

کنترل وضعیت موتور DC تحریک جداگانه با کمک کنترلر شبکه عصبی NARMA-L2

جهان جمشیدی^{۱*}، رضا حق مرام^۲

۱- دانشجوی کارشناسی ارشد، ۲- استادیار، دانشگاه جامع امام حسین(ع)، دانشکده و پژوهشکده فناوری اطلاعات و ارتباطات، گروه الکترونیک

E-mail: Jamshidi.Jahan@gmail.com

(دریافت: ۸۸/۰۳/۳۰، پذیرش: ۸۹/۱۱/۰۶)

چکیده

به دلیل کاربرد وسیع موتورهای dc، کنترل وضعیت یا سرعت آنها اهمیت فراوانی پیدا می‌کند. در این مقاله هدف، ارائه روشی جدید برای کنترل هوشمند وضعیت موتور dc تحریک جداگانه با استفاده از شبکه‌های عصبی مصنوعی است. برای این منظور از کنترلر شبکه عصبی NARMA-L2 استفاده شده است. در کنترل وضعیت با استفاده از شبکه عصبی، وضعیت مبنای شبکه عصبی همان وضعیت ورودی (مطلوب) در نظر گرفته شده است و پس از انطباق وضعیت خروجی موتور بر وضعیت مبنای، ولتاژ آرمیچر صفر شده و موتور از کار می‌ایستد. در این روش، شبکه عصبی مدل موتور dc مورد استفاده برای کنترل وضعیت را می‌آموزد. سپس ورودی را برای تطبیق وضعیت خروجی مدل بر وضعیت مطلوب بهینه می‌سازد. ویژگی روش پیشنهادی، اصلاح ضرایب شبکه عصبی در حین کارکردن موتور و مقاوم بودن پاسخها نسبت به تغییرات لختی دورانی (I) و اصطکاک (B) است. برخلاف کنترلر PID که در اثر تغییر یکی از پارامترهای مدل، نظیر J و B کارایی بهینه خود را از دست می‌دهد و نیاز به تنظیم مجدد پارامترهای کنترلر دارد، روش پیشنهادی نیاز به طراحی مجدد ندارد و در صورت تغییر، این پارامترها به طور خودکار اصلاح می‌شود. روش پیشنهادی با استفاده از اطلاعات یک موتور dc تحریک جداگانه توسط نرم‌افزار MATLAB و در محیط SIMULINK شبیه‌سازی شده است. نتایج نشان می‌دهند که به واسطه‌ی استفاده از روش پیشنهادی، نوسان در پاسخها به شدت کاهش یافته، همچنین زمان رسیدن به وضعیت مطلوب نیز کم شده است.

کلیدواژه‌ها: موتور dc تحریک جداگانه؛ کنترل وضعیت؛ کنترلر شبکه عصبی NARMA-L2

Position Control of Separately Excited DC Motor Using NARMA-L2 Neural Network Controller

J. Jamshidi^{1*}, R. Haghmaram²

Department of Electronic, Imam Hossein University

E-mail: Jamshidi.Jahan@gmail.com

Abstract

Due to the widespread use of dc motors, their position or speed control becomes very important. The aim in this paper is providing a new method for intelligent position control of separately excited dc motor using neural networks. For this purpose the NARMA-L2 neural network controller is used. In the position control using neural network, based position for neural network intended as input position (ideal), and after adapting of output position of motor to base position, Armature voltage become zero and motor stops working. In this way, neural network learns the used model of dc motors to control the position, then the input optimized for matching of model output position to ideal position. Advantage of the proposed neural network method is, correction coefficients during the working engine and Replies robustness to changes inertia moment (J) and friction (B). Unlike the PID controllers that will lose their optimization Performance with changing one of the model parameters, such as J & B, and need to reset the parameters of the controller, the proposed method needs not to redesign, and these parameters repair automatically. Proposed method is simulated using information from a separately excited dc motor in SIMULINK environment of MATLAB software. Results show that by using of proposed method, fluctuations in response and need times to reach the ideal situation decreases.

Keywords: Separately Excited DC Motor; Position Control; NARMA-L2 Neural Network Controller

۱. مقدمه

موتورهای dc و ac به طور وسیعی در سیستم های کنترلی مورد استفاده قرار می گیرند اما هر کدام دارای مشخصات خاصی هستند. موتورهای dc از اولین موتورهای الکتریکی در صنعت بوده اند که بواسطه سهولت کنترل آنها در توان های چند وات تا چند صد کیلووات و با بازه وسیع ولتاژ و در سرعت های نامی مختلف ساخته شده اند. مزیت اصلی موتورهای dc کنترل پذیری آسان وضعیت یا سرعت و بازه وسیع تنظیم پذیری آنها است [۱]. بنابراین موتورهای dc کاربردهای وسیعی داشته و در لوازم صنعتی، تجاری، نظامی، خانگی و درمانی از جمله بازوی انواع ربات ها، دریل های الکتریکی، اره ها، لوازم صوتی و تصویری، آسانسورها، بالابر خودروها، انواع پمپ ها و فن ها و ماشین آلات کشاورزی به کار می روند. از دیگر موارد استفاده موتورهای dc می توان به کاربرد آنها در راه اندازی و کنترل جهت و سرعت انواع موشک ها، بخش های مختلف رادارهای ردیاب و فاصله یاب، بخش های مختلف بالگرد و هواپیما، اندام های مصنوعی کنترل پذیر مانند دست و پای مصنوعی، پمپ های تزریق مواد شیمیایی، سوختی و دارویی اشاره کرد.

هدف از طراحی کنترلر، افزایش پایداری و کاهش زمان رسیدن به حالت مطلوب در برابر اعمال اغتشاش در یک پروسه است. به عنوان مثال در بازوی ربات ها بازه وسیعی از حرکت لازم است تا در شرایط بار متغیر، یک مسیر از پیش تعیین شده را دنبال کند؛ از اینرو وجود یک کنترلر کارآمد برای کنترل موتورهای dc و در نتیجه کنترل بازوی ربات، اهمیت زیادی می یابد. عمده ترین کنترلر موتورهای dc با کنترل ولتاژ آرمیچر و یا کنترل میدان صورت می پذیرد. برای این منظور یک ورودی مرجع به آن داده می شود و خروجی باید از آن پیروی کند. میزان مطابقت خروجی با ورودی به عملکرد کنترلر بستگی دارد [۱ و ۴-۷ و ۱۰].

روش های ارائه شده برای کنترل وضعیت موتورهای dc به طور کلی به سه دسته تقسیم می شوند:

(۱) روش های کلاسیک مثل استفاده از کنترلرهای PI و PID [۲ و ۱۱]

(۲) روش های مدرن مثل روش های تطبیقی و بهینه [۳ و ۴]

(۳) روش های هوشمند مثل کاربرد تئوری فازی و شبکه های عصبی [۱۲ و ۱۳]

موتورهای dc معمولاً با سیستم های خطی مدل می شوند. با در

نظر گرفتن واکنش آرمیچر و سیم پیچ های جبران ساز، موتورهای dc معمولاً سیستم های غیرخطی محسوب می شوند. اکثر کنترلرهای خطی موجود به دلیل بازه وسیع شرایط کاری منجر به پیروی و تنظیم سیستم خوبی نمی شوند. در سال های اخیر پیشرفت های قابل توجهی در طراحی کنترلر غیر خطی در سروسیستم های با عملکرد خوب صورت گرفته است. روش های کنترلی برای سیستم های غیرخطی با فرض این که پارامترهای سیستم شناخته شده و معلوم باشند توسعه یافته اند. این روش ها هنگامی به کار می روند که عوامل غیرخطی سیستم بتوانند بر حسب توابع خطی معین مقیاس بندی و بر حسب پارامترهای نامعین بیان شوند. محیط شبکه های عصبی شامل مسیرهای امیدوارکننده ای برای حل مشکلات مربوط به سیستم های غیرخطی نامعین است؛ بنابراین شبکه های عصبی ابزار قدرتمندی برای آموزش و یادگیری سیستم های ایستا و سیستم های دینامیک به شدت غیرخطی هستند. قابلیت موازی سازی، تطابق بسیار سریع و توانایی های ذاتی شبکه های عصبی باعث شده است که تحقیقات وسیعی در زمینه بکارگیری آنها در مهندسی سیستم، بخصوص در زمینه کنترل و تشخیص سیستم صورت پذیرد.

در سال های اخیر تلاش های وسیعی برای استفاده از شبکه های عصبی مصنوعی، در علم کنترل صورت گرفته است. ویژگی مهم و حیاتی شبکه های عصبی مصنوعی توانایی آنها در خوب مدل کردن سیستم های غیرخطی، گستردگی زیاد، ساختار موازی شان، پیاده سازی آسان آنها با سخت افزار و نرم افزار و توانایی یادگیری و تطبیق خودشان با رفتار یک سیستم واقعی است. شبکه های عصبی می توانند هم با داده های بدست آمده از سیستم و هم هنگامی که به یک سیستم متصل می شوند، آموزش ببینند. همچنین به دلیل ساختار موازی شان به زمان پردازش کمتری نیاز دارند. شبکه های عصبی انعطاف پذیری بیشتری از لحاظ فیلتر کردن ورودی های نویزی فراهم می کنند. همچنین دارای حساسیت کمتر و قدرت رویارویی بیشتری در هنگام تغییر در پارامترهای سیستم به نسبت سایر کنترلر ها هستند [۵].

در ادامه و در بخش دوم، مدل سازی موتور dc عنوان شده است. بخش سوم به توضیح در مورد شبکه عصبی خودسازمانده و بلوک های کنترلی مورد استفاده اختصاص داده شده است. در بخش چهارم، روش های پیشنهادی توضیح داده شده اند و نتایج

بدست آمده در بخش آخر ارائه شده‌اند.

پارامترهای موتور dc شبیه‌سازی شده در جدول (۱) نشان داده شده است.

۲. مدل‌سازی موتور DC

موتورهای جریان مستقیم انواع گوناگونی دارند و برای هر یک روش‌های متعددی برای کنترل وضعیت ارائه شده است. در این مقاله موتور dc تحریک جداگانه^۱ برای کنترل وضعیت انتخاب شده و این اقدام با کنترل ولتاژ تغذیه صورت گرفته است. برای بدست آوردن مدار معادل یک موتور dc باید معادلات دیفرانسیل حاکم بر موتور dc را حل کرد. معادلات حاکم بر این موتور عبارتند از:

$$v_t = L_a \frac{di_a}{dt} + R_a i_a + E_a \quad (1)$$

$$E_a = K \omega_a \quad (2)$$

$$J \frac{d^2 \theta}{dt^2} + B \frac{d\theta}{dt} - T_L = K i_a \quad (3)$$

$$\omega_a = \frac{d\theta}{dt} \quad (4)$$

که در آن‌ها R_a و L_a و i_a و V_t به ترتیب مقاومت، اندوکتانس، جریان و ولتاژ آرمیچر، E_a نیروی ضد محرکه موتور، ω_a سرعت زاویه‌ای، θ وضعیت، T_L گشتاور بار، J و B نیز لختی دورانی و ضریب اصطکاک معادل در محور موتور هستند. با توجه به روابط فوق مدل موتور dc مورد استفاده برای کنترل وضعیت، در شکل (۱) نمایش داده شده است که در آن T_m ثابت زمانی مکانیکی موتور ($T_m = J/B$ (s)) و T_a ثابت زمانی جریان آرمیچر موتور ($T_a = L_a/R_a$ (s)) است [۶ و ۱۰]. مشخصات و

۳. سیستم‌های کنترلی و شبکه‌های عصبی

شبکه‌های عصبی در تعریف و کنترل سیستم‌های دینامیک عملکرد موفقی داشته‌اند. توانایی‌های تخمینی و جامع شبکه‌های پرسپترون چند لایه آن‌ها را گزینه مناسبی برای مدل‌سازی سیستم‌های غیرخطی و پیاده‌سازی سیستم‌های کنترلی کرده است. سه معماری پرتعداد شبکه‌های عصبی که برای پیشگویی و کنترل به کار می‌رود عبارتند از:

۱. کنترلر پیشگو

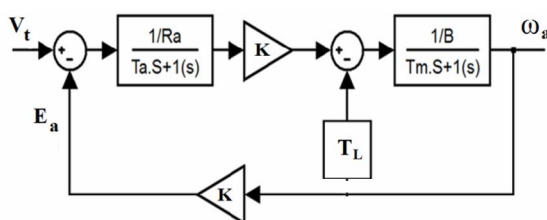
۲. کنترلر NARMA-L2

۳. کنترلر مدل مرجع

در این مدل‌ها ابتدا شبکه عصبی مدل سیستم را شناسایی می‌کند و سپس با استفاده از الگوریتم‌هایی، ورودی سیستم را برای انطباق خروجی سیستم بر مرجع، بهینه می‌سازد. عموماً دو گام برای استفاده از شبکه‌های عصبی وجود دارد:

۱. تعریف سیستم

۲. طراحی کنترلر

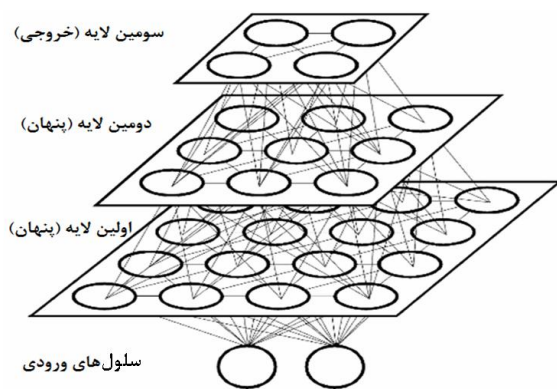


شکل ۱. مدل موتور dc تحریک جداگانه [۶]

جدول ۱. مشخصات و پارامترهای موتور dc شبیه‌سازی شده

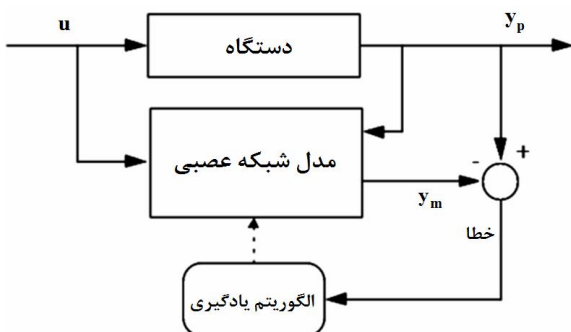
$V_t = 110 \text{ V}$	$P_{out} = 2/5 \text{ hp}$	$n_{sync} = 1800 \text{ rpm}$	$i_a = 20 \text{ A}$	$L_a = 46 \text{ Mh}$
$K = 0.55 \text{ V.s/rad}$	$R_a = 1 \Omega$	$J = 0.093 \text{ kg.m}^2$	$B = 0.008 \text{ N.m.s/rad}$	

1. Separately Excited DC Motor
2. Nonlinear Autoregressive Moving Average



شکل ۲. نمایش یک شبکه عصبی پرسپترون سه لایه [۹]

دینامیک پیشرو از یک دستگاه است. خطای پیشگویی بین خروجی دستگاه و خروجی شبکه عصبی برای آموزش شبکه عصبی مورد استفاده قرار می‌گیرد که این روند در شکل (۳) نشان داده شده است. شبکه عصبی از ورودی و خروجی مرحله قبل دستگاه برای پیشگویی مقادیر آینده خروجی دستگاه استفاده می‌کند. ساختار شبکه عصبی مورد استفاده در شکل (۴) نشان داده شده است. این شبکه عصبی می‌تواند به صورت آفلاین^۴ و دسته‌ای آموزش داده شود. داده‌های آموزشی با توجه به کارکرد دستگاه گردآوری می‌شوند. هر یک از الگوریتم‌های یادگیری پس‌انتشار در فرایند یادگیری قابل استفاده هستند. در این مقاله از کنترلر NARMA-L2 برای کنترل موتور dc استفاده شده است.



شکل ۳. روند آموزش شبکه عصبی [۱۴]

در مرحله تعریف سیستم یک مدل شبکه عصبی برای سیستمی که می‌خواهیم کنترل کنیم ایجاد می‌کنیم. در مرحله طراحی کنترلر از شبکه عصبی مدل سیستم، برای طراحی یا آموزش کنترلر استفاده می‌کنیم. در هر یک از معماری های کنترلی بیان شده، تعریف سیستم دارای روال ثابتی است؛ اما مرحله طراحی کنترلر برای هر یک از این مدل‌ها متفاوت است. [۷ و ۱۴].

۳-۱. شبکه‌های عصبی پرسپترون^۱

یکی از انواع شبکه‌های عصبی، پرسپترون است که به صورت‌های پرسپترون تک لایه^۲ و پرسپترون چند لایه^۳ موجود است. شبکه‌های عصبی پرسپترون جزء شبکه‌های عصبی پیشخور طبقه‌بندی می‌شوند. پرسپترون تک لایه تنها می‌تواند مسایل مجزای خطی را دسته‌بندی کند و برای مسایل پیچیده‌تر لازم است که از تعداد بیشتری لایه استفاده شود. شبکه‌های پیشخور چند لایه از یک یا تعداد بیشتری لایه‌های میانی تشکیل شده‌اند. شبکه‌های پرسپترون چند لایه از نوع شبکه‌های پیشخور چند لایه‌ای هستند که در لایه اول (لایه ورودی) اطلاعات ورودی سیستم به شبکه تغذیه می‌شود و در لایه خروجی، خروجی‌های شبکه محاسبه می‌شود. لایه‌های بین لایه ورودی و لایه خروجی لایه های مخفی نامیده می‌شوند و پردازش داده‌ها در آن‌ها صورت می‌گیرد.

پرسپترون چندلایه یک شبکه کاملاً به هم مرتبط است؛ چرا که هر سلول در یک لایه به تمامی سلول‌های لایه بعدی مرتبط است. اگر بعضی از این ارتباطات وجود نداشته باشد، شبکه، یک شبکه مرتبط ناقص است. زمانی که می‌گوییم شبکه از n لایه تشکیل شده است، در واقع تنها لایه‌های میانی و لایه خارجی را شمرده ایم و لایه ورودی را به حساب نیاورده‌ایم؛ چرا که سلول‌ها در این لایه، محاسبه ای را انجام نمی‌دهند. بنابراین شبکه تک لایه شبکه‌ای با تنها یک لایه خارجی است [۸]. در شکل (۲) نمونه‌ای از یک پرسپترون سه لایه نشان داده شده است.

۳-۲. تعریف سیستم

اولین گام در کنترل، آموزش شبکه عصبی برای ارائه یک

1. Perceptron
2. Single-Layer Perceptron
3. Multi-Layer Perceptron

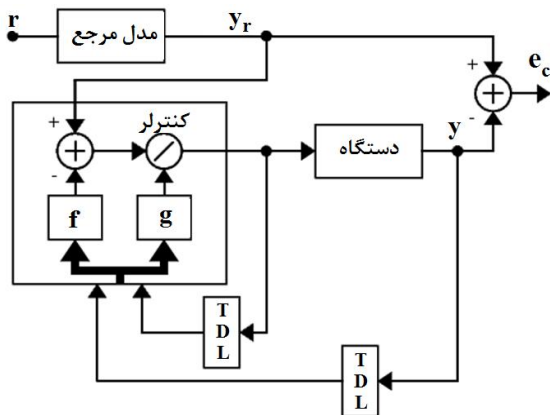
که در آن تابع G خطای مجموع مربعات را حداقل می‌کند. مشکل این نوع کنترلر این است که اگر لازم باشد شبکه عصبی برای ایجاد تابع G آموزش یابد، باید از پس‌انتشار پویا استفاده شود و این کار بسیار کند انجام می‌شود. یک راه‌حل برای این مشکل تخمین مدل است. کنترلر مورد استفاده در اینجا بر مبنای مدل تخمینی NARMA-L2 عمل می‌کند:

$$\hat{y}(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)] + g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)].u(k) \quad (7)$$

پس با استفاده از NARMA-L2 می‌توان کنترلر شکل (۵) را ایجاد کرد:

$$u(k+1) = y_r(k+d) - f[y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)] / g[y(k), y(k-n+1), u(k), \dots, u(k-n+1)] \quad (8)$$

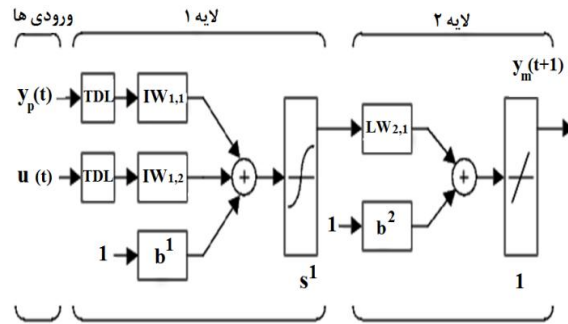
که در آن $d \geq 2$ است. شکل (۵) بلوک مربوط به کنترلر NARMA-L2 و شکل (۶) ساختار شبکه عصبی مورد استفاده در این کنترلر را نشان می‌دهد.



شکل ۵. کنترلر NARMA-L2 [۱۴]

۴. کنترل وضعیت به کمک کنترلر NARMA-L2

هدف از طراحی کنترلر، افزایش پایداری و کاهش زمان رسیدن به حالت مطلوب در برابر اعمال اغتشاش در یک پروسه است. در این مقاله از کنترلر PID برای مقایسه نتایج روش پیشنهادی استفاده شده است. عمده اشکال کنترلر PID علاوه بر دشواری



شکل ۴. ساختار شبکه عصبی مورد استفاده برای پیشگویی مقادیر آینده خروجی دستگاه [۱۴]

۳-۳. کنترلر NARMA-L2

کنترلر NARMA-L2 از شبکه‌های عصبی برای پیشگویی رفتار یک دستگاه غیرخطی استفاده می‌کند. کنترلر، ورودی‌های کنترلی را برای افزایش کارایی دستگاه روی محور زمان محاسبه می‌کند و در بین مدل‌های کنترلی دارای حداقل محاسبات است که آموزش آن به صورت دسته‌ای و آفلاین انجام می‌شود. تنها محاسبات آنلاین^۱ در گذر رو به جلوی شبکه عصبی است.

اولین گام در استفاده از این کنترلر تعریف سیستمی است که باید کنترل شود. شبکه عصبی برای نمایش یک دینامیک پیشرو از سیستم، آموزش داده می‌شود که گام اول، انتخاب ساختار مدل مورد استفاده است. همچنین این کنترلر به سادگی مدل دستگاه را دوباره‌چینی می‌کند. یک مدل استاندارد مورد استفاده در سیستم‌های غیرخطی گسسته، مدل NARMA به صورت زیر است:

$$y(k+d) = N[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-n+1)] \quad (5)$$

که در آن $u(k)$ ورودی سیستم و $y(k)$ خروجی آن است. در فاز تعریف می‌توانیم شبکه عصبی را برای تخمین تابع غیرخطی N آموزش دهیم و اگر بخواهیم خروجی سیستم از یک خط سیر مرجع پیروی کند ($y(k+d) = y_r(k+d)$)، باید در گام بعدی یک کنترلر غیرخطی را توسعه دهیم.

$$u(k) = G[y(k), y(k-1), \dots, y(k-n+1), y_r(k+d), u(k-1), \dots, u(k-m+1)] \quad (6)$$

1. Online

۵. نتایج

با اعمال تغییرات زاویه مرجع θ_{SOURCE} و همچنین J و B ، حالت‌های مختلف کارکرد موتور با استفاده از کنترلر شبکه عصبی NARMA-L2 بررسی، و نتایج بدست آمده از کنترلر پیشنهادی با نتایج بدست آمده از کنترلر PID مقایسه شده است.

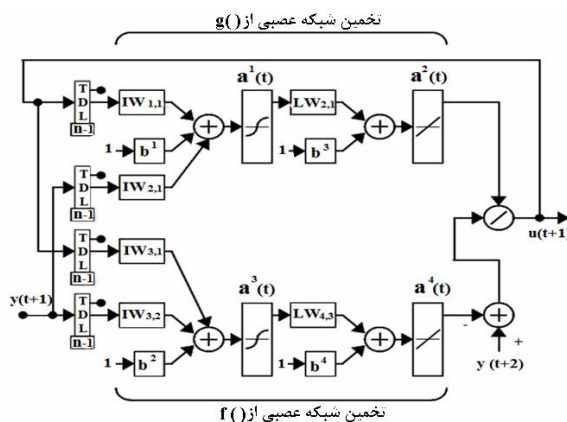
۵-۱. کنترل وضعیت در حالت ثابت بودن پارامترهای موتور

شکل (۸) نتایج کنترل وضعیت موتور dc در حالت ثابت بودن پارامترهای موتور برای دو کنترلر پیشنهادی و PID را نشان می‌دهد. نتایج نشان می‌دهد که در این حالت هر دو کنترلر دارای نتایج بدون نوسانی هستند؛ اما کنترلر NARMA-L2 سریعتر به وضعیت نهایی می‌رسد. همان‌طور که در شکل (۸) می‌بینیم کنترلر NARMA-L2 در مدت زمان ۰/۳۷ ثانیه به وضعیت نهایی می‌رسد اما کنترلر PID در مدت زمان ۱/۵ ثانیه به این وضعیت دست پیدا می‌کند.

۵-۲. کنترل وضعیت در حالت متغیر بودن پارامترهای موتور

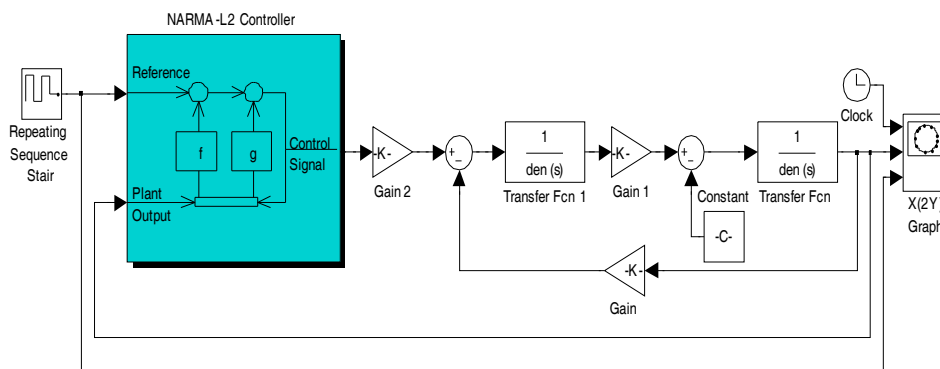
شکل‌های (۹)، (۱۰) و (۱۱) نتایج کنترل وضعیت موتور dc در حالت متغیر بودن پارامترهای موتور برای کنترلر پیشنهادی و کنترلر PID را نشان می‌دهند. این نتایج در سه حالت مقایسه شده‌اند: الف) $T_{m1} = 70\% T_m$ و ب) $T_{m2} = 50\% T_m$ و ج) $T_{m3} = 30\% T_m$ در آن‌ها T_m ثابت زمانی مکانیکی در شرایط ثابت بودن پارامترهای سیستم است.

تنظیم پارامترها برای داشتن پاسخ مطلوب، نیاز به اصلاح آن‌ها به واسطه تغییر شرایط کارکرد موتور است که عملاً تنظیم مجدد پارامتر در حین کارکردن موتور امکان‌پذیر نیست. برای برطرف کردن این مشکل روش کنترل هوشمند با استفاده از شبکه‌های عصبی پیشنهاد شده است.

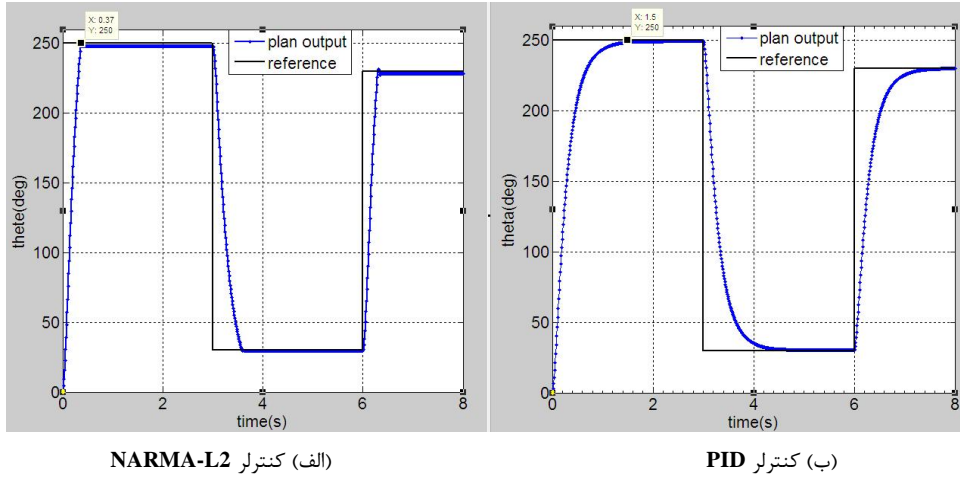


شکل ۶. ساختار شبکه عصبی مورد استفاده در کنترلر NARMA-L2 [۱۴]

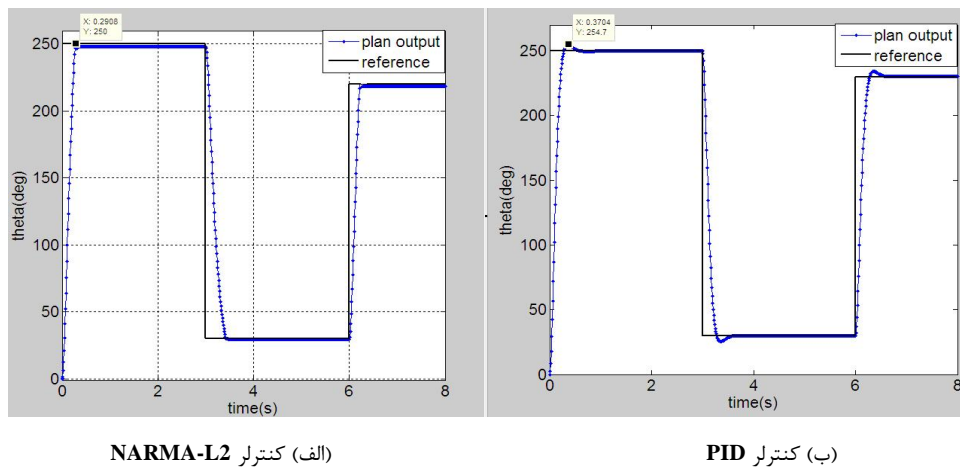
کنترلر NARMA-L2 باعث طراحی یک روش مفید برای کنترل مطلوب وضعیت در برابر تغییرات پارامترهای ماشین می‌شود. شکل (۷) چگونگی استفاده از کنترلر NARMA-L2 برای کنترل وضعیت موتور dc را نشان می‌دهد. شبکه عصبی موجود در بلوک NARMA-L2 رابطه ورودی و خروجی موتور dc را شناسایی، و سپس با استفاده از الگوریتم‌های بهینه‌سازی، ورودی موتور dc را برای انطباق وضعیت خروجی بر وضعیت مرجع، بهینه می‌کند.



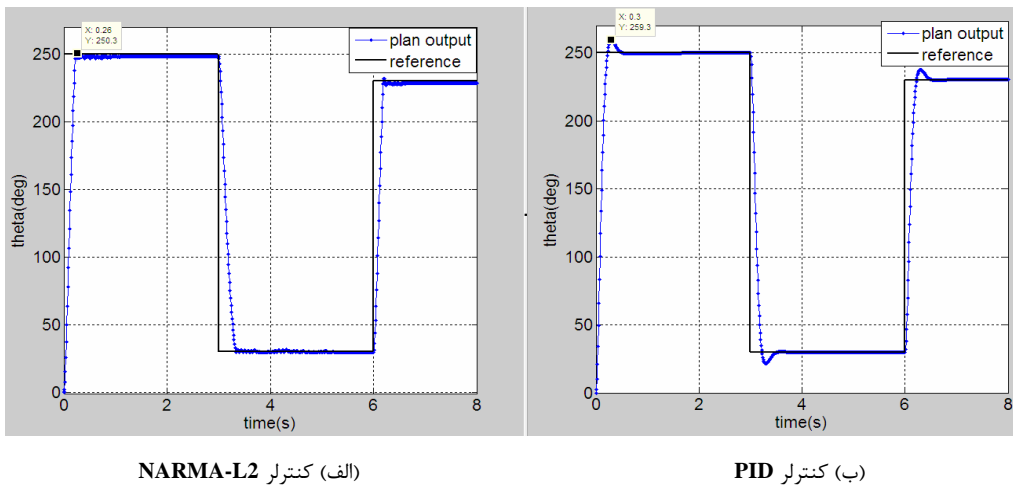
شکل ۷. نحوه استفاده از کنترلر NARMA-L2 برای کنترل وضعیت موتور DC



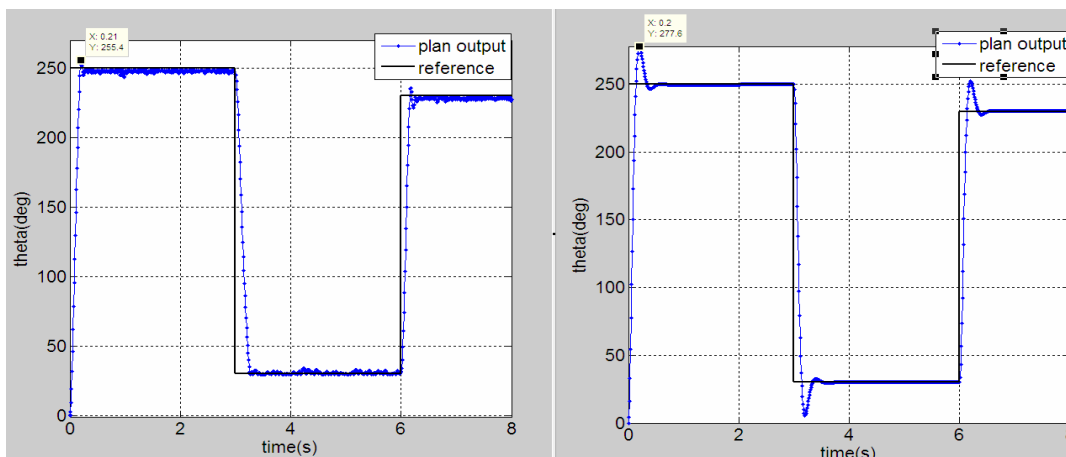
شکل ۸. نتایج کنترل وضعیت در شرایط ثابت بودن پارامترهای موتور DC



شکل ۹. نتایج کنترل وضعیت موتور DC در حالت $T_m = 70\%$



شکل ۱۰. نتایج کنترل وضعیت موتور DC در حالت $T_m = 50\%$



الف) کنترلر NARMA-L2

ب) کنترلر PID

شکل ۱۱. نتایج کنترل وضعیت موتور DC در حالت $T_{m3} = 30\% T_m$

۶. نتیجه گیری

در این مقاله از کنترلر NARMA-L2 که در آن از شبکه‌های عصبی استفاده شده است، برای کنترل وضعیت یک موتور dc استفاده و نتایج با یک کنترلر PID مقایسه شد. ویژگی و برتری روش پیشنهادی، اصلاح ضرایب شبکه عصبی در حین کارکردن موتور و مقاوم بودن پاسخها نسبت به تغییرات لختی دورانی (J) و اصطکاک (B) است. برخلاف کنترلر PID که در اثر تغییر یکی از پارامترهای مدل نظیر J و B، کارایی بهینه خود را از دست می‌دهد و نیاز به تنظیم مجدد پارامترهای کنترلر دارد، روش پیشنهادی نیاز به طراحی مجدد ندارد و در برابر تغییرات، پارامترهای کنترلر بطور خودکار اصلاح می‌شود. همان‌طور که نشان داده شد، به‌ازای تغییرات در مقدار T_m ، کنترلر NARMA-L2 کارایی بهینه خود را حفظ می‌کند در حالی که کنترلر PID کارایی بهینه خود را از دست می‌دهد و بالازدگی و پایین‌زدگی شدیدی در پاسخهای آن مشاهده می‌شود. مزیت و برتری دیگر کنترلر پیشنهادی این است که در مدت زمان کمتری به حالت پایدار می‌رسد و این نتایج نشان می‌دهد که کنترلر NARMA-L2 می‌تواند گزینه مناسبی برای جایگزین کردن کنترلرهای کلاسیک مثل کنترلر PID در کارهای کنترلی باشد.

۵-۲-۱. $T_{m1} = 70\% T_m$

همان‌طور که در شکل (۹) دیده می‌شود در این حالت خروجی کنترلر بدون هیچ‌گونه پایین‌زدگی^۱ یا بالازدگی^۲ و در مدت زمان ۰/۲۹ ثانیه به مقدار نهایی خود می‌رسد، در حالی که کنترلر PID با ۰/۲ بالازدگی و پس از حدود ۰/۷ ثانیه به این مقدار می‌رسد.

۵-۲-۲. $T_{m2} = 50\% T_m$

مطابق شکل (۱۰) در این حالت خروجی کنترلر NARMA-L2 تقریباً بدون پایین‌زدگی و بالازدگی و در مدت زمان ۰/۲۶ ثانیه به مقدار نهایی خود می‌رسد در حالی که کنترلر PID با حدود ۰/۴ بالازدگی و پس از حدود ۰/۶ ثانیه به مقدار نهایی خود می‌رسد.

۵-۲-۳. $T_{m3} = 30\% T_m$

طبق شکل (۱۱) در این حالت خروجی کنترلر NARMA-L2 با حدود ۰/۲ بالازدگی و در مدت زمان ۰/۲۱ ثانیه به مقدار نهایی خود می‌رسد در حالی که کنترلر PID با حدود ۰/۱۱ بالازدگی و پس از حدود ۰/۵۶ ثانیه به مقدار نهایی خود می‌رسد.

1. Undershoot
2. Overshoot

۷. مراجع

- [1] Horng, J. H. "Neural Adaptive Tracking Control of a DC Motor."; International Journal of Information Science 1999, 118, 1-13.
- [2] Tang, K. L.; Mulholland, R. J. "Comparing Fuzzy Logic with Classical Controller Designs."; IEEE Trans. SMC. 1987, 17(6), 1085-1087.
- [3] Rico, R. M.; Anderson, J. S.; Kebrekidis, I. G. "Continuous-Time Nonlinear Signal Processing: a Neural Network Based Approach for Gray Box Identification."; IEEE Workshop on Neural Network for Signal Processing 1994, 596-605.
- [4] La, Y. F.; Lau, C. C. "Development of Fuzzy Algorithms for Servo Systems."; IEEE International Conference on Robotics and Automation, Philadelphia 1988, 24-29, 65-71.
- [5] Nouri, K. H.; Dhaouadi, R.; Braiek N. B. "Adaptive Control of a Nonlinear DC Motor Drive Using Recurrent Neural Networks."; Applied Soft Computing 2008, 78, 371-382.
- [6] Dzung, P. Q.; Phuong, L. M. "ANN Control System DC Motor."; International Symposium on Electrical- Electronics Engineering ISEE 2005.
- [7] George, M. "Speed Control of Separately Excited DC Motor."; American Journal of Applied Sciences 2008, 5(3), 227-233.
- [8] Hagan, M. T.; Demuth, H. B.; Beale, M. H. "Neural Network Design."; PWS Publishing, Boston, 1996.
- [9] Longo, G.; Brescia, M. "Multilayer Perceptron."; <http://Voneural.Na.Infn.It/Mlp.Html>.
- [۱۰] شریف خضری، حمیدرضا و صدیقی انارکی، علیرضا "کنترل هوشمند وضعیت موتور dc"، هشتمین کنفرانس دانشجویی مهندسی برق ایران، دانشگاه باهنر کرمان، ص ۱-۲، شهریور ۱۳۸۴.
- [۱۱] اوگاتا، کاتسو هیکو، ترجمه دیانی، محمد "مهندسی کنترل"، انتشارات نص، چاپ چهارم، سال ۱۳۸۱.
- [۱۲] مختاری، م، ترجمه صمدی، و. "کاربرد matlab و simulink در مهندسی"، انتشارات خراسان، چاپ اول، سال ۱۳۸۰.
- [۱۳] منهاج، محمد باقر "هوش محاسباتی، مبانی شبکه‌های عصبی"، انتشارات دانشگاه صنعتی امیرکبیر، چاپ پنجم، سال ۱۳۸۷.
- [۱۴] کیا، مصطفی "شبکه‌های عصبی در matlab"، خدمات نشر کیان رایانه سبز، چاپ دوم، ص ۱۳۸-۱۲۱، سال ۱۳۸۷.

