مجله کنترل
ISSN (print) 2008-8345
ISSN (online) 2538-3752
جلد ۱۳، شماره ۲، تابستان ۱۳۹۸، صفحه ۶۶-۵۳

شناسایی و کنترل تطبیقی موقعیت و سرعت موتور DC مغناطیس دائم با مشخصه غیرخطی ناحیه مرده مبتنی بر ماشین‌های بردار پشتیبان

محمود حسن‌پور دهنوی [۱]، سید کمال حسینی ثانی [۲]

[۱] فارغ‌التحصیل کارشناسی ارشد مهندسی برق، گروه کنترل، دانشگاه فردوسی مشهد، m.hasanpur.dehnavi@gmail.com

[۲] دانشیار، دانشکدهٔ مهندسی برق، گروه کنترل، دانشگاه فردوسی مشهد، k.hosseini@um.ac.ir

دریافت: ۱۳۹۷/۰۶/۲۲ ویرایش: ۱۳۹۷/۱۰/۱۵ پذیرش: ۱۳۹۷/۱۱/۲۴

چکیده: در این مقاله نوع جدیدی از شبکه‌های عصبی به نام ماشین‌های بردار پشتیبان حداقل مربعات که در سال‌های اخیر به منظور شناسایی سیستم‌های غیرخطی مورد توجه زیادی قرار گرفته‌اند، جهت شناسایی سیستم موتور DC با مشخصه غیرخطی ناحیه مرده به کارگرفته شده است. سیستم شناسایی شده پس از خطی‌سازی در هر واحد زمانی به صورت روی خط اطلاعات مدل را در اختیار کنترل‌کننده پیش‌بین موقعیت و سرعت به منظور دنبال‌کردن مسیر مطلوب موقعیت و سرعت قرار می‌دهد. در روش پیشنهادی حلقه‌های کنترل گشتاور، سرعت و موقعیت به صورت کاملا خودکار و براساس مدل شناسایی شده بسته می‌شوند. روش پیشنهادی برروی سرودرایور ساخته‌شده پیاده‌سازی شده است و نتایج عملی ترسیم و تحلیل شده‌اند. مزیت بزرگ این روش عدم نیاز به تنظیم پارامترهای کنترل‌کننده‌های جریان، سرعت و موقعیت می‌باشد. شناسایی روی خط سیستم امکان دنبال‌کردن تغییرات دینامیکی فرآیند را فراهم می‌نماید. علاوه برآن ساختار پیشنهادی توانایی غلبه‌بر اصطکاک کولمب به ویژه در سرعت‌های پایین را دارا بوده و قادر است گشتاور، سرعت و موقعیت موتور DC مغناطیس دائم را به طور دقیقی کنترل نماید.

کلمات کلیدی: ماشین‌های بردار پشتیبان حداقل مربعات، کنترل‌کننده پیش‌بین تعمیم یافته، کنترل‌کننده سری، شناسایی روی خط، مشخصه غیرخطی ناحیه مرده.

Identification and Adaptive Position and Speed Control of Permanent Magnet DC Motor with Dead Zone Characteristics Based on Support Vector Machines

Mahmoud Hasanpour Dehnavi, Seyed Kamal Hosseini Sani

Abstract: In this paper a new type of neural networks known as Least Squares Support Vector Machines which gained a huge fame during the recent years for identification of nonlinear systems has been used to identify DC motor with nonlinear dead zone characteristics. The identified system after linearization in each time span, in an online manner provide the model data for Model Predictive Controller of position and speed in order to tracking the desired references trajectory. In this method all the cascaded controllers including current, speed and position has been automatically tuned based on the identified model. The offered method has been tested on the servo-drive made specifically for this purpose, and all the results are practically examined and analyzed. The biggest advantage of this method is the self-tuning behavior which insulates the user for tuning any of the controller's parameters. The online identification of the system provides the possibility to keep track of the changes in dynamics of the system as well as tackling the coulomb's friction specifically in low speeds with accurate controlling of the speed and position for DC motors.

Keywords: Least Square Support Vector Machines, Generalized Predictive Control (GPC), Cascaded Controller, Online identification, Nonlinear dead zone characteristics.

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

۱- مقدمه

مدل مرسوم موتور DC مغناطیس دائم یک مدل خطی مرتبه دو می‌باشد که در آن از اثرات غیرخطی ناشی از اصطکاک صرف نظر می‌شود. مشخصه غیرخطی ناحیه مرده[1] که در درجه اول ناشی از اصطکاک غیرخطی کولمب می‌باشد [۱] مانع از حرکت روتور در سرعت‌های پایین می‌شود، تا زمانی که گشتاور موتور به اندازه کافی بزرگ شود تا بر آن غلبه کند.

برای جبران اثر مخرب اصطکاک در عملکرد سیستم سرو بایستی سیستم موتور DC با مشخصه غیرخطی اصطکاک را به خوبی شناسایی کرده و با استفاده از کنترل‌کننده‌ای مناسب اثر آن را جبران نماییم. برای شناسایی سیستم‌های غیرخطی از مدل‌های وینر[2] و همرستین[3] به طور وسیعی به منظور مدل‌سازی فرآیندهای غیرخطی استفاده می‌شود[۲,۳]. در مدل همرستین یک بلوک غیرخطی استاتیک به صورت سری قبل از یک بلوک خطی دینامیک قرار می‌گیرد، در حالی که مدل وینر ساختاری عکس مدل همرستین دارد[۴]. به منظور دستیابی به عملکرد مطلوب و قابلیت تطبیق پذیری، شبکه‌های عصبی در ساختار مدل وینر و همرستین ادغام می‌شوند [۸-۲,۵]. کارا [۹] به منظور شناسایی سیستم موتور DC با اصطکاک کولمب و ناحیه مرده از مدل همرستین کمک گرفته به طوری که برای شناسایی بلوک خطی و غیرخطی از روش حداقل مربعات بازگشتی استفاده کرده است. مدل همرستین به خوبی قادر است سیستم موتور DC با مشخصه غیرخطی ناحیه مرده را شناسایی کند اما استفاده از این مدل مناسب کنترل تطبیقی موتور به صورت آنلاین نیست. پنگ و دابی [۱۰] با بیان مدل وینر به صورت یک شبکه عصبی با استفاده از الگوریتم پس انتشار خطا سیستم موتور DC با مشخصه غیرخطی ناحیه مرده را به صورت روی خط شناسایی کرده و سرعت موتور DC را به صورتی تطبیق پذیر کنترل کرده‌اند.

برای کنترل موتور DC کنترل‌کننده تناسبی، مشتقی و انتگرالی به خاطر ساختار ساده و عدم نیاز به اطلاع دقیق از دینامیک سیستم یکی از رایج‌ترین روش‌ها در صنعت می‌باشد[۱۱]. با این وجود به هنگام استفاده از ساختار PID تعیین دقیق بهره‌های کنترل‌کننده به خاطر وجود عدم قطعیت و مشخصه‌های غیرخطی(تغییرات بار، اشباع، اصطکاک و...) که باعث کاهش عملکرد سیستم کنترلی می‌شوند، مشکل است. مقاله‌های پژوهشی مختلفی بر روی کنترل‌کننده PID تطبیقی[۱۵-۱۲]، کنترل-کننده PID خود تنظیم[۳۲-۱۶,۱۷]، کنترل‌کننده PID خود تنظیم پیش‌بین[۱۸]، و... تمرکز کرده‌اند. در کنترل‌کننده PID تطبیقی و خود تنظیم، پارامترهای کنترل‌کننده براساس تغییرات پارامترهای فرآیند به صورت خودکار تنظیم می‌شوند. اگرچه این کنترل‌کننده‌ها عملکرد رضایت بخشی دارند ولی هنوز در مواجهه با سیستم‌های غیرخطی ناتوان می‌باشند.

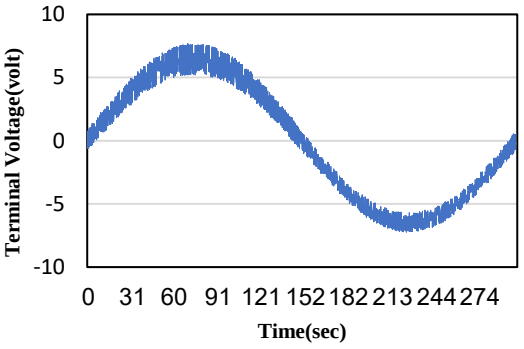
در سال‌های اخیر شبکه‌های عصبی مبتنی‌بر تکنیک‌های کنترلی در کاربردهای صنعتی مختلفی به کار گماشته شده‌اند. مارتین [۱۹] و چن [۲۰] از شبکه‌های عصبی برای تنظیم ضرایب کنترل‌کننده PID استفاده کرده‌اند. یوان و وانگ [۲۱] یک شبکه عصبی مبتنی‌بر کنترل‌کننده PID خود یاد گیرنده ارائه کرده‌اند که در آن پارامترهای کنترل‌کننده PID به صورت وزن‌های شبکه عصبی بیان شده و توسط الگوریتم شبکه عصبی تنظیم می‌گردند. در سیستم موتور DC، ناحیه مرده معمولا به دلیل اصطکاک و سایش مکانیکی به وجود می‌آید. برای حذف اثر ناحیه مرده، پژوهش‌های زیادی انجام شده است از قبیل: رویکردهای تطبیقی[۲۷-۲۲]، شبکه‌های عصبی [۳۱-۲۸] و سیستم‌های فازی[۳۴-۳۳].

در این مقاله نوع جدیدی از شبکه‌های عصبی به نام ماشین‌های بردار پشتیبان حداقل مربعات که در سال‌های اخیر به منظور شناسایی سیستم‌های غیرخطی مورد توجه زیادی قرار گرفته‌اند[۴۱-۳۵]، جهت شناسایی سیستم موتور DC با مشخصه غیرخطی ناحیه مرده به کار گرفته شده است. سیستم شناسایی شده در هر واحد زمانی به صورت روی خط اطلاعات مدل را در اختیار کنترل‌کننده پیش‌بین تطبیقی به منظور دنبال‌کردن مسیر مرجع مطلوب موقعیت و سرعت قرار می‌دهد.

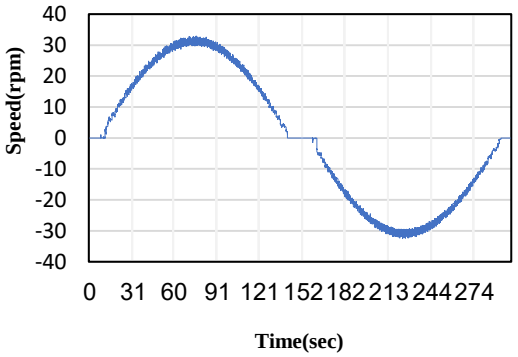
در شکل۱ ولتاژ ترمینال و سرعت شفت خروجی یک موتور DC مغناطیس دائم را مشاهده می‌کنید. یک شکل موج سینوسی ولتاژ با دوره تناوب ۳۰۰ ثانیه(ولتاژ اعمالی را بسیار فرکانس پایین و کم دامنه در نظر گرفته‌ایم تا بتوانیم اثر مخرب اصطکاک را افزایش دهیم) به موتور تحت آزمایش اعمال شده و با استفاده از یک انکودر افزایشی[4] ۱۰۰۰ پالس که به ته شفت موتوری با نسبت گیربکس ۱ به ۲۴ متصل شده است، سرعت موتور به صورت حلقه باز بر حسب دور بر دقیقه ترسیم شده است. همان طور که مشخص است در سرعت‌های پایین اصطکاک غیرخطی کولمب مانع از حرکت شفت موتور شده است. به ناحیه‌ای که در آن با اعمال ولتاژ شفت موتور حرکت نمی‌کند ناحیه مرده می‌گویند. لازم به ذکر است که نوسانات شکل موج ولتاژ موتور ناشی از مدولاسیون پهنای باند[5] در قسمت سوئیچینگ سرودرایور می‌باشد. این نوسانات پس از دو مرحله فیلترینگ آنالوگ و دیجیتال تا حد امکان کاهش یافته‌اند(فیلترینگ بیش از حد باعث افزایش تاخیر در حلقه کنترلی و کاهش حاشیه پایداری سیستم می‌شود). همچنین نوسانات شکل موج سرعت به خاطر ارتعاشات منتقل شده از موتور به انکودر از طریق کوپلینگ می‌باشند.

[1] Dead zone
[2] Wiener
[3] Hammerstein
[4] Incremental Encoder
[5] Pulse Width Modulation

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]



(۱)

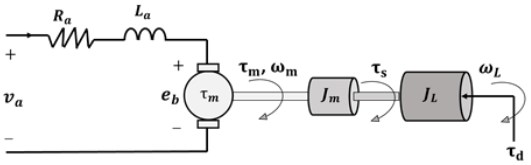


(۲)

شکل۱: ولتاژ ترمینال موتور(۱)، سرعت موتور برحسب دور بردقیقه(۲)

۲- مدل دینامیکی موتور DC مغناطیس دائم

در این قسمت به بیان مدل دینامیکی موتور DC مغناطیس دائم به همراه بار متصل به شفت آن می‌پردازیم. شکل2 شماتیک کامل سیستمی که در ادامه توضیح داده خواهد شد را نشان می‌دهد[۱۰].



شکل۲: مدل الکترومکانیکی موتور DC متصل به بار

معادلات الکتریکی و مکانیکی موتور DC مغناطیس دائم را می‌توان به ترتیب با روابط (۱-الف) و (۱-ب) بیان کرد:

$$v_a = R_a i_a + L_a \frac{di_a}{dt} + e_b \qquad (۱\text{-الف})$$

$$J_m \frac{d\omega_m}{dt} = \tau_m - B_m \omega_m - \tau_s \qquad (۱\text{-ب})$$

که در رابطه (۱) v_a ولتاژ آرمیچر موتور، R_a مقاومت سیم‌پیچ آرمیچر، L_a اندوکتانس سیم‌پیچ آرمیچر، i_a جریان آرمیچر، e_b ولتاژ برگشتی نیروی محرکه الکتریکی[۱]، J_m ممان اینرسی[۲] موتور، ω_m سرعت زاویه‌ای موتور، τ_m گشتاور موتور، B_m ضریب اصطکاک ویسکوز[۳] موتور و τ_s گشتاور انتقالی شفت می‌باشد. در موتور DC مغناطیس دائم ولتاژ برگشتی نیروی محرکه الکتریکی با یک ثابت الکتریکی[۴] وابسته به سرعت زاویه‌ای موتور و گشتاور موتور با یک ثابت مکانیکی[۵] وابسته به جریان موتور می‌باشد:

$$e_b = K_e \omega_m \qquad (۲\text{-الف})$$

$$\tau_m = K_m i_a \qquad (۲\text{-ب})$$

که K_e و K_m به ترتیب ثابت‌های الکتریکی و مکانیکی می‌باشند. معادلات مکانیکی مربوط به ممان اینرسی بار نیز به صورت رابطه (۳) قابل بیان می‌باشد:

$$\begin{aligned} &J_l \frac{d\omega_l}{dt} = \tau_s - B_l \omega_l - \tau_d \\ &\tau_s = K_s(\theta_m - \theta_l) + B_s(\omega_m - \omega_l) \\ &with \ \frac{d\theta_m}{dy} = \omega_m \quad , \quad \frac{d\theta_l}{dt} = \omega_l \end{aligned} \qquad (۳)$$

که J_l ممان اینرسی بار، ω_l سرعت زاویه‌ای بار، B_l ضریب اصطکاک ویسکوز بار، τ_d گشتاور اغتشاش بار، K_s قابلیت ارتجاعی شفت، B_s ضریب میرایی داخلی شفت و θ_m و θ_l به ترتیب جابجایی زاویه‌ای موتور و بار می‌باشند. به منظور طراحی یک سیستم کنترل حرکت با کارایی بالا اطلاع دقیق از دینامیک سیستم الکترومکانیکی شامل رفتارهای خطی و غیرخطی سیستم ضروری می‌باشد. اصطکاک یکی از مشخصه‌های نامطلوب و غیرقابل اجتناب در سیستم‌های مکانیکی می‌باشد که بایستی اثر آن را در مدل‌سازی و کنترل لحاظ کرد. یک فرم کلی از اصطکاک در حالت دورانی به صورت رابطه (۴) قابل بیان است:

$$\tau_f(\omega) = \left(\tau_C + (\tau_S - \tau_C)e^{-\left(\left|\frac{\omega}{\omega_s}\right|\right)^i}\right) sign(\omega) \qquad (۴)$$

که شامل اصطکاک ایستایی[۶]، کولمب[۷] و استریبک[۸] می‌باشد[1]. اکنون بایستی معادلات دینامیکی خطی موتور DC با اضافه‌کردن رفتارهای غیرخطی به آن اصلاح شود. معادلات اصلاح شده به صورت رابطه (۵) می‌باشد:

$$J_m \frac{d\omega_m}{dt} = \tau_m - B_m \omega_m - \tau_s - \tau_f(\omega_m) \qquad (۵\text{-الف})$$

$$J_l \frac{d\omega_l}{dt} = \tau_s - B_l \omega_l - \tau_d - \tau_f(\omega_l) \qquad (۵\text{-ب})$$

۳- ماشین‌های بردار پشتیبان حداقل مربعات

ماشین‌های بردار پشتیبان حداقل مربعات نسخه‌ای اصطلاح شده از ماشین‌های بردار پشتیبان استاندارد هستند که در آن به‌جای حل مسئله بهینه‌سازی برنامه‌ریزی درجه دو، از تابع هزینه مجموع مربعات خطا (SSE) و قیود تساوی استفاده می‌کنند[۴۲]. این امر منجر به یک سری معادلات خطی شده که به طور قابل توجهی حجم محاسبات را کاهش و سرعت پردازش را افزایش می‌دهند و این نسخه از ماشین‌های بردار پشتیبان

[1] Back Electromotive Force
[2] Moment of Inertia
[3] Viscous Friction
[4] Electrical Constant
[5] Mechanical Constant
[6] Static Friction(Stiction)
[7] Coulomb Friction
[8] Stribeck Friction

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

را مناسب شناسایی آنلاین می‌نمایند. ماشین‌های بردار پشتیبان حداقل مربعات در سال‌های اخیر جهت شناسایی سیستم‌های غیرخطی مورد توجه زیادی قرار گرفته‌اند[۳۶-۴۰]. اگر مدل LS-SVM شناسایی شده را به صورت رابطه (۶) در نظر بگیریم:

$$f(x) = w^T.\varphi(x) + b \qquad (۶)$$

که $\varphi(x)$ یک نگاشت غیرخطی از فضای ورودی[1] به فضای ویژگی[2] با ابعاد بالاتر می‌باشد که با x بردار ورودی مدل LS-SVM می‌باشد که با توجه به داده‌های مجموعه آموزش[3] و به صورت رابطه (۷) تشکیل می‌شود:

$$x(k) = [u(k-1), \dots, u(k - n_u), y(k), \dots, y(k - n_y + 1)]^T \qquad (۷)$$

که u و y به ترتیب ورودی و خروجی سیستم و n_u و n_y مرتبه دینامیکی سیستم می‌باشند. پارامترهای مدل یعنی w و b براساس اصل مینیمم سازی ریسک عملیاتی (ERM) تخمین زده می‌شوند. تابع بهینه سازی به صورت تابع تلفات مربعی با قیود تساوی به صورت رابطه (۸) تعریف می‌شود:

$$\min_{w,b,e} J(w,b,e) = \frac{1}{2}w^T w + C\frac{1}{2}\sum_{i=1}^{n} e_i^2$$
$$s.t.\ \ y_i = w^T\varphi(x_i) + b + e_i \ \ , i = 1,2,\dots,n \qquad (۸)$$

که در رابطه (۸) ، C پارامتر تنظیم و e خطای بین خروجی واقعی و خروجی مدل می‌باشد. ازآنجایی‌که حل مسئله بهینه‌سازی به صورت رابطه (۸) دشوار است با استفاده از ضرب‌کننده‌های لاگرانژ قید را در تابع هزینه به صورت رابطه (۹) ترکیب می‌کنیم:

$$L(w,b,e;\alpha) = \frac{1}{2}w^T w + C\frac{1}{2}\sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n}\alpha_i[w^T\varphi(x_i) + b + e_i - y_i] \qquad (۹)$$

که α_i ضرب‌کننده‌های لاگرانژ می‌باشند که بایستی از طریق آموزش شبکه عصبی از روی مجموعه آموزش محاسبه شوند. برای محاسبه مقادیر α_i و b، شرایط بهینه با گرفتن مشتقات جزئی از رابطه (۹) به صورت رابطه (۱۰) نسبت به هر یک از پارامترها بدست می‌آیند.

$$\frac{\partial L}{\partial w} = 0 \ \rightarrow \ \ w = \sum_{i=1}^{n}\alpha_i\varphi(x_i)$$
$$\frac{\partial L}{\partial b} = 0 \rightarrow \ \ \sum_{i=1}^{n}\alpha_i = 0$$
$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \ \alpha_i = Ce_i \qquad i = 1,2,\dots,n \qquad (۱۰)$$
$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T\varphi(x_i) + b + e_i - y_i = 0\ ; i = 1,2,\dots,n$$

از شرایط رابطه (۱۰) به وضوح مشخص است که مقادیر ضرب‌کننده‌های لاگرانژ در هرگام وابسته به خطای e_i می‌باشند. با حذف w و e از رابطه (۱۰) داریم:

$$\begin{bmatrix} 0 & r_v^T \\ r_v & \emptyset + \frac{I}{C} \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \qquad (۱۱)$$

که $y = [y_1; y_2; \dots; y_n]$ ، $r_v = [1; 1; \dots; 1_n]$ ماتریس I ، واحد n×n، $\alpha = [\alpha_1; \alpha_2; \dots; \alpha_n]$ و $\emptyset$ ماتریس متقارن n×n می باشد که به صورت رابطه (۱۲) تعریف می‌شود:

$$\emptyset_{ij} = \varphi(x_i)\varphi(x_j) = K(x_i, x_j) \qquad i,j = 1,2,\dots,n \qquad (۱۲)$$

با حل دستگاه معادلات خطی رابطه (۱۱) مقادیر ضرب‌کننده‌های لاگرانژ و عبارت بایاس بدست می‌آیند و در نهایت مدل LS-SVM به صورت رابطه (۱۳) خواهد بود:

$$y(x) = \sum_{i=1}^{n}\alpha_i K(x, x_i) + b \qquad (۱۳)$$

٤- طراحی کنترلر پیش بین تعمیم یافته براساس مدل LS-SVM

ایده کنترل پیش بین مبتنی بر مدل در اوایل دهه ۱۹۷۰ به عنوان یک روش کنترل صنعتی کارآمد به وجود آمد[۴۳]. کنترلگرهای پیش‌بین متکی بر مدل فرآیند هستند که اغلب مدل‌ها از طریق روش‌های شناسایی تجربی حاصل می‌شوند. کنترل‌کننده پیش بین در هرگام نمونه برداری با توجه به مسیر مرجع و با استفاده از مدل فرآیند و با بهره‌گیری از ورودی‌ها و خروجی‌های گذشته سیگنال‌های کنترلی آینده را با بهینه‌سازی یک تابع هزینه پیش بینی می‌کند. سپس سیگنال کنترلی پیش بینی شده یک گام بعد به سیستم اعمال می‌شود. در لحظه بعدی نمونه برداری روند بالا دوباره تکرار خواهد شد. به عبارتی ساده‌تر در کاربردهایی که رفتار مطلوب

[1] Input Space
[2] Feature Space
[3] training set

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

سیستم(مسیر مرجع) معلوم باشد(به عنوان مثال ربات‌هایی که کارهایی مشخص و تکراری انجام می‌دهند) کنترل پیش بین قادر است با استفاده از مدل فرآیند، ورودی‌های آینده سیستم را با توجه به بهینه‌سازی یک تابع هزینه محاسبه نماید که این امر موجب بهبود کیفیت کنترل خواهد شد.

مدل‌های استفاده شده در کنترل‌کننده‌های پیش بین در حالت کلی مبتنی بر پاسخ ضربه، پاسخ پله، فضای حالت و یا تابع تبدیل فرآیند می‌باشند. در این پژوهش از مدل تابع تبدیل فرآیند به منظور پیاده‌سازی کنترل‌کننده پیش‌بین استفاده خواهد شد. کنترل پیش‌بین تعمیم یافته یکی از روش‌های کنترل پیش‌بین مدل می‌باشد که به طور صریح از مدل تابع تبدیل فرآیند جهت محاسبه سیگنال کنترل استفاده می‌کند. کنترل GPC اولین بار توسط کلارک در سال ۱۹۸۷ مطرح شد[۴۳]. مدل CARIMA را به صورت رابطه (۱۴) در نظر بگیرید:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \quad (۱۴)$$

اگر در مدل رابطه (۱۳) ورودی x را به صورت رابطه (۱۵) در نظر بگیریم:

$$x(k) = [u(k-1), \dots, u(k-n_u), y(k), \dots, y(k-n_y)] \quad (۱۵)$$

که در آن n_u و n_y مرتبه دینامیکی فرآیند(و یا به عبارتی حافظه سیستم) می‌باشند. اکنون با استفاده از کرنل خطی می‌توان مدل شبکه عصبی LS-SVM رابطه (۱۳) را به صورت رابطه (۱۶) باز نویسی کرد:

$$\begin{aligned} y(x) &= \sum_{i=1}^{n}\alpha_i(k)\{x_i^T(k)[u(k-1), \dots, u(k-n_u), y(k), \dots, y(k-n_y)]\} + b(k) \\ &= \sum_{i=1}^{n}\alpha_i(k)\{x_{i,1}(k)u(k-1) + \cdots + x_{i,n_u}(k)u(k-n_u) + x_{i,n_u+1}(k)y(k-1) + \cdots + x_{i,n_u+n_y}(k)y(k-n_y)\} + b(k) \\ &= \sum_{i=1}^{n}[\alpha_i(k)(x_{i,1}(k) + \cdots + x_{i,n_u}(k)z^{-n_u+1})]u(k-1) + [\alpha_i(k)(x_{i,n_u+1}(k)z^{-1} + \cdots + x_{i,n_u+n_y}(k)z^{-n_y})]y(k) + b(k) \end{aligned} \quad (۱۶)$$

رابطه (۱۶) را می‌توان به صورت رابطه (۱۷) نوشت که همان مدل ARMA می‌باشد:

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + b(k) \quad (۱۷)$$

که در آن چند جمله‌ای‌های A و B به صورت رابطه (۱۸) بدست می‌آیند:

$$\begin{aligned} A(z^{-1}) &= 1 - [\sum_{i=1}^{n}\alpha_i(k)(x_{i,n_u+1}(k)z^{-1} + \cdots + x_{i,n_u+n_y}(k)z^{-n_y})] \\ B(z^{-1}) &= \sum_{i=1}^{n}\alpha_i(k)(x_{i,1}(k) + \cdots + x_{i,n_u}(k)z^{-n_u+1}) \end{aligned} \quad (۱۸)$$

۵- طراحی و ساخت سرو درایور

هدف نهایی یک مهندس تبدیل دانش تئوری و خام به محصول تجاری و صنعتی می‌باشد. تفاوت اساسی رشته‌های مهندسی با رشته‌های محض در نوع نگاه ویژه و منحصر به فرد یک مهندس به دانش و بروز و نمود آن در قالب یک محصول است. طراحی و ساخت سرودرایور[1] پس از سه نسخه طراحی و انجام تست‌های مختلف بعد از ۸ ماه منجر به محصول نهایی شد. انتخاب پروتکل ارتباطی و میکروکنترلر یکی از مراحل مهم طراحی و ساخت سرودرایور بود چرا که انتخاب نادرست آنها می‌توانست باعث افزایش تعداد نسخه‌ها و اتلاف وقت شود. پس از تحقیقات لازم پروتکل ارتباطی شبکه و میکروکنترلر STM32F407VGT6 ساخت شرکت ST انتخاب شدند.

شکل۳ (۱) نمایی از طراحی ۳ بعدی سرودرایور را در نرم افزار آلتیوم[2] نشان می‌دهد. در شکل۳ (۲) می‌توانید سرودرایور ساخته شده نهایی را مشاهد نمایید. همچنین شکل۴ تست ستاپ ساخته‌شده جهت سنجش عملکرد ساختار کنترلی پیشنهادی را نشان می‌دهد. تست ستاپ شامل یک موتور DC مغناطیس دائم ۲۵۰ ولت با انکودر افزایشی ۱۰۰۰ پالس و نسبت گیربکس ۱ به ۲۴ می‌باشد که به یک دیسک دوار متصل شده است. دیسک دوار شامل یک شیار بوده که از آن جهت اتصال بار و تست عملکرد کنترل‌کننده نسبت به اغتشاش بار استفاده می‌شود.



(۱)

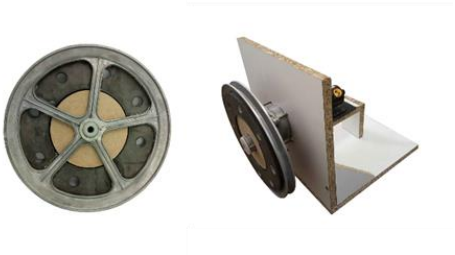


(۲)

شکل۳: نمایی از طراحی ۳ بعدی سرودرایور در نرم افزار آلتیوم(۱)، سرو درایور نهایی(۲)

[1] Servo Driver
[2] Altium Designer

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

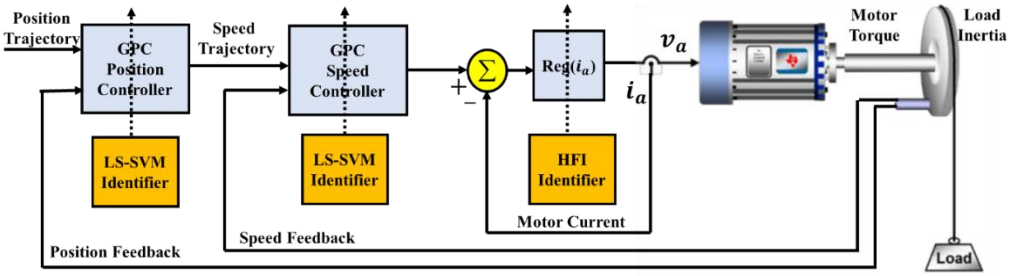


شکل۴: تست ستاپ ساخته شده جهت انجام آزمایش‌ها

مطابق شکل۵ حلقه کنترل جریان داخلی‌ترین حقله کنترلی بوده و استفاده از آن به منظور دستیابی به اهداف زیر می‌باشد:

- کنترل جریان راه‌اندازی
- محافظت از سرودرایور و موتور به هنگام اضافه بار
- بهبود کیفیت کنترل
- تضمین پایداری داخلی‌ترین حلقه کنترلی

حلقه کنترل جریان را به صورت شکل۶ که در آن از کنترل‌کننده PI با ساختار سری استفاده شده‌است در نظر بگیرید.



شکل۵: ساختار کنترل‌کننده پیشنهادی

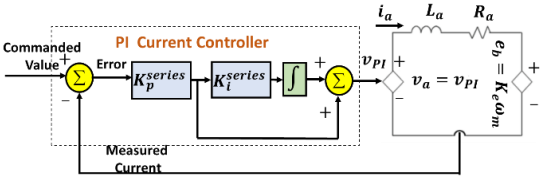
٦- اهداف کنترل

در این مقاله ساختاری کنترلی پیشنهاد می‌شود که در گام اول قادر است پارامترهای مربوط به کنترل‌کننده گشتاور، سرعت و موقعیت موتور DC مغناطیس دائم را به صورتی کاملا خود تنظیم تعیین نماید. علاوه بر آن ساختار پیشنهادی توانایی غلبه بر اصطکاک کولمب که باعث رفتار غیرخطی موتور در سرعت‌های پایین می‌شود را دارا می‌باشد. همچنین شناسایی روی خط امکان دنبال‌کردن تغییرات دینامیکی فرآیند را میسر می‌سازد. در مرحله طراحی و ساخت، ساختار پیشنهادی برروی سخت افزار پیاده‌سازی شده و در نهایت سرودرایوری ساخته شده‌است که قادر است گشتاور، سرعت و موقعیت هر موتور DC دلخواهی را به طور دقیق کنترل نماید. سرودرایور ساخته شده مسیر مطلوب سرعت و موقعیت را از طریق پروتکل ارتباطی شبکه از کامپیوتر دریافت نموده و تمامی فرآیندهای کنترلی در داخل آن انجام می‌شود.

۷ - بررسی کلی ساختار کنترلی پیشنهادی

روش پیشنهادی به منظور کنترل گشتاور، سرعت و موقعیت موتور DC مغناطیس دائم از ساختار کنترل‌کننده سری[1] استفاده می‌کند. در این روش از حلقه کنترل جریان PI با شناسایی مبتنی بر روش HFI و از حلقه کنترل پیش‌بین سرعت و موقعیت با شناسایی روی خط مبتنی‌بر ماشین‌های بردار پشتیبان حداقل مربعات مطابق شکل۵ استفاده می‌شود.

۸ - پیاده سازی حلقه کنترل جریان



شکل۶: حلقه کنترل‌کننده جریان

تابع تبدیل حلقه باز کنترل‌کننده جریان به صورت رابطه (۱۹) می‌باشد:

$$G_{loop}(s) = PI(s) \times \frac{I_a(s)}{V_a(s)} = \left(\frac{K_p^{series} K_i^{series}\left(1+\frac{s}{K_i^{series}}\right)}{s}\right) \times \left(\frac{\frac{1}{R_a}}{1+\frac{L_a}{R_a}s}\right) \quad (۱۹)$$

از آنجایی‌که ولتاژ محرکه الکتریکی e_b وابسته به سرعت روتور (که یک پارامتر مکانیکی و لخت است) می‌باشد، تغییرات آن در مقایسه با جریان سیم‌پیچ (که پارامتری الکتریکی و سریع است) کندتر بوده و در تابع تبدیل جریان به ولتاژ موتور از آن صرف نظر شده‌است. اکنون ضریب K_i^{series} کنترل‌کننده PI جریان را به صورت رابطه (۲۰) در نظر می‌گیریم:

$$K_i^{series} = \frac{R_a}{L_a} \quad (۲۰)$$

به عبارتی K_i^{series} را به گونه‌ای انتخاب می‌کنیم تا صفر کنترلر PI جریان، قطب الکتریکی تابع تبدیل جریان به ولتاژ موتور را خنثی نماید. این امر موجب می‌شود تا تابع تبدیل حلقه بسته کنترل‌کننده جریان یک تابع تبدیل

[1] Cascaded Controller

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

مرتبه اول شده و پایداری حلقه کنترل جریان داخلی تضمین شود. ضریب K_p^{series} نیز به صورت رابطه (۲۱) محاسبه می‌شود:

$$G_{current}(s) = \frac{1}{1 + \frac{L_a}{K_p^{series}} s} \qquad (۲۱)$$

$$\Rightarrow \quad K_p^{series} = L_a \times \text{Bandwidth}$$

طبق روابط (۲۰) و (۲۱) ضرایب کنترل‌کننده جریان با توجه به مقاومت و اندوکتانس سیم‌پیچ آرمیچر تعیین می‌شوند. جهت شناسایی مقاومت و اندوکتانس سیم‌پیچ آرمیچر از روش HFI استفاده می‌کنیم به این صورت که یک ولتاژ کسینوسی با فرکانس ۱۰۰ هرتز و دامنه ۵ ولت و به مدت ۵ ثانیه به موتور اعمال می‌کنیم و سپس با اندازه‌گیری جریان و ولتاژ ترمینال موتور، مقاومت و اندوکتانس سیم‌پیچ آرمیچر را جهت محاسبه ضرایب کنترل‌کننده جریان بدست می‌آوریم. ولتاژ اندازه‌گیری شده ترمینال موتور را به صورت رابطه (۲۲) در نظر بگیرید:

$$v_m = V_{m0} \cos(\omega_0 t + \theta) \qquad (۲۲)$$

فرکانس ولتاژ کسینوسی اعمالی به گونه‌ای انتخاب شده‌است که تنها پارامترهای الکتریکی موتور را تحریک نماید و بنابراین شفت موتور به هنگام اعمال ولتاژ کسینوسی با فرکانس ۱۰۰ هرتز حرکت نخواهد کرد و ولتاژ e_b صفر خواهد بود. بنابراین تابع تبدیل جریان به ولتاژ موتور در حالت دائمی سینوسی به صورت رابطه (۲۳) می‌باشد:

$$G(j\omega) = \frac{I_a(j\omega)}{V_a(j\omega)} = \frac{\frac{1}{R_a}}{1 + j\frac{L_a}{R_a}\omega} \qquad (۲۳)$$

بنابراین جریان اندازه‌گیری شده موتور نیز به صورت رابطه (۲۴) خواهد بود:

$$i_m = V_{m0}\,|G(j\omega_0)| cos(\omega_0 t + \theta + \measuredangle G(j\omega_0)) \qquad (۲۴)$$

که در رابطه (۲۴) اندازه و فاز تابع تبدیل جریان به ولتاژ موتور به صورت رابطه (۲۵) می‌باشند:

$$|G(j\omega_0)|^2 = \frac{1}{R_a{}^2(1 + (\frac{L_a\omega_0}{R_a})^2)} \qquad (۲۵\text{-الف})$$

$$\measuredangle G(j\omega_0) = -\tan^{-1}(\frac{L_a\omega_0}{R_a}) \qquad (۲۵\text{-ب})$$

چنانچه ولتاژ ترمینال موتور در رابطه(۲۲) را به صورت رابطه(۲۶) بسط دهیم:

$$v_m = V_{m0}\cos(\omega_0 t + \theta) = V_{m0}\cos(\theta)\cos(\omega_0 t) - V_{m0}\sin(\theta)\sin(\omega_0 t) \qquad (۲۶)$$

اگر در رابطه(۲۶) تعریف کنیم:

$$V_{m0}\cos(\theta) = a \qquad (۲۷\text{-الف})$$

$$-V_{m0}\sin(\theta) = b \qquad (۲۷\text{-ب})$$

داریم:

$$v_m = a\cos(\omega_0 t) + b\sin(\omega_0 t) \qquad (۲۸)$$

حال اگر v_m را در $\cos(\omega_0 t)$ ضرب کرده و از آن روی کل زمان اعمال شکل موج ولتاژ به موتور طبق رابطه(۲۹) انتگرال بگیریم خواهیم داشت:

$$\begin{aligned}
&\int_{t=0}^{T} v_m \cos(\omega_0 t)\, dt \\
&= \int_{t=0}^{T} (a\cos(\omega_0 t) + b\sin(\omega_0 t))\cos(\omega_0 t)\, dt = \\
&\int_{t=0}^{T} (a\cos(\omega_0 t)^2 + b\sin(\omega_0 t)\cos(\omega_0 t))dt \\
&= \int_{t=0}^{T} \left(a\frac{(1+\cos(2\omega_0 t))}{2} + \frac{1}{2}b\sin(2\omega_0 t)\right)dt = \\
&\frac{1}{2}\int_{t=0}^{T} a\,dt + \frac{1}{2}\int_{t=0}^{T} a\cos(2\omega_0 t)\,dt \\
&\qquad + \frac{1}{2}\int_{t=0}^{T} b\sin(2\omega_0 t)\,dt \\
&\qquad = \frac{T}{2}a
\end{aligned} \qquad (۲۹)$$

اکنون اگر v_m را در $\sin(\omega_0 t)$ ضرب کرده و عملیات رابطه (۲۹) را به طور مشابهی تکرار کنیم داریم:

$$\int_{t=0}^{T} v_m \sin(\omega_0 t)\, dt = \frac{T}{2}b \qquad (۳۰)$$

با توجه به روابط (۲۹) و (۳۰) و استفاده از تعاریف رابطه (۲۷) می‌توانیم روابط مهم (۳۱) را بدست آوریم:

$$\begin{aligned}
&\left(\frac{T}{2}a\right)^2 + \left(\frac{T}{2}b\right)^2 \\
&= \left(\frac{T}{2}\right)^2 ((V_{m0}\cos(\theta))^2 + (-V_{m0}\sin(\theta))^2) \\
&= \left(\frac{T}{2}\right)^2 ((V_{m0}\cos(\theta))^2 + (-V_{m0}\sin(\theta))^2) = \left(\frac{T}{2}\right)^2 {V_{m0}}^2
\end{aligned} \qquad (۳۱\text{-الف})$$

$$\frac{(\frac{T}{2}b)}{(\frac{T}{2}a)} = -\tan\theta \qquad (۳۱\text{-ب})$$

در مرحله بعد مراحل روابط(۲۶) تا (۳۱) را برای جریان موتور نیز تکرار می‌کنیم. بنابراین اگر رابطه(۲۴) را به صورت رابطه(۳۲) بسط دهیم:

$$\begin{aligned}
i_m &= V_{m0}\,|G(j\omega_0)|\cos(\omega_0 t + \theta + \measuredangle G(j\omega_0)) \\
&= V_{m0}|G(j\omega_0)|\,cos(\theta + \measuredangle G(j\omega_0))\,cos(\omega_0 t) \\
&\quad -V_{m0}|G(j\omega_0)|sin\,(\theta + \measuredangle G(j\omega_0))sin\,(\omega_0 t)
\end{aligned} \qquad (۳۲)$$

اگر در رابطه(۳۲) تعریف کنیم:

$$V_{m0}|G(j\omega_0)|\cos(\theta + \measuredangle G(j\omega_0)) = c \qquad (۳۳\text{-الف})$$

$$-V_{m0}|G(j\omega_0)|\sin(\theta + \measuredangle G(j\omega_0)) = d \qquad (۳۳\text{-ب})$$

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

داریم:

$$i_m = c\cos(\omega_0 t) + d\sin(\omega_0 t) \qquad (۳۴)$$

چنانچه طرفین رابطه(۳۴) را یک بار در $\cos(\omega_0 t)$ و بار دیگر در $\sin(\omega_0 t)$ ضرب کنیم و از طرفین رابطه روی کل زمان اعمال شکل موج ولتاژ به موتور انتگرال بگیرم خواهیم داشت:

$$\int_{t=0}^{T} i_m \cos(\omega_0 t)\, dt = \frac{T}{2}c \qquad (۳۵\text{-الف})$$

$$\int_{t=0}^{T} i_m \sin(\omega_0 t) dt = \frac{T}{2}d \qquad (۳۵\text{-ب})$$

با توجه به روابط (۳۵) و استفاده از تعاریف رابطه (۳۳) می‌توانیم روابط مهم(۳۶) را بدست آوریم:

$$\left(\frac{T}{2}c\right)^2 + \left(\frac{T}{2}d\right)^2 = \left(\frac{T}{2}\right)^2 \Big(\big(V_{m0}|G(j\omega_0)|\cos(\theta + \measuredangle G(j\omega_0))\big)^2 + \big(-V_{m0}|G(j\omega_0)|\sin(\theta + \measuredangle G(j\omega_0))\big)^2\Big) = \left(\frac{T}{2}\right)^2 {V_{m0}}^2 |G(j\omega_0)|^2 \qquad (۳۶\text{-الف})$$

$$\frac{(\frac{T}{2}d)}{(\frac{T}{2}c)} = -\tan(\theta + \measuredangle G(j\omega_0)) \qquad (۳۶\text{-ب})$$

با تقسیم رابطه (۳۶-الف) بر رابطه (۳۱-الف) و استفاده از رابطه (۲۵-الف) داریم:

$$\frac{\left(\frac{T}{2}c\right)^2 + \left(\frac{T}{2}d\right)^2}{\left(\frac{T}{2}a\right)^2 + \left(\frac{T}{2}b\right)^2} = |G(j\omega_0)|^2 = \frac{1}{{R_a}^2\left(1 + (\frac{L_a\omega_0}{R_a})^2\right)} \qquad (۳۷)$$

در رابطه(۳۷) مقادیر $\frac{T}{2}a$، $\frac{T}{2}b$، $\frac{T}{2}c$ و $\frac{T}{2}d$ معلوم می‌باشند. در رابطه (۳۷) تعریف می‌کنیم:

$$\frac{\left(\frac{T}{2}c\right)^2 + \left(\frac{T}{2}d\right)^2}{\left(\frac{T}{2}a\right)^2 + \left(\frac{T}{2}b\right)^2} = M \quad , \quad \frac{L_a\omega_0}{R_a} = X \qquad (۳۸)$$

بنابراین داریم:

$$\mathrm{R_a} = \sqrt{\frac{1}{\mathrm{M}(1+\mathrm{X}^2)}} \qquad (۳۹)$$

همچنین چنانچه رابطه(۳۶-ب) را براساس قواعد مثلثاتی بسط داده و از روابط (۲۵-ب) و (۳۱-ب) استفاده کنیم داریم:

$$\frac{\left(\frac{\mathrm{T}}{2}\mathrm{d}\right)}{\left(\frac{\mathrm{T}}{2}\mathrm{c}\right)} = -\tan\left(\theta + \measuredangle \mathrm{G}(\mathrm{j}\omega_0)\right) = -\frac{\tan(\theta)+\tan\left(\measuredangle \mathrm{G}(\mathrm{j}\omega_0)\right)}{1-\tan(\theta)\tan\left(\measuredangle \mathrm{G}(\mathrm{j}\omega_0)\right)} = -\frac{-\frac{\left(\frac{\mathrm{T}}{2}\mathrm{b}\right)}{\left(\frac{\mathrm{T}}{2}\mathrm{a}\right)} - \frac{\mathrm{L_a}\omega_0}{\mathrm{R_a}}}{1-\left(-\frac{\left(\frac{\mathrm{T}}{2}\mathrm{b}\right)}{\left(\frac{\mathrm{T}}{2}\mathrm{a}\right)}\right)\left(-\frac{\mathrm{L_a}\omega_0}{\mathrm{R_a}}\right)} \qquad (۴۰)$$

در رابطه (۴۰) مقادیر $\frac{\mathrm{T}}{2}\mathrm{a}$ ، $\frac{\mathrm{T}}{2}\mathrm{b}$ ، $\frac{T}{2}c$ و $\frac{T}{2}d$ معلوم می‌باشند. چنانچه در رابطه(۴۰) تعریف کنیم:

$$\frac{(\frac{T}{2}d)}{(\frac{T}{2}c)} = A \quad , \quad \frac{(\frac{T}{2}b)}{(\frac{T}{2}a)} = B \quad , \quad \frac{L_a\omega_0}{R_a} = X \qquad (۴۱)$$

با جایگذاری تعاریف رابطه(۴۱) در رابطه(۴۰) و حل آن داریم:

$$\mathrm{X} = \frac{\mathrm{A}-\mathrm{B}}{1+\mathrm{AB}} \qquad (۴۲)$$

با قرار دادن مقدار X در رابطه(۳۹) مقاومت سیم‌پیچ روتور محاسبه خواهد شد. و در نهایت اندوکتانس سیم‌پیچ روتور را می‌توان به صورت رابطه (۴۳) بدست آورد:

$$\frac{\mathrm{L_a}\omega_0}{\mathrm{R_a}} = \mathrm{X} \quad \Longrightarrow \quad \mathrm{L_a} = \frac{\mathrm{R_a X}}{\omega_0} \qquad (۴۳)$$

بنابراین با استفاده از روابط (۳۹) و (۴۳) به ترتیب مقاومت و اندوکتانس سیم پیچ موتور را شناسایی کرده و با در نظر گرفتن پهنای باند مطلوب کنترل‌کننده جریان به اندازه ۱ کیلوهرتز، با استفاده از روابط (۲۰) و (۲۱) ضرایب کنترل‌کننده جریان را محاسبه کرده و حلقه کنترل جریان را می‌بندیم.

۹- اعمال ورودی جریان مناسب و آموزش آفلاین شبکه عصبی

به منظور پیاده‌سازی کنترل‌کننده پیش‌بین سرعت و موقعیت بایستی مدل فرآیند را شناسایی نماییم. از آنجایی‌که حلقه کنترل جریان را در مرحله قبل بسته‌ایم بنابراین فرآیند را می‌توانیم به صورت شکل۷ در نظر بگیریم. با توجه به شکل۷ تابع تبدیل سرعت به جریان و موقعیت به جریان موتور به ترتیب به صورت روابط (۴۴) و (۴۵) حاصل می‌شوند:

$$\frac{\omega_{\mathrm{m}}(\mathrm{s})}{\mathrm{I_c}(\mathrm{s})} = \frac{1}{1+\frac{s}{BW_c}} \times K_m \times \frac{1}{\mathrm{Js}+\mathrm{B}} = \frac{K_m}{(1+\frac{s}{BW_c})(\mathrm{Js}+\mathrm{B})} \qquad (۴۴)$$

$$\frac{\theta_{\mathrm{m}}(\mathrm{s})}{\mathrm{I_c}(\mathrm{s})} = \frac{K_m}{s(1+\frac{s}{BW_c})(\mathrm{Js}+\mathrm{B})} \qquad (۴۵)$$

جهت آموزش شبکه عصبی LS-SVM یک پالس مربعی جریان با دامنه ۰/۵ آمپر به فرآیند شکل۷ اعمال کرده و اطلاعات مربوط به سرعت و موقعیت موتور را ذخیره می‌کنیم (مجموعه آموزش) و از آنها جهت آموزش شبکه عصبی در نرم‌افزار متلب بهره خواهیم گرفت.

۱۰- پیاده سازی حلقه کنترل پیش‌بین سرعت

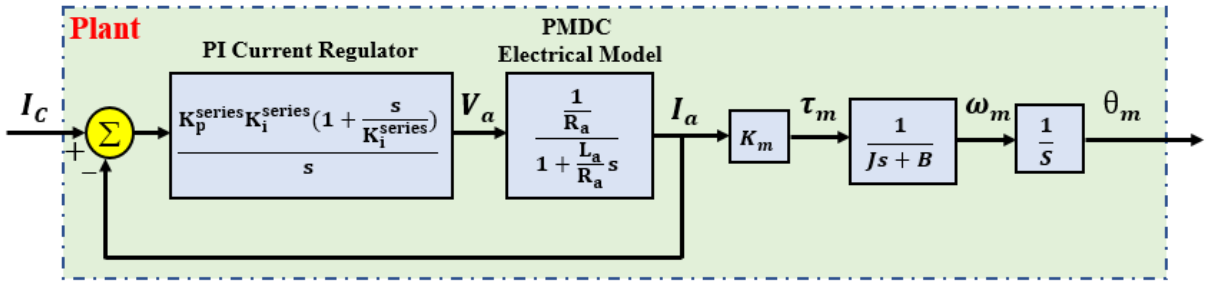
به منظور پیاده‌سازی کنترل‌کننده پیش‌بین سرعت و موقعیت بایستی مدل خطی مناسبی را از مدل شبکه عصبی LS-SVM استخراج نماییم. نحوه استخراج مدل خطی ARMA از مدل LS-SVM را در بخش ۳ مطرح کردیم. جهت استخراج مدل ARMA بایستی بردار ورودی رابطه (۱۵) را از روی معادل گسسته تابع تبدیل سرعت به جریان فرآیند تشکیل دهیم.

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

چنانچه معادل گسسته تابع تبدیل سرعت به جریان رابطه (۴۴) را با استفاده از روش نگهدارنده مرتبه صفر[1] بدست آوریم خواهیم داشت:

$$\frac{\omega_m(z^{-1})}{I_c(z^{-1})}=\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \qquad (۴۶)$$

نظر می‌گیریم تا بتوانیم از یک تابع معکوس‌سازی واحد در میکروکنترلر استفاده کنیم.



شکل۷: مدل فرآیند پس از بستن حلقه کنترل جریان

در مدل رابطه (۴۶) ثوابت b_1، b_2، a_1 و a_2 پارامترهای فرآیند می‌باشند که بایستی از طریق آموزش شبکه عصبی بدست آیند. بردار ورودی شبکه عصبی را به صورت رابطه (۴۷) در نظر می‌گیریم.

$$x(k)=[I_c(k-1),I_c(k-2),\omega_m(k-1),\omega_m(k-2)] \qquad (۴۷)$$

با استفاده از مجموعه آموزش و با توجه به بردار ورودی رابطه (۴۷) کرنل خطی داده‌های مجموعه آموزش را حساب کرده و با حل دستگاه معادلات خطی رابطه (۱۱) مقادیر ضرب‌کننده‌های لاگرانژ و ترم بایاس را بدست می‌آوریم. سپس با استفاده از رابطه (۱۸) چند جمله‌ای‌های A و B و درنتیجه پارامترهای مدل رابطه (۴۶) را محاسبه می‌کنیم. مقادیر پارامترهای مدل رابطه (۴۶) در سه بار آموزش شبکه‌عصبی LS-SVM با استفاده از داده‌های مجموعه آموزش مختلف به صورت جدول۱ می‌باشند.

جدول۱: پارامترهای تابع تبدیل سرعت به جریان

آموزش	a_1	a_2	b_1	b_2
1	-1.2573	0.2572	0.0007654	0.0004897
2	-1.2570	0.2577	0.0007655	0.0004881
3	-1.2567	0.2574	0.0007651	0.0004868

پس از آموزش شبکه عصبی و استخراج پارامترهای جدول۱ مربوط به تابع تبدیل سرعت به جریان در نرم افزار متلب، اطلاعات مدل را از طریق پروتکل شبکه به سرودرایور جهت پیاده‌سازی کنترل‌کننده پیش‌بین سرعت منتقل می‌کنیم. هنگام پیاده‌سازی کنترل‌کننده پیش‌بین سرعت براساس مدل استخراج شده از تربیت آفلاین شبکه عصبی، به ازای افق کنترل و افق پیش-بین ۵ کنترل‌کننده پیش‌بین عملکرد مطلوبی داشت ولی همانطور که در قسمت آنلاین کردن روش شناسایی مطرح خواهیم کرد حداقل طول مجموعه آموزش جهت دنبال‌کردن صحیح تغییرات دینامیکی فرآیند ۱۰ زوج سرعت و جریان می‌باشد. بنابراین افق کنترل و پیش‌بین را نیز ۱۰ در

جهت معکوس کردن یک ماتریس ۱۰×۱۰ ابتدا معکوس یک ماتریس ۵×۵ را در نرم‌افزار Maple به صورت پارامتری حساب کرده و سپس با استفاده از لم معکوس سازی رابطه (۴۸) معکوس ماتریس ۱۰×۱۰ را حساب می‌کنیم.

$$M=\begin{bmatrix}A & B\\ C & D\end{bmatrix}\Rightarrow M^{-1}=\begin{bmatrix}W & X\\ Y & Z\end{bmatrix} \qquad (۴۸)$$

که در رابطه (۴۸) ماتریس‌های W، X، Y و Z به صورت رابطه (۴۹) محاسبه می‌شوند:

$$W=(A-BD^{-1}C)^{-1} \qquad (۴۹\text{-الف})$$

$$Y=-D^{-1}CW \qquad (۴۹\text{-ب})$$

$$Z=(D-CA^{-1}B)^{-1} \qquad (۴۹\text{-پ})$$

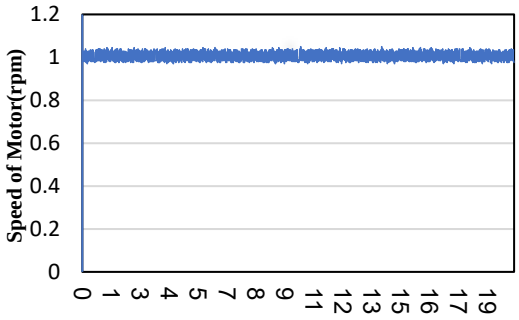
$$X=-A^{-1}BZ \qquad (۴۹\text{-ت})$$

به طورخلاصه نحوه عملکرد کنترل‌کننده پیش‌بین سرعت به این صورت می‌باشد که در هر بار اجرای حلقه کنترلی سیگنال‌های کنترلی تا ۱۰ افق پیش‌رو باتوجه به مدل استخراج شده از آموزش آفلاین شبکه عصبی و ورودی‌ها و خروجی‌های گذشته و مسیر مرجع آینده محاسبه می‌شوند. سپس سیگنال کنترلی در لحظه t به فرآیند اعمال می‌شود. در اجرای بعدی حلقه کنترلی تمامی مراحل فوق مجدد تکرار می‌شوند.

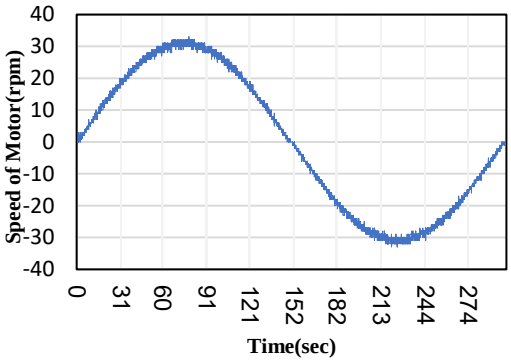
جهت سنجش عملکرد کنترل‌کننده پیش بین سرعت یک ورودی پله یک دور بر دقیقه از طریق نرم افزار متلب و به وسیله پروتکل شبکه به کنترل‌کننده پیش‌بین سرعت طراحی شده در سرودرایور اعمال می‌کنیم. شکل ۸ (۱) سرعت واقعی موتور را نمایش می‌دهد. در سرو درایور ساخته شده حداقل گام تنظیم سرعت یک دور بر دقیقه می باشد که مشاهده می‌کنیم سرو درایور به ورودی پله یک دور بر دقیقه پاسخ مطلوبی داده است. همچنین یک شکل موج سینوسی سرعت با دامنه ۳۰ دور بر دقیقه و فرکانس ۰/۰۰۳۳ هرتز(دوره تناوب ۳۰۰ ثانیه) فرکانس شکل موج اعمالی را بسیار پایین درنظرگرفته‌ایم تا بتوانیم رفتار غیرخطی اصطکاک کولمب را به خوبی نشان دهیم. شکل۸ (۲) سرعت واقعی موتور را نمایش می‌دهد.

[1] Zero Older Hold

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]



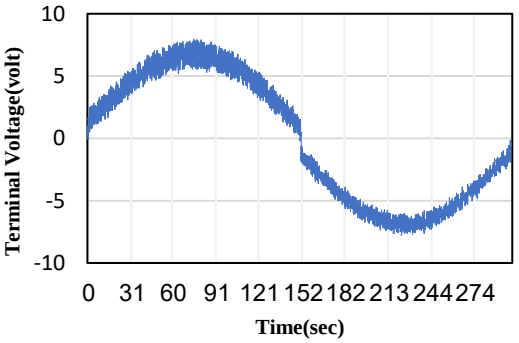
(۱)



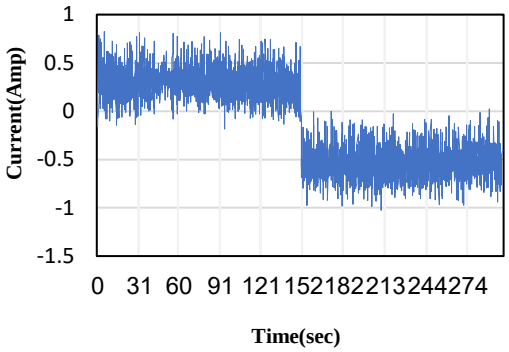
(۲)

شکل۸: پاسخ کنترل‌کننده پیش‌بین سرعت مبتنی بر شناسایی شکبه عصبی LS-SVM به ورودی پله(۱)، ورودی سینوسی سرعت(۲)

از مقایسه شکل۸ (۲) با شکل۱ (۲) مشاهده می‌کنید کنترل‌کننده پیش-بین سرعت به خوبی توانسته است رفتار نامطلوب ناشی از اصطکاک غیرخطی کولمب را جبران نماید به طوریکه موتور مسیر مرجع سرعت را به خوبی حتی در سرعت‌های پایین دنبال کرده است. شکل موج ولتاژ و جریان موتور در کنترل‌کننده پیش‌بین سرعت مبتنی بر مدل LS-SVM در پاسخ به ورودی سینوسی نیز به ترتیب در شکل۹ (۱) و شکل۹ (۲) نشان داده شده است.



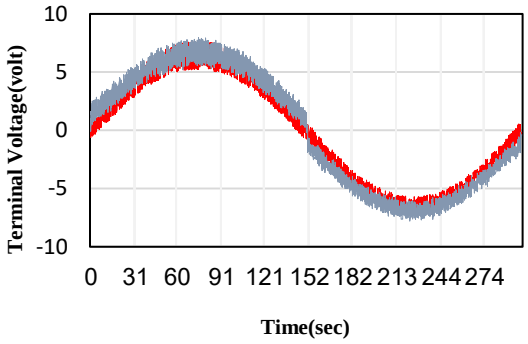
(۱)



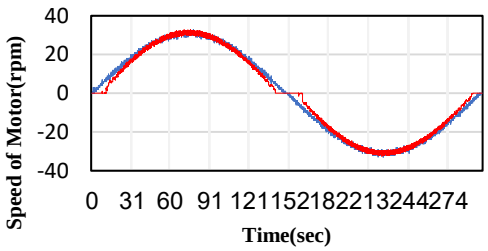
(۲)

شکل ۹ : شکل موج ولتاژ ترمینال موتور(سیگنال کنترل) (۱) ، شکل موج جریان موتور (۲)

در شکل ۱۰ (۱) و شکل ۱۰ (۲) عملکرد کنترل‌کننده پیش‌بین سرعت مبتنی‌بر مدل LS-SVM با کنترل‌کننده سری با حلقه کنترل PI سرعت با یکدیگر مقایسه شده‌اند. کنترل‌کننده پیش‌بین برخلاف کنترل‌کننده PI به خوبی توانسته است رفتار نامطلوب ناشی از اصطکاک کولمب را جبران نماید. همچنین از مقایسه ولتاژ ترمینال موتور(سیگنال کنترلی) در کنترلکننده پیش‌بین با کنترلکننده سری با حلقه کنترل سرعت PI یک تفاوت اساسی وجود دارد و آن رفتار متفاوت ولتاژ ترمینال موتور(سیگنال کنترلی) به هنگام نزدیک شدن به ناحیه مرده است که نشان‌دهنده تلاش کنترل‌کننده پیش‌بین به منظور غلبه‌بر اصطکاک غیرخطی کولمب می‌باشد.



(۱)



(۲)

شکل ۱۰ : مقایسه ولتاژ ترمینال(سیگنال کنترلی) کنترل‌کننده پیش بین سرعت مبتنی بر مدل LS-SVM با کنترل‌کننده سری با حلقه کنترل سرعت PI (۱) مقایسه عملکرد کنترل‌کننده پیش بین سرعت مبتنی بر مدل LS-SVM با کنترل‌کننده سری با حلقه کنترل سرعت PI (۲)

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

جدول۲ : پارامترهای تابع تبدیل موقعیت به جریان

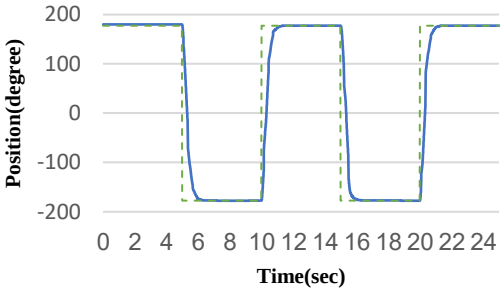
آموزش	a_1	a_2	a_3	b_1	b_2	b_3
1	-2.1941	1.5140	-0.2572	-1.3871×10^{-7}	-4.0867×10^{-7}	-7.0520×10^{-8}
2	-2.2186	1.4719	-0.2125	-1.2307×10^{-7}	-4.1493×10^{-7}	-6.9871×10^{-8}
3	-2.1875	1.3927	-0.2394	-1.2967×10^{-7}	-4.1085×10^{-7}	-7.1174×10^{-8}

۱۱ - پیاده سازی حلقه کنترل پیش‌بین موقعیت

پس از طراحی حلقه کنترل سرعت بایستی حلقه کنترل موقعیت را طراحی و پیاده‌سازی کنیم. روند مشابه بستن حلقه کنترل سرعت می‌باشد با این تفاوت که خروجی کنترل‌کننده موقعیت به عنوان مسیر سرعت مرجع به کنترل‌کننده سرعت اعمال خواهد شد(شکل۵). معادل گسسته تابع تبدیل موقعیت به جریان رابطه (۴۴) با استفاده از روش نگهدارنده مرتبه صفر به صورت رابطه (۴۹) می‌باشد:

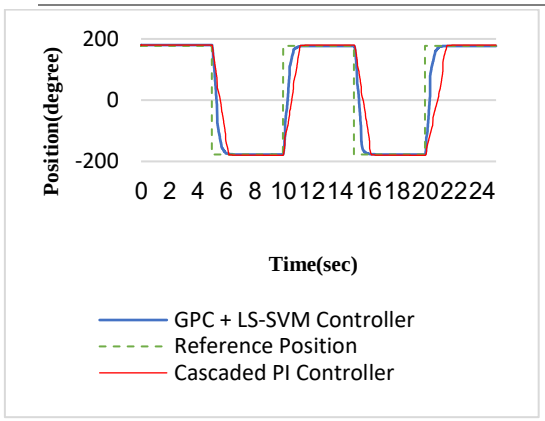
$$\frac{\theta_m(z^{-1})}{I_c(z^{-1})}=\frac{b_1z^{-1}+b_2z^{-2}+b_3z^{-3}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}} \qquad (۴۹)$$

پس از آموزش آفلاین شبکه عصبی LS-SVM با استفاده از کرنل خطی، پارامترهای مدل رابطه (۴۹) به صورت جدول۲ شناسایی شدند. با استفاده از پارامترهای شناسایی شده جدول۲ کنترل‌کننده پیش‌بین موقعیت را با افق پیش‌بین و افق کنترل ۱۰ می‌بندیم. جهت سنجش عملکرد کنترلکننده پیش‌بین موقعیت یک مسیر مرجع موقعیت مربعی با نرم افزار متلب و از طریق کابل شبکه به کنترل‌کننده طراحی شده برروی سرودرایور اعمال می‌کنیم. شکل۱۱ پاسخ کنترل‌کننده را نشان می‌دهد.

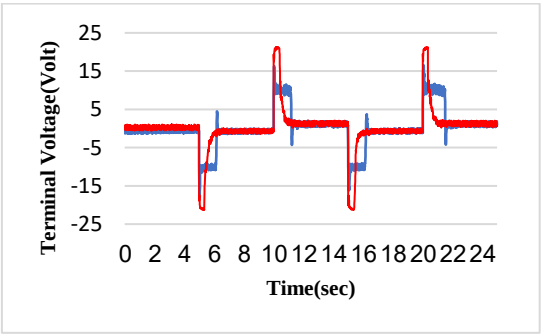


شکل۱۱ : پاسخ کنترل‌کننده پیش بین موقعیت مبتنی بر مدل شبکه عصبی

در شکل۱۲ (۱) عملکرد عملکرد کنترل‌کننده پیش‌بین موقعیت سری مبتنی بر مدل LS-SVM با کنترلر سری با حلقه‌های کنترل PI با یکدیگر مقایسه شده اند. زمان صعود پاسخ کنترل‌کننده پیش‌بین موقعیت کمتر بوده و با رفتار مناسب‌تری به حالت ماندگار خود می‌رسد. همچنین خطای حالت ماندگار که در درجه اول ناشی از اصطکاک کولمب می‌باشد در کنترل-کننده پیش‌بین موقعیت کمتر از کنترل‌کننده سری با حلقه‌های کنترل PI می‌باشد. در شکل۱۲ (۲) نیز ولتاژ ترمینال موتور(سیگنال کنترلی) کنترل-کننده پیش‌بین سری موقعیت مبتنی بر مدل LS-SVM با کنترل‌کننده سری موقعیت با حلقه‌های کنترل PI با یکدیگر مقایسه شده‌اند.



(۱)



(۲)

شکل۱۲ : مقایسه عملکرد کنترل‌کننده پیش‌بین موقعیت سری مبتنی بر مدل LS-SVM با کنترل‌کننده سری با حلقه‌های کنترل PI (۱)
مقایسه ولتاژ ترمینال موتور (۲)

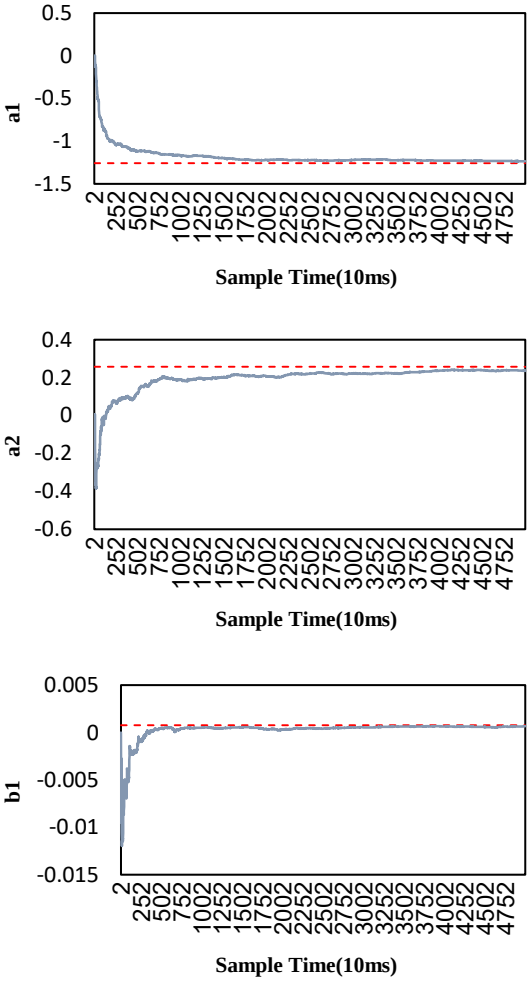
۱۲ - آنلاین کردن شناسایی به منظور تطبیق پذیر کردن کنترل‌کننده پیش بین موقعیت و سرعت

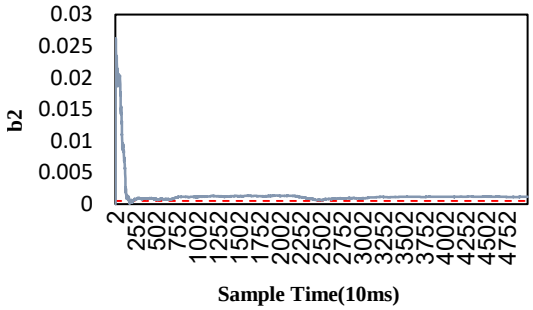
در قسمت‌های قبل ابتدا با اعمال یک ورودی جریان مناسب و ذخیره-سازی داده‌های سرعت و موقعیت موتور، یک شبکه عصبی LS-SVM با هسته‌های خطی را در نرم‌افزار متلب به صورت آفلاین تربیت کردیم و سپس با ارسال مدل شناسایی شده به سرودرایور کنترل‌کننده پیش‌بین سرعت و موقعیت را پیاده سازی کردیم.

به منظور قابلیت دنبال‌کردن تغییرات دینامیکی فرآیند بایستی روش شناسایی را آنلاین کنیم. برای این منظور اولا بایستی طول مجموعه آموزش را به اندازه‌ای در نظر بگیریم تا در هر بار اجرای حلقه کنترلی فرآیند آموزش کامل شود و ثانیا داده‌های مجموعه آموزش را در هر بار اجرای حلقه کنترلی به‌روزرسانی کرده تا کنترل‌کننده بتواند تغییرات دینامیکی فرآیند را دنبال نماید. با توجه به محدودیت سرعت

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

میکروکنترلر(۱۶۰ مگاهرتز) حداکثر طول مجموعه آموزش قابل پیاده-سازی برروی آن ۱۰ بدست آمد. همچنین جهت به روز رسانی مجموعه آموش در هر بار اجرای حلقه کنترل قدیمی‌ترین زوج آموزش را حذف کرده و جدیدترین زوج آموزش را به مجموعه آموزش اضافه می‌کنیم. پیاده‌سازی روش شناسایی آنلاین به این صورت می‌باشد که همزمان با پیاده‌سازی کنترل‌کننده پیش‌بین سرعت براساس مدل شناسایی شده آفلاین، روش آنلاین در داخل سرودرایور اجرا می‌شود. زمانی که اختلاف پارامترهای شناسایی شده از روش آنلاین با روش آفلاین به زیر ۲۰ درصد رسید، کنترل‌کننده پیش‌بین براساس مدل شناسایی شده آنلاین عمل خواهد کرد. روند همگرایی پارامترهای تابع تبدیل سرعت به جریان مدل آنلاین به مدل آفلاین به صورت شکل ۱۳ می‌باشد.





شکل۱۳ همگرایی پارامترهای مدل آنلاین به آفلاین تابع تبدیل سرعت به جریان

در نرم افزار متلب هر ۱۰ میلی ثانیه مقدار پارامترهای مدل را از طریق کابل شبکه می‌خوانیم. مطابق شکل ۱۳ روند شناسایی پارامترها در روش آنلاین کند می‌باشد و حدود ۱۰ ثانیه طول می‌کشد. دلیل این امر کم بودن طول مجموعه آموزش می‌باشد. مسلما با افزایش طول مجموعه آموزش سرعت شناسایی آنلاین بیشتر خواهد شد.

۱۱ - نتیجه‌گیری

در این مقاله کنترل‌کننده پیش‌بین موقعیت و سرعت مبتنی بر مدل شبکه عصبی ماشین‌های بردار پشتیبان حداقل مربعات را با ساختار سری طراحی کرده و برروی سرودرایور پیاده‌سازی کردیم. در این روش حلقه کنترل گشتاور را براساس روش شناسایی HFI و حلقه‌های کنترل موقعیت و سرعت را براساس مدل شبکه عصبی بستیم. مزیت بزرگ این روش این است که تمامی حلقه‌های کنترلی به صورت خودکار و براساس مدل فرآیند طراحی شدند. علاوه براین روش پیشنهادی به خوبی توانست اثر مخرب ناشی از اصطکاک غیرخطی کولمب را در سرعت‌های پایین و حالت ماندگار جبران نماید. همچنین روش شناسایی آنلاین مطرح شده توانایی دنبال‌کردن تغییرات دینامیکی کند در فرآیند را دارا می‌باشد و با بهبود سرعت پردازش میکروکنترلر و طول مجموعه آموزش می‌توان سرعت شناسایی را افزایش داد.

مراجع

[1] B. Armstrong, B. Dupont, and C.D. Wit, “A survey of models, analysis tools and compensation methods for the control of machines with friction,” Automatica, vol. 30, no. 7, pp. 1083-1138, 1994.

[2] Guanrong Chen, Ying Chen, and H. Ogmen, “Identifying chaotic systems via a Wiener-type cascade model, IEEE Control Systems, vol. 17, no. 5, pp. 29-36, 1997.

[3] Ming Xu, Guanrong Chen, and Yan-TaoTian, “Identifying chaotic systems using Wiener and Hammerstein cascade models,” Mathematical and Computer Modelling, vol. 33, no. 4-5, pp. 483-493, 2001.

[4] Norquay SJ, Palazoglu A, Romagnoli JA, “Model predictive control based on Wiener

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

models," Chemical Engineering Science, vol. 53, no. 1, pp. 75-84, 1998.

[5] H. Al-Duwaish, M.N. Karim, V. Chandrasekar, "Use of multilayer feedforward neural networks in identification and control of Wiener model," IEE Proceedings Control Theory and Applications, vol. 143, no. 3, pp. 255-258, 1996.

[6] A. anczak, "Neural network approach for identification of Hammerstein systems," International Journal of Control, vol. 76, no. 17, pp. 1749–1766, 2003.

[7] M.A. Mahini, M. Teshnehlab, and M. A. khanehsar, "Nonlinear System Identifcation Using Hammerstein-Wiener Neural Network and subspace algorithms," Journal of Advances in Computer Engineering and Technology, vol. 1, no. 3, 2015.

[8] Mingyong Cui, Haifang Liu, Zhonghui Li, Yinggan Tang, and Xinping Guan , "Identification of Hammerstein model using functional link artificial neural network," Elsevier Science Publishers Neurocomputing, vol. 142, pp. 419-428, 2014.

[9] T. Kara, I. Eker, "Nonlinear modeling and identification of a DC motor for bidirectional operation with real time experiments," Energy Conversion and Management, vol. 45, no. 7-8, pp. 1087-1106, 2004.

[10] J. Peng, R. Dubay, "Identification and adaptive neural network control of a DC motor system with dead-zone characteristics," ISA Transactions, vol. 50, no. 4, pp. 588-598, 2011.

[11] S. Bennett, "Development of the PID controller," IEEE Control Systems Magazine vol. 13, no. 6, pp. 58-62, 1993.

[12] M.N. Howell, T.J. Gordon, M.C. Best, "The application of continuous action reinforcement learning automata to adaptive PID tuning," IEEE Seminar on learning systems for control, 2000.

[13] Chun-Fei Hsu, Bore-Kuen Lee, "FPGA-based adaptive PID control of a DC motor driver via sliding-mode approach," Expert Systems with Applications, vol. 38, no. 9, pp. 11866-11872, 2011.

[14] Ruben Tapia-Olvera, Francisco Beltran-Carbajal, Omar Aguilar-Mejia, and Antonio Valderrabano-Gonzalez, "An Adaptive Speed Control Approach for DC Shunt Motors," Energies, 2016.

[15] A.Aziz Khater, Mohammad El-Bardini, and Nabila M. El-Rabaie, "Embedded Adaptive Fuzzy Controller Based on Reinforcement Learning for DC Motor with Flexible Shaft," Arabian Journal for Science and Engineering, vol. 40, no. 8, pp. 2389-2406, 2015.

[16] F. Cameron, D.E. Seborg, "A self-tuning controller with a PID structure," International Journal of Control, vol. 38, no. 2, pp. 401–17, 1983.

[17] Reza Akbari Hasanjani, Shahram Javadi, Reza Sabbaghi Nadooshan, "DC motor speed control by self-tuning fuzzy PID algorithm, " Transactions of the Institute of Mesearment and Control, vol. 37, no. 2, 2015.

[18] P. Vega, C. Prada, V. Aleixander, "Self-tuning predictive PID controller," IEE Proceedings, Control Theory and Applications, vol. 138, no. 3, pp. 303–311, 1991.

[19] F.G. Martins, A.N. Coelho, "Application of feed-forward artificial neural networks to improve process control of PID-based control algorithms," Computers and Chemical Engineering, vol. 24, no. 2-7, pp. 853–858, 2000.

[20] J. Chen, T.C. Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control," Journal of Process Control, vol. 14, no. 2, pp. 211–230, 2004.

[21] X.F. Yuan, Y.N. Wang, "Neural networks based self-learning PID control of electronic throttle," Nonlinear Dynamics, vol. 55, no. 4, pp. 385–393, 2009.

[22] Xing-Song Wang, Chun-Yi Su, Henry Hong, "Robust adaptive control of a class of nonlinear systems with unknown dead-zone," Automatica, vol. 40, no. 3, pp. 407–413, 2004.

[23] W. Zhonghua, Y. Bo, C. Lin, Z. Shusheng, "Robust adaptive deadzone compensation of DC servo system," IEE Proceedings - Control Theory and Applications, vol. 153, no. 6, pp. 709-713, 2006.

[24] J. Zhou, C. Wen, Y. Zhang, "Adaptive output control of nonlinear systems with uncertain deadzone nonlinearity," IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 504–511, 2006.

[25] S. Ibrir, W.F. Xie, C.Y. Su, "Adaptive tracking of nonlinear systems with nonsymmetric deadzone input," Automatica, vol. 43, no. 3, pp. 522–530, 2007.

[26] Seong Ik Han, Kwon Soon Lee, Min Gyu Park, and Jang Myung Lee, "Robust adaptive deadzone and friction compensation of robot manipulator using RWCMAC network," Journal of Mechanical Science and

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]

Technology, vol. 25, no. 6, pp. 1583–1594, 2011.

[27] Jianyong Yao, Zongxia Jiao, and Dawei Ma, "Adaptive Robust Control of DC Motors with Extended State Observer," IEEE Transactions on Industrial Electronics, vol. 61, no. 7, pp. 3630–3637, 2014.

[28] J.O. Jang, G.J. Jeon, "A parallel neuro-controller for DC motors containing nonlinear friction," Neurocomputing, vol. 30, no. 1–4, pp. 233–248, 2000.

[29] R.R. Selmic, F.L. Lewis, "Deadzone compensation in motion control systems using neural networks," IEEE Transactions on Automatic Control, 2000.

[30] T.P. Zhang, S.S. Ge, "Adaptive neural control of MIMO nonlinear state time-varying delay systems with unknown dead-zones and gain signs," Automatica, vol. 43, no. 6, pp. 1021–1033, 2007.

[31] Lei Liu, Yan-Jun Liu, and C.L. Philip Chen, "Adaptive Neural Network Control for a DC Motor System with Dead-Zone," Nonlinear Dynamics, vol. 72, no. 1–2, pp. 141–147, 2013.

[32] Liping Fan, Yi Liu, "Fuzzy Self-Tuning PID Control of the Main Drive System for Four-High Hot Rolling Mill," Journal of Advanced Manufacturing Systems, vol. 14, no. 1, pp. 11-22, 2015.

[33] S.Y. Oh, D.J. Park, "Design of new adaptive fuzzy logic controller for nonlinear plants with unknown or time-varying dead zones," IEEE Transactions on Fuzzy Systems, vol. 6, no. 4, pp. 482–491, 1998.

[34] F.L. Lewis, W.K. Tim, L.Z. Wang, and Z.X. Li, "Deadzone compensation in motion control systems using adaptive fuzzy logic control," IEEE Transactions on Control Systems Technology, vol. 7, no. 6, pp.731–742, 1999.

[35] Luting Miao, Yuxin Sun, Huangqiu Zhu, AND Xianxing Liu, "Decoupling Control of Bearingless Induction Motor Based on LS-SVM Inverse System," Mechanics and Materials, vol. 703, pp. 331–334, 2015.

[36] Hossam Mohammad Khalil, and Mohamad El Bardini, "Support Vector Machines Based Adaptive Controller for Piston Hydraulic Motor," International Journal of Control and Automation, vol. 4, no. 3, 2011.

[37] Stefan Tötterman, Hannu T. Toivonen, "Support vector method for identification of Wiener models," Journal of Process Control, vol. 19, no. 7, pp. 1174–1181, 2009.

[38] I.B. Tijani, Rini Akmeliawati, "Support vector regression-based friction modeling and compensation in motion control system," Engineering Applications of Artificial Intelligence, vol. 25, no. 5, pp. 1043–1052, 2012.

[39] Serdar Iplikci, "A support vector machine-based control application to the experimental three tank system," ISA Transactions, vol. 49, no. 3, pp. 376–386, 2010.

[40] Vincent Laurain, Roland Tóth, Dario Piga, and Wei Xing Zheng, "An instrumental least squares support vector machine for nonlinear system identification," Automatica, vol. 54, pp. 340–347, 2015.

[41] He Yanzhao, Zheng Shiqiang, and Fang Jiancheng, "Start-up current adaptive control for sensorless high-speed brushless DC motors based on inverse system method and internal mode controller," Chinese Journal of Aeronautics, vol. 28, 2016.

[42] Johan Suykens, "Least Squares Support Vector Machines, " NATO-ASI Learning Theory and Practice Leuven July 2002.

[43] Ruchika, Neha Raghu, "Model Predictive Control: History and Development," International Journal of Engineering Trends and Technology(IJETT), vol. 4, no. 6, pp. 2600–2602, 2013.

[44] D.W. Clarke, "Generalized predictive control," Automatica, vol. 23, no. 2, pp. 137–148, 1987.

[Downloaded from joc.kntu.ac.ir on 2026-11-07]
[DOR: 20.1001.1.20088345.1398.13.2.7.0]
[DOI: 10.29252/joc.13.2.53]