

شبیه‌سازی و پیاده‌سازی آزمایشگاهی کنترل کننده سرعت موتور BLDC با مراجع دینامیکی، مبتنی بر بُرد ARDUINO MEGA

احمد انتظاری، آرش دهستانی کلاگر و محمدرضا علیزاده پهلوانی

مخلوط کننده، ششوار، دوربین فیلمبرداری، پنکه خانگی، تهویه مطبوع و یخچال کاملاً گسترده است. علاوه بر این، موتورهای BLDC در صنعت، عمدتاً در زمینه مهندسی ساخت یا طراحی اتوماسیون صنعتی به کار می‌روند. در تولید، موتورهای DC بدون جاروبک عمدتاً در سیستم‌های کنترل حرکت، موقعیت‌یابی یا محرک‌ها استفاده می‌شوند.

رایج‌ترین کاربردهای صنعتی موتورهای DC بدون جاروبک، شامل موتورهای خطی، سروموتورها، محرک برای ربات‌های صنعتی، موتورهای محرک اکسترودر و کنترل ابزارهای CNC می‌باشند. کاربردهای معمول موتورهای BLDC در هوافضا شامل کنترل بازوی رباتیک، ژيروسکوپ و راه‌انداز سکان^۱ و غیره می‌باشد. این کاربردها عموماً به کنترل با دقت بالا و پاسخ دینامیکی سریع نیاز دارند؛ بنابراین سیستم‌های مربوطه با بازخورد سرعت حلقه بسته کنترل می‌شوند و اکثر آنها از الگوریتم‌های پیشرفته کنترل فرایند استفاده می‌کنند.

موتورهای DC بدون جاروبک به یک انتخاب موتوری محبوب برای هواپیماهای مدل تبدیل شده‌اند که شامل بالگردها و پهپادها می‌شود. نسبت نیرو به وزن و طیف وسیع اندازه‌های موجود، بازار پرواز مدل‌های برقی را متحول کرده است و تقریباً جایگزین تمام موتورهای DC معمولی شده‌اند. کموتاسیون مکانیکی با روش کموتاسیون الکترونیکی در این موتور جایگزین شده است. برای کموتاسیون الکترونیکی از ترانزیستورهای قدرت استفاده می‌شود. ترانزیستورهای قدرت با توجه به موقعیت روتور، روشن و خاموش می‌شوند و باعث چرخش موتور می‌گردند [۱]. در این نوع کموتاسیون از حسگرهای موقعیت برای به دست آوردن موقعیت روتور استفاده می‌شود. یک موتور سه‌فاز BLDC به اطلاعاتی در مورد موقعیت روتور در هر ۶۰ درجه الکتریکی نیاز دارد و از این رو از سه حسگر اثر هال استفاده می‌شود. شکل موج ولتاژ ضدمحرکه یک موتور BLDC در حالت ایده‌آل به صورت ولتاژ ضدمحرکه دوزنقه‌ای می‌باشد [۲]. مقالات زیادی برای کنترل سرعت موتور BLDC ارائه شده است. در [۱] راه‌انداز BLDC بر اساس مدل ریاضی موتور معرفی شده و سخت‌افزار و نرم‌افزار راه‌انداز تحلیل گردیده است. مدار اصلی از یک پردازنده سیگنال دیجیتال (DSP) تشکیل شده است. روش کنترل سرعت BLDC در این مقاله، استراتژی کنترلی جدیدی را با استفاده از مقدار از پیش برچسب‌گذاری شده سرعت متغیر، بیان می‌کند تا سرعت چرخش را با دقت در محدوده بزرگی تشخیص دهد. این مقاله یک استراتژی کنترلی تضعیف شار را در سرعت بالا برای BLDC تحلیل می‌کند. در [۳] به موتورهای DC بدون جاروبک (BLDC) پرداخته می‌شود که در منطقه با سرعت بالا و بدون هیچ حسگر سرعت یا موقعیت کار می‌کنند. یک تکنیک کموتاسیون

چکیده: موتورهای BLDC به دلیل کارایی، قابلیت اطمینان و قابلیت کنترل در صنایع و زمینه‌های مختلف کاربرد دارند. از جمله این کاربردها می‌توان به وسایل نقلیه الکتریکی و هیبریدی، سیستم‌های رباتیک و اتوماسیون، لوازم الکترونیکی خانگی، صنایع هوافضا، تجهیزات پزشکی، توربین‌های بادی و سیستم‌های خورشیدی اشاره نمود. در بسیاری از این کاربردها به کنترل سرعت موتور به صورت دینامیکی نیاز می‌باشد. در این مقاله با توجه به اهمیت موضوع در ابتدا به کمک نرم‌افزار PLECS و با استفاده از روش PWM و کنترل کننده PID، شبیه‌سازی کنترل سرعت دینامیکی موتور BLDC با مراجع سرعت دینامیکی انجام گرفته و سپس پیاده‌سازی آزمایشگاهی راه‌انداز موتور BLDC با استفاده از بُرد میکروکنترلر Arduino Mega در شرایط تغییرات دینامیکی مراجع سرعت صورت پذیرفته و نتایج متناظر ارائه شده است. جهت نمایش و ثبت داده‌های سرعت در رایانه از نرم‌افزار Labview استفاده شده است. نتایج شبیه‌سازی و آزمایشگاهی نشان می‌دهند که موتور BLDC در ردیابی مراجع سرعت دینامیکی تا فرکانس دینامیکی ۳/۰ هرتز دارای کارایی مطلوب می‌باشد.

کلیدواژه: موتور BLDC، کموتاسیون الکترونیکی، کنترل سرعت دینامیکی، برد Arduino Mega.

۱- مقدمه

موتور DC بدون جاروبک (BLDC) با توجه به نبود جاروبک و کموتاتور و چگالی توان بالا، دارای راندمان بالا و پاسخ دینامیکی مناسبی است. این موتور دارای ساختار ساده، قابلیت اطمینان بالا، نویز کم، مشخصه سرعت-گشتاور بهتر و نگهداری راحت‌تر است؛ بنابراین به طور گسترده در صنایع مختلف استفاده می‌شود. برخی از کاربردهای موتورهای DC بدون جاروبک بدین شرح می‌باشد [۱]: با توجه به ساختار ساده موتور BLDC و عملکرد پایدار و تنظیمی سرعت به صورت مناسب، این موتور به طور فزاینده‌ای در زمینه خودرو کاربرد دارند. این موتور علاوه بر اینکه جزء اصلی راه‌انداز خودروهای الکتریکی است، می‌تواند برای سیستم تهویه مطبوع خودرو، برف‌پاک‌کن، درب‌های برقی، کیسه هوا، صندلی‌های برقی و غیره نیز استفاده شود. فناوری استفاده از این موتورها در لوازم خانگی مانند DVD، VCD، دوچرخه برقی، جاروبرقی،

این مقاله در تاریخ ۱۰ تیر ماه ۱۴۰۳ دریافت و در تاریخ ۱۶ مهر ماه ۱۴۰۳ بازنگری شد.

احمد انتظاری، مجتمع دانشگاهی برق و کامپیوتر، دانشگاه صنعتی مالک اشتر، تهران، ایران، (email: Ahmad_entezari6@yahoo.com).

آرش دهستانی کلاگر (نویسنده مسئول)، مجتمع دانشگاهی برق و کامپیوتر، دانشگاه صنعتی مالک اشتر، تهران، ایران، (email: a_dehestani@mut.ac.ir).

محمدرضا علیزاده پهلوانی، مجتمع دانشگاهی برق و کامپیوتر، دانشگاه صنعتی مالک اشتر، تهران، ایران، (email: mr_alizadehp@mut.ac.ir).

جریان بر اساس تولید جریان شبه‌مربعی برای سه فاز موتور به کار می‌رود. برخلاف روش کنترل برداری که پیاده‌سازی آن پیچیده است، این تکنیک ساده، جریان سه‌فاز را در حالت تعادل حفظ می‌کند و جریان، تنها از طریق یک سیگنال DC که حداکثر مقدار دامنه جریان شبه‌مربعی (I_{max}) را نشان می‌دهد، کنترل می‌شود. این تکنیک با الگوریتم کنترل (PI) و روش مقایسه حامل مثلثی برای تولید سیگنال مدولاسیون عرض پالس (PWM) انجام می‌شود. در [۱۱] یک کنترل‌کننده PID مبتنی بر منطق فازی نوع ۲ (ITYPIDFLC) برای کنترل سرعت موتور DC بدون جاروبک (BLDC) پیشنهاد شده است. عملکرد کنترل‌کننده پیشنهادی به ترتیب با کنترل‌کننده‌های PID معمولی و کنترل‌کننده PID مبتنی بر منطق فازی نوع ۱ در محیط MATLAB/SIMULINK مقایسه می‌شود. نتایج شبیه‌سازی، عملکرد برتر ITYPIDFLC را نسبت به دو نمونه دیگر نشان می‌دهد. در [۱۲] یک کنترل‌کننده سرعت تطبیقی برای موتور DC بدون جاروبک ارائه شده است. روش کنترل تطبیقی مبتنی بر مدل مرجع برای موتور BLDC اعمال می‌شود. این مدل، امکان جبران ریبیل گشتاور و تغییرات گشتاور بار را فراهم می‌کند. پایداری سیستم با استفاده از تابع لیاپانوف ثابت شده است. در [۱۳] روش‌های کنترل بدون حسگر موتور BLDC بررسی می‌شود. پیشرفت‌های اخیر در این زمینه با مزایا و معایب ذاتی آنها معرفی می‌شوند. این مطالعه شامل مروری عمیق بر روش‌های جدید سنجش ولتاژ ضد‌محرکه موتور شامل سنجش ولتاژ ترمینال، انتگرال‌گیری ولتاژ هارمونیک سوم، سنجش جریان ترمینال، انتگرال‌گیری از ولتاژ ضد‌محرکه و استراتژی‌های PWM می‌باشد. همچنین مرتبط‌ترین تکنیک‌های مبتنی بر تخمین و مدل‌ها به طور خلاصه در این مقاله تحلیل می‌شوند. مشاهده‌گر حالت لغزشی، فیلتر کالمن توسعه‌یافته، سیستم تطبیقی مدل مرجع، مشاهده‌گر تطبیقی و شبکه‌های عصبی مصنوعی از جمله این روش‌ها هستند.

در صنعت و کنترل فرایندهای فیزیکی، موارد متعددی وجود دارد که عملکرد سیستم کنترلی، تحت تأثیر پارامترهای دینامیکی قرار می‌گیرند. به عنوان مثال فرض کنید موتور BLDC با اتصال مناسب به شیر هیدرولیکی متصل شده و نیاز به کنترل فشار خروجی شیر هیدرولیکی به صورت سینوسی، پله‌ای (مربعی) و ... باشد. کنترل فشار دینامیکی در این مورد می‌تواند تراست یک محفظه احتراق با مرجع دینامیکی مناسب را کنترل نماید. با توجه به این توضیحات در این مقاله عملکرد دینامیکی موتور BLDC با مراجع دینامیکی مختلف مورد مطالعه قرار گرفته است. با توجه به اهمیت پاسخ دینامیکی موتور BLDC در این مقاله به بررسی، شبیه‌سازی و پیاده‌سازی راه‌انداز موتور BLDC برای کنترل سرعت دینامیکی موتور با مراجع سینوسی، مربعی، مثلثی و دندان‌اره‌ای پرداخته می‌شود. شبیه‌سازی و پیاده‌سازی برای یک موتور BLDC با توان ۸۴ وات و دارای گیربکس خورشیدی با نسبت کاهنده ۵۰ انجام شده است. بیشینه فرکانس مرجع دینامیکی ۰/۴ هرتز می‌باشد و حداکثر دامنه سرعت ۱۲۰ دور بر دقیقه (با اعمال گیربکس) در نظر گرفته شده است. در بیشتر سیستم‌های اتوماسیون و کنترل صنعتی که با کنترل فرایندهای فیزیکی در ارتباط است، دور موتور تا ۱۲۰ rpm جوابگوی بیشتر نیازها می‌باشد و برای بیشینه دور کنترلی کفایت می‌نماید. به عنوان مثال در کنترل فشار یک مدار هیدرولیکی، با حدکثر این دور می‌توان کنترل بهینه و مناسبی بر روی پارامتر فشار خروجی یک شیر هیدرولیکی انجام داد. از طرف دیگر با توجه به اینکه ساختار موتور به صورت مکانیکی می‌باشد، بنابراین پهنای باند مکانیکی عدد کمی است و محدوده کنترل پارامتر مکانیکی (خود موتور و یا فشار سیال) از محدوده فرکانس حداکثر ۰/۴ هرتز نمی‌تواند

شش مرحله‌ای برای کنترل موتور BLDC با استفاده از فاز قطع‌شده برای سنجش ولتاژ القایی استفاده می‌شود. هدف اصلی این مقاله توصیف محدودیت کنترل کموتاسیون شش‌مرحله‌ای معمولی در حالت بدون حسگر در سرعت‌های بالا و معرفی یک رویکرد جدید است که تکنیک مرسوم را برای مناطق با سرعت بالا گسترش می‌دهد. در [۴] راه‌انداز موتور BLDC با استفاده از کنترلر FPGA و با استفاده از روش PWM کنترل می‌شود. این مقاله به معرفی کنترل سرعت موتور BLDC از طریق تغییر سیکل کاری کلیدهای قدرت اینورتر می‌پردازد. در [۵] طراحی سیستم کنترل موتور BLDC با استفاده از MATLAB/SIMULINK برای الگوریتم (PID) توصیف می‌شود که می‌تواند به طور مؤثری کنترل سرعت این نوع موتورها را بهبود بخشد. هدف این مقاله ارائه یک نمای کلی در مورد عملکرد و طراحی کنترل‌کننده PID است. نهایتاً این مطالعه تحت آزمایش‌هایی قرار می‌گیرد که نشان می‌دهد تنظیم‌کننده PID در مقایسه با سایر کنترل‌کننده‌ها بسیار کاربردی‌تر، عملیاتی‌تر و مؤثرتر در دستیابی به عملکرد مناسب کنترل موتور عمل می‌نماید. مقاله [۶] بررسی و تحلیل چالش‌های مهم موتورهای BLDC را انجام می‌دهد. علاوه بر این، در این مقاله بر روی انواع روش‌های کنترل پیشرفته موتور BLDC، مانند کنترل تراسس خطا، کاهش تداخل الکترومغناطیسی، کنترل جهت‌گیری میدان (FOC)، کنترل مستقیم گشتاور (DTC)، شکل‌دهی جریان، کنترل ولتاژ ورودی، کنترل هوشمند، اصل عملکرد کنترلی در کاهش ریبیل گشتاور و نیز توپولوژی راه‌انداز-اینورتر به تفصیل بحث شده است. این مقاله همچنین تاریخچه، انواع، ساختار، مدل‌سازی ریاضی و استانداردهای موتور BLDC را برای کاربردهای مختلف مورد بحث قرار می‌دهد. در [۷] بر عملکرد جدیدی از یک موتور DC بدون جاروبک (BLDC) تمرکز می‌شود. این موتور توسط یک مبدل باک-بوست و کنترل‌کننده (PI) کنترل می‌گردد. یک باتری، توان مورد نیاز برای به حرکت درآوردن موتور BLDC را فراهم می‌کند. سیستم محرک موتور BLDC پیشنهادی برای سرعت ثابت و پله‌ای در ولتاژ گذرگاه DC، توسط یک کنترل‌کننده PI مبتنی بر سیستم FPGA کنترل می‌گردد. این سیستم، عملکرد را بهبود می‌بخشد و کنترل بهینه را تحت شرایط ناپایدار و شرایط با بار متغیر برای استفاده در خودروهای الکتریکی و وسایل نقلیه الکتریکی ترکیبی فراهم می‌کند. هدف [۸] کنترل سرعت موتور BLDC با مدل‌سازی حالت‌های دینامیکی موتور BLDC با استفاده از نرم‌افزار MATLAB/SIMULINK و طراحی یک کنترل‌کننده PID با طراحی یک جبران‌کننده مناسب توسط جعبه ابزار SISO است. در [۹] کنترل‌کننده منطق فازی PI (PF-PI) تنظیم‌شده با الگوریتم ازدحام ذرات ارائه شده که این الگوریتم برای سیستم کنترل سرعت اعمال می‌شود. هدف این مقاله، تنظیم بهینه پارامترهای کنترل‌کننده PI برای به دست آوردن بهترین پاسخ راه‌انداز است. عامل‌های مقیاس با استفاده از کنترل‌کننده PI بهینه‌سازی شده ازدحام ذرات (P-PI) و کنترل‌کننده PF-PI بهینه می‌شوند. سه شاخص عملکردی خطای مطلق زمانی انتگرالی (ITAE)، خطای مربع زمانی انتگرالی (ITSE) و خطای مربع انتگرالی (ISE) برای اندازه‌گیری اثربخشی بهینه‌سازی کنترل‌کننده PF-PI استفاده می‌شوند. نتایج نشان می‌دهند منحنی‌های ریبیل گشتاور و پاسخ سرعت بهینه با استفاده از ITAE به عنوان شاخص عملکرد به دست می‌آیند. همچنین نتایج نشان می‌دهد که روش پیشنهادی، عملکرد دینامیکی برتری را برای موتور BLDC ارائه می‌دهد. رویکرد [۱۰] توسعه تکنیک مدولاسیون دقیق و ساده برای کنترل جریان موتور DC بدون جاروبک (BLDC) می‌باشد. این رویکرد با استفاده از یک کنترل‌کننده

$$V_{an} = R_a i_a + \frac{d}{dt}(L_a i_a + M_{ab} i_b + M_{ac} i_c) + e_a \quad (۱)$$

$$V_{bn} = R_b i_b + \frac{d}{dt}(L_b i_b + M_{ba} i_a + M_{bc} i_c) + e_b \quad (۲)$$

$$V_{cn} = R_c i_c + \frac{d}{dt}(L_c i_c + M_{ca} i_a + M_{cb} i_b) + e_c \quad (۳)$$

چون سیم‌پیچ‌های سه‌فاز یکسان و متقارن فرض می‌شوند، مقدار مقاومت فاز، اندوکتانس خودی و اندوکتانس متقابل آنها یکسان است؛ پس داریم

$$R_a = R_b = R_c = R_s \quad (۴)$$

$$L_a = L_b = L_c = L \quad (۵)$$

$$M_{ab} = M_{ba} = M_{bc} = M_{cb} = M_{ac} = M_{ca} = M \quad (۶)$$

با قراردادن (۴) تا (۶) در (۱) تا (۳) خواهیم داشت

$$V_{an} = R_s i_a + (L - M) \frac{d}{dt} i_a + e_a \quad (۷)$$

$$V_{bn} = R_s i_b + (L - M) \frac{d}{dt} i_b + e_b \quad (۸)$$

$$V_{cn} = R_s i_c + (L - M) \frac{d}{dt} i_c + e_c \quad (۹)$$

که L اندوکتانس خودی، M اندوکتانس متقابل و $L_s = L - M$ اندوکتانس معادل هر فاز می‌باشد. در موتورهای BLDC، ولتاژ ضدمحرکه دوزنقه‌ای با روابط زیر بیان می‌شوند

$$e_a = k_e \omega_m f_a(\theta_e) \quad (۱۰)$$

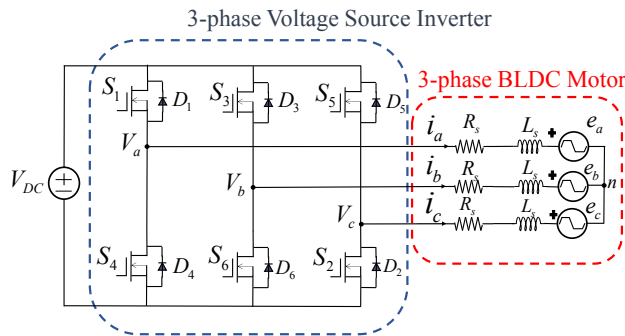
$$e_b = k_e \omega_m f_b(\theta_e) \quad (۱۱)$$

$$e_c = k_e \omega_m f_c(\theta_e) \quad (۱۲)$$

که در روابط فوق k_e ثابت ولتاژ ضدمحرکه و ω_m سرعت مکانیکی بر حسب rad/sec بوده و $f_a(\theta_e)$ ، $f_b(\theta_e)$ و $f_c(\theta_e)$ نیز شکل موج‌های دوزنقه‌ای سه‌فاز با اندازه واحد هستند که به صورت زیر قابل بیان هستند

$$f_a(\theta_e) = \begin{cases} \frac{\pi}{6} - \theta_e & 0 < \theta_e < \frac{\pi}{6} \\ 1 & \frac{\pi}{6} < \theta_e < \frac{5\pi}{6} \\ -\frac{\pi}{6} - \theta_e + \pi & \frac{5\pi}{6} < \theta_e < \frac{7\pi}{6} \\ -1 & \frac{7\pi}{6} < \theta_e < \frac{11\pi}{6} \\ \frac{\pi}{6} - \theta_e - \pi & \frac{11\pi}{6} < \theta_e < 2\pi \end{cases} \quad (۱۳)$$

$$f_b(\theta_e) = \begin{cases} -1 & 0 < \theta_e < \frac{\pi}{6} \\ \frac{\pi}{6} - \theta_e - \pi & \frac{\pi}{6} < \theta_e < \frac{5\pi}{6} \\ 1 & \frac{5\pi}{6} < \theta_e < \frac{7\pi}{6} \\ -\frac{\pi}{6} - \theta_e + \pi & \frac{7\pi}{6} < \theta_e < \frac{11\pi}{6} \\ -1 & \frac{11\pi}{6} < \theta_e < 2\pi \end{cases} \quad (۱۴)$$



شکل ۱: مدار معادل موتور BLDC که توسط اینورتر سه‌فاز راه‌اندازی شده است.

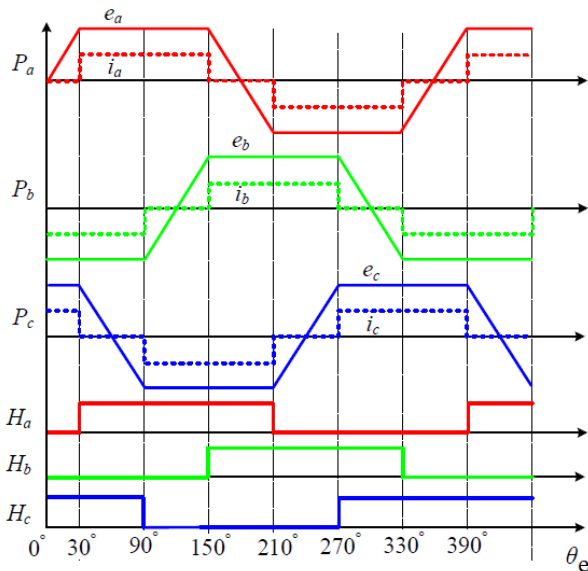
بیشتر شود (چرا که خود موتور و یا فشار سیال، پارامترهای لختی هستند). لذا دور و فرکانس شبیه‌سازی و تست آزمایشگاهی در این محدوده قرار دارند و بنابراین شبیه‌سازی‌ها و تست‌های عملیاتی با این شرایط صورت پذیرفته است. شبیه‌سازی راه‌انداز موتور در نرم‌افزار PLECS انجام شده و از روش PWM و کنترل‌کننده PID استفاده شده است. در این شبیه‌سازی از یک روش جدید برای به دست آوردن مقدار بیشینه جریان موتور از طریق آشکارساز مقدار بیشینه^۱ و استفاده از جریان بیشینه در حلقه کنترل سرعت استفاده شده است. برای پیاده‌سازی راه‌انداز موتور از یک راه‌انداز ۲۰۰ وات استفاده شده است. روش PWM و کنترل‌کننده PID، توسط برد Arduino Mega پیاده‌سازی شده و کنترل دینامیکی سرعت انجام شده است. برد Arduino Mega از طریق پورت سریال با رایانه و نرم‌افزار Labview ارتباط دارد و سرعت مرجع و سرعت واقعی موتور از این طریق در رایانه نمایش داده شده است. همچنین اطلاعات سرعت مرجع و سرعت واقعی در یک LCD کاراکتری نیز نمایش داده می‌شود. بخش‌های مختلف این مقاله بدین ترتیب بیان خواهد شد. در بخش دوم اصول عملکرد موتور BLDC و روابط حاکم بر رفتار آن بیان خواهد شد. بخش سوم بلوک دیاگرام کلی سیستم شبیه‌سازی کنترل‌کننده سرعت موتور BLDC با مراجع دینامیکی را بررسی می‌نماید. در بخش چهارم فلوجارت روش کنترل سرعت دینامیکی موتور در Arduino و جزئیات آن بیان می‌شود. در این بخش، نحوه نمایش و ذخیره داده‌های سرعت مرجع و دور موتور در نرم‌افزار Labview بررسی می‌شوند. در بخش پنجم، نتایج شبیه‌سازی و نتایج آزمایشگاهی حاصل شده نشان داده خواهند شد و در انتها نیز نتیجه‌گیری مقاله ارائه می‌گردد.

۲- اصول عملکرد موتور BLDC و روابط حاکم بر آن

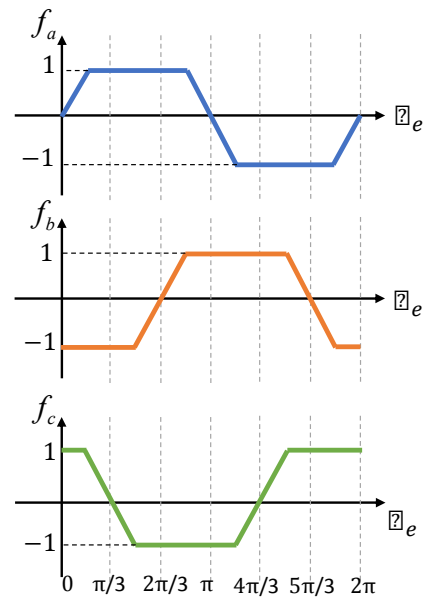
مدار معادل ساده‌شده موتور BLDC در شکل ۱ آمده که در آن سیم‌پیچی استاتور هر فاز با مقاومت R_s و اندوکتانس معادل L_s نشان داده شده است. ولتاژ ضدمحرکه هر فاز (e_a, e_b, e_c) در موتورهای BLDC به شکل دوزنقه هستند. ولتاژ ضدمحرکه فازها به اندازه ۱۲۰ درجه الکتریکی با یکدیگر اختلاف فاز دارند و موتور توسط یک اینورتر سه‌فاز هدایت می‌شود. موقعیت میدان مغناطیسی روتور برای کلیدزنی مناسب اینورتر سه‌فاز، با کمک سه حسگر اثر هال نصب‌شده در موتور BLDC و بر روی قاب استاتور شناسایی می‌شود.

مجموعه معادلات زیر را می‌توان به عنوان مدل ریاضی موتور BLDC در مورد موتورهای با ولتاژ ضدمحرکه دوزنقه‌ای استفاده کرد. ولتاژ پایانه‌های موتور به ترتیب برای فازهای a ، b و c عبارتند از

1. Peak Detector



شکل ۳: جریان‌ها، ولتاژ ضدمحرکه و سیگنال حسگرهای اثر هال در حالت ایده‌آل در موتور BLDC.



شکل ۴: توابع $f_a(\theta_e)$ ، $f_b(\theta_e)$ و $f_c(\theta_e)$ در مدل ولتاژ ضدمحرکه موتور BLDC.

گرفته‌اند، در شکل ۳ نشان داده شده‌اند. در این شکل P_a ، P_b و P_c مربوط به جریان‌ها و ولتاژهای ضدمحرکه مربوط به فازهای a ، b و c و H_a ، H_b و H_c نیز سیگنال حسگرهای اثر هال مربوط به این فازها می‌باشند و دارای ۶ وضعیت مختلف در دوره الکتریکی موتور هستند؛ لذا با یک اینورتر شش مرحله‌ای روبه‌رو هستیم. حاصل ضرب e_a در i_a ، e_b در i_b و e_c در i_c نمایانگر توان ایجادشده توسط هر فاز در موتور می‌باشد؛ به همین دلیل در شکل ۳ محورهای عمودی با P_a ، P_b و P_c مشخص شده‌اند. در این شکل θ_e زاویه الکتریکی روتور بوده و جریان‌های مرجع برای فازهای موتور نیز با توجه به حلقه کنترل سرعت و زاویه θ_e به دست می‌آیند. خروجی حسگرهای اثر هال با توجه به زاویه الکتریکی روتور، θ_e ، ایجاد می‌شوند. به عنوان مثال در زاویه 30° درجه با وصل کلید S_1 در شکل ۱، جریان i_a در جهت مثبت در موتور جاری شده و همزمان با آن سیگنال H_a برابر با ۱ می‌شود. این سیگنال به اندازه 180° درجه الکتریکی ۱ بوده و در زاویه 210° درجه، همزمان با وصل کلید S_2 در شکل ۱ و جاری شدن جریان i_a در جهت منفی در موتور، صفر خواهد شد. در واقع، لبه بالارونده سیگنال اثر هال با لحظه جاری شدن جریان فاز در جهت مثبت و لبه پایین‌رونده آن نیز با لحظه جاری شدن جریان فاز در جهت منفی همزمان می‌باشد. برای بقیه فازها نیز سیگنال حسگرهای اثر هال به همین صورت به دست می‌آید.

۳- بلوک دیاگرام شبیه‌سازی کنترل کننده سرعت موتور BLDC با مراجع دینامیکی

برای شبیه‌سازی کنترل کننده سرعت موتور BLDC با مراجع دینامیکی از نرم‌افزار PLECS استفاده شده و بلوک دیاگرام کنترلی در شکل ۴ آمده است. منبع تغذیه ورودی ۲۴ ولت می‌باشد. این ولتاژ به اینورتر منبع ولتاژی متصل می‌شود. برای اندازه‌گیری جریان از مقاومت ۱ اهم و برای ایجاد تقارن در اینورتر از ۳ مقاومت استفاده گردیده است. برای به دست آوردن بیشینه دامنه جریان از مدار آشکارساز بیشینه دامنه جریان بهره گرفته شده است. این مدار از مقاومت ۱ اهم، دیود D_1 ، خازن C_1 و مقاومت R_1 تشکیل شده است. جریان جاری شده در ساق اینورتر در مقاومت ۱ اهم ساق به ولتاژ تبدیل می‌شود. دیود D_1 ولتاژ را یکسو کرده و خازن C_1 مقدار پوش آن را در خود ذخیره می‌نماید. مقاومت R_1 نیز

$$f_c(\theta_e) = \begin{cases} 1 & 0 < \theta_e < \frac{\pi}{6} \\ -\frac{\pi}{6}\theta_e + 2 & \frac{\pi}{6} < \theta_e < \frac{3\pi}{6} \\ -1 & \frac{3\pi}{6} < \theta_e < \frac{7\pi}{6} \\ \frac{\pi}{6}\theta_e - 8 & \frac{7\pi}{6} < \theta_e < \frac{9\pi}{6} \\ 1 & \frac{9\pi}{6} < \theta_e < 2\pi \end{cases} \quad (15)$$

توابع $f_a(\theta_e)$ ، $f_b(\theta_e)$ و $f_c(\theta_e)$ در شکل ۲ نشان داده شده‌اند. گشتاور الکترومغناطیسی ایجادشده توسط موتور BLDC نیز با معادله زیر بیان می‌شود

$$T_{em} = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega_m} \quad (16)$$

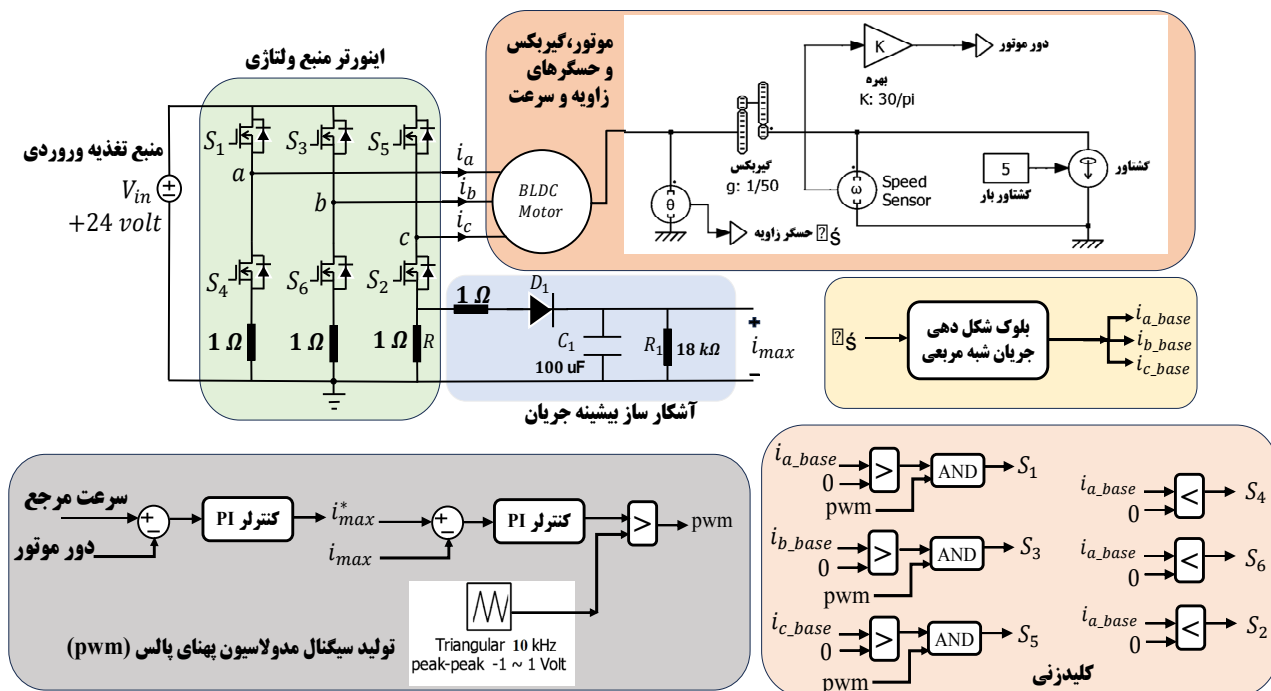
گشتاور الکترومغناطیسی تولیدشده، طبق معادله نیوتن، باعث حرکت و شتاب گرفتن موتور می‌شود. این معادله در (۱۷) بیان شده است

$$T_{em} = T_L + J_m \frac{d\omega_m}{dt} + B\omega_m \quad (17)$$

که J_m ممان اینرسی، B ضریب اصطکاک و T_L گشتاور بار است. مشتق موقعیت روتور نسبت به زمان بیانگر سرعت روتور می‌باشد

$$\omega_e = \frac{d\theta_e}{dt} = \frac{p}{2} \omega_m \rightarrow \theta_e = \int \omega_e dt \quad (18)$$

متغیر p تعداد قطب‌های موتور BLDC می‌باشد. برای $p=2$ سرعت الکتریکی با سرعت مکانیکی روتور برابر است. با توجه به رابطه گشتاور الکترومغناطیسی (رابطه (۱۶))، مجموع مقادیر $e_a i_a$ ، $e_b i_b$ و $e_c i_c$ باید ثابت باشد تا گشتاور الکترومغناطیسی ثابتی به دست آید. برای ولتاژهای ضدمحرکه دوزنقه‌ای ایده‌آل که دارای بخش مسطحی به اندازه 120° درجه می‌باشند، جریان‌های مرجع فاز شبه‌مربعی، گشتاور الکترومغناطیسی ثابت در هر بخش را نتیجه می‌دهد. جریان‌ها، ولتاژهای ضدمحرکه و نیز سیگنال حسگرهای اثر هال که در زوایای 120° درجه در اطراف روتور قرار



شکل ۴: بلوک دیاگرام کنترلی جهت شبیه‌سازی کنترل کننده سرعت دینامیکی موتور BLDC در نرم‌افزار PLECS.

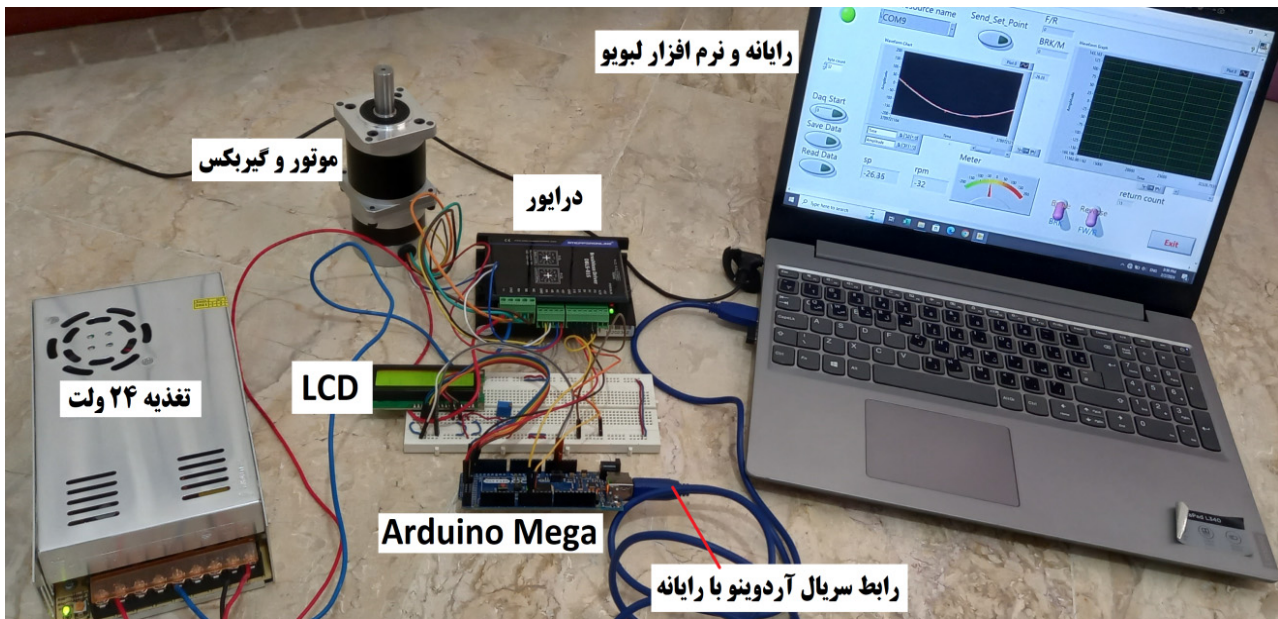
جریان‌های پایه از مقدار صفر بزرگ‌تر باشند و سیگنال PWM یک باشد (AND منطقی)، کلیدهای بالای اینورتر روشن می‌شوند. اگر جریان‌های پایه از مقدار صفر کمتر باشند، کلیدهای پایین اینورتر روشن می‌شوند. بدین ترتیب کلیدزنی اینورتر انجام می‌شود و سرعت واقعی موتور، سرعت دینامیکی مرجع را دنبال می‌نماید. نتایج شبیه‌سازی این راه‌انداز در بخش‌های بعد نشان داده می‌شوند.

۴- پیاده‌سازی سیستم آزمایشگاهی کنترل کننده سرعت دینامیکی موتور BLDC

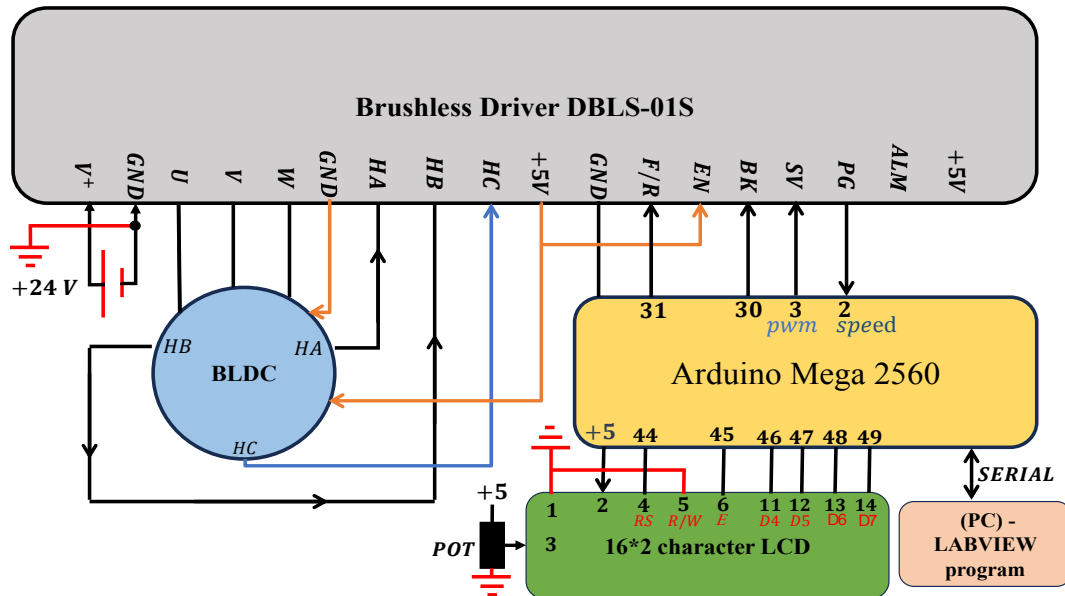
شکل ۵ سیستم آزمایشگاهی پیاده‌سازی شده برای کنترل سرعت دینامیکی موتور BLDC را نشان می‌دهد. این سیستم از یک تغذیه ۲۴ ولت، راه‌انداز ۲۰۰ وات موتور BLDC، موتور BLDC با ولتاژ نامی ۲۴ ولت، توان ۸۴ وات و گیربکس کاهنده ۱ به ۵۰، برد Arduino Mega، LCD کاراکتری و رایانه همراه با نرم‌افزار Labview تشکیل شده است. نحوه اتصالات برای این سیستم آزمایشگاهی در شکل ۶ نشان داده شده است. تغذیه ۲۴ ولت به پایه V+ و GND متصل می‌شود. پایه‌های موتور به پایه‌های U، V و W از راه‌انداز متصل شده‌اند. تغذیه حسگرهای اثر هال از طریق ولتاژ +۵ ولت و GND راه‌انداز تأمین می‌شود. خروجی حسگرهای اثر هال موتور به پایه‌های HA، HB و HC وصل می‌شوند تا ناحیه عملکردی موتور (زاویه الکتریکی) مشخص شود و کلیدزنی مناسب موتور بدین صورت به طور صحیح انجام پذیرد. پایه F/R در راه‌انداز برای تعیین جهت چرخش موتور به صورت راستگرد و یا چپگرد به کار می‌رود که به پایه ۳۱ برد Arduino Mega متصل می‌شود و از این طریق و برنامه برد Arduino Mega کنترل می‌شود. برای فعال بودن راه‌انداز، پایه EN به ولتاژ +۵ متصل شده است. پایه BK برای ترمز موتور مورد استفاده قرار می‌گیرد که از طریق پایه ۳۰ در Arduino Mega کنترل می‌شود. پایه ۳ برد Arduino Mega، یکی از پایه‌های PWM می‌باشد که برای کنترل سرعت موتور BLDC به پایه SV راه‌انداز متصل می‌شود. این پایه برای کنترل سرعت موتور از طریق سیگنال PWM به کار می‌رود. در پایه‌های راه‌انداز، پایه PG حاوی اطلاعات سرعت می‌باشد و

برای دشارژ خازن و برقراری تعادل در مدار آشکارساز بیشینه به کار می‌رود. اگر مقاومت R_۱ نباشد در زمان کاهش جریان، خازن راهی برای دشارژ ندارد و بنابراین مدار نمی‌تواند مقدار بیشینه را به درستی تشخیص دهد. مقادیر مقاومت و خازن مدار به نحوی تعیین شده‌اند که از طرفی سرعت تشخیص مقدار بیشینه مناسب باشد و از طرف دیگر نوسانات جریان بیشینه بیش از حد معمول نباشد تا بتوان کار کنترل جریان را به خوبی انجام داد. مقدار بیشینه جریان اینورتر که همان جریان موتور BLDC است با i_{max} مشخص شده است. سه ساق اینورتر به موتور BLDC متصل می‌شوند و جریان‌های ساق در موتور جاری می‌شوند و باعث چرخش موتور می‌شوند. حسگر زاویه به موتور متصل شده تا زاویه الکتریکی روتور θ را مشخص نماید. موتور به گیربکس کاهنده ۱ به ۵۰ متصل شده است. خروجی گیربکس به گشتاور بار با مقدار ۵ N.m متصل شده و حسگر سرعت زاویه‌ای نیز به خروجی گیربکس متصل گردیده است. سرعت زاویه‌ای با ضریب تبدیل π/۳۰ به سرعت موتور بر حسب RPM^۱ تبدیل می‌شود. برای کنترل دور موتور، سرعت مرجع ورودی و دور موتور از هم کم شده و نتیجه، پس از عبور از کنترل کننده PI به عنوان جریان مرجع بیشینه شناخته می‌شود. این مرجع و جریان بیشینه به دست آمده از آشکارساز بیشینه، از هم کم شده و پس از عبور از کنترل کننده PI به بخش مقایسه کننده وارد می‌شود. در این بخش، سیگنال به دست آمده با سیگنال موج مثلثی با فرکانس ۱۰ KHz و دامنه ۱- تا ۱+ مقایسه می‌شود و بدین ترتیب سیگنال PWM^۲ ایجاد می‌گردد. زاویه θ وارد بلوک شکل دهی جریان‌های شبه‌مربعی می‌شود. این بلوک، جریان‌های شبه‌مربعی شکل ۳ را پیاده‌سازی می‌نماید. به عنوان مثال بین زاویه ۳۰ تا ۹۰ درجه، جریان i_a مقدار +۱، جریان i_b مقدار -۱ و جریان i_c مقدار صفر را دارند و یا بین زاویه ۱۵۰ تا ۲۱۰ درجه، جریان i_a مقدار صفر، جریان i_b مقدار +۱ و جریان i_c مقدار -۱ را دارند. برای بقیه زوایا نیز به همین صورت، جریان‌های پایه شبه‌مربعی به دست می‌آیند. اگر

1. Revolution per Minute
2. Pulse Width Modulation



شکل ۵: سیستم آزمایشگاهی پیاده‌سازی شده برای کنترل سرعت دینامیکی موتور BLDC.



شکل ۶: نحوه اتصالات سیستم آزمایشگاهی پیاده‌سازی شده برای کنترل سرعت دینامیکی موتور BLDC.

می‌شوند. سرعت موتور نیز در Arduino Mega اندازه‌گیری می‌گردد (از طریق وقفه سخت‌افزاری - پایه ۲). داده‌های سرعت مرجع و دور موتور از طریق ارتباط سریال به رایانه و نرم‌افزار Labview ارسال می‌شوند و به صورت عددی، Meter (دور موتور) و نیز گراف نمایش داده می‌شوند. این مقادیر با شروع داده‌برداری (زدن کلید Daq Start) در یک آرایه به صورت موقت ذخیره می‌گردند. با زدن کلید Save Data مقادیر آرایه در یک فایل دلخواه به صورت دائمی ذخیره می‌گردند. با کلید Read Data می‌توان این مقادیر را در Waveform Graph نمایش داد تا مورد پردازش بیشتر و دقیق‌تر قرار گیرند. در موارد مورد نیاز می‌توان برنامه را به صورتی تغییر داد که بتوان جهت حرکت موتور و نیز اعمال ترمز را به صورت دستی از طریق نرم‌افزار انجام داد.

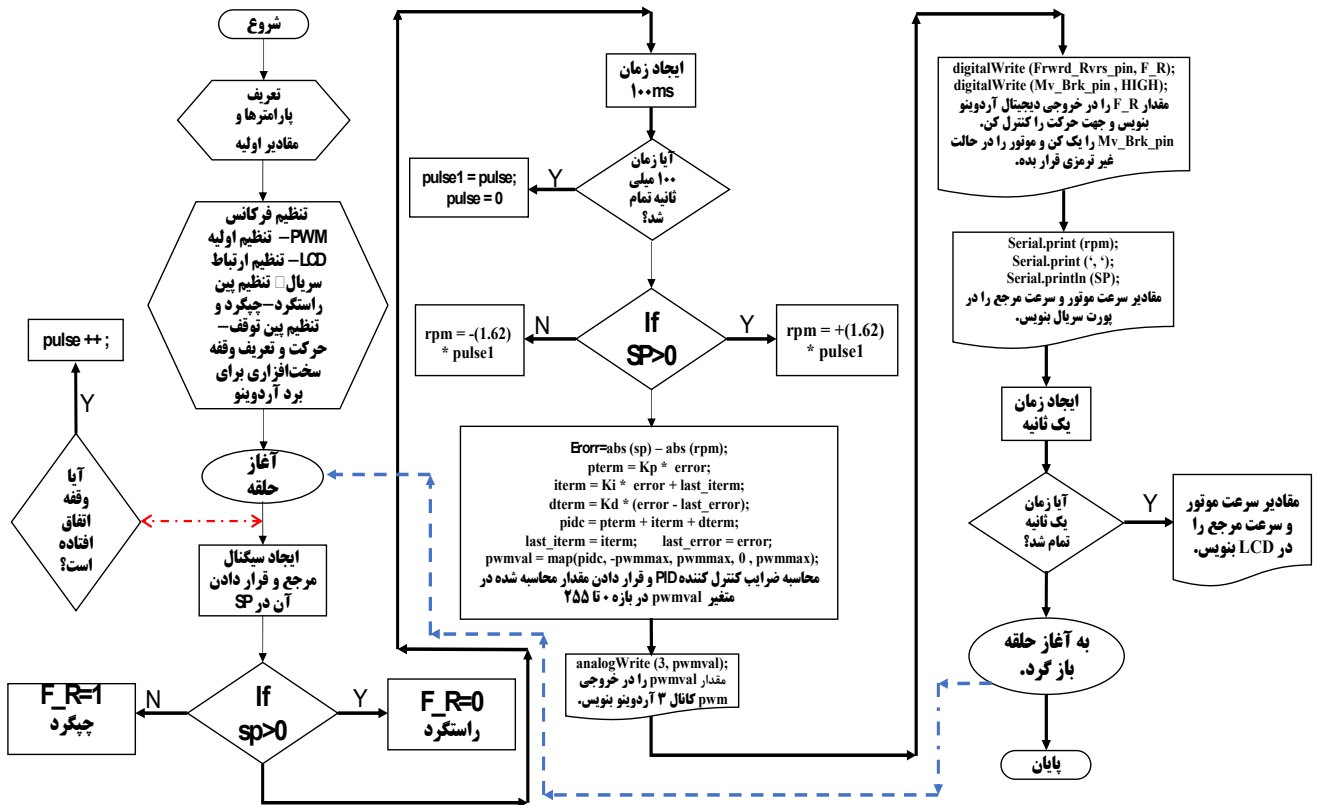
۴-۱ فلوجارت روش کنترل سرعت دینامیکی موتور BLDC در Arduino Mega

شکل ۸ فلوجارت روش کنترل سرعت دینامیکی موتور BLDC در

فرکانس این پایه با تغییرات سرعت، تغییر می‌نماید. این پایه به پایه ۲ برد Arduino Mega متصل می‌شود که پایه وقفه سخت‌افزاری است و از این طریق می‌توان فرکانس سیگنال ورودی که معادل سرعت موتور است را در برد Arduino Mega به دست آورد. برای نمایش سرعت مرجع و سرعت واقعی موتور از یک LCD کاراکتری ۱۶ در ۲ استفاده شده است. پایه‌های LCD طبق شکل ۶ به برد Arduino Mega متصل شده‌اند. علاوه بر آن، سرعت مرجع و سرعت واقعی موتور از طریق پورت سریال به رایانه ارسال می‌شوند. در رایانه از طریق نرم‌افزار Labview، داده‌ها از هم جدا شده و سرعت مرجع و سرعت موتور در نرم‌افزار به صورت عددی و گراف نمایش داده می‌شوند. شکل ۷ صفحه طراحی شده در نرم‌افزار Labview را نشان می‌دهد. در این صفحه، ابتدا پورت سریال مورد نظر جهت ارتباط با برد Arduino Mega انتخاب شده و برنامه اجرا می‌شود. سرعت مرجع در برد Arduino Mega با فرکانس و دامنه مشخص ایجاد می‌شود. سرعت مرجع می‌تواند شکل موج‌های سینوسی، مربعی، مثلثی، دندان‌اره‌ای و یا شکل‌های دیگری باشد که در Arduino تولید



شکل ۷: صفحه طراحی شده در نرم افزار Labview برای سیستم آزمایشگاهی کنترل سرعت دینامیکی موتور BLDC.



شکل ۸: فلوجارت روش کنترل سرعت دینامیکی موتور BLDC در Arduino.

```

pterm = Kp * error;
item = Ki * error + last_itterm;
if (item > pwmmax) item = pwmmax;
if (item < -pwmmax) item = -pwmmax;
dterm = Kd * (error - last_error);
if (dterm > pwmmax) dterm = pwmmax;
if (dterm < -pwmmax) dterm = -pwmmax;

pidc = 1 * (pterm + item + dterm);
if (pidc > pwmmax) pidc = pwmmax;
if (pidc < -pwmmax) pidc = -pwmmax;
last_itterm = item;
last_error = error;
    
```

محاسبه مقدار تناسبی
محاسبه مقدار انتگرالی
ایجاد محدودیت برای مقدار انتگرالی
ایجاد محدودیت برای مقدار انتگرالی
محاسبه مقدار مشتق گیر
تعیین محدودیت برای مقدار مشتق گیر
تعیین محدودیت برای مقدار مشتق گیر
محاسبه مقدار
تعیین محدودیت برای مقدار
تعیین محدودیت برای مقدار
ذخیره مقدار انتگرالی در متغیر
ذخیره مقدار خطا در متغیر

Arduino Mega را نشان می‌دهد. در این برنامه، ابتدا پارامترهای ثابت و متغیر تعریف شده و مقادیری اولیه می‌گردند. متغیرهای Kp ، Ki و Kd مربوط به ضرایب تناسبی، انتگرالی و مشتق گیر می‌باشند. متغیرهای $error$ ، $item$ ، $dterm$ ، $last_error$ ، $last_itterm$ و $pidc$ مربوط به محاسبات کنترل کننده PID هستند که متغیر $error$ برای خطا، $pterm$ مقدار تناسبی، $item$ مقدار انتگرالی، $dterm$ مقدار مشتق گیر، $last_error$ آخرین مقدار خطا، $last_itterm$ آخرین مقدار $item$ و $pidc$ مقدار نهایی PID را در خود ذخیره می‌نمایند. این روابط در برد Arduino Mega به صورت زیر پیاده‌سازی شده‌اند:

$error = abs(sp) - abs(rpm);$ محاسبه خطا

باشد $F_R = 0$ و اگر مرجع منفی باشد $F_R = 1$ قرار داده می‌شود. در ادامه برنامه هر ۱۰۰ میلی‌ثانیه، متغیر *pulse* در متغیر *pulse1* قرار می‌گیرد و *pulse* صفر می‌شود تا برای محاسبه دوباره فرکانس به کار رود. با توجه به اینکه مرجع مثبت و یا منفی باشد با اعمال ضریب کالیبراسیون مقدار دور موتور *rpm* به دست می‌آید. در ادامه برنامه، محاسبات مربوط به کنترل‌کننده *PID* انجام می‌شود. با توجه به خطای موجود بین مقدار مرجع و دور واقعی موتور و همچنین ضرایب تناسبی، مشتق‌گیر و انتگرالی، محاسبه *iterm* و *dterm* انجام می‌شود و نهایتاً مقدار *pidc* از جمع این موارد به دست می‌آید. این مقدار به عدد بین ۰ تا ۲۵۵ تبدیل می‌شود تا قابل اعمال به خروجی *pwm* برد *Arduino* باشد. در ادامه مقدار کنترل‌کننده *PID* درون متغیر *pwmvalue* قرار می‌گیرد و به کانال ۳ که مربوط به خروجی *pwm* است، نوشته می‌شود تا سرعت موتور را کنترل نماید. همچنین مقدار F_R برای کنترل جهت چرخش موتور به خروجی دیجیتال مربوط به کنترل جهت ارسال می‌شود. موتور نیز در حالت غیرترمزی قرار می‌گیرد. مقادیر متغیرهای *rpm* و *sp* از طریق پورت سریال به رایانه ارسال می‌شود تا این مقادیر در نرم‌افزار *Labview* دریافت، ذخیره و پردازش شوند. مقادیر سرعت موتور و مرجع نیز هر یک ثانیه یک بار در داخل *LCD* کاراکتری نمایش داده می‌شوند.

۴-۲ نحوه نمایش و ذخیره داده‌های سرعت مرجع و دور

موتور BLDC در *Labview*

شکل ۹ پیکربندی ساختار پیاده‌سازی شده در نرم‌افزار *Labview* جهت نمایش و ذخیره داده‌های سرعت مرجع دینامیکی و دور واقعی موتور را نشان می‌دهد. در این برنامه، خارج از حلقه، پورت سریال همراه با نام آن و نرخ تبادل ۱۱۵۲۰۰ تعریف می‌شود. در مورد ارتباط سریال بالاترین عدد *BaudRate* برای ارتباط سریال در نظر گرفته شده تا سرعت انتقال داده‌ها به رایانه و به‌روزرسانی نمایش گرافها از کیفیت خوبی برخوردار باشند. سرعت‌های پایین‌تر کیفیت نمایش داده‌ها را پایین آورده و از این جهت مطلوب نمی‌باشد. پایه‌های ۳۰ و ۳۱ برد *Arduino* به صورت خروجی در نظر گرفته می‌شوند و در انتها نیز وقفه سخت‌افزاری صفر که مربوط به پایه ۲ برد *Arduino* هست تعریف می‌شود. اگر پالسی به ورودی پایه ۲ اعمال شود، با تغییرات در این پایه، تابعی اجرا می‌شود که در این تابع مقدار متغیر *pulse* اضافه می‌شود تا تعداد تغییرات در بازه زمانی مشخص (در اینجا بازه زمانی ۱۰۰ ms) به دست آید. این تغییرات نشان‌دهنده فرکانس سیگنال ورودی می‌باشد. این فرکانس با یک ضریب به دور واقعی موتور تبدیل می‌شود. در واقع متغیر *pulse1* یک متغیر کمکی است که هر ۱۰۰ میلی‌ثانیه مقدار *pulse* را در خود ذخیره می‌کند و برای محاسبه دور استفاده می‌نماید. در ضمن هر ۱۰۰ میلی‌ثانیه مقدار *pulse* صفر می‌شود تا برای شمارش دوباره تعداد پالس‌ها به کار رود. ضریب ۱/۶۲ مربوط به کالیبراسیون دور موتور می‌باشد. پایه *PG* در درایو، فرکانس متناسب با دور موتور تولید می‌کند. در *Arduino* تعداد پالس‌های این سیگنال در زمان ۱۰۰ میلی‌ثانیه توسط وقفه سخت‌افزاری محاسبه می‌شوند تا بدین وسیله فرکانس سیگنال به دست آید. برای تبدیل مقدار فرکانس به دور موتور به ضریب کالیبراسیون نیاز است. با علامت‌گذاری بر روی شفت موتور و به دست آوردن تعداد دورها در زمان ۶۰ ثانیه مقدار *RPM* برای موتور به دست می‌آید. این ضریب به گونه‌ای تعیین شده که مقدار نمایش داده‌شده در رایانه و *LCD* با دور محاسبه‌شده در ۶۰ ثانیه برابر گردد و دور واقعی موتور به طور صحیح نمایش داده شود. در آغاز حلقه، سیگنال مرجع ایجاد شده و در متغیر *sp* قرار می‌گیرد. با توجه به اینکه مرجع مثبت و یا منفی باشد، جهت حرکت موتور با توجه به متغیر F_R تعیین می‌شود. اگر مرجع مثبت

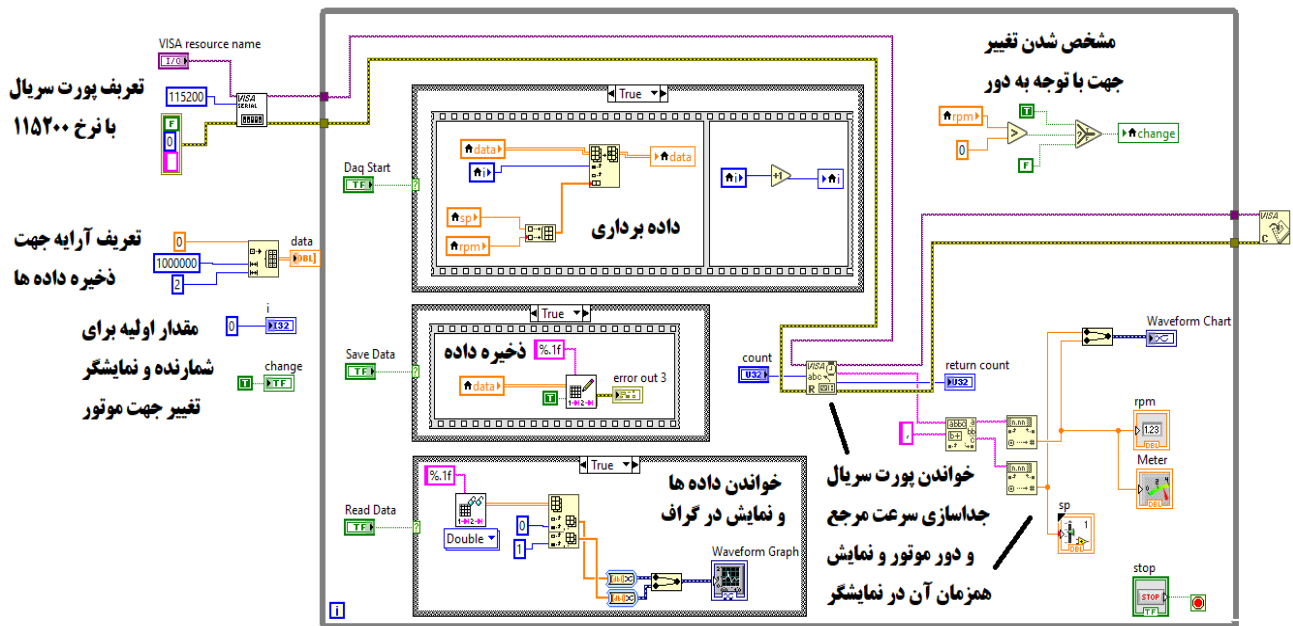
$pwmval = \text{map}(pidc, -pwmmax, pwmmax, 0, pwmmax);$

قراردادن خروجی کنترل‌کننده در محدوده ۰ تا ۲۵۵

$\text{analogWrite}(3, pwmval);$

ارسال مقدار خروجی مدولاسیون پهنای پالس به پایه خروجی ۳

متغیر *Mv_Brk_pin*، پین *Arduino*، مربوط به ترمز و حرکت موتور را مشخص می‌نماید. *Frwrdr_Rvrs_pin* پین آردوینو مربوط به جهت حرکت موتور را مشخص می‌نماید. از آنجا که پایه *PWM* در *Arduino*، هشتمینی می‌باشد، حداکثر مقدار *PWM* برابر ۲۵۵ است. از این رو متغیر *pwmmax* این مقدار را در خود ذخیره می‌کند تا در محاسبات *PID* از آن استفاده گردد. متغیر F_R مقدار مربوط به جهت حرکت موتور را در خود ذخیره می‌نماید. *pwmmax* مقدار خروجی *PWM* است که از پایه ۳ آردوینو خارج می‌شود. *pulse* و *pulse1* مقدار فرکانس اندازه‌گیری شده از پایه ۲ (وقفه سخت‌افزاری) آردوینو را در خود ذخیره می‌کنند. *sp* مقدار مرجع سرعت دینامیکی را در خود ذخیره می‌کند و *rpm* دور واقعی موتور را که از روی *pulse1* و ضریب کالیبراسیون به دست آمده است، در خود ذخیره می‌نماید. در بخش بعد فرکانس سیگنال *PWM* برابر ۳۹۲۱ هرتز تنظیم می‌شود. در برد *Arduino Mega* با تنظیم رجیستر کنترلی تایمر می‌توان فرکانس *PWM* را تغییر داد. فرکانس *PWM* معمولاً در توان‌های در حدود ۱۰۰ وات برابر چند کیلوهرتز در نظر گرفته می‌شود. با آزمایش‌هایی که بر روی درایو و مدار کنترلی صورت پذیرفت، فرکانس کلیدزنی *PWM* حدود ۴ کیلوهرتز در نظر گرفته شده است. *LCD* به صورت کاراکتری ۱۶ در ۲ تعریف می‌شود. مقدار *baudrate* در ارتباط سریال برابر ۱۱۵۲۰۰ تنظیم می‌شود. در مورد ارتباط سریال بالاترین عدد *baudrate* برای ارتباط سریال در نظر گرفته شده تا سرعت انتقال داده‌ها به رایانه و به‌روزرسانی نمایش گرافها از کیفیت خوبی برخوردار باشند. سرعت‌های پایین‌تر کیفیت نمایش داده‌ها را پایین آورده و از این جهت مطلوب نمی‌باشد. پایه‌های ۳۰ و ۳۱ برد *Arduino* به صورت خروجی در نظر گرفته می‌شوند و در انتها نیز وقفه سخت‌افزاری صفر که مربوط به پایه ۲ برد *Arduino* هست تعریف می‌شود. اگر پالسی به ورودی پایه ۲ اعمال شود، با تغییرات در این پایه، تابعی اجرا می‌شود که در این تابع مقدار متغیر *pulse* اضافه می‌شود تا تعداد تغییرات در بازه زمانی مشخص (در اینجا بازه زمانی ۱۰۰ ms) به دست آید. این تغییرات نشان‌دهنده فرکانس سیگنال ورودی می‌باشد. این فرکانس با یک ضریب به دور واقعی موتور تبدیل می‌شود. در واقع متغیر *pulse1* یک متغیر کمکی است که هر ۱۰۰ میلی‌ثانیه مقدار *pulse* را در خود ذخیره می‌کند و برای محاسبه دور استفاده می‌نماید. در ضمن هر ۱۰۰ میلی‌ثانیه مقدار *pulse* صفر می‌شود تا برای شمارش دوباره تعداد پالس‌ها به کار رود. ضریب ۱/۶۲ مربوط به کالیبراسیون دور موتور می‌باشد. پایه *PG* در درایو، فرکانس متناسب با دور موتور تولید می‌کند. در *Arduino* تعداد پالس‌های این سیگنال در زمان ۱۰۰ میلی‌ثانیه توسط وقفه سخت‌افزاری محاسبه می‌شوند تا بدین وسیله فرکانس سیگنال به دست آید. برای تبدیل مقدار فرکانس به دور موتور به ضریب کالیبراسیون نیاز است. با علامت‌گذاری بر روی شفت موتور و به دست آوردن تعداد دورها در زمان ۶۰ ثانیه مقدار *RPM* برای موتور به دست می‌آید. این ضریب به گونه‌ای تعیین شده که مقدار نمایش داده‌شده در رایانه و *LCD* با دور محاسبه‌شده در ۶۰ ثانیه برابر گردد و دور واقعی موتور به طور صحیح نمایش داده شود. در آغاز حلقه، سیگنال مرجع ایجاد شده و در متغیر *sp* قرار می‌گیرد. با توجه به اینکه مرجع مثبت و یا منفی باشد، جهت حرکت موتور با توجه به متغیر F_R تعیین می‌شود. اگر مرجع مثبت



شکل ۹: پیکره‌بندی ساختار پیاده‌سازی شده در نرم‌افزار Labview جهت نمایش و ذخیره داده‌های سرعت مرجع دینامیکی و دور واقعی موتور BLDC.

صفر و یک فایل ذخیره شده‌اند؛ بنابراین با استخراج داده‌های فایل به صورت ستونی می‌توان آنها را استخراج و در گراف نمایش داد.

۵- نتایج شبیه‌سازی و آزمایشگاهی

۵-۱ نتایج شبیه‌سازی

برای شبیه‌سازی از نرم‌افزار PLECS استفاده شده و مشخصات موتور شبیه‌سازی شده در جدول ۱ آمده است. شبیه‌سازی‌ها برای سیگنال‌های مرجع سینوسی، مثلثی، دندان‌اره‌ای و مربعی انجام شده است. برای مرجع سینوسی با فرکانس ۰/۱ تا ۰/۴ هرتز و دامنه ۱۰۰ RPM، دور موتور مقدار مرجع را دنبال کرده و این موضوع در شکل ۱۰ نشان داده شده است. در این شکل رنگ قرمز مرجع و رنگ آبی دور موتور را نشان می‌دهد. شکل ۱۱ مقادیر i_{max}^* و i_{max} را در زمان شبیه‌سازی با مرجع سینوسی با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان می‌دهد. ملاحظه می‌گردد که بیشینه جریان موتور، مقدار مرجع خود را به خوبی دنبال نموده است. در این شکل، رنگ قرمز جریان مرجع و رنگ آبی بیشینه جریان واقعی موتور می‌باشد. شکل ۱۲ جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی را برای مرجع سینوسی با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان داده است.

برای مرجع مثلثی با فرکانس ۰/۱ تا ۰/۴ هرتز و دامنه ۱۲۰ RPM، دور موتور، مقدار مرجع را دنبال کرده و این موضوع در شکل ۱۳ نشان داده شده است. در این شکل رنگ قرمز مرجع و رنگ آبی دور موتور را نشان می‌دهد. شکل ۱۴ مقادیر i_{max}^* و i_{max} را در زمان شبیه‌سازی با مرجع مثلثی با فرکانس ۰/۱ هرتز و دامنه ۱۲۰ RPM نشان می‌دهد. ملاحظه می‌گردد که بیشینه جریان موتور مقدار مرجع خود را به خوبی دنبال نموده است. شکل ۱۵ جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی را برای مرجع مثلثی با فرکانس ۰/۱ هرتز و دامنه ۱۲۰ RPM نشان داده است.

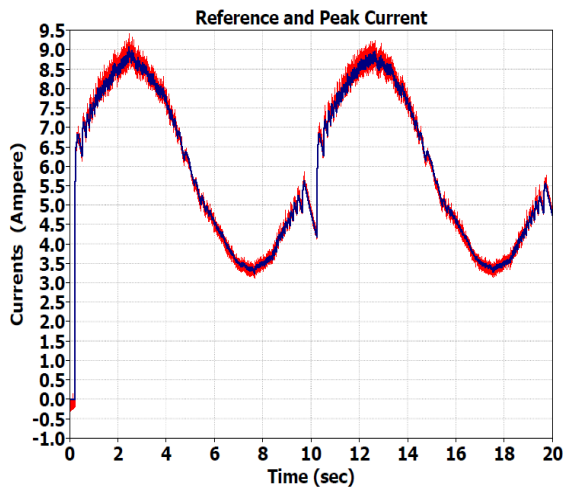
برای مرجع دندان‌اره‌ای با فرکانس ۰/۱ تا ۰/۴ هرتز و دامنه ۱۰۰ RPM، دور موتور، مقدار مرجع را دنبال کرده است. این موضوع در شکل ۱۶ نشان داده شده است. در این شکل رنگ قرمز مرجع و رنگ آبی دور موتور را نشان می‌دهد. شکل ۱۷ مقادیر i_{max}^* و i_{max} را در زمان

جدول ۱: مشخصات موتور شبیه‌سازی شده در نرم‌افزار PLECS.

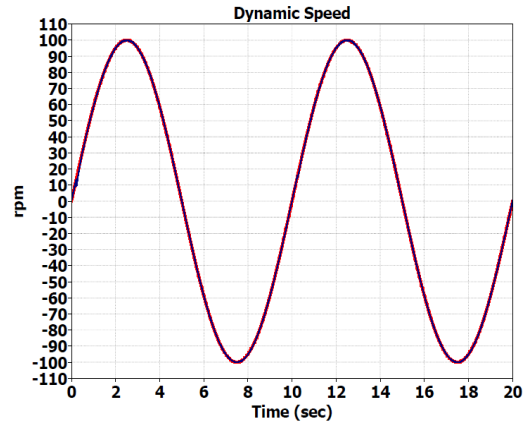
پارامتر	مقدار
ولتاژ نامی	۲۴ V
دور نامی	۶۰۰۰ rpm
مقاومت فاز	۰/۵ Ω
اندوکتانس فاز	۰/۸ mH
تعداد قطب‌ها	۲ عدد (یک جفت)
گشتاور نامی	۰/۲ N.m
ضریب ولتاژ	۰/۲ V/Krpm
ضدمحرکه	

```
void loop () {
    x = x + 0.0001;
    y1 = 120 * sin(x);
    sp = y1;
    if (sp > 0) rpm = (1.62) * pulse1;
    if (sp < 0) rpm = (-1.62) * pulse1;
    Serial.print(rpm);
    Serial.print(',');
    Serial.println(y1);
}
```

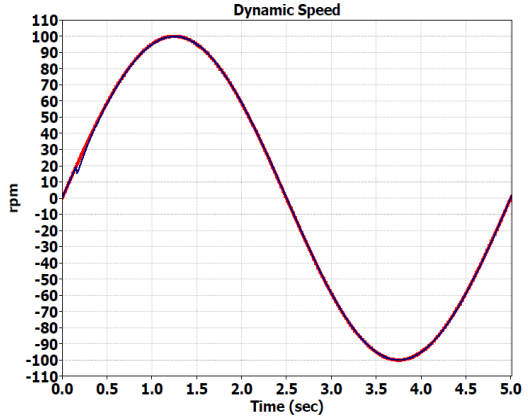
همان طور که مشاهده می‌شود، دور موتور و مرجع دینامیکی به صورت سریال و با یک کاراکتر « , » از هم جدا شده‌اند؛ بنابراین در Labview ابتدا مقدار دور خوانده می‌شود و وقتی رشته سریال به کاراکتر « , » رسید، مقدار بعد از آن به عنوان سیگنال مرجع در نظر گرفته می‌شود و بدین ترتیب جداسازی دور موتور و مقدار مرجع از هم صورت می‌پذیرد. مقدار متغیر *change* با توجه به اینکه سرعت موتور مثبت و یا منفی است، تغییر وضعیت می‌دهد. با شروع داده‌برداری متغیرهای *rpm* و *sp* با افزایش شمارنده *i* در سطرهای آرایه *data* قرار می‌گیرند. با پایان داده‌برداری آرایه *data* آماده ذخیره‌شدن است و با زدن دکمه *Save Data* می‌توان *data* را در فایلی با نام دلخواه ذخیره نمود. با زدن دکمه *Read Data* می‌توان داده‌های ذخیره شده سرعت مرجع و دور موتور را جهت پردازش بیشتر در *WaveformGraph* نمایش داد و مورد تحلیل قرار داد. بدین ترتیب که هر یک از داده‌های سرعت مرجع و دور موتور در ستون‌های



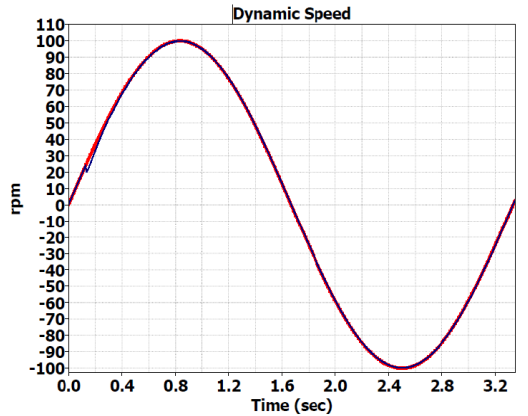
شکل ۱۱: جریان بیشینه مرجع (قرمز) و بیشینه جریان واقعی موتور (آبی) در شبیه‌سازی با مرجع سینوسی با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.



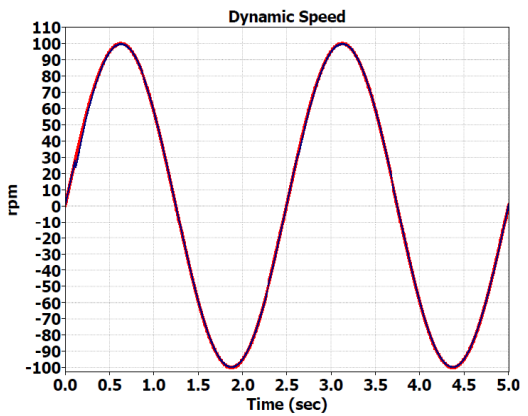
(الف)



(ب)

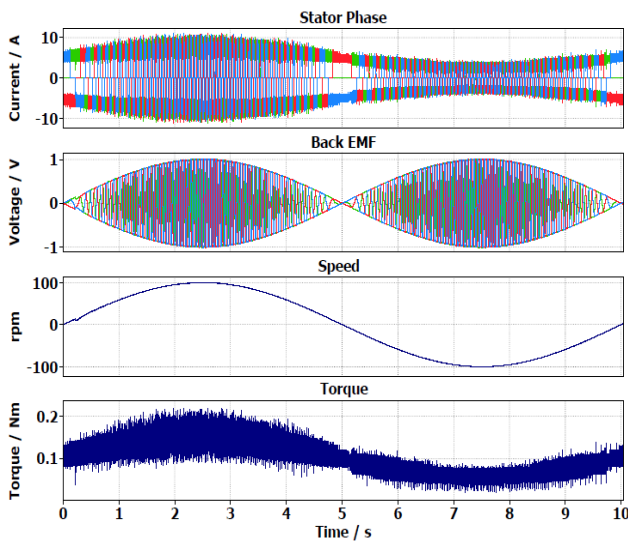


(ج)



(د)

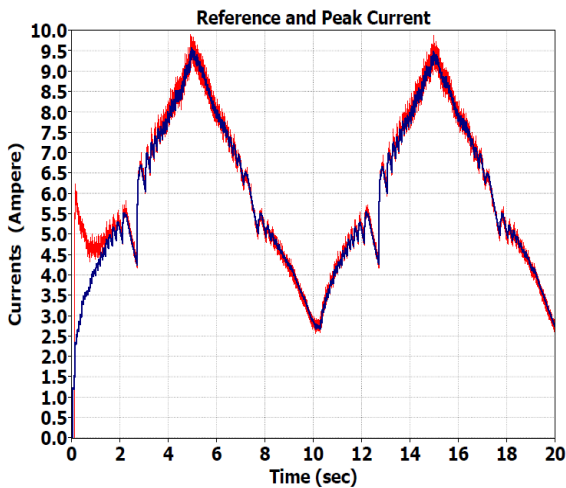
شکل ۱۰: نتایج شبیه‌سازی برای مرجع سینوسی با دامنه ۱۰۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ قرمز مرجع و رنگ آبی دور واقعی موتور است.



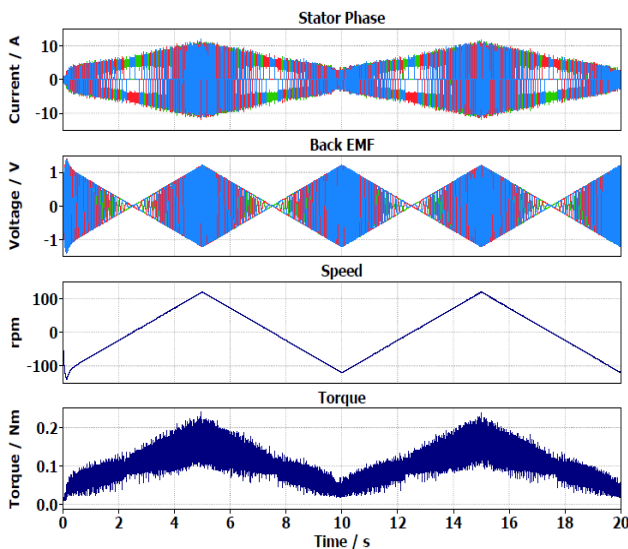
شکل ۱۲: جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی برای مرجع سینوسی با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.

شبیه‌سازی با مرجع دندان‌اره‌ای با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان می‌دهد. ملاحظه می‌گردد که بیشینه جریان موتور مقدار مرجع خود را به خوبی دنبال نموده است. مقدار جریان مرجع در حلقه کنترل سرعت بیانگر مقدار جریان مرجع بیشینه است که همواره دارای مقادیر مثبت می‌باشد (چه در جهت راستگرد و چه در جهت چپگرد موتور). همان طور که در بلوک دیاگرام کنترلی شکل ۴ مشاهده می‌شود، مقدار بیشینه جریان موتور با استفاده از مدار آشکارساز بیشینه به دست آمده است که همواره مقداری مثبت می‌باشد و دارای آفست است. این جریان با جریان مرجع بیشینه مقایسه می‌شود و در نهایت سیگنال *PWM* برای کنترل اینورتر استخراج می‌گردد. شکل ۱۸ جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی را برای مرجع دندان‌اره‌ای با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان داده است.

برای مرجع مربعی با فرکانس ۰/۱ تا ۰/۴ هرتز و دامنه ۱۰۰ RPM، دور موتور مقدار مرجع را دنبال کرده و این موضوع در شکل ۱۹ نشان داده شده است. در این شکل رنگ قرمز مرجع و رنگ آبی دور موتور را نشان می‌دهد. شکل ۲۰ مقادیر i_{max}^* و i_{max} را در زمان شبیه‌سازی با مرجع مربعی با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان می‌دهد. ملاحظه می‌گردد که بیشینه جریان موتور مقدار مرجع خود را به خوبی



شکل ۱۴: جریان بیشینه مرجع (قرمز) و بیشینه جریان واقعی موتور (آبی) در شبیه‌سازی با مرجع مثلثی با دامنه ۱۲۰ rpm و فرکانس ۰/۱ هرتز.



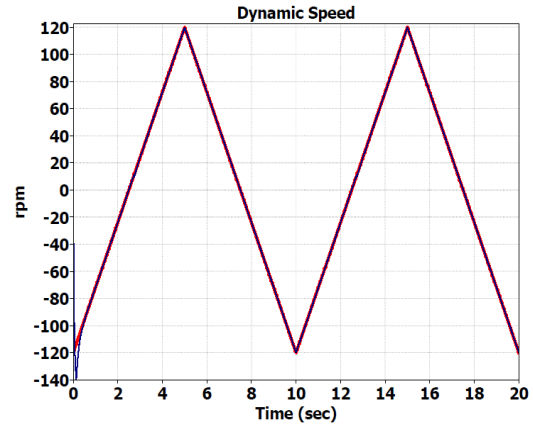
شکل ۱۵: جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی برای مرجع مثلثی با دامنه ۱۲۰ rpm و فرکانس ۰/۱ هرتز.

دنبال نموده است. با توجه به اینکه تغییرات دور مرجع مربعی بسیار سریع اتفاق می‌افتد، جریان مرجع (قرمز) در شکل ۲۰ تغییرات شدیدی دارد و بدیهی است که جریان بیشینه (آبی) با سرعت بالا نمی‌تواند مرجع را دنبال نماید (به دلیل وجود خازن و مقاومت در مدار آشکارساز بیشینه جریان). بعد از زمان کوتاهی جریان بیشینه موتور جریان مرجع را دنبال می‌نماید. شکل ۲۱ جریان‌های استاتور، ولتاژهای ضدمحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی را برای مرجع مربعی با فرکانس ۰/۱ هرتز و دامنه ۱۰۰ RPM نشان داده است.

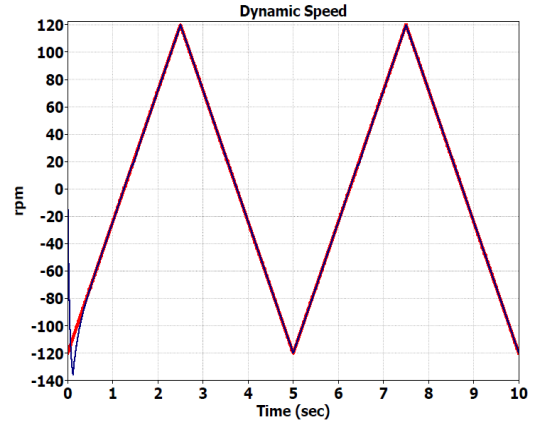
۲-۵ نتایج آزمایشگاهی

تست‌های آزمایشگاهی برای مرجع سینوسی، مثلثی، دندان‌اره‌ای و مربعی با فرکانس‌های ۰/۱ تا ۰/۴ هرتز انجام شده است. تست‌های انجام‌شده با یک راه‌انداز ۲۰۰ وات و یک موتور BLDC با توان ۸۴ وات همراه با گیربکس کاهنده ۱ به ۵۰ صورت گرفته است. مشخصات موتور BLDC استفاده‌شده در تست‌های آزمایشگاهی در جدول ۲ آمده است.

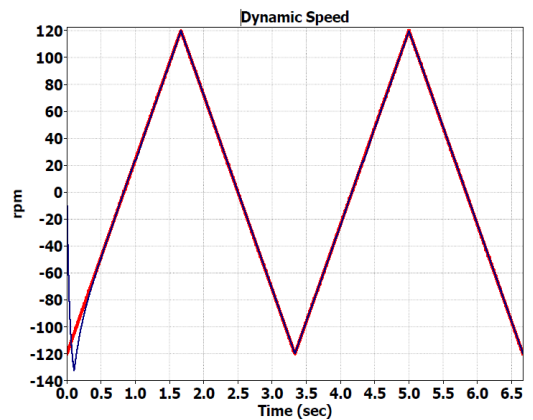
برای پیاده‌سازی سرعت مرجع به صورت سینوسی به صورت زیر عمل می‌شود:



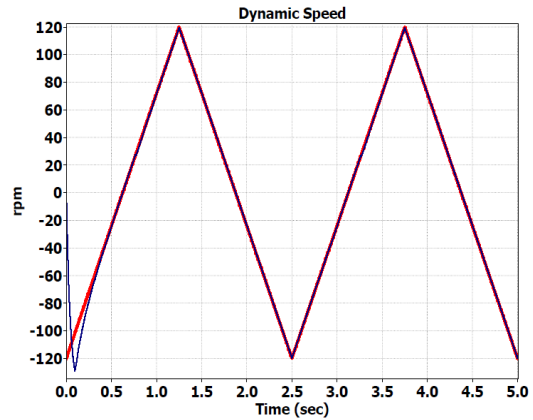
(الف)



(ب)

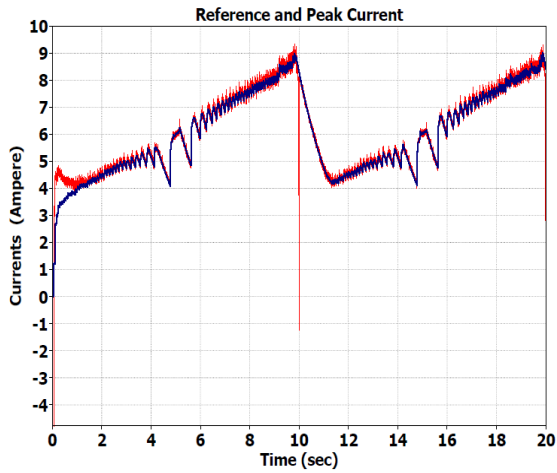


(ج)

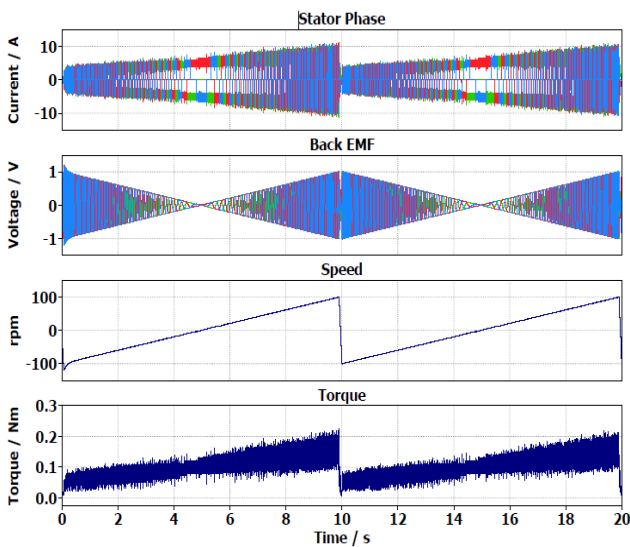


(د)

شکل ۱۳: نتایج شبیه‌سازی برای مرجع مثلثی با دامنه ۱۲۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ قرمز مرجع و رنگ آبی دور واقعی موتور است.



شکل ۱۷: جریان بیشینه مرجع (قرمز) و بیشینه جریان واقعی موتور (آبی) در شبیه‌سازی با مرجع دندان‌اره‌ای با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.



شکل ۱۸: جریان‌های استاتور، ولتاژهای ضد‌محرکه، دور واقعی موتور و گشتاور الکترومغناطیسی برای مرجع دندان‌اره‌ای با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.

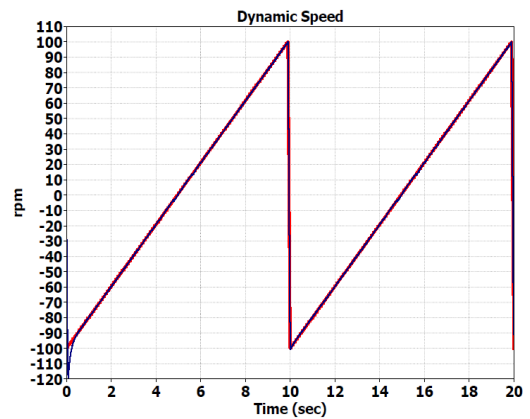
جدول ۲: مشخصات موتور BLDC استفاده‌شده در تست‌های آزمایشگاهی.

پارامتر	مقدار
ولتاژ نامی	۲۴ V
جریان نامی	۵ A
بازده	٪۷۰
دور نامی	۶۰۰ RPM
مقاومت فاز	۰/۴۲ Ω
اندوکتانس فاز	۰/۷ mH
تعداد قطب‌ها	۴ عدد (۲ جفت)
گشتاور نامی	۰/۲ N.m

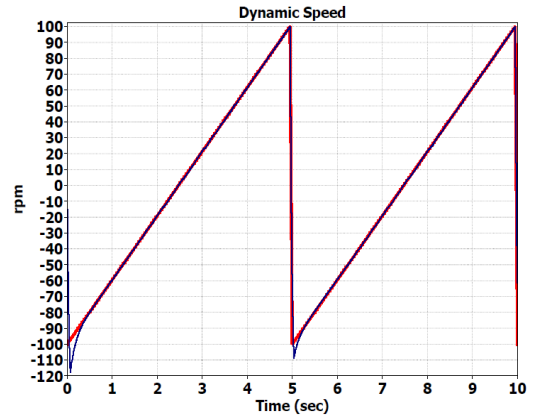
```

VOID LOOP () {
    X = X + 0.0001;
    Y1 = 120 * SIN(X);
    SP = Y1;
}
    
```

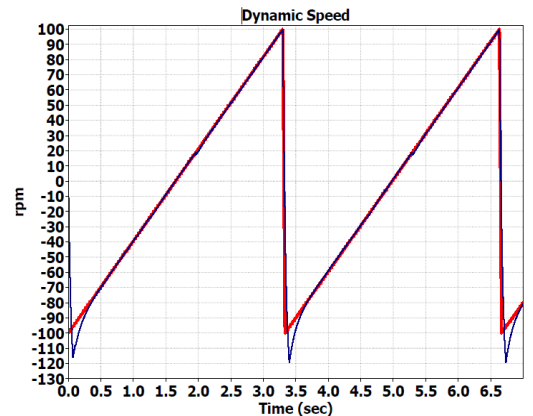
در این برنامه، متغیر X با نرخ مشخص (بستگی به فرکانس سیگنال مرجع) افزایش یافته و سپس مقدار سینوس آن محاسبه می‌شود و در متغیر SP قرار می‌گیرد. برای پیاده‌سازی سرعت مرجع به صورت مربعی به صورت زیر عمل می‌شود:



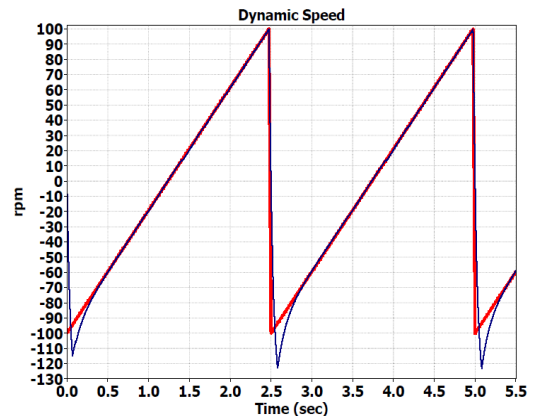
(الف)



(ب)

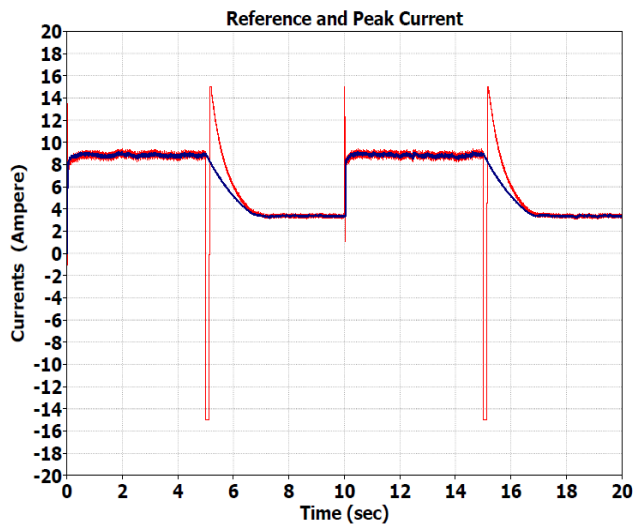


(ج)

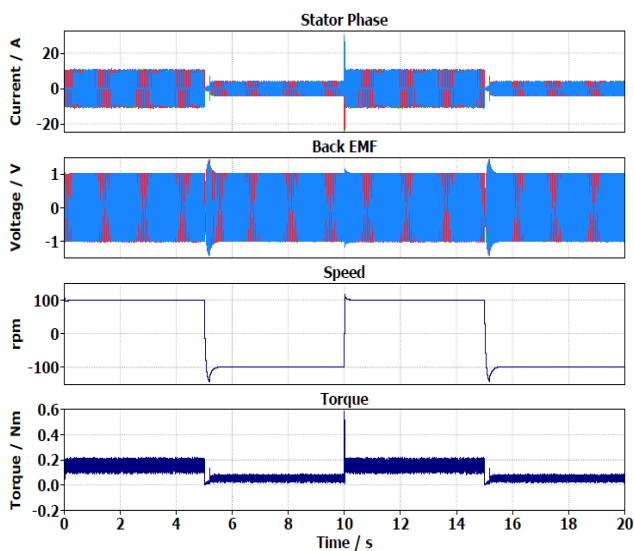


(د)

شکل ۱۶: نتایج شبیه‌سازی برای مرجع دندان‌اره‌ای با دامنه ۱۰۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ قرمز مرجع و رنگ آبی دور واقعی موتور است.



شکل ۲۰: جریان بیشینه مرجع (قرمز) و بیشینه جریان واقعی موتور (آبی) در شبیه‌سازی با مرجع مربعی با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.

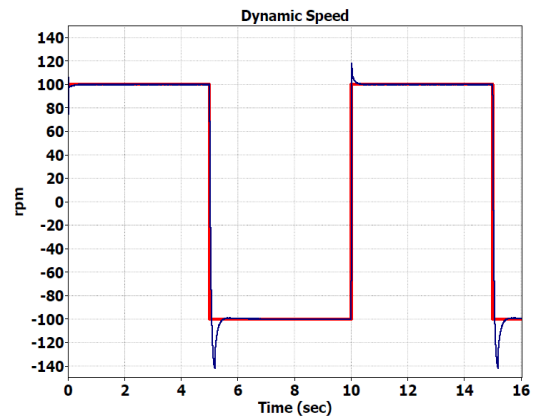


شکل ۲۱: جریان‌های استاتور، ولتاژهای ضدحرکه، دور واقعی موتور و گشتاور الکترومغناطیسی برای مرجع مربعی با دامنه ۱۰۰ rpm و فرکانس ۰/۱ هرتز.

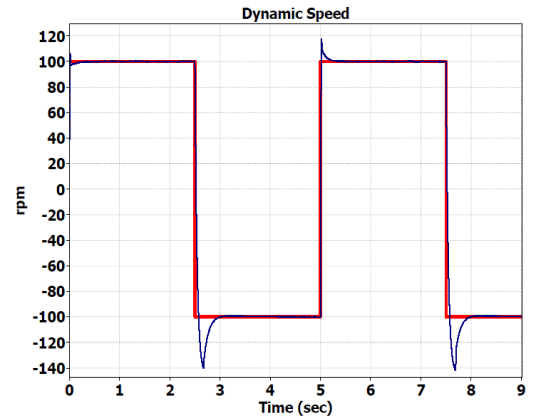
```
SIGN = 1;
VOID LOOP () {
    NOWTIME = MILLIS ();
    IF (NOWTIME > LASTTIME + 1000) {
        LASTTIME = NOWTIME;
        SIGN = -1 * SIGN;
        Y1 = SIGN*100;
    }
    SP = Y1;
}
```

در این برنامه برای ایجاد فرکانس ۰/۵ هرتز، هر ۱ ثانیه یک بار مقدار *SIGN* تغییر می‌نماید و این مقدار در ۱۰۰ ضرب می‌شود و بدین ترتیب موج مربعی با فرکانس ۰/۵ هرتز و دامنه ± 100 ایجاد می‌شود. برای ایجاد سیگنال سرعت به صورت مثلثی به صورت زیر عمل شده است:

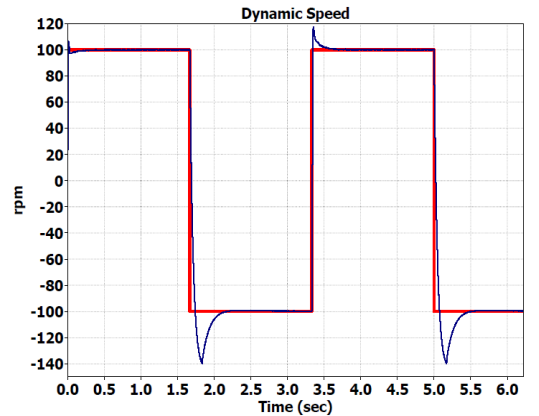
```
Y1 = 0;
VOID LOOP () {
    Y1 = Y1 + S;
    IF (Y1 > 100) {
```



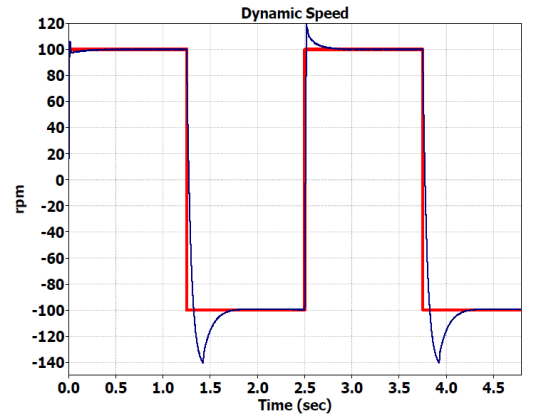
(الف)



(ب)

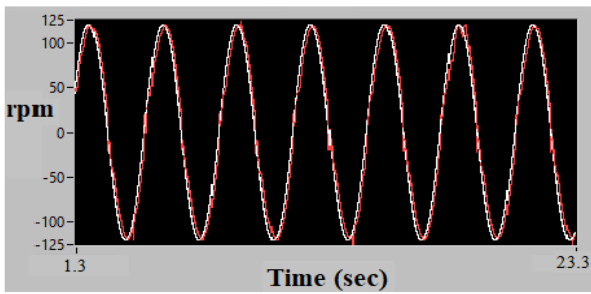


(ج)

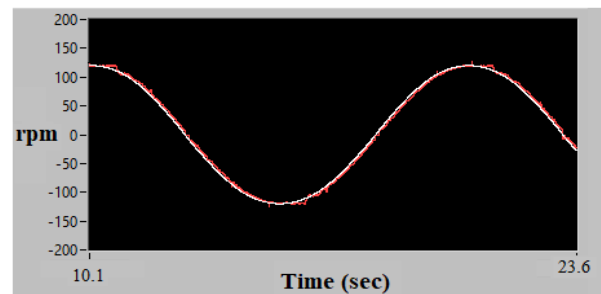


(د)

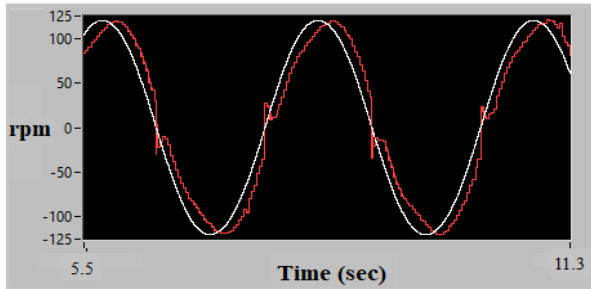
شکل ۱۹: نتایج شبیه‌سازی برای مرجع مربعی با دامنه ۱۰۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ قرمز مرجع و رنگ آبی دور واقعی موتور است.



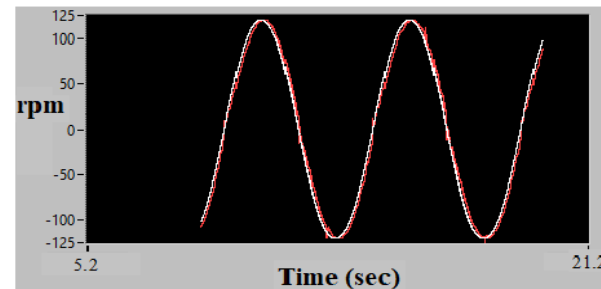
(ج)



(الف)



(د)



(ب)

شکل ۲۲: نتایج آزمایشگاهی برای مرجع سینوسی با دامنه ۱۲۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ سفید مرجع و رنگ قرمز دور واقعی موتور است.

مشخص شده‌اند. مشاهده می‌گردد که در فرکانس‌های ۰/۱ و ۰/۲ هرتز، دور موتور به خوبی مرجع را دنبال نموده است. در دور ۰/۳ هرتز، دور موتور کمی اعوجاج پیدا کرده است. در دور ۰/۴ هرتز، دور موتور با اختلاف فاز و مقداری اعوجاج، مرجع را دنبال کرده است.

شکل ۲۴ نتایج حاصل از تست آزمایشگاهی برای مرجع دندان‌اره‌ای با دامنه ۱۰۰ rpm و ۱۲۰ rpm و با فرکانس‌های ۰/۱ تا ۰/۴ هرتز را نشان می‌دهد. در این شکل، مرجع با رنگ سفید و دور واقعی موتور با رنگ قرمز مشخص شده‌اند. مشاهده می‌گردد که در فرکانس‌های ۰/۱ و ۰/۲ هرتز، دور موتور به خوبی مرجع را دنبال نموده است. البته در حالت تغییر سریع موج دندان‌اره‌ای، دور موتور دارای کمی نوسان می‌باشد. در مرجع ۰/۳ هرتز، دور موتور کمی اختلاف فاز پیدا کرده و در دور ۰/۴ هرتز، دور موتور با اختلاف فاز بیشتری، مرجع را دنبال کرده است.

شکل ۲۵ نتایج حاصل از تست آزمایشگاهی برای مرجع مربعی با دامنه ۳۰ rpm و ۶۰ rpm و با فرکانس‌های ۰/۱ تا ۰/۴ هرتز را نشان می‌دهد. در این شکل، مرجع با رنگ سفید و دور واقعی موتور با رنگ قرمز مشخص شده‌اند. همان‌طور که ملاحظه می‌گردد، در فرکانس‌های ۰/۱ تا ۰/۳ هرتز، دور موتور به خوبی مرجع را دنبال نموده است؛ البته دارای حالت‌های نوسانی گذرا می‌باشد. در مرجع ۰/۴ هرتز، دور موتور با نوسانات بیشتری مقدار مرجع را دنبال می‌نماید.

۶- نتیجه‌گیری

با توجه به مزایایی که موتور BLDC دارد، می‌توان از آن برای کنترل سرعت و به تبع آن موقعیت به صورت دینامیکی در کاربردهای مختلف صنعتی از جمله صنایع رباتیک، اتوماسیون صنعتی، صنایع هوایی، صنایع هوافضا و ... استفاده نمود. با توجه به این موضوع و اهمیت آن در این مقاله، شبیه‌سازی‌هایی بر روی راه‌انداز موتور BLDC با استفاده از نرم‌افزار PLECS صورت گرفت. در این شبیه‌سازی‌ها مراجع سرعت دینامیکی مختلف با شکل‌های سینوسی، مثلثی، دندان‌اره‌ای و مربعی و با فرکانس‌های ۰/۱ تا ۰/۴ هرتز و دامنه بیشینه ۱۲۰ RPM به راه‌انداز موتور اعمال گردید. نتایج شبیه‌سازی نشان داد که موتور BLDC با مشخصات

```

Y1 = 100;
S = -0.01;
}
IF (Y1 < 1) {
Y1 = 1;
S = 0.01;
}
}

```

مقدار اولیه مرجع در این برنامه، صفر در نظر گرفته می‌شود. برای شیب مثبت، مقدار S مثبت در نظر گرفته می‌شود و مقدار $Y1$ با این نرخ افزایش می‌یابد. وقتی که $Y1$ به مقدار ۱۰۰ رسید، مقدار S منفی شده و $Y1$ با نرخ S کاهش می‌یابد. این روند در یک حلقه تکرار می‌شود. برای تولید مرجع دندان‌اره‌ای به صورت زیر عمل می‌شود:

```

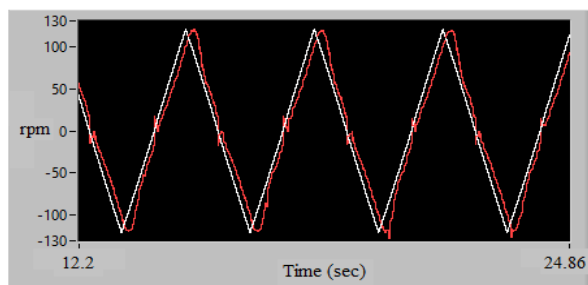
Y1 = -100;
VOID LOOP () {
Y1 = Y1 + 0.01;
IF (Y1 > 100) {
Y1 = -100;
}
}
}

```

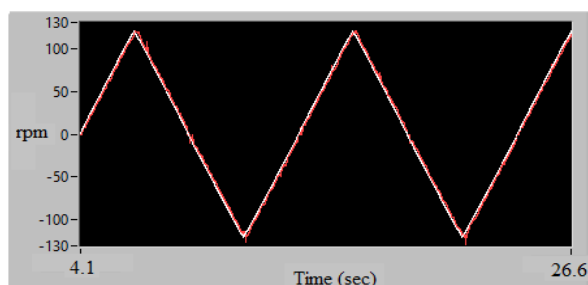
در این برنامه ابتدا مقدار خروجی -100 در نظر گرفته می‌شود و سپس با یک نرخ مشخص افزایش می‌یابد. وقتی مقدار خروجی به مقدار $+100$ (نهایی) رسید، خروجی به مقدار کمینه بازمی‌گردد.

شکل ۲۲ نتایج حاصل از تست آزمایشگاهی برای مرجع سینوسی با دامنه ۱۲۰ RPM و با فرکانس‌های ۰/۱ تا ۰/۴ هرتز را نشان می‌دهد. در این شکل، مرجع با رنگ سفید و دور موتور با رنگ قرمز مشخص شده‌اند. در فرکانس‌های ۰/۱ تا ۰/۳ هرتز، دور موتور به خوبی مرجع را دنبال نموده است. در مرجع ۰/۴ هرتز، دور موتور با کمی اختلاف فاز و مقداری اعوجاج، مرجع را دنبال کرده است.

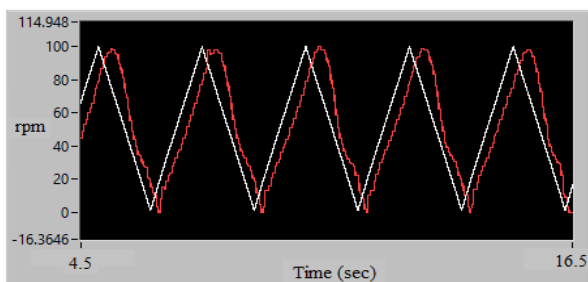
شکل ۲۳ نتایج حاصل از تست آزمایشگاهی برای مرجع مثلثی با دامنه ۱۰۰ RPM و ۱۲۰ RPM و با فرکانس‌های ۰/۱ تا ۰/۴ هرتز را نشان می‌دهد. در این شکل، مرجع با رنگ سفید و دور موتور با رنگ قرمز



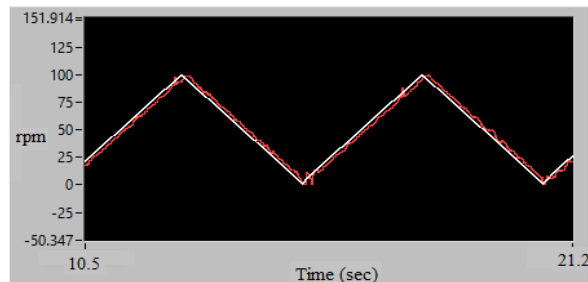
(ج)



(الف)

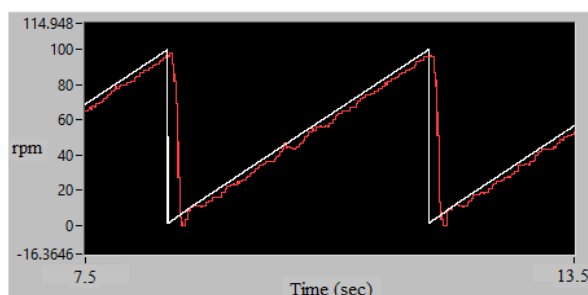


(د)

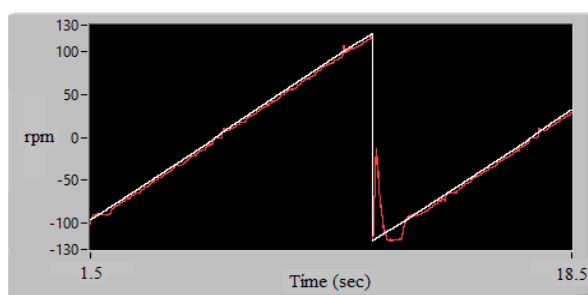


(ب)

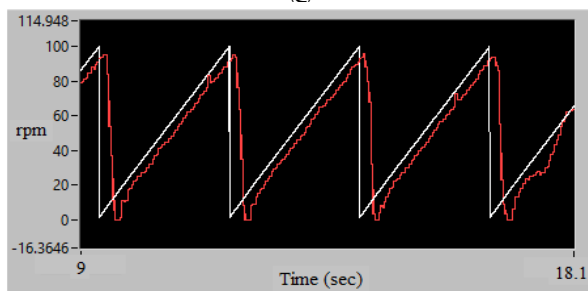
شکل ۲۳: نتایج آزمایشگاهی برای مرجع مثلثی با دامنه ۱۰۰ rpm و ۱۲۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ سفید مرجع و رنگ قرمز دور واقعی موتور است.



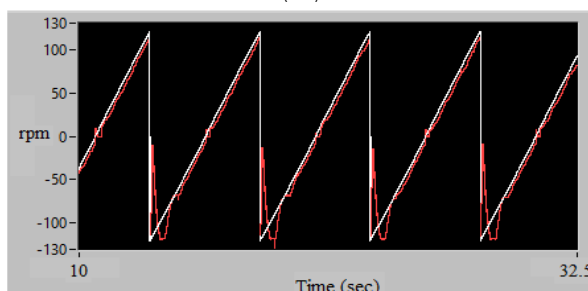
(ج)



(الف)



(د)

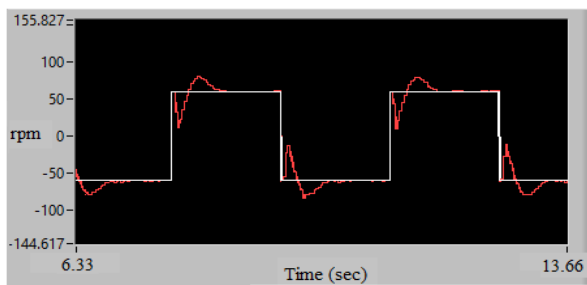


(ب)

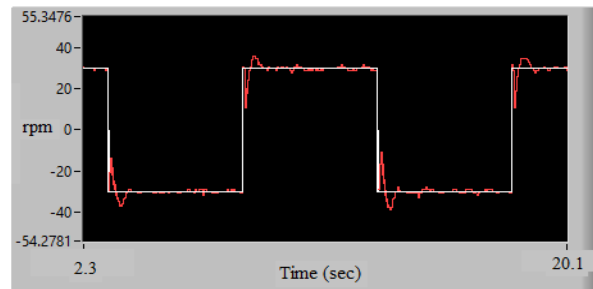
شکل ۲۴: نتایج آزمایشگاهی برای مرجع دندان‌اره‌ای با دامنه ۱۰۰ rpm و ۱۲۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ سفید مرجع و رنگ قرمز دور واقعی موتور است.

جداسازی در نرم‌افزار Labview نمایش داده می‌شود. در Labview، برنامه‌ای برای نمایش داده‌ها، ذخیره آنها در فایل و بازیابی مجدد آنها در گراف نوشته شده است. در این مقاله، برنامه‌های نوشته‌شده در نرم‌افزار Arduino و Labview به تفصیل بررسی گردیده است. تست‌های آزمایشگاهی برای مراجع سینوسی، مثلثی، دندان‌اره‌ای و مربعی با دامنه بیشینه ۱۲۰ RPM و فرکانس بیشینه ۰/۴ هرتز صورت پذیرفته است. نتایج آزمایشگاهی نشان می‌دهد که موتور BLDC با مشخصات ارائه‌شده در این مقاله می‌تواند مرجع سرعت دینامیکی تا فرکانس ۰/۳ هرتز را به خوبی دنبال نماید. در رابطه با مرجع با فرکانس ۰/۴ هرتز، در دور واقعی موتور مقدار کمی اختلاف فاز و یا اعوجاج مشاهده می‌گردد.

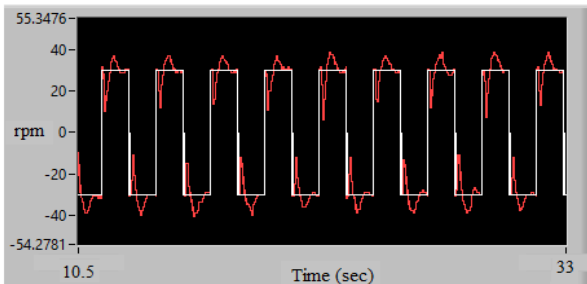
ارائه‌شده در این مقاله به خوبی مقادیر خود را تا فرکانس ۰/۴ هرتز دنبال می‌نماید. در این شبیه‌سازی از یک روش جدید برای به دست آوردن مقدار بیشینه جریان موتور از طریق آشکارساز مقدار بیشینه و استفاده از جریان بیشینه در حلقه کنترل سرعت استفاده شده است. همچنین تست‌های آزمایشگاهی برای کنترل سرعت دینامیکی با استفاده از یک راه‌انداز ۲۰۰ وات و موتور BLDC با توان ۸۴ وات و گیربکس کاهنده ۱ به ۵۰ انجام گردیده است. پیاده‌سازی کنترل‌کننده PID، کنترل جهت چرخش موتور از طریق ارتباط با راه‌انداز موتور BLDC، همراه با تولید مرجع سرعت و محاسبه دور موتور و ارسال داده‌های مرجع و دور موتور به رایانه توسط برد Arduino Mega صورت گرفته است. داده‌های مرجع و سرعت موتور در رایانه از طریق پورت سریال دریافت می‌شود و پس از



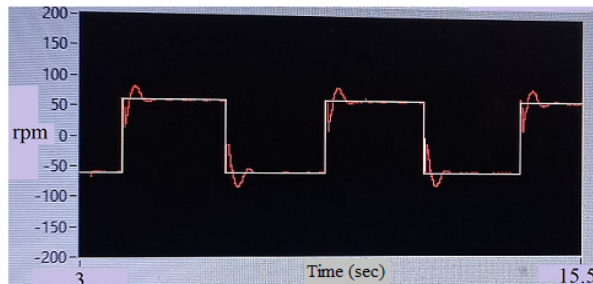
(ج)



(ب)



(د)



(ا)

شکل ۲۵: نتایج آزمایشگاهی برای مرجع مربعی با دامنه ۳۰ rpm و ۶۰ rpm و فرکانس‌های (الف) ۰/۱ هرتز، (ب) ۰/۲ هرتز، (ج) ۰/۳ هرتز و (د) ۰/۴ هرتز. رنگ سفید مرجع و رنگ قرمز دور واقعی موتور است.

- [12] J. Bernat and S. Stepien, "The adaptive speed controller for the BLDC motor using MRAC technique," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 4143-4148, Jun. 2011.
- [13] J. Carlos Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil, "Position and speed control of brushless DC motors using sensorless techniques and application trends," *Sensors*, vol. 10, no. 7, pp. 6901-6947, Jul. 2010.

احمد انتظاری در سال ۱۳۸۴ مدرک کارشناسی مهندسی برق- الکترونیک خود را از دانشگاه صنعتی شاهرود دریافت نمود و پس از چند سال کار تخصصی در صنعت، در سال ۱۳۸۹ در دوره کارشناسی ارشد مهندسی برق- الکترونیک با گرایش مدارات مجتمع آنالوگ در دانشگاه صنعتی مالک اشتر تهران پذیرفته شد و در سال ۱۳۹۱ فارغ التحصیل گردید. وی هم‌اکنون در حال گذراندن دوره دکتری مهندسی برق با گرایش الکترونیک- قدرت در دانشگاه صنعتی مالک اشتر تهران می‌باشد. زمینه‌های علمی و کاری مورد علاقه ایشان عبارتند از: تحلیل و طراحی مدارات الکترونیک آنالوگ و دیجیتال، مدارات شکل‌دهی سیگنال، ابزار دقیق، طراحی و کنترل مبدل‌های الکترونیک قدرت و درایو موتورهای الکتریکی.

آرش دهستانی کلاگر در سال ۱۳۸۴ مدرک کارشناسی مهندسی برق خود را از دانشگاه تهران و در سال ۱۳۸۶ مدرک کارشناسی ارشد مهندسی برق خود را از دانشگاه اصفهان و مدرک دکتری خود را در سال ۱۳۹۲ از دانشگاه علم و صنعت ایران دریافت نمود. نام‌برده از سال ۱۳۹۳ به‌عنوان عضو هیأت علمی در دانشگاه صنعتی مالک اشتر در تهران مشغول به فعالیت گردید. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: الکترونیک قدرت، مبدل‌های توان بالا، فیلترهای اکتیو، کوره‌های قوس الکتریکی و سیستم‌های مغناطیسی.

محمد رضا علیزاده پهلوانی در سال ۱۳۷۶ مدرک کارشناسی مهندسی برق خود را از دانشگاه شهید چمران اهواز و در سال ۱۳۸۰ مدرک کارشناسی ارشد مهندسی برق خود را از دانشگاه صنعتی مالک اشتر در تهران دریافت نمود. از سال ۱۳۷۷ الی ۱۳۸۸ نام‌برده به عنوان محقق سیستم‌های قدرت در مرکز تحقیقات کنترل دانشگاه صنعتی مالک اشتر مشغول به کار بود. در سال ۱۳۸۲ به دوره دکتری مهندسی برق در دانشگاه علم و صنعت ایران وارد گردید و در سال ۱۳۸۸ موفق به اخذ درجه دکتری مهندسی برق از دانشگاه مذکور گردید. ایشان از سال ۱۳۸۸ در مجتمع دانشگاهی برق و کامپیوتر دانشگاه صنعتی مالک اشتر در تهران مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشگاه با مرتبه استادی می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده متنوع بوده و شامل موضوعاتی مانند ماشین‌های الکتریکی و الکترونیک قدرت، سیستم پالسی، شبکه‌های الکتریکی و کنترل می‌باشد.

مراجع

- [1] H. X. Wu, S. K. Cheng, and S. M. Cui, "A controller of brushless DC motor for electric vehicle" *IEEE Trans. on Magnetics*, vol. 41, no. 1, pp. 509-512, Jan. 2005.
- [2] M. Pandi Maharajan and S. Arockia Edwin Xavier, "Design of speed control and reduction of torque ripple factor in BLDC motor using spider based controller," *IEEE Trans. on Power Electronics*, vol. 34, no. 8, pp. 7826-7837, Nov. 2018.
- [3] M. Štulrajter, P. Makyšn, and P. Rafajidus, "Sensorless control of high speed BLDC," in *Proc. IEEE Int. Symp. on Sensorless Control for Electrical Drives*, pp. 225-230, Catania, Italy, 18-19 Sept. 2017.
- [4] R. M. Pindoriya, S. Rajendran, and P. J. Chauhan, "Speed control of BLDC motor using PWM technique," *International J. of Advance Engineering and Research Development*, vol. 6, no. 4, pp. 18-21, Mar. 2014.
- [5] M. Mahmud, S. M. A. Motakabber, A. H. M. Zahirul Alam, and A. Nurashikin Nordin, "Control BLDC motor speed using PID controller," *International J. of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 477-481, 2020.
- [6] D. Mohanraj, R. Arulavid, R. Verma, K. Sathyasekar, A. Bakr Barnawi, B. Chokkalingam, and L. Mihet-popa, "A review of BLDC motor: state of art, advanced control techniques, and applications," *IEEE Access*, vol. 10, pp. 54833-54869, 2022.
- [7] P. Yadav, R. Poola, and K. Najumudeen, "High dynamic performance of a BLDC motor with a front end converter using an FPGA based controller for electric vehicle application," *Turkish J. of Electrical Engineering and Computer Sciences*, vol. 24, no. 3, pp. 1636-1651, Jun. 2016.
- [8] E. Anitha and P. Vairaprakash, "Design of dynamic states and digital speed control of BLDC motor," in *Proc. Int. Conf. on Circuit, Power and Computing Technologies*, 6 pp., Nagercoil, India, 18-19 Mar. 2016.
- [9] J. Shi, Q. Mi, W. Cao, and L. Zhou1, "Optimizing BLDC motor drive performance using particle swarm algorithm? tuned fuzzy logic controller," *SN Applied Sciences*, vol. 4, Article ID: 293, Oct. 2022.
- [10] M. A. A. Hassan, A. R. Abdullah, and N. Bahari, "Speed control analysis of brushless DC motor based on maximum amplitude DC current feedback," in *Proc. 4th Int. Conf. on Production, Energy and Reliability, MATEC Web of Conferences*, vol. 13, 6 pp., 2014.
- [11] I. Faraji Davoudkhani and M. Akbari, "Adaptive speed control of brushless DC (BLDC) motor based on interval type-2 fuzzy logic," in *Proc. 24th Iranian Conf. on Electrical Engineering, ICEE'16*, pp. 1119-1124, Shiraz, Iran, 10-12 May 2016.