of the uncertainties and external disturbances, the parameters of the neural network are adjusted based on the adaption laws such that the sliding condition is satisfied. The simulation results in different conditions such as, at the presence of disturbance and time delay, are compared to the conventional sliding mode control method. The simulation results show that the proposed control method results in good performance and can be used in different applications.

Keywords

Neural controller,
Excitation neurons,
Inhibition neurons,
Sliding mode,
Feedback linearization,
Adaptive control,
Robot.

۱- مقدمه

ربات‌های جابه‌جا کننده اشیاء از جمله سیستم‌های مکانیکی کنترل‌پذیر می‌باشند که در صنعت کاربرد فراوانی دارند. به علت دینامیک‌های غیرخطی این نوع ربات‌ها، جزء مسائل کنترلی چالش برانگیزند. این ربات‌ها باید بارهایی را با اندازه‌ها، وزن‌ها و توزیع وزن‌های متفاوتی جابه‌جا کنند. امروزه به علت نیاز به افزایش دقت در فرایندهای صنعتی، کنترل بازوهای مکانیکی روز به روز اهمیت بیشتری پیدا کرده است. طراحان با بالا بردن دقت مدل و پیچیده‌تر کردن کنترل‌کننده، سعی در بهبود عملکرد این ربات‌ها دارند. هدف مطلوب این است که این ربات‌ها بارهایی را با پارامترهای نامعلوم، با سرعت و دقت بالا جابه‌جا کنند.

درسال‌های اخیر مساله کنترل حرکت بازوهای مکانیکی با در نظر گرفتن دینامیک‌های محرک‌های الکتریکی به خصوص در سرعت‌های بالا و بارهای متغیر، توجه بسیاری از محققان علم رباتیک را به خود جلب کرده است. به عنوان نمونه در مرجع [۱]، کنترل‌کننده تطبیقی ارائه شده است که در آن، قوانین تطبیقی برای بهره پسخور بدست آورده شده است. در مرجع [۲]، کنترل‌کننده پسخور خطا با بهره تطبیقی ارائه شده و برای آنالیز مقاومت و پایداری از روش مدل‌غزشی استفاده شده است. در مرجع [۳]، برخی نامعینی‌های پارامتری در نظر گرفته شده و عملکرد کنترل‌کننده تطبیقی ارائه شده با شبیه‌سازی روی یک بازوی ربات نشان داده شده است. روش کنترلی پسگام^۱ برای بازوی‌های رباتیک در مرجع [۴]، مطالعه شده است. کنترل‌کننده مدل‌غزشی تطبیقی به دلیل سادگی و پایدار بودن مورد توجه بسیاری از محققان بوده است. این روش کنترلی برای بازوهای رباتیک در مراجع [۵-۹]، بررسی شده است. در مرجع [۱۰]، پارامترهای نامعین در دینامیک یک بازوی رباتیک با استفاده از روش نیوتن-اولر تخمین زده شده و با تحلیل لیاپانوف همگرایی روش ارائه شده نشان داده شده است. کنترل پیش‌بین غیرخطی در مرجع [۱۱]، برای یک بازوی ربات طراحی و پیاده‌سازی شده است. به طور کلی در روش‌های کلاسیک معمولاً دینامیک سیستم معلوم فرض می‌شود و کنترل‌کننده بر اساس این دینامیک طراحی می‌شود. برای آنالیز پایداری کران عدم قطعیت به صورت محافظه‌کارانه انتخاب می‌شود که در برخی موارد منجر به نوسان سیگنال کنترلی می‌شود.

برای حل مشکل نامعینی دینامیکی، شبکه‌های عصبی و سیستم‌های فازی در بسیاری از مطالعات مورد توجه قرار گرفته‌اند. به عنوان نمونه در مرجع [۱۲]، کنترل‌کننده مدل‌غزشی انتگرالی با شبکه‌های عصبی پرسپترون چند لایه ترکیب شده است؛ که در آن، شبکه‌های عصبی برای تخمین نامعینی‌ها و اغتشاش به کار رفته‌اند. در مرجع [۱۳]، کنترل‌کننده مدل‌غزشی مرتبه کسری برای بازوی رباتیک طراحی شده است که در آن، پارامترهای کنترل‌کننده با استفاده از الگوریتم تکاملی مورچگان^۲ بهینه‌سازی شده است. در مرجع [۱۴]، کنترل‌کننده فازی لغزشی پیشنهاد شده است که در آن، ابتدا بر اساس تکنیک خطی‌سازی پسخور، دینامیک معلوم سیستم حذف می‌شود؛ سپس برای آنالیز پایداری، کنترل‌کننده مدل‌غزشی به کار گرفته می‌شود. سیستم فازی سوگنو در این مرجع برای بهبود عملکرد کنترل‌کننده مدل‌غزشی معمولی بکار رفته و با استفاده از شبیه‌سازی روی یک بازوی ربات نشان داده شده است که روش پیشنهادی مفید است. در مرجع [۱۵]، رویکردی مشابه مرجع [۱۴]، بکار رفته است که در آن، پارامترهای کنترل‌کننده فازی لغزشی با استفاده از الگوریتم مورچگان اصلاح شده بهینه‌سازی شده است. روش کنترلی پسگام با استفاده از شبکه‌های عصبی -

^۱Backstepping

^۲Ant colony optimization algorithm

فازی در مرجع [۱۶] ارائه شده است. در این مرجع، قوانین تطبیقی برای آموزش پارامترهای شبکه عصبی-فازی، از تحلیل پایداری لیاپانوف بدست می‌آیند. کنترل‌کننده فازی-لغزشی انتگرالی در مرجع [۱۷] مطالعه شده است. در این مرجع، برای حذف پدیده نوسان سیگنال کنترلی، از سیستم فازی برای تعیین تطبیقی کران نامعینی استفاده شده است. در مرجع [۱۸]، به منظور بهبود عملکرد روش کنترلی مدل‌لغزشی، کنترل‌کننده تناسبی-انتگرالی-مشتقی (PID) بر اساس سیستم‌های فازی با کنترل‌کننده مدل‌لغزشی ترکیب شده است. در مرجع [۱۹]، روش کنترلی PID با استفاده از سیستم‌های فازی-عصبی برای دسته‌ای از بازوهای رباتیک صنعتی بررسی شده است. در مرجع [۲۰]، روش کنترلی عصبی تطبیقی برای کلاسی از بازوهای رباتیک به کار رفته است که در آن، شبکه‌های عصبی پرسپترون چندلایه برای تخمین نامعینی‌ها استفاده می‌شود. در مرجع [۲۱]، کنترل‌کننده مد لغزشی با استفاده از شبکه‌های عصبی شعاعی برای کلاسی از بازوهای رباتیک طراحی شده است.

در برخی روش‌های ارائه شده با استفاده از شبکه‌های عصبی، از الگوریتم‌های بهینه‌سازی تکاملی برای تنظیم پارامترهای شبکه عصبی استفاده شده است که این روش‌ها تضمین پایداری و همگرایی ندارند و همچنین حجم محاسبات بالایی دارند که مناسب کاربردهای صنعتی نیست. در بیشتر روش‌های کنترلی ارائه شده بر اساس شبکه‌های عصبی-فازی، از شبکه‌های عصبی ساده مثل پرسپترون چند لایه یا سیستم‌های فازی استفاده شده است. سیستم‌های فازی برای اینکه کارایی قابل قبولی داشته باشند باید ساختار مناسبی داشته باشند. به عبارت دیگر باید پایگاه قوانین خوبی داشته باشند و معمولاً این قوانین به صورت سعی و خطا نوشته می‌شوند و در طول زمان فرم آنها ثابت است. از طرف دیگر شبکه‌های عصبی پرسپترون چندلایه ساده برای کارایی مناسب نیاز به تعداد نرون‌های بیشتری دارند.

همانطور که گفته شد، کنترل‌کننده مدل‌لغزشی به دلیل سادگی و مقاوم بودن، مورد توجه بسیاری از محققان بوده است و مطالعات برای بهبود عملکرد این روش کنترلی ادامه دارد. در حالت کلاسیک، برای برقرار شرط لغزشی از یک تابع اشباع با نوار مرزی کوچک استفاده می‌شود. اگر پهنای لایه مرزی کوچک باشد، منجر به لرزش سیگنال کنترلی می‌شود که در حالت کلی بسیار نامطلوب است؛ چون که در برگیرنده فعالیت کنترلی زیاد است و نیز ممکن است دینامیک‌های فرکانس بالای صرف‌نظر شده در هنگام مدل‌سازی را تحریک کند. اگر پهنای لایه مرزی بزرگ انتخاب شود، دقت ردیابی پایین می‌آید.

در این مقاله، برای حل این مساله از شبکه عصبی پیشنهادی که بر پایه نرون‌های پویاست، به همراه کنترل‌کننده‌های کلاسیک استفاده می‌شود. سیگنال کنترلی با ترکیبی از خطی‌سازی پسخور و شبکه عصبی پیشنهادی بدست می‌آید. پارامترهای شبکه عصبی بر اساس تابع لیاپانوف برگرفته از مدل‌لغزشی، آموزش بینند. مزیت این روش عدم نیاز به مدل ریاضی دقیق و دانش زیاد از سیستم و مقاوم بودن در برابر تغییر پارامترهای مختلف سیستم و اغتشاش است.

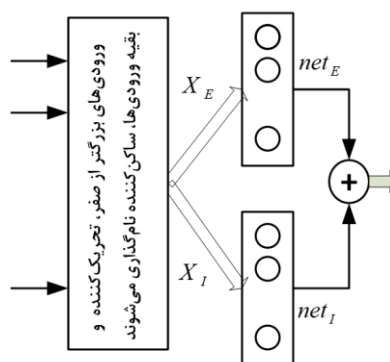
روش کنترلی پیشنهادی، به خاطر توابع سیگموئید به کار رفته در لایه خروجی و میانی شبکه عصبی، ناپیوستگی سیگنال کنترلی در حالت کلاسیک را هموار کرده و یک لایه مرزی متغیر بازمان حول سطح لغزشی ایجاد می‌کند. در لحظات اولیه که خطای ردیابی زیاد است، پهنای این لایه مرزی بزرگ بوده و با ردیابی کامل به صفر می‌رسد. کنترل کلاسیک، توانایی مقابله با آشفتگی‌هایی با شدت بیش از کران پیش‌بینی شده در ضمن طراحی قانون کنترلی را ندارد؛ در حالی که کنترل‌کننده هوشمند

طراحی شده در این مقاله به خاطر قابلیت تطبیقی که در شبکه عصبی به کار گرفته شده است، توانایی جوابگویی به این آشفتگی‌ها را دارد.

در ادامه این مقاله در بخش ۲، ابتدا شبکه عصبی استفاده شده معرفی شده و روابط تنظیم وزن‌ها بیان می‌شود. در بخش ۳-۱، روش آموزش شبکه براساس تابع لیاپانوف تشریح شده و روابط مربوط برای آموزش وزن‌ها بدست آورده می‌شود، در بخش ۳-۲، ساختار کنترلی بیان شده و روابط مربوط به کنترل‌کننده بدست آورده می‌شود. در بخش ۴ نتایج شبیه‌سازی آورده شده و در بخش ۵ نتیجه‌گیری می‌شود.

۲- شبکه عصبی پیشنهادی

حجم محاسبات بالا در شبکه‌های عصبی پرسپترون چندلایه^۱ و دیگر شبکه‌های عصبی به‌خصوص در حالت‌هایی که سیستم دارای تاخیر می‌باشد یا وابستگی زیادی بین داده‌های هر لحظه با لحظات قبل وجود دارد، باعث بزرگ شدن ساختار شبکه عصبی می‌شود. این بزرگ شدن ساختار شبکه عصبی، دقت محاسبات را در شبکه عصبی با تعداد تکرار کم آموزش، کاهش می‌دهد. به همین منظور تفکر استفاده از شبکه‌های عصبی پویا (دینامیک) مطرح گردیده است، این گونه شبکه‌ها علاوه بر حل مشکلات فوق، در مورد سیستم‌هایی که دینامیک هر لحظه به لحظات قبل مربوط است یا هنگامی که همبستگی بین داده‌ها وجود دارد، یا در حالت وجود اینرسی در سیستم، کارایی خوبی دارند. در این مقاله برای افزایش سرعت محاسبات و کاهش پیچیدگی، شبکه عصبی پویا به صورتی که در بخش ۲-۱ توضیح داده می‌شود، اصلاح شده است. در این شبکه پیشنهادی یک پیش پردازش روی ورودی‌های شبکه انجام شده و ورودی‌ها به دو دسته تحریک کننده و ساکن کننده تقسیم می‌شوند. همچنین، نرون‌های لایه میانی به دو دسته تحریک کننده و ساکن کننده تقسیم می‌شوند. ایده اصلی از شبکه عصبی فوکوشیما گرفته شده است [۲۲]. در این مرجع نشان داده شده است که پردازش اولیه روی ورودی‌ها باعث افزایش سرعت تخمین و کاهش تعداد نرون‌ها می‌شود. در ادامه، الگوریتم پیشرو شبکه عصبی پیشنهادی توضیح داده می‌شود. ساختار نرون پیشنهادی در شکل ۱، نشان داده شده است.



شکل ۱. ساختار نرون پیشنهادی

۲-۱ الگوریتم پیشرو

الگوریتم پیشرو برای لایه اول شبکه عصبی پیشنهاد شده به صورت زیر توصیف می‌شود:

^۱ Multi-Layer Perceptron

$$\begin{aligned} net_E^1(t) &= a^1 X_E(t) \\ net_I^1(t) &= b^1 X_I(t) \\ net_T^1(t) &= net_E^1(t) + net_I^1(t) \end{aligned} \quad (1)$$

که در آن X_E و X_I ، ورودی های تحریک کننده و ساکن کننده می باشند، net_E^1 و net_I^1 ، به ترتیب مجموع ورودی های وزندار نرون های تحریک کننده و ساکن کننده لایه فعال اول، a^1 و b^1 به ترتیب وزن های مربوط به نرون های تحریک کننده و ساکن کننده در لایه فعال اول است، همچنین برای لایه فعال دوم روابط زیر برقرار است:

$$\begin{aligned} net_E^2(t) &= a^2 O_E^1(t) \\ net_I^2(t) &= b^2 O_I^1(t) \\ net_T^2(t) &= net_E^2(t) + net_I^2(t) \end{aligned} \quad (2)$$

که در آن O_E^1 و O_I^1 به ترتیب خروجی های تحریک کننده و ساکن کننده لایه فعال اول، به عنوان ورودی های لایه فعال ساز دوم می باشند، net_E^2 و net_I^2 به ترتیب نرون های تحریک کننده و ساکن کننده لایه فعال ساز دوم، a^2 و b^2 به ترتیب وزن های مربوط به نرون های تحریک کننده و ساکن کننده در لایه فعال دوم است. حال مقدار محاسبه شده net_T^2 در رابطه (۳)، در تابع فعالساز f قرار داده می شود. لازم به ذکر است که تمامی متغیرهای بالا به صورت برداری می باشند.

$$o_{E/I}(t) = f_{E/I}(t) = \frac{1 - \exp(-net_T(t))}{1 + \exp(-net_T(t))} \quad (3)$$

تابع فوق از توابع فعالساز دوقطبی و مشتق پذیر می باشد.

۲-۲ الگوریتم آموزش شبکه عصبی پیشنهادی

الگوریتم پسخور این شبکه عصبی با استفاده از روش گرادیان نزولی و مشتق گیری زنجیره ای به شرح زیر، محاسبه می شود. تابع هزینه به صورت رابطه (۴) توصیف می شود:

$$E = \frac{1}{2} \sum_{i=1}^{n_2} (d_i - o_i)^2 = \frac{1}{2} \sum_{i=1}^{n_2} e_i^2 \quad (4)$$

که در آن d_i, o_i^2 به ترتیب خروجی های شبکه عصبی و مطلوب نرون i -ام می باشد و n_2 تعداد نرون های لایه فعالساز خروجی می باشد.

برای وزن های لایه فعالساز دوم، a^2 داریم:

$$\begin{aligned} \Delta a^2 &= -\eta \frac{\partial E}{\partial a^2(t)} = \\ &= -\eta \frac{\partial E}{\partial e} \frac{\partial e}{\partial o^2} \frac{\partial o^2}{\partial net_T^2} \frac{\partial net_T^2}{\partial net_E^2} \frac{\partial net_E^2}{\partial a^2} \\ \Delta a^2 &= \eta e F'^2(\cdot) \cdot 1 \cdot F^1 \\ &= \eta e F'^2(\cdot) \cdot o_E^1 \\ &= \eta \delta^2(k) o_E^1(k) \\ \delta^2 &= e F'^2 \end{aligned} \quad (5)$$

که در آن o_E^1 خروجی نرون‌های تحریک کننده لایه میانی است

همچنین برای وزن‌های لایه فعالساز دوم، b^2 داریم:

$$\begin{aligned} \Delta b^2 &= -\eta \frac{\partial E}{\partial b^2(t)} \\ &= \eta \delta^2(k) o_i^1(k) \end{aligned} \quad (6)$$

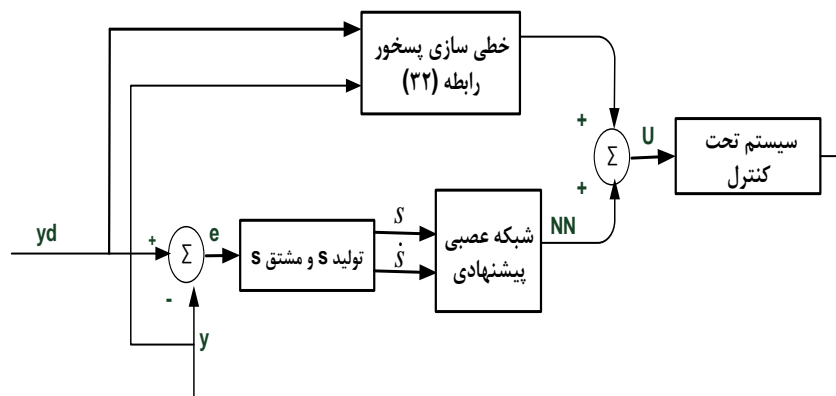
پارامترهای آموزش عمومی برای لایه فعالساز میانی به صورت (۷) است:

$$\begin{aligned} \Delta a^1 &= -\eta \frac{\partial E}{\partial a^1(t)} = \\ &= -\eta \frac{\partial E}{\partial e} \frac{\partial e}{\partial o^2} \frac{\partial o^2}{\partial net_T^2} \frac{\partial net_T^2}{\partial net_E^2} \frac{\partial net_E^2}{\partial o_E^1} \frac{\partial o_E^1}{\partial net_T^1} \frac{\partial net_T^1}{\partial net_E^1} \frac{\partial net_E^1}{\partial a^1} \\ \Delta a^1 &= \eta e F'^2(\cdot) \cdot 1 \cdot a^2 \cdot F''(\cdot) X_E = \eta \delta^1(k) X_E(k) \\ \delta^1(k) &= \delta^2(k) a^2 \cdot F'' \\ \Delta b^1 &= -\eta \frac{\partial E}{\partial b^1(t)} \\ &= \eta \delta^1(k) X_i(k) \\ \delta^1(k) &= \delta^2(k) b^2 \cdot F'' \end{aligned} \quad (7)$$

که در آن $\delta^2(k)$ خطای محلی لایه فعالساز دوم است.

۳- ساختار کنترلی

ساختار کنترلی به صورت شکل ۲، در نظر گرفته می‌شود. درحالی که عدم قطعیت درمدل سیستم وجود دارد هدف پایدارسازی و قرار دادن سیستم در وضعیت مطلوب و ردیابی ورودی مرجع است. در بلوک خطی سازی پسخور، خطی سازی پسخور برای ورودی- خروجی انجام می‌شود، که دربخش ۳-۳ توضیح داده شده است. برای پایدارسازی و برقرار کردن شرط لغزشی از شبکه عصبی پیشنهادی دربخش ۲، استفاده می‌شود، ورودی این شبکه عصبی \dot{s} ، s هستند که براساس لیپانوف آموزش می‌بیند. در بخش ۳ راجع به این ورودی‌ها و روند آموزش توضیحات کامل داده می‌شود.



شکل ۲. بلوک دیاگرام کنترلی

۳-۱ آموزش شبکه عصبی براساس تابع لیاپانوف

آموزش شبکه عصبی معرفی شده در بخش ۲ بر مبنای استفاده از مشتق تابع لیاپانوف در کنترل مُد لغزشی می باشد، بدین صورت که از مشتق تابع لیاپانوف به عنوان تابع هزینه جهت آموزش عصبی کنترل کننده استفاده می گردد و سعی بر کمینه کردن آن می شود.

کنترل مد لغزشی برای سیستم هایی که دارای عدم قطعیت می باشند، تعریف می شود. در این روش با تعریف یک سطح لغزش بر اساس حالت های سیستم سعی بر کنترل سیستم می باشد. یعنی اگر حالت های سیستم در این سطح قرار گیرند، سیستم پایدار می باشد. ابتدا بر اساس کنترل کننده کلاسیک مُد لغزشی باید معادلات سیستم را به صورت نرمال (۸) نوشت:

$$\dot{x}_p = f(X_p) + g(X_p)u(t) \quad (8)$$

که توابع $f(X_p), g(X_p)$ در مورد سیستم در نظر گرفته شده در این مقاله، که یک ربات جابه جا کننده دو بازویی است، کاملاً معلوم نیستند. در واقع مدل سیستم با استفاده از روابط فیزیکی حاکم بر آن معلوم است، اما به خاطر بارها با جرم های متفاوتی که این ربات باید جابه جا کند، پارامترهای مجهول متغیر با زمان در معادلات سیستم وجود دارد. یعنی عدم قطعیت پارامتری در سیستم وجود دارد. در بخش ۳-۲، در مورد دینامیک های این سیستم توضیح داده شده می شود. معادله سطح لغزش به صورت زیر تعریف می شود [۲۳]:

$$s(t) = \left(\frac{d}{dt} + \delta \right)^{n-1} e(t) \quad (9)$$

که سیگنال خطا به صورت زیر است:

$$e = x - x_d \quad (10)$$

که در آن x حالت های سیستم، x_d حالت های مطلوب سیستم، n مرتبه سیستم و δ یک اسکالر مثبت است. برای سیستم مرتبه دو سطح لغزش به صورت (۱۱) محاسبه می شود:

$$s(t) = \dot{e}(t) + \delta e(t) \quad (11)$$

پس مشتق $s(t)$ برای سیستم مرتبه n به صورت زیر می باشد:

$$\begin{aligned} \dot{s}(t) &= \left(\frac{d}{dt} + \delta \right)^n e(t) \\ &= \dot{e}^n(t) + k_1 e^{n-1}(t) + \dots + k_{n-2} \ddot{e}(t) + k_{n-1} \dot{e}(t) + k_n e(t) \end{aligned} \quad (12)$$

اگر از رابطه (۱۰)، n بار مشتق گرفته و رابطه (۸) در آن جایگذاری شود، داریم:

$$\begin{aligned} e^n &= x^n - x_d^n \\ &= f(X_p) + g(X_p)u(t) - x_d^n \end{aligned} \quad (13)$$

با جایگذاری رابطه (۱۳) در معادله (۱۲) خواهیم داشت:

$$\begin{aligned} \dot{s}(t) = & \\ & f(x_p) + g(x_p)u(t) - x_d^n(t) + k_1 e^{n-1}(t) \\ & + \dots + k_{n-2} \ddot{e}(t) + k_{n-1} \dot{e}(t) + k_n e(t) \end{aligned} \quad (14)$$

تابع هزینه آموزش شبکه عصبی، مشتق تابع لیاپانوف به صورت رابطه (۱۵) در نظر گرفته می شود:

$$\begin{aligned} V(t) &= \frac{1}{2} s^2(t) \\ \dot{V}(t) &= \dot{s}(t)s(t) < 0 \end{aligned} \quad (15)$$

در حالت چند ورودی - چند خروجی تابع هزینه به صورت (۱۶) در نظر گرفته می شود:

$$\dot{V}(t) = (s^T(t)s(t))' < 0 \quad (16)$$

که برای حالت دو ورودی - دو خروجی، داریم:

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (17)$$

حال آموزش وزن ها بر اساس روش پس انتشار خطا به صورت زیر انجام می شود:

در ابتدا برای وزن های لایه خروجی داریم:

$$\begin{aligned} \Delta a^2 &= -\eta \frac{\partial \dot{V}(k)}{\partial a^2(t)} \\ &= -\eta \frac{\partial s(k)\dot{s}(k)}{\partial \dot{s}(k)} \frac{\partial \dot{s}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o^2(k)} \frac{\partial o^2}{\partial net_T^2} \frac{\partial net_T^2}{\partial net_E^2} \frac{\partial net_E^2}{\partial a^2} \\ \frac{\partial s(k)\dot{s}(k)}{\partial \dot{s}(k)} &= \frac{\partial s(k)\dot{s}(k)}{\partial \dot{s}_1(k)} + \frac{\partial s(k)\dot{s}(k)}{\partial \dot{s}_2(k)} \\ \Delta a^2 &= -\eta (s_1(k)\hat{g}_1(k) + s_2(k)\hat{g}_2(k)) F'^2 \cdot 1 \cdot o_E^1(k) \\ &= -\eta \delta^2(k) o_E^1(k) \\ \delta^2 &= (s_1(k)\hat{g}_1(k) + s_2(k)\hat{g}_2(k)) F'^2 \end{aligned} \quad (18)$$

رابطه برای وزن های b^2 به صورت (۱۹) است:

$$\begin{aligned} \Delta b^2 &= -\eta \frac{\partial \dot{V}(k)}{\partial b^2(t)} \\ &= -\eta \delta^2(k) o_I^1(k) \end{aligned} \quad (19)$$

که در آن تخمینی از g_1 و g_2 تخمینی از g_2 است. برای وزن های لایه اول داریم:

$$\begin{aligned}
 \Delta a^1 &= -\eta \frac{\partial \dot{V}(k)}{\partial a^1(t)} \\
 &= -\eta \frac{\partial s(k) \dot{s}(k)}{\partial \dot{s}(k)} \frac{\partial \dot{s}(k)}{\partial u(k)} \frac{\partial u(k)}{\partial o^2(k)} \frac{\partial o^2}{\partial net_T^2} \\
 &\quad \frac{\partial net_T^2}{\partial net_E^2} \frac{\partial net_E^2}{\partial o_E^1} \frac{\partial o_E^1}{\partial net_T^1} \frac{\partial net_T^1}{\partial net_E^1} \frac{\partial net_E^1}{\partial a^1} \\
 &= -\eta \delta^2 a^2 F'' X_E \\
 &= -\eta \delta^1 X_E \\
 \delta^1 &= \delta^2 a^2 F''
 \end{aligned} \tag{20}$$

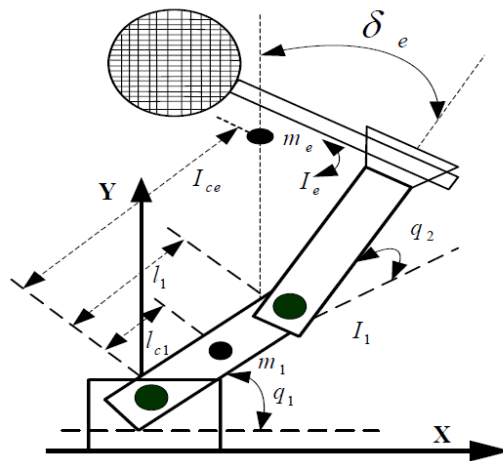
به طریق مشابه، روابط برای وزن های b^1 به صورت (۲۱) بدست می آیند:

$$\begin{aligned}
 \Delta b^1 &= -\eta \frac{\partial \dot{V}(k)}{\partial b^1(t)} \\
 &= -\eta \delta^1 X_I \\
 \delta^1 &= \delta^2 b^2 F''
 \end{aligned} \tag{21}$$

۲-۳ معادلات سیستم

یک بازوی رباتیک n درجه آزادی، که در شکل ۳، از مرجع [۲۴]، نشان داده شده است، به طور کلی دارای مدل دینامیک مرتبه دو به صورت رابطه (۲۲) است:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{22}$$



شکل ۳. ربات جابجاکننده دوبازویی [۲۴]

که در آن $H(q)$ ماتریس 2×2 لختی جابه جا کننده (که مثبت معین متقارن است)، $C(q, \dot{q})$ یک بردار دو بعدی شامل گشتاورهای مرکزگرا و کوریولیس (باماتریس 2×2 $C(q, \dot{q})$)، و $g(q)$ یک بردار دو بعدی گشتاورهای گرانشی است. معادلات دینامیکی یک جابجا کننده دو بازویی که در شکل ۳، از مرجع [۲۴] نشان داده شده است، به صورت زیر می باشد:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -h\dot{q}_2 & -h(\dot{q}_2 + \dot{q}_1) \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{23}$$

که در آن:

$$\begin{aligned} H_{11} &= \alpha_1 + 2\alpha_3 \cos q_2 + 2\alpha_4 \sin q_2 a \\ H_{12} = H_{21} &= \alpha_2 + \alpha_3 \cos q_2 + \alpha_4 \sin q_2 \\ H_{22} &= \alpha_2 \\ h &= \alpha_3 \sin q_2 - \alpha_4 \cos q_2 \end{aligned} \quad (24)$$

که در آن پارامترهای $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ به صورت زیر تعریف شده‌اند:

$$\begin{aligned} \alpha_1 &= I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \\ \alpha_2 &= I_e + m_e l_{ce}^2 \\ \alpha_3 &= m_e l_1 l_{ce} \cos \delta_e \\ \alpha_4 &= m_e l_1 l_{ce} \sin \delta_e \end{aligned} \quad (25)$$

۳-۳ خطی سازی پسخور

در این بخش به بررسی خطی سازی ورودی-خروجی سیستم‌های غیرخطی که با نمایش فضای حالت زیر توصیف می‌شوند پرداخته می‌شود.

$$\begin{aligned} \dot{x} &= f(X_p) + g(X_p)u(t) \\ y &= h(x) \end{aligned} \quad (26)$$

روش اساسی در خطی سازی ورودی-خروجی، به طور ساده این است که از خروجی y مکرراً آنقدر مشتق گرفته شود تا ورودی u ظاهر شود، سپس u طوری طراحی می‌شود که اثرات غیرخطی را حذف کند. اگر مدل دینامیکی سیستم دقیق باشد، خطی سازی پسخور کار سراسری است. با در نظر گرفتن τ به شکل (۲۷):

$$\tau = H(q)v + C(q, \dot{q})\dot{q} + g(q) \quad (27)$$

که در آن v یک وردی جدید است، داریم: $\ddot{q} = v$

با تعریف خطای ردیابی، $\tilde{q} = q - q_d$ و فرض:

$$v = \ddot{q}_d - 2\lambda\dot{\tilde{q}} - \lambda^2\tilde{q} \quad (\lambda > 0)$$

دینامیک حلقه بسته پایدار نمایی (۲۸) حاصل می‌شود:

$$\ddot{\tilde{q}} + 2\lambda\dot{\tilde{q}} + \lambda^2\tilde{q} = 0 \quad (28)$$

اما چون سیستم به خاطر جابه‌جایی بارهایی با جرم‌های متفاوت دارای عدم قطعیت پارامتری است برای بدست آوردن کنترل کننده مقاوم، به صورت زیر عمل می‌شود.

سیستم به صورت نرمال (۲۹) در نظر گرفته می‌شود:

$$\ddot{q} = f(q, \dot{q}) + g(q, \dot{q})\tau \quad (29)$$

طوری که دینامیک های g و f معلوم نبوده و با \hat{g} , \hat{f} تخمین زده می شوند. خطای تخمین بر روی g و f با توابع معلوم F و G

$$|f - \hat{f}| \leq F, |g - \hat{g}| \leq G \quad \text{محدود شده است.}$$

باتوجه به روابط (۲۳) و (۲۹) روابط زیر بدست می آیند:

$$\hat{f} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ \hat{H}_{21} & \hat{H}_{22} \end{bmatrix}^{-1} \begin{bmatrix} -\hat{h}\dot{q}_2 & -\hat{h}(\dot{q}_2 + \dot{q}_1) \\ \hat{h}\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (30)$$

$$\hat{g} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ \hat{H}_{21} & \hat{H}_{22} \end{bmatrix}^{-1}$$

برای اینکه سیستم $x_d(t)$ را تعقیب کند، یک سطح لغزشی به صورت (۳۱) تعریف می شود:

$$\dot{s} = \ddot{q} - \ddot{q}_d + \lambda \tilde{q} = f + gu - \ddot{q}_d + \lambda \tilde{q} \quad (31)$$

پس، بهترین تقریب قانون کنترل که بتواند $\dot{s} = 0$ را برآورده کند، عبارت است از:

$$\hat{\tau} = \hat{g}^{-1}(-\hat{f} + \ddot{q}_d - \lambda \tilde{q}) \quad (32)$$

برای اینکه شرط لغزشی علی رغم عدم قطعیت در مدل برقرار باشد، یک جمله دیگر به صورت (۳۳) به قانون کنترل اضافه می شود:

$$\tau = \hat{\tau} + U_{NN} \quad (33)$$

که U_{NN} خروجی شبکه عصبی است.

شرایط لغزشی بایستی، یک به یک در هر درجه آزادی با $s_i = \dot{\tilde{q}}_i + \lambda_i \tilde{q}_i$ تأیید شود، تابع هزینه آموزش شبکه عصبی مشتق تابع لیاپانوف به صورت (۳۴) در نظر گرفته می شود:

$$\dot{V}(t) = (s^T s)' < 0 \quad (34)$$

نحوه آموزش وزن ها بر اساس لیاپانوف در قسمت (۳-۱) تشریح شد.

در واقع وزن های شبکه عصبی با تابع هزینه در نظر گرفته شده چنان تغییر می کنند که شرط لغزشی برقرار شود. در مقایسه با کنترل کلاسیک شبکه عصبی ضمن برقراری شرط لغزشی یک لایه مرزی متغیر با زمان حول سطح $s = 0$ ایجاد می کند، در لحظات اولیه که خطای ردیابی زیاد است، پهنای این لایه مرزی بزرگ است و با ردیابی کامل به صفر می رسد.

۴- شبیه سازی

در شبیه سازی ها از پارامترها با مقادیر بیان شده در جدول (۱) استفاده شده است. در طراحی که کنترل کننده فرض شده

است حداکثر عدم قطعیت خواص جرم ۲۵٪ می باشد. ورودی مرجع به صورت زیر می باشد:

$$\begin{aligned} q_{d1}(t) &= 30^\circ \cos(t) \\ q_{d2}(t) &= 45^\circ \cos(t) \end{aligned} \quad (35)$$

نمودار ردیابی ورودی مرجع و خطای ردیابی به صورت شکل ۴ و شکل ۵ می باشد. همانطور که دیده می شود خروجی سیستم به خوبی توانسته است ورودی مرجع را علی رغم عدم قطعیت در مدل سیستم دنبال کند. در حالت کنترل مد لغزشی کلاسیک، نمودارها به صورت شکل ۶ می باشند. همانطور که دیده می شود خطای ردیابی با کنترل کننده مطرح شده در این مقاله خیلی کمتر است و ضمن اینکه شبکه عصبی به کار رفته جوابگوی دینامیک های ناشناخته سیستم نیز می باشد. در روش کلاسیک اگر حد بالا و پایین تابع اشباع کم شود خطای ردیابی کم، اما نوسانات سیگنال کنترل زیاد می شود و این خود باعث تحریک دینامیک های مدل نشده فرکانس بالا می شود. خروجی های شبکه عصبی در شکل ۷ نشان داده شده است.

جدول ۱. مقادیر در نظر گرفته شده برای پارامترهای سیستم در شبیه سازی

مقدار	پارامتر	مقدار	پارامتر
0.12	I_1	1	m_1
0.5	l_{c1}	1	l_1
0.25	I_e	2	m_e
0.6	l_{ce}	30°	δ_e

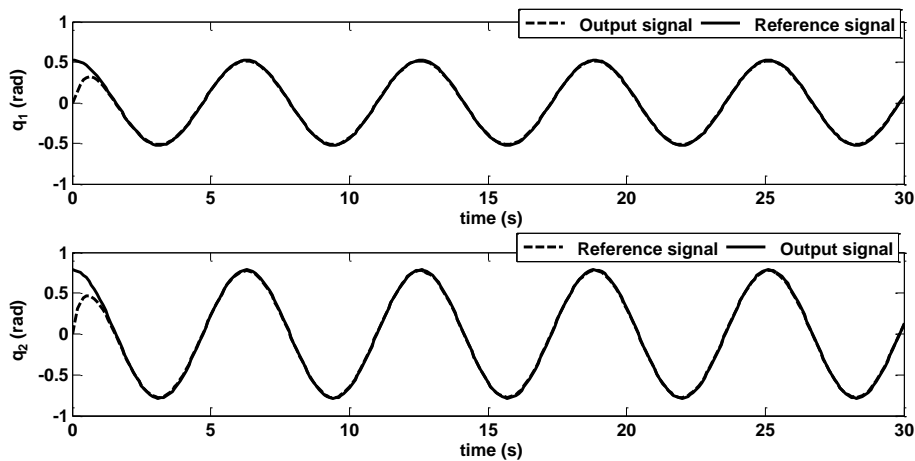
صفر شدن خروجی شبکه عصبی به این معنی است که شرط لغزشی برقرار شده است و حالت های سیستم به وضعیت مطلوب رسیده اند همانطور که دیده می شود شبکه عصبی پیشنهادی خیلی سریع توانسته است این کار را انجام دهد. نمودار خروجی ها در حالت کنترل وضعیت، با کنترل کننده مطرح شده در این مقاله به صورت شکل ۸ می باشد. خروجی های سیستم با سرعت قابل توجه و بدون نوسان در وضعیت مطلوب قرار گرفته اند. نمودار کنترل وضعیت با روش کنترل کلاسیک مد لغزشی به صورت شکل ۹ می باشد. از مقایسه شکل های ۸ و ۹ دیده می شود روش که کنترلی طراحی شده در این مقاله کارایی بهتری نسبت به روش های کلاسیک دارد.

برای اینکه این تفاوت بهتر دیده شود، شبیه سازی ها در حضور اغتشاش تکرار می شوند، لازم به توضیح است که اغتشاش به صورت نویزی با واریانس ۰/۱ و میانگین صفر از لحظه ۱۵ به سیستم وارد می شود. اغتشاش با ورودی کنترل جمع می شود. نمودار کنترل وضعیت با روش مد لغزشی کلاسیک در حضور اغتشاش به صورت شکل ۱۰ می باشد. نمودار کنترل وضعیت با کنترل کننده مطرح شده در این مقاله در حضور اغتشاش به صورت شکل ۱۱ است. از مقایسه شکل های ۱۰ و ۱۱ دیده می شود که کنترل کننده هوشمند طراحی شده در این مقاله به خوبی توانسته است اغتشاش را رفع کند. نمودار ردیابی ورودی مرجع، با روش کنترلی طراحی شده در این مقاله در حضور اغتشاش به صورت شکل ۱۲ می باشد.

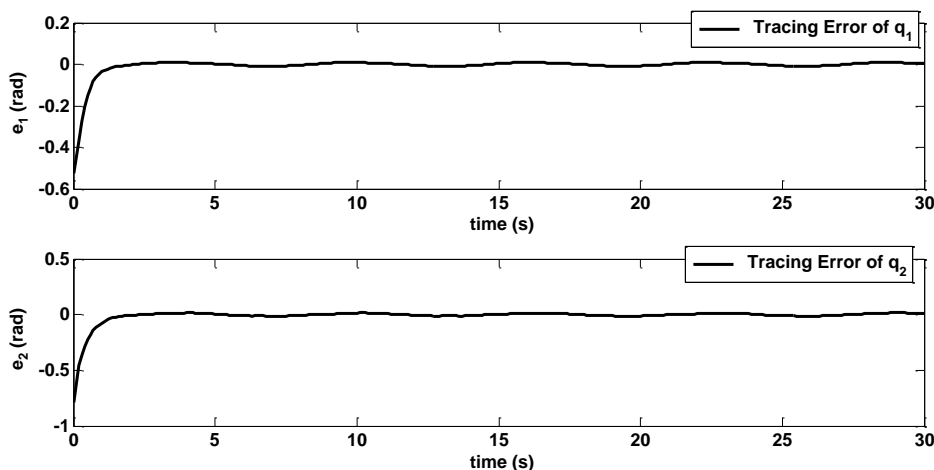
همانطور که دیده می شود، در حضور اغتشاش سیستم توانسته است ورودی مرجع را به خوبی دنبال کند، در واقع امتیاز خوب روش کنترلی به کار رفته در این مقاله این است که علاوه بر این که بدون نوسان و خیلی سریعتر از روش های معمول، خروجی را در وضعیت مطلوب قرار می دهد، توانایی مقاومت در برابر اغتشاش و دینامیک های ناشناخته را نیز داراست.

برای اینکه مؤثر بودن روش کنترلی بهتر دیده شود، برای ورودی مرجع سینوسی، عملکرد روش کنترلی پیشنهادی با روش کنترلی مد لغزشی فازی نوع دوم ارائه شده در مرجع [۲۵]، مقایسه شده است. مقادیر کمترین مربعات خطا برای خطاهای ردیابی در

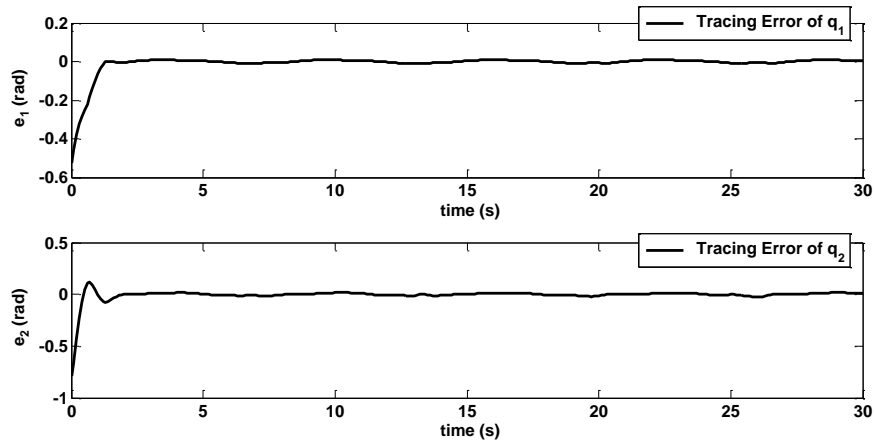
شرایط مختلف در جدول ۲ آورده شده است. میزان تأخیر، پنج زمان نمونه برداری است. در شبیه سازی ها زمان نمونه برداری ۰/۰۱ ثانیه در نظر گرفته شده است. همانطور که دیده می شود، اغتشاش و تأخیر کمترین تأثیر را در روش کنترلی پیشنهادی داشته اند. با توجه به اینکه در روش کنترلی پیشنهادی، قیده های مهم عملی در نظر گرفته شده است و نتایج شبیه سازی ها مؤثر بودن این روش را نشان می دهند، کنترل کننده پیشنهادی می تواند در موارد مختلف عملی به کار گرفته شود. همچنین حجم محاسبات در روش پیشنهادی در مقایسه با سایر روش هایی که از الگوریتم هایی مثل ژنتیک [۲۶]، مورچگان [۲۷ و ۲۸]، تجمعی پرندگان [۲۹ و ۳۰] برای بهینه سازی کنترل کننده استفاده می کنند، کمتر بوده و برای کاربردهای عملی مناسبتر است. از طرف دیگر در مراجعی مثل [۲۵ و ۳۱] که از سیستم های فازی برای بهبود عملکرد کنترل کننده استفاده شده است، ساختار سیستم های فازی به صورت سعی و خطا تعیین می شود و فرم قوانین در طول زمان ثابت است. کنترل کننده پیشنهادی برای پیاده سازی های غیرمتمرکز مناسب است. از مهمترین معایب سیستم های کنترلی متمرکز این است که در صورت بروز مشکل در سیستم کنترل مرکزی، کل سیستم از کار می افتد.



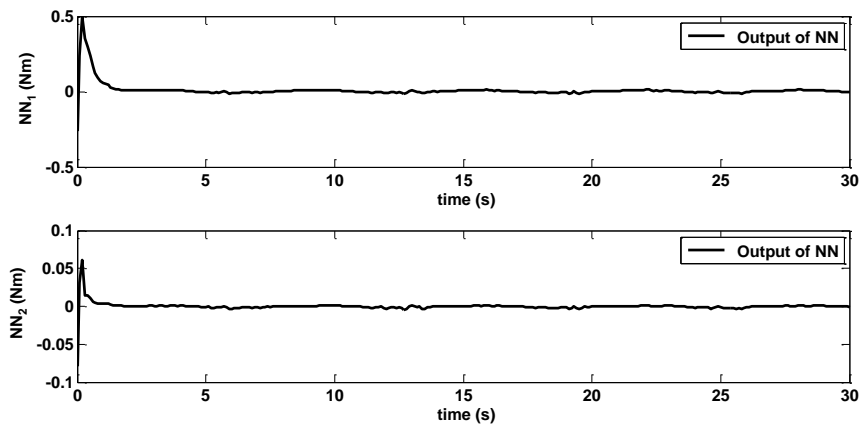
شکل ۴. نمودارهای خروجی با روش کنترلی پیشنهادی.



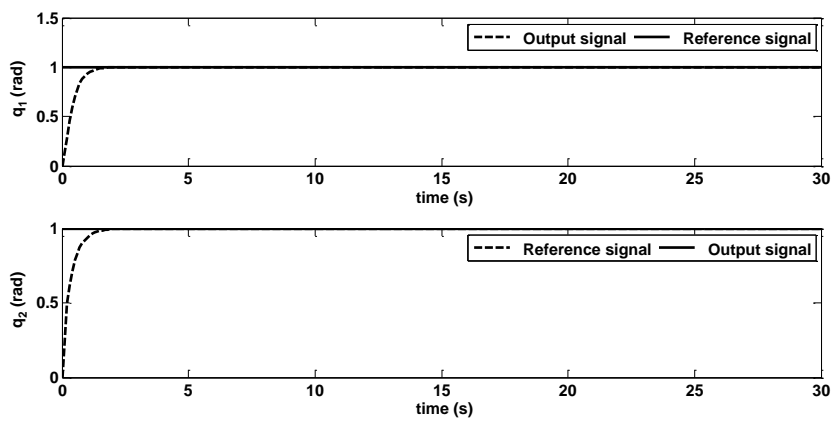
شکل ۵. نمودارهای خطاهای ردیابی با روش کنترلی پیشنهادی.



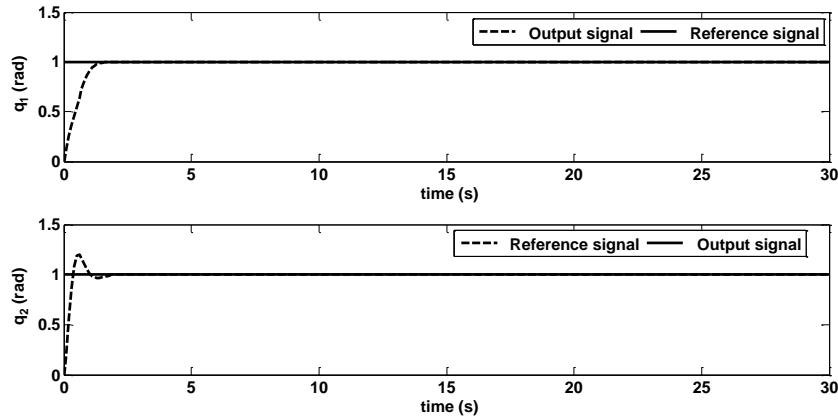
شکل ۶. ردیابی با روش مد لغزشی کلاسیک.



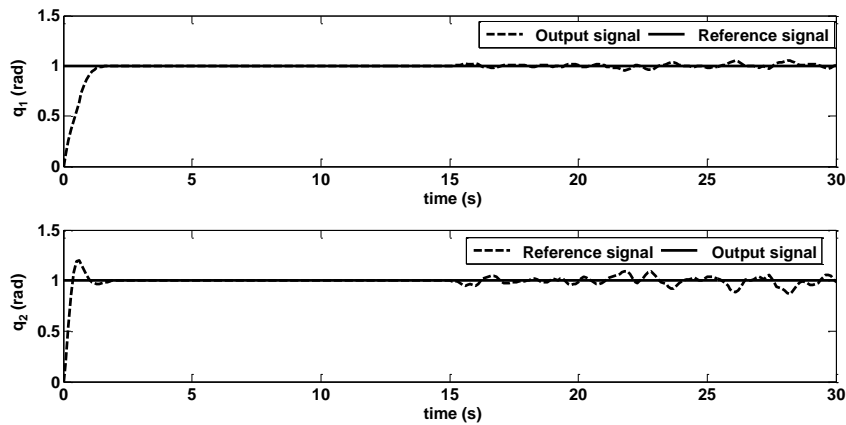
شکل ۷. نمودار خروجی های شبکه عصبی



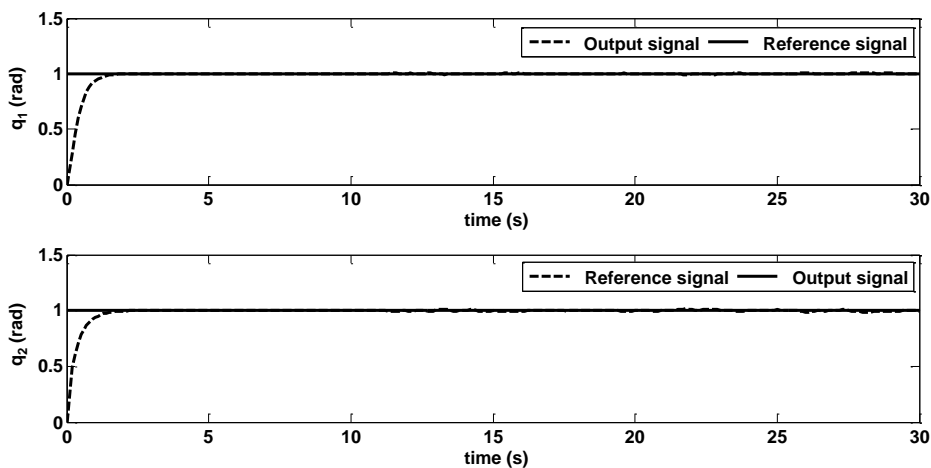
شکل ۸. کنترل وضعیت با کنترل کننده پیشنهادی.



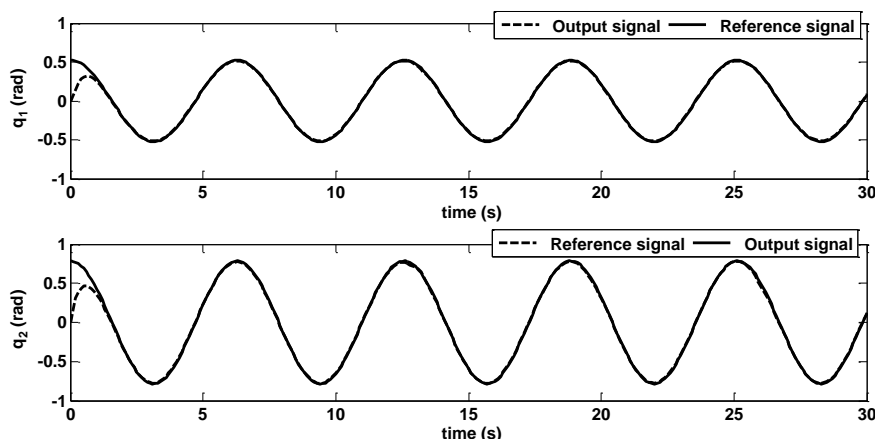
شکل ۹. کنترل وضعیت با روش کلاسیک مد لغزشی



شکل ۱۰. نمودار خروجی های سیستم در حضور اغتشاش، باروش کنترلی کلاسیک



شکل ۱۱. نمودار خروجی های سیستم در حضور اغتشاش، باروش کنترلی هوشمند



شکل ۱۲. نمودار خروجی، در حضور اغتشاش..

جدول ۲. مقایسه عددی کمترین مربعات خطا، در شرایط مختلف

در حضور تأخیر		در حضور اغتشاش		بدون حضور اغتشاش		
q_2	q_1	q_2	q_1	q_2	q_1	
۰/۴۳۰۰	۰/۲۷۳۷	۰/۱۱۴۶	۰/۰۷۵۳	۰/۱۱۵۶	۰/۰۷۵۶	روش کنترلی فازی لغزشی
۰/۳۴۰۱	۰/۱۳۲۰	۰/۰۷۱۳	۰/۰۵۲۷	۰/۰۷۱۵	۰/۰۵۲۸	روش پیشنهادی

۵- نتیجه گیری

در این مقاله، یک کنترل کننده جدید با استفاده از شبکه های عصبی پویا و خطی سازی پسخور برای یک بازوی ربات صنعتی ارائه شد. کنترل کننده پیشنهادی برای ردیابی مسیر و کنترل وضعیت بازوی ربات، در حضور عدم قطعیت پارامتری و اغتشاش خارجی طراحی شد. شبیه سازی های مختلفی برای نشان دادن کارایی روش پیشنهادی ارائه شد. همانطور که در نتایج شبیه سازی ها دیده شد، علی رغم عدم قطعیت در مدل سیستم، با کنترل کننده طراحی شده، بازوهای ربات بدون نوسان در وضعیت مطلوب قرار می گیرند و ورودی مرجع را به خوبی حتی در حضور اغتشاش دنبال می کنند. در واقع شبکه عصبی پیشنهادی به خوبی توانسته است با برقرار کردن شرط لغزشی حالت های سیستم را در وضعیت مطلوب قرار دهد. روش بیان شده کاملاً عملی است و می تواند جایگزین بسیار مناسبی برای کنترل کلاسیک در صنعت باشد.

مراجع

- [1] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, pp. 948-954, 2017.
- [2] J. Baek, S. Cho, and S. Han, "Practical Time-Delay Control With Adaptive Gains for Trajectory Tracking of Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 5682-5692, 2018.
- [3] M. Li, Y. Li, S. S. Ge, and T. H. Lee, "Adaptive control of robotic manipulators with unified

- motion constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, pp. 184-194, 2017.
- [4] A. Brahmi, M. Saad, G. Gauthier, W.-H. Zhu, and J. Ghommam, "Tracking control of mobile manipulator robot based on adaptive backstepping approach," *International Journal of Digital Signals and Smart Systems*, vol. 1, pp. 224-238, 2017.
- [5] J. Lee, P. H. Chang, and M. Jin, "Adaptive integral sliding mode control with time-delay estimation for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 6796-6804, 2017.
- [6] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 3628-3637, 2016.
- [7] Y. Wang, L. Gu, Y. Xu, and X. Cao, "Practical tracking control of robot manipulators with continuous fractional-order nonsingular terminal sliding mode," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 6194-6204, 2016.
- [8] S. I. Han and J. Lee, "Finite-time sliding surface constrained control for a robot manipulator with an unknown deadzone and disturbance," *ISA transactions*, vol. 65, pp. 307-318, 2016.
- [9] S. M. H. Zadeh, S. Khorashadizadeh, M. M. Fateh, and M. Hadadzarif, "Optimal sliding mode control of a robot manipulator under uncertainty using PSO," *Nonlinear Dynamics*, vol. 84, pp. 2227-2239, 2016.
- [10] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 8112-8123, 2018.
- [11] J. Wilson, M. Charest, and R. Dubay, "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 23-30, 2016.
- [12] M. Rahmani, A. Ghanbari, and M. M. Etefagh, "Hybrid neural network fraction integral terminal sliding mode control of an Inchworm robot manipulator," *Mechanical Systems and Signal Processing*, vol. 80, pp. 117-136, 2016.
- [13] M. Rahmani, A. Ghanbari, and M. M. Etefagh, "Robust adaptive control of a bio-inspired robot manipulator using bat algorithm," *Expert Systems with Applications*, vol. 56, pp. 164-176, 2016.
- [14] M. R. Soltanpour, M. H. Khooban, and M. Soltani, "Robust fuzzy sliding mode control for tracking the robot manipulator in joint space and in presence of uncertainties," *Robotica*, vol. 32, pp. 433-446, 2014.
- [15] M. Veysi, M. R. Soltanpour, and M. H. Khooban, "A novel self-adaptive modified bat fuzzy sliding mode control of robot manipulator in presence of uncertainties in task space," *Robotica*, vol. 33, pp. 2045-2064, 2015.
- [16] R.-J. Wai and R. Muthusamy, "Design of fuzzy-neural-network-inherited backstepping control for robot manipulator including actuator dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 22, pp. 709-722, 2014.
- [17] M. R. Soltanpour, P. Otadolajam, and M. Soltani, "Optimal adaptive fuzzy integral sliding model control for electrically driven SCARA robot manipulator," *The Modares Journal of Electrical Engineering*, vol. 12, pp. 31-48, 2015.
- [18] S. Tayebihaghighi, F. Piltan, and J.-M. Kim, "Control of an Uncertain Robot Manipulator Using an Observation-based Modified Fuzzy Sliding Mode Controller," *International Journal of Intelligent Systems and Applications*, vol. 10, p. 41, 2018.

- [19]H. Chaudhary, V. Panwar, R. Prasad, and N. Sukavanam, "Adaptive neuro fuzzy based hybrid force/position control for an industrial robot manipulator," *Journal of Intelligent Manufacturing*, vol. 27, pp. 1299-1308, 2016.
- [20]W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE transactions on cybernetics*, vol. 46, pp. 620-629, 2016.
- [21]T. Van Tran, Y. Wang, H. Ao, and T. K. Truong, "Sliding mode control based on chemical reaction optimization and radial basis functional link net for de-icing robot manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, p. 051009, 2015.
- [22]K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, pp. 119-130, 1988.
- [23]S. Zhang, Y. Dong, Y. Ouyang, Z. Yin, and K. Peng, "Adaptive neural control for robotic manipulators with output constraints and uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.
- [24]K. J. Åström and B. Wittenmark, *Adaptive control*: Courier Corporation, 2013.
- [25]J.-H. Hwang, Y.-C. Kang, J.-W. Park, and D. W. Kim, "Advanced Interval Type-2 Fuzzy Sliding Mode Control for Robot Manipulator," *Computational intelligence and neuroscience*, vol. 2017, 2017.
- [26]S. Refoufi and K. Benmahammed, "Control of a manipulator robot by neuro-fuzzy subsets form approach control optimized by the genetic algorithms," *ISA transactions*, vol. 77, pp. 133-145, 2018.
- [27]E. Zakeri, S. A. Moezi, and M. Eghtesad, "Tracking Control of Ball on Sphere System Using Tuned Fuzzy Sliding Mode Controller Based on Artificial Bee Colony Algorithm," *International Journal of Fuzzy Systems*, pp. 1-14, 2017.
- [28]M. Y. Hassan and A. F. Sugban, "Ant Colony Optimization Based Type-2 Fuzzy Force-Position Control for Backhoe Excavator Robot," *Al-Nahrain Journal for Engineering Sciences*, vol. 21, pp. 1-11, 2018.
- [29]A. Jalali, F. Piltan, A. Gavahian, and M. Jalali, "Model-free adaptive fuzzy sliding mode controller optimized by particle swarm for robot manipulator," *International Journal of Information Engineering and Electronic Business*, vol. 5, p. 68, 2013.
- [30]M. Vijay and D. Jena, "PSO based neuro fuzzy sliding mode control for a robot manipulator," *Journal of Electrical Systems and Information Technology*, vol. 4, pp. 243-256, 2017.
- [31]V. T. Yen, W. Y. Nan, and P. Van Cuong, "Recurrent fuzzy wavelet neural networks based on robust adaptive sliding mode control for industrial robot manipulators," *Neural Computing and Applications*, pp. 1-14, 2018.