

الگوریتمی نوین برای طراحی و پیاده‌سازی سخت‌افزاری شبیه‌ساز بلادرنگ^۱ ماشین‌های الکتریکی بر بستر FPGA به منظور تعامل بهینه در سیستم‌های سخت‌افزار در حلقه^۲

علی‌هادی زاده^۱ متین‌هاشمی^۲ مصطفی پرنیانی^۳

۱- دانش آموخته کارشناسی ارشد- دانشکده مهندسی برق- دانشگاه صنعتی شریف- تهران- ایران
hadizadeh_ali@ee.sharif.edu

۲- استادیار- دانشکده مهندسی برق- دانشگاه صنعتی شریف - تهران- ایران
matin@sharif.edu

۳- استاد- دانشکده مهندسی برق- دانشگاه صنعتی شریف - تهران- ایران
parniani@sharif.edu

چکیده: ماشین‌های الکتریکی از مهم‌ترین عناصر موجود در سیستم‌های انرژی الکتریکی هستند. جهت طراحی و ساخت بهینه ماشین‌های الکتریکی و یا سیستم‌های متصل به آن می‌توان از شبیه‌سازهای بلادرنگ به‌صورت سخت‌افزار در حلقه استفاده نمود. شبیه‌ساز مورد بحث مقاله حاضر یک دستگاه سخت‌افزاری است که به‌صورت بلادرنگ با تجهیزات فیزیکی پیرامونی در تعامل است و جایگزینی برای ماشین الکتریکی در انجام آزمایشها محسوب می‌شود. در واقع در این نوع از شبیه‌سازی امکان بررسی تعامل ماشین و سیستم متصل به آن قبل از ساخت نمونه فیزیکی مهیا می‌گردد. در نتیجه، باعث کاهش زمان ساخت، صرفه‌جویی در هزینه و رسیدن به طراحی بهینه عناصر یک سیستم می‌گردد. در این مقاله به معماری یک شبیه‌ساز بلادرنگ برای ماشین القایی سه‌فاز پرداخته شده‌است. از تکنیک‌های به کار رفته در این مقاله می‌توان در انواع دیگر ماشین نیز استفاده نمود. در این مقاله معادلات ماشین در قالب dq استخراج و سپس به روش اویلر پسرو گسسته شده است. نوآوری مقاله در بهره‌گیری از الگوریتم نوین حل معادلات دیفرانسیل بر مبنای روش شرم-موریسون-وودبریو معماری سخت‌افزاری خلاقانه و بهینه FPGA می‌باشد که معادلات گسسته شده ماشین به روش اویلر پسرو را با دقت مضاعف و در گام زمانی بسیار کوچک ۱۱۰ نانو ثانیه ای حل می‌کند.

کلمات کلیدی: شبیه‌ساز بلادرنگ، سخت‌افزار در حلقه، ماشین الکتریکی، معماری FPGA.

تاریخ ارسال مقاله: ۱۳۹۶/۱۰/۹

تاریخ پذیرش مشروط مقاله: ۱۳۹۷/۰۷/۲۴

تاریخ پذیرش مقاله: ۱۳۹۷/۸/۳۰

نام نویسنده‌ی مسئول: دکتر متین‌هاشمی

نشانی نویسنده‌ی مسئول: ایران - تهران - خیابان آزادی - دانشگاه صنعتی شریف - دانشکده مهندسی برق

۱- مقدمه

شبیه‌سازی بلادرنگ عناصر مختلف شبکه قدرت همواره یکی از نیازهای صنعت برق بوده است. این شبیه‌سازها از زمان طراحی، تولید تا زمان بهره‌برداری و نگهداری یک سیستم قدرت کاربرد دارند. از تولیدکنندگان تجاری این محصول می‌توان به شرکت های RTDS و Opal-RT اشاره کرد.

از مهم ترین شاخص‌های شبیه‌ساز بلادرنگ امکان جایگزینی آن‌ها به جای اجزای اصلی سیستم واقعی و بررسی رفتار سیستم است. به طور مثال، در یک سیستم قدرت که برای اندازه‌گیری موتور الکتریکی از درایو و کنترلرها استفاده می‌شود، اگر بخواهیم درایو جدیدی طراحی و عملکرد آن را بررسی کنیم، در صورت وجود کوچکترین مشکلی در درایو ممکن است به موتور صدمه وارد شود. لذا بهترین کار قرار دادن یک شبیه‌ساز بلادرنگ به جای موتور در سیستم است تا بتواند رفتار موتور را دقیقاً شبیه‌سازی کرده، صحت عملکرد درایو جدید را تایید کند و اگر خطایی در عملکرد آن وجود دارد به راحتی آن را نمایش دهد. تمامی این امور با کمک یک شبیه‌ساز بلادرنگ و با بهره‌گیری از تکنیک سخت‌افزار در حلقه بدون آنکه صدمه‌ای به تجهیزات گران قیمت سیستم مانند موتور وارد گردد میسر می‌شود [۱].

یکی دیگر از مزیت‌های سخت‌افزار در حلقه، کمک به طراحان جهت طراحی اجزایی خاص از یک سیستم است به طوری که عضو جدید بتواند تعاملی بهینه با سیستم داشته باشد. اگر در یک سیستم قدرت قصد طراحی یک موتور الکتریکی را داشته باشیم، جهت یافتن مشخصات و ویژگی‌های خاص این موتور می‌توانیم از این نوع شبیه‌سازها استفاده کنیم. با قرار دادن شبیه‌ساز در سیستم و تغییر پارامترهای داخلی موتور امکان دستیابی به طراحی بهینه و ارزیابی عملکرد موتور برای سیستم واقعی بسیار راحت‌تر فراهم خواهد شد. در صورت نبود این نوع شبیه‌ساز و استفاده از شبیه‌سازهای برون‌خط طراحان مجبور به مدل کردن رفتار بقیه سیستم و طراحی شبیه‌سازی جامع برای کل سیستم هستند که رفتارهای ناشی از بقیه مدل‌سازی‌های سیستم را دریافت کند، اثر آن را بر مدل موتور پیشنهادی بررسی کرده و نتایجی را نشان دهد. در صورتی که یک شبیه‌ساز بلادرنگ با سیستم واقعی مستقیماً در تعامل است و نتایجی که از آن حاصل می‌شود می‌تواند به مراتب سریع‌تر و دقیق‌تر باشد [۳،۲]. فناوری سخت‌افزار در حلقه بسیار قابل اطمینان و موثر در کاهش هزینه‌های طراحی و ارزیابی سیستم است و به جهت بلادرنگ بودن، نتایج آن بسیار سریع‌تر از سایر شبیه‌سازها در دسترس خواهد بود.

در این سیستم، مدت زمان انجام محاسبات برای تولید یک خروجی مطلوب، کمتر و یا حداقل برابر زمان شبیه‌سازی شده است. به عنوان مثال، اگر شبیه‌ساز با گام‌های زمانی ۱۰۰ نانو ثانیه‌کار کند، یعنی اگر هر ۱۰۰ نانو ثانیه یکبار خروجی‌های خود را به روز کند، برای انجام

محاسبات و تولید خروجی صحیح حداکثر ۱۰۰ نانو ثانیه فرصت در اختیار دارد.

شبیه‌سازهای برون‌خط که مدت‌ها قبل طراحی و مورد استفاده قرار می‌گرفتند در مقایسه با شبیه‌سازهای بلادرنگ محدودیت خاصی از لحاظ زمان اجرا و پاسخگویی به کاربر نداشتند؛ لذا در این نوع شبیه‌سازها ممکن است محاسبه نتیجه یک شبیه‌سازی که عملاً چند ثانیه از زمان کارکرد سیستم را شبیه‌سازی می‌کند تا ساعت‌ها به طول انجامد. از این رو، این گونه سیستم‌ها با اینکه نتایج بعضاً دقیقی در اختیار کاربران خود قرار می‌دهند اما به دلیل طولانی بودن زمان محاسبات نمی‌توانند با یک سیستم واقعی در تعامل قرار گیرند و رفتارهای درستی از سیستم شبیه‌سازی شده را ارائه دهند.

با پیشرفت‌های حاصل شده در چند دهه اخیر و افزایش توان پردازشی، امکان شبیه‌سازی بلادرنگ سیستم‌ها و اجزای آن‌ها با دقت و سرعت بالادر FPGA فراهم شده است. در مقالات [۱،۲] شبیه‌ساز بلادرنگ برای ماشین‌های الکتریکی معرفی شده. این شبیه‌سازها دارای مدل با دقت مناسب ولی گام‌های زمانی بزرگ هستند که عملاً از تحلیل وضعیت ماشین در تعامل با اغتشاشات فرکانس بالا و نیز اجزای دیگر مدار مانند مبدل الکترونیک قدرت متصل به ماشین ناتوان می‌مانند. معمولاً ماشین در کنار ادوات دیگر مخصوصاً مبدل الکترونیک قدرت متصل به آن بررسی می‌شود که در این حالت به علت اینکه فرکانس سوییچینگ مبدل خیلی بالاتر است لازم است کل شبیه‌سازی بلادرنگ با گام زمانی بسیار کوچکتری انجام شود. البته شبیه‌سازهای بلادرنگی وجود دارند که اجزای مختلف سیستم را با گام‌های زمانی مختلفی شبیه‌سازی کنند، ولی عموم شبیه‌سازهای بلادرنگ سیستم را با یک گام زمانی ثابت و یکسان تحلیل می‌کنند.

در این مقاله، الگوریتم نوینی جهت حل سریع معادلات دیفرانسیل مربوط به ماشین القایی سه‌فاز پیشنهاد شده است. این الگوریتم که برای بار اول در شبیه‌سازی بلادرنگ ماشین‌های الکتریکی استفاده می‌شود، نیازمند طراحی سخت‌افزاری جدید جهت انجام محاسبات است که این سخت‌افزار پیشنهادی بر روی FPGA از خانواده Xilinx Virtex 6 پیاده‌سازی شده است. استفاده از این الگوریتم و معماری نوین سخت‌افزاری باعث دستیابی به گام‌های زمانی کوچک در عین حال دقت مضاعف می‌شود.

به منظور استفاده از مزیت‌های سیستم سخت‌افزار در حلقه، برد FPGA باید در مدار وارد شود. قدم اول خارج کردن سخت‌افزار مورد شبیه‌سازی از سیستم است. سپس پایانه‌های ورودی سخت‌افزار مورد شبیه‌سازی با کمک مبدل‌های آنالوگ به دیجیتال نمونه‌برداری می‌شوند. این نمونه‌ها در ثباتی جهت استفاده در شبیه‌ساز نگهداری خواهند شد. FPGA این ورودی‌ها را از ثبات خوانده و با توجه به مقادیر اندازه‌گیری شده در آزمایش به صورت بلادرنگ با گام زمانی مشخص، خروجی‌ها را محاسبه خواهد کرد. خروجی‌های محاسبه شده در ثباتی در خروجی ذخیره خواهند شد تا مقادیر آن توسط مبدل

دیجیتال به آناگول به مدار اصلی تحویل داده شود. از آنجا که این شبیه‌سازی بسیار سریع است، مدار حلقه بسته، جایگزین شدن سخت‌افزار اصلی را با شبیه‌ساز احساس نخواهد کرد.

در ادامه این مقاله در بخش دوم به بررسی مدل ماشین القایی پرداخته شده، در فصل سوم بررسی روش‌های گسسته‌سازی مدل صورت گرفته است. در بخش چهارم الگوریتم پیشنهادی جهت حل دستگاه معادلات دیفرانسیل مربوطه ارائه شده است و در بخش پنجم معماری ابتکاری برای پیاده‌سازی بهینه سخت‌افزاری شبیه‌ساز بلادرنگ بر بستر FPGA معرفی شده و در بخش بعد، به تحقیق عملکرد شبیه‌ساز و آزمون و ارزیابی آن پرداخته شده است. در بخش پایانی مقایسه کارهای مشابه بحث شده است.

۲- مدل سازی ماشین القایی:

مدل‌های مداری و ریاضی ماشین‌های کلاسیک سال‌ها مورد بررسی و تحقیق قرار گرفته و برای شبیه‌سازی حالت دائم و گذرای ماشین‌های الکتریکی از دقت مناسبی برخوردار هستند. برای آنالیز ویژگی‌های گذرای الکترومغناطیسی یک سیستم قدرت معمولاً ۳ روش به کار می‌رود [۴-۱].

۱- روش فاز- دامنه (PD)

۲- روش ولتاژ قبل از راکتانس (VBR)

۳- روش تبدیل پارک (DQ)

در نوشتن معادلات ماشین معمولاً از معادلات حالت استفاده شده و متغیرهای حالت را معمولاً مجموعه شارهای الکتریکی درون ماشین و یا جریان‌های آن فرض می‌کنند [۴]. لذا ماتریسی از اندوکتانس‌ها مشخص کننده وضعیت سیستم است. در دو روش اول، ماتریس اندوکتانس ماتریسی متغیر با زمان بوده و در هر گام زمانی باید این ماتریس به روز رسانی شده، معکوس آن محاسبه شده و ضرب‌های ماتریسی برای یافتن خروجی‌های مورد نظر انجام شود. اما، در روش سوم یا تبدیل پارک ماتریس اندوکتانس ثابت خواهد بود. در نتیجه، ماتریس دیگر نیازی به به‌روز شدن ندارد و معکوس آن هم از قبل محاسبه شده است. از این رو، دو روش اول در شبیه‌سازهای برون خط بیشتر کاربرد داشته و خارج از بحث این مقاله هستند. با این وجود، لازمه استفاده از تبدیل پارک، نگاشت ورودی‌های ماشین-ولتاژ ورودی سه‌فاز - به فضای DQ است که نگاشتی غیر خطی است. همچنین، بعد از حل یک گام زمانی می‌بایست وارون تبدیل پارک بر کمیت‌های خروجی اعمال شود تا این متغیرها از فضای DQ به abc نگاشته شوند؛ چراکه در یک سیستم واقعی بر اساس ولتاژها و جریان‌ها در فضای abc اظهار نظر می‌شود [۵]. در نتیجه به دلیل محدود بودن زمان برای حل یک گام زمانی در شبیه‌سازی بلادرنگ و تمایل به بار محاسباتی کمتر، معمولاً در شبیه‌سازی با استفاده از مدل فشرده، از روش DQ و تبدیل پارک برای مدل‌سازی استفاده شده است.

با این حال، تبدیل از فضای abc به DQ به دلیل روش محاسبه و استفاده از نسبت‌های مثلثاتی تقریب‌هایی دارد [۶، ۱]. طبق نتایج شبیه‌سازی اگر برای نسبت‌های مثلثاتی یک جدول ۴۰۹۶ نقطه‌ای برای نیم دوره در نظر بگیریم، هر گام زمانی باید کوچکتر از ۵۰ میکرو ثانیه باشد تا شبیه‌ساز دچار عدم همگرایی نشود. از آنجا که گام‌های زمانی برای شبیه‌سازی در حد چند دهم میکرو ثانیه است غالباً این مسئله چالشی جدی در پیاده‌سازی نخواهد بود؛ چرا که با استفاده از FPGA امکان انجام محاسبات با سرعت بالا در نتیجه حل گام‌های زمانی کوتاه میسر است. با این وجود شایان ذکر است که FPGA بسیار گران‌قیمت بوده و یافتن روشی که در آن به توان گام‌های زمانی بزرگتر را بدون واگرایی نتایج محقق ساخت بسیار ارزشمند خواهد بود؛ چراکه می‌توان به جای استفاده از FPGA از سخت‌افزارهای ارزان‌تری مانند GPGPU یا CPU^۵ نیز استفاده کرد.

در تبدیل متغیرها از فضای abc به DQ روابط زیر به کار گرفته می‌شوند. [۴]

$$f_{dq0s} = K_s f_{abcs} \quad (1)$$

که در (۱) داریم:

$$(f_{dq0s})^T = [f_{qs} \quad f_{ds} \quad f_{0s}] \quad (2)$$

$$(f_{abcs})^T = [f_{as} \quad f_{bs} \quad f_{cs}] \quad (3)$$

$$K_s = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

به منظور طراحی شبیه‌ساز بلادرنگ از مرحله مدل‌سازی تا مرحله ارزیابی، ماشین القایی سه فاز را که از پراستفاده‌ترین ماشین‌های الکتریکی در صنعت است به عنوان مرجع طراحی انتخاب می‌کنیم. مراحل طی شده برای انواع دیگر ماشین نیز به همین صورت است. متغیرهای مورد استفاده در معادلات ماشین در جدول (۱) آمده‌اند.

جدول (۱): تعریف پارامترهای ماشین

متغیر	تعریف	متغیر	تعریف
r_s	مقاومت استاتور	r_r	مقاومت روتور
X_{ls}	رئاکتانس استاتور	X_{lr}	رئاکتانس روتور
X_M	رئاکتانس متقابل	ω_b	فرکانس پایه
p	عملگر دیفرانسیل	ω	سرعت روتور
J	اینرسی	ω_r	سرعت چرخش پنجره گردان
P	تعداد قطب‌ها	B	ضریب اصطکاک

با توجه به (۱) معادلات ولتاژ - جریان ماشین با توجه به مدار معادل مدل ماشین القایی سه‌فاز به صورت زیر به دست خواهد آمد [۴، ۷].

$$C = \begin{bmatrix} \frac{p}{\omega_b} X_M & \frac{(\omega - \omega_r)}{\omega_b} X_M & 0 \\ -\frac{(\omega - \omega_r)}{\omega_b} X_M & \frac{p}{\omega_b} X_M & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} r'_r + \frac{p}{\omega_b} X'_{rr} & \frac{(\omega - \omega_r)}{\omega_b} X'_{rr} & 0 \\ -\frac{(\omega - \omega_r)}{\omega_b} X'_{rr} & r'_r + \frac{p}{\omega_b} X'_{rr} & 0 \\ 0 & 0 & r'_r + \frac{p}{\omega_b} X'_{lr} \end{bmatrix}$$

معادله (۷) بیانگر رابطه دیفرانسیلی بین ولتاژها و جریان‌ها در یک ماشین القایی است. در این معادلات جریان‌ها به عنوان متغیر حالت در نظر گرفته شده‌اند. برای محاسبه شار پیوندی از روی جریان‌ها نیز (۸) به کار گرفته می‌شود [۸].

$$\begin{bmatrix} \psi_{qs} \\ \psi_{ds} \\ \psi_{0s} \\ \psi'_{qr} \\ \psi'_{dr} \\ \psi'_{0r} \end{bmatrix} = \begin{bmatrix} X_{ss} & 0 & 0 & X_M & 0 & 0 \\ 0 & X_{ss} & 0 & 0 & X_M & 0 \\ 0 & 0 & X_{ls} & 0 & 0 & 0 \\ X_M & 0 & 0 & X'_{rr} & 0 & 0 \\ 0 & X_M & 0 & 0 & X'_{rr} & 0 \\ 0 & 0 & 0 & 0 & 0 & X'_{lr} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i'_{qr} \\ i'_{dr} \\ i'_{0r} \end{bmatrix} \quad (۸)$$

۳- گسسته‌سازی معادلات ماشین

به منظور شبیه‌سازی ماشین القایی، معادله (۷) که در بخش قبل به دست آمده است، باید در هر گام زمانی حل گردد. به منظور حل این معادله در سیستم‌های دیجیتال اولین قدم گسسته‌سازی معادلات است. پس از گسسته‌سازی، قابلیت حل عددی معادلات بر بسترهای دیجیتالی از قبیل CPU, GPU, FPGA و... فراهم می‌گردد [۹].

۳-۱- گسسته‌سازی معادلات ماشین القایی به منظور حل سخت‌افزاری

از مهمترین نکات در مورد روش‌های گسسته‌سازی معادلات در شبیه‌سازهای بلادرنگ، سادگی، سرعت و همچنین پایداری آن‌ها است. برای مثال، روش‌هایی که منجر به حل معادلات به شکل بازگشتی شده و یا باعث ناپایداری شوند روش‌های مناسبی برای گسسته‌سازی معادلات نیستند. از این رو به دلیل سادگی و سرعت در حل معادلات، روش اوپلر [۵] در این مقاله انتخاب شده است. روش اوپلر با ساده سازی معادلات و همچنین خطای قابل قبول بهترین روش برای گسسته‌سازی در راستای حل بلادرنگ می‌باشد. اگر چه روش‌هایی مانند رانگ کوتا و دوزنقه‌ای [۱۰] ممکن است در گام‌های زمانی بزرگ دقتی بالاتر از

$$v_{qs} = r_s i_{qs} + \frac{\omega}{\omega_b} \psi_{ds} + \frac{p}{\omega_b} \psi_{qs}, \quad (۴)$$

$$v_{ds} = r_s i_{ds} - \frac{\omega}{\omega_b} \psi_{qs} + \frac{p}{\omega_b} \psi_{ds},$$

$$v_{0s} = r_s i_{0s} + \frac{p}{\omega_b} \psi_{0s},$$

$$v'_{qr} = r'_r i'_{qr} + \left(\frac{\omega - \omega_r}{\omega_b} \right) \psi'_{dr} + \frac{p}{\omega_b} \psi'_{qr}, \quad (۵)$$

$$v'_{dr} = r'_r i'_{dr} - \left(\frac{\omega - \omega_r}{\omega_b} \right) \psi'_{qr} + \frac{p}{\omega_b} \psi'_{dr},$$

$$v'_{0r} = r'_r i'_{0r} + \frac{p}{\omega_b} \psi'_{0r}.$$

برای رسیدن به معادلات ولتاژ-جریان کافیت شار پیوندی را بر حسب جریان‌های مدار بدست آوریم. به این منظور:

$$\psi_{qs} = X_{ls} i_{qs} + X_M (i_{qs} + i'_{qr}), \quad (۶)$$

$$\psi_{ds} = X_{ls} i_{ds} + X_M (i_{ds} + i'_{dr}),$$

$$\psi_{0s} = X_{ls} i_{0s},$$

$$\psi'_{qr} = X'_{lr} i'_{qr} + X_M (i_{qs} + i'_{qr}),$$

$$\psi'_{dr} = X'_{lr} i'_{dr} + X_M (i_{ds} + i'_{dr}),$$

$$\psi'_{0r} = X'_{lr} i'_{0r}.$$

در نهایت با جایگذاری (۶) در (۵) معادلات دیفرانسیلی حوزه زمان ماشین با توجه به معادلات بالا به صورت (۷) در دسترس خواهد بود.

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{0s} \\ v'_{qr} \\ v'_{dr} \\ v'_{0r} \end{bmatrix} = \begin{bmatrix} A_{3 \times 3} & B_{3 \times 3} \\ C_{3 \times 3} & D_{3 \times 3} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i'_{qr} \\ i'_{dr} \\ i'_{0r} \end{bmatrix} \quad (۷)$$

$$A = \begin{bmatrix} r_s + \frac{p}{\omega_b} X_{ss} & \frac{\omega}{\omega_b} X_{ss} & 0 \\ -\frac{\omega}{\omega_b} X_{ss} & r_s + \frac{p}{\omega_b} X_{ss} & 0 \\ 0 & 0 & r_s + \frac{p}{\omega_b} X_{ls} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{p}{\omega_b} X_M & \frac{\omega}{\omega_b} X_M & 0 \\ -\frac{\omega}{\omega_b} X_{ss} & \frac{p}{\omega_b} X_M & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} r_s + \frac{X_{ss}}{\omega_b \Delta t} & \frac{\omega}{\omega_b} X_{ss} & 0 \\ -\frac{\omega}{\omega_b} X_{ss} & r_s + \frac{X_{ss}}{\omega_b \Delta t} & 0 \\ 0 & 0 & r_s + \frac{X_{ls}}{\omega_b \Delta t} \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{X_M}{\omega_b \Delta t} & \frac{\omega}{\omega_b} X_M & 0 \\ -\frac{\omega}{\omega_b} X_{ss} & \frac{X_M}{\omega_b \Delta t} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{X_M}{\omega_b \Delta t} & \frac{(\omega - \omega_r)}{\omega_b} X_M & 0 \\ -\frac{(\omega - \omega_r)}{\omega_b} X_M & \frac{X_M}{\omega_b \Delta t} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} r_r' + \frac{X_{rr}'}{\omega_b \Delta t} & \frac{(\omega - \omega_r)}{\omega_b} X_{rr}' & 0 \\ -\frac{(\omega - \omega_r)}{\omega_b} X_{rr}' & r_r' + \frac{X_{rr}'}{\omega_b \Delta t} & 0 \\ 0 & 0 & r_r' + \frac{X_{lr}'}{\omega_b \Delta t} \end{bmatrix}.$$

در (۱۲) بردار b به شکل (۱۳) محاسبه می‌گردد.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} \frac{X_{ss}}{\omega_b} & 0 & 0 & \frac{X_M}{\omega_b} & 0 & 0 \\ 0 & \frac{X_{ss}}{\omega_b} & 0 & 0 & \frac{X_M}{\omega_b} & 0 \\ 0 & 0 & \frac{X_{ls}}{\omega_b} & 0 & 0 & 0 \\ \frac{X_M}{\omega_b} & 0 & 0 & \frac{X_{rr}'}{\omega_b} & 0 & 0 \\ 0 & \frac{X_M}{\omega_b} & 0 & 0 & \frac{X_{rr}'}{\omega_b} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{X_{lr}'}{\omega_b} \end{bmatrix} \times \begin{bmatrix} i_{qs}(k-1) \\ i_{ds}(k-1) \\ i_{0s}(k-1) \\ i_{qr}'(k-1) \\ i_{dr}'(k-1) \\ i_{0r}'(k-1) \end{bmatrix} + \begin{bmatrix} v_{qs}(k-1) \\ v_{ds}(k-1) \\ v_{0s}(k-1) \\ v_{qr}'(k-1) \\ v_{dr}'(k-1) \\ v_{0r}'(k-1) \end{bmatrix} \quad (13)$$

روش اویلر داشته باشند اما با توجه هدف نهایی که رسیدن به گام‌های زمانی کوچک است، این روش‌ها علی‌رغم محاسبات پیچیده‌تر از نظر دقت، تفاوت چندانی نخواهند داشت اما بار محاسباتی را به شدت افزایش می‌دهند. در نتیجه، روش اویلر برای این نوع شبیه‌ساز مناسب‌تر خواهد بود [۵].

به این ترتیب، در ادامه معادلات ماشین با روش اویلر پس‌روگسسته می‌شود. در این روش برای جملات دیفرانسیلی از تقریبی به‌صورت (۹) استفاده می‌گردد [۱۱].

$$py(k) = \frac{d}{dt} y(k) = \frac{y(k) - y(k-1)}{\Delta t} \quad (9)$$

روند گسسته‌سازی معادلات برای معادله سطر اول دستگاه معادلات دیفرانسیل انجام شده‌است. معادله حوزه زمان به شکل زیر است.

$$v_{qs} = r_s i_{qs} + \frac{X_{ss}}{\omega_b} \frac{d}{dt} (i_{qs}) + \frac{\omega}{\omega_b} X_{ss} i_{ds} + \frac{X_M}{\omega_b} \frac{d}{dt} (i_{qr}') + \frac{\omega}{\omega_b} X_M i_{dr}' \quad (10)$$

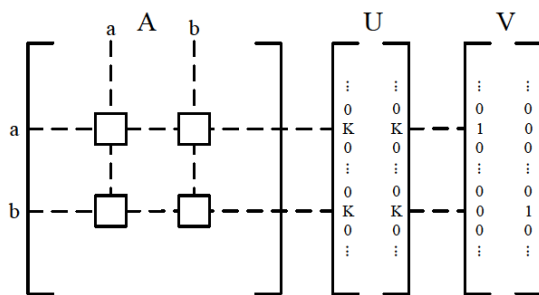
معادله بالا در فرآیند گسسته‌سازی به شکل معادله زیر در می‌آید.

$$v_{qs}(k) = r_s i_{qs}(k) + \frac{X_{ss}}{\omega_b} \left(\frac{i_{qs}(k) - i_{qs}(k-1)}{\Delta t} \right) + \frac{\omega}{\omega_b} X_{ss} i_{ds}(k) + \frac{X_M}{\omega_b} \left(\frac{i_{qr}'(k) - i_{qr}'(k-1)}{\Delta t} \right) + \frac{\omega}{\omega_b} X_M i_{dr}'(k) \quad (11)$$

به همین ترتیب در نهایت مجموعه معادلات توصیف کننده ماشین

به‌صورت (۱۲) در دسترس خواهند بود.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} A_{3 \times 3} & B_{3 \times 3} \\ C_{3 \times 3} & D_{3 \times 3} \end{bmatrix} \begin{bmatrix} i_{qs}(k) \\ i_{ds}(k) \\ i_{0s}(k) \\ i_{qr}'(k) \\ i_{dr}'(k) \\ i_{0r}'(k) \end{bmatrix} \quad (12)$$



شکل (۱): نحوه تولید ماتریس U و V

در شکل (۱) مقدار K برابر مقدار تغییر آن عنصر متناظر در ماتریس A است. برای مثال اگر درایه ماتریس A از ۲ به ۵ در ماتریس جدید تغییر پیدا کند، مقدار k برای المان متناظر در ماتریس U برابر ۳ خواهد بود.

همچنین در (۱۴) ماتریس D که ابعاد آن مناسب با تعداد ستون‌های تغییر یافته در ماتریس A می‌باشد، به صورت زیر محاسبه می‌گردد.

$$D = I + VA_{old}^{-1}U \quad (15)$$

مزیت اصلی این روش در محاسبه معکوس ماتریس‌هایی می‌باشد که درایه‌های محدودی از آن‌ها در هر گام زمانی تغییر می‌کنند. یاب‌به‌صورت دیگر به توان تغییرات ماتریس را به صورت جداگانه و در پالس‌های ساعت متفاوت به سیستم اعمال نمود. چراکه اگر تعداد ستون‌های جدید در ماتریس A زیاد باشد سائز ماتریس D که در هر مرحله معکوس آن باید حساب گردد، به همان میزان افزایش خواهد یافت.

اما مشکل اصلی و اساسی روش شرم-موریسون ناپایداری عددی این روش در استفاده‌های مکرر است. همان طور که از شکل (۲) مشخص است میزان خطای محاسبه ماتریس معکوس به این روش بعد از چند هزار گام زمانی به شکل قابل توجهی افزایش خواهد یافت. لذا به منظور جلوگیری از این خطای ناگهانی در زمان نامشخص روش شرم-موریسون اصلاح شده را ارائه می‌نماییم.

۴-۲- الگوریتم پیشنهادی شرم-موریسون اصلاح شده

در این روش به منظور جلوگیری از خطای محاسبات مکرر در روش شرم با توجه به معادلات ماشین سه مرجع مختلف جهت اصلاح محاسبات در نظر می‌گیریم.

متغیر اصلی در ماتریس A سرعت ماشین است لذا این مراجع با توجه به سرعت ماشین انتخاب می‌گردند. هر مرجع دارای یک محدوده عملکرد خواهد بود و تاثیر این مراجع در محاسبات این گونه خواهد بود که شبیه‌ساز اصلاح ماتریس A را از یک مرجع شروع کرده و تا زمانی که در محدوده عملکرد این مرجع است، پس از هر هزار بار محاسبه مکرر دوباره به مرجع برگشته و محاسبات خود را از مرجع پیگیری می‌کند. به این صورت از اتفاق افتادن خطای زیاد در محاسبات جلوگیری شده و در صورت اتفاق افتادن خطایی اندک هم در زمانی بسیار کم این خطا اصلاح می‌گردد.

اکنون معادلات توصیف کننده رفتار ماشین به شکل گسسته قابلیت پیاده‌سازی بر بسترهای دیجیتالی را خواهند داشت. در مرحله بعدی به طراحی الگوریتمی بهینه و سریع جهت حل این معادلات در FPGA پرداخته می‌شود.

۴- الگوریتم پیشنهادی جهت حل دستگاه معادلات دیفرانسیل به صورت بلادرنگ

به طور کلی حل دستگاه معادلات خطی زمان برترین قسمت شبیه‌سازی بلادرنگ است. روش‌های مختلفی برای حل دستگاه معادلات خطی پیشنهاد شده‌اند. این روش‌ها را می‌توان به دو دسته مستقیم و تکراری تقسیم کرد. در میان روش‌های حل مستقیم دستگاه معادلات روش‌های حذفی گاوس، حذفی گاوس-جردن، تجزیه LU، تجزیه QR و تجزیه Cholesky معروف‌ترین روش‌ها هستند [۱۱] [۱۲-۱۴]. روش‌های تکراری نیز به دلیل زمان بر بودن مناسب استفاده در شبیه‌سازهای بلادرنگ نیستند.

در این مقاله برای اولین بار از روش شرم-موریسون-وودبری برای شبیه‌سازی بلادرنگ ماشین‌های الکتریکی استفاده شده است. این روش به جای آن‌که در هر گام زمانی کل دستگاه معادلات را حل کند، ماتریس معکوس معادلات را به روز رسانی می‌کند.

در واقع در روش SMW یک بار لازم است که ماتریس معکوس دستگاه معادلات (A^{-1}) محاسبه و در حافظه ذخیره شود. سپس در هر مرحله می‌توان با توجه به تغییرات ماتریس اصلی (A)، ماتریس معکوس (A^{-1}) را به روز کرد [۱۵].

این روش برای کاربردهایی مناسب است که تغییرات درایه‌های ماتریس معادلات در گام‌های زمانی متوالی محدود باشد. از آنجایی که در کاربردهای الکترونیک قدرت، تغییرات دستگاه معادلات در گام‌های زمانی متوالی کم است، می‌توان از این روش استفاده کرد. از طرفی با توجه به محاسبات مستقل این روش، امکان پیاده‌سازی موازی آن نیز وجود دارد که امکان دستیابی به گام زمانی کوچک را مهیا می‌کند.

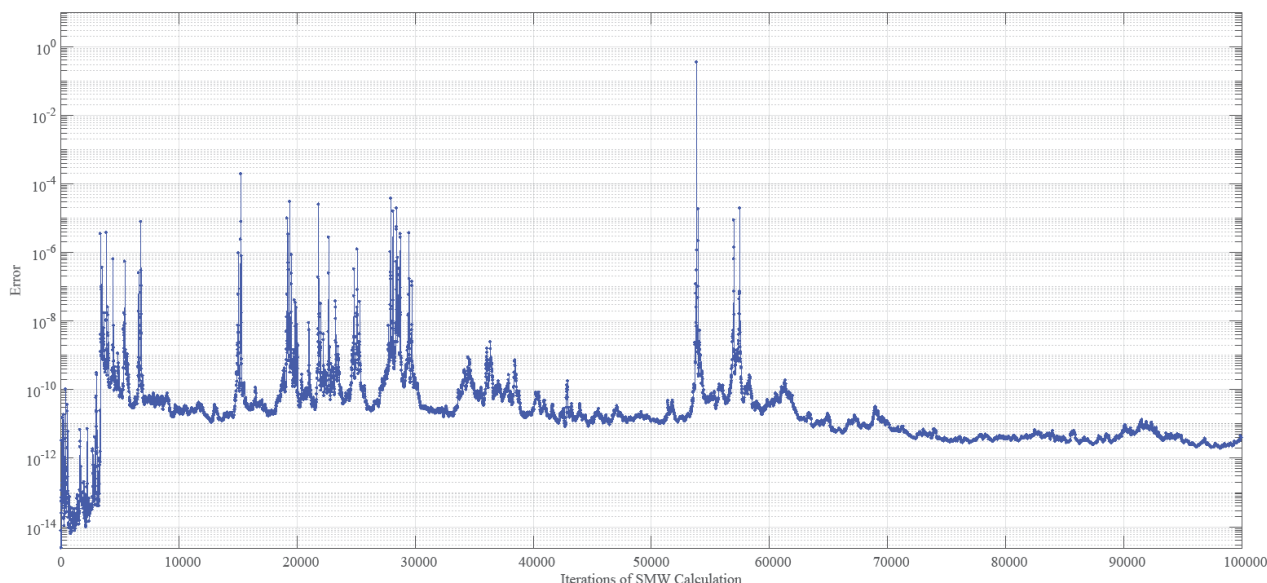
۴-۱- الگوریتم شرم-موریسون-وودبری

روش محاسبه ماتریس معکوس در این الگوریتم در (۱۴) آمده است.

$$A_{new}^{-1} = (A_{old} - UV^T)^{-1} = A_{old}^{-1} - A_{old}^{-1}UD^{-1}V^T A_{old}^{-1} \quad (14)$$

در (۱۴) لازم است ماتریس‌های (بردارهای) U و V در هر گام زمانی با توجه به تغییرات ماتریس جدید نسبت به وضعیت پیشین به دست آیند.

شکل (۱) روش به دست آوردن ماتریس U و V را با توجه به تغییرات ماتریس A قدیمی نسبت به جدید با فرض تغییر چهار درایه در سطر و ستون‌های a و b را نشان می‌دهد.



شکل (۲): نمودار خطا در استفاده مکرر از الگوریتم شرم - موریسون

درایه‌های در سطر چهارم و پنجم ماتریس A هستند، در نتیجه با توجه به خاصیت (۱۶) می‌توان از ترانهاده ماتریس A جهت معکوس‌سازی استفاده کرد.

$$(A^T)^{-1} = (A^{-1})^T \Rightarrow A^{-1} = ((A^T)^{-1})^T \quad (15)$$

در نتیجه در هر گام زمانی با توجه به سرعت جدید ماشین، یک ماتریس معکوس 2×2 باید محاسبه گردد.

همچنین از (۱۲) پیداست رابطه ولتاژ و جریان مولفه صفر تنها به خود وابسته‌اند، لذا این دو معادله را جهت ساده‌سازی از معادلات ماشین حذف کرده و آن‌ها را به شکل موازی هم‌زمان با فرآیند ماتریس معکوس محاسبه می‌کنیم. در نتیجه معادلات ماشین از اندازه ۶ در ۶ به ۴ در ۴ کاهش پیدا می‌کنند.

طراحی و پیاده‌سازی سخت‌افزاری

در این بخش معماری سخت‌افزاری مناسب جهت محاسبه ماتریس معکوس با روش شرم-موریسون اصلاح شده و سپس حل معادلات ماشین ارائه می‌گردد. در این روش با کمک روش شرم در هر گام زمانی عملیات با ترتیب زیر را انجام خواهیم داد.

۱. کنترلر مدار مرجع محاسبات شرم - موریسون را با توجه به سرعت روتور مشخص کرده و ماتریس معکوس متناظر با مرجع در خروجی حافظه در دسترس خواهد بود.
۲. با توجه به سرعت و مرجع استخراج شده، مقادیر U و V تعیین می‌گردند.
۳. با داشتن ماتریس V, U و ماتریس مرجع مقدار D^{-1} محاسبه خواهد شد.

مراجع و محدوده عملکرد آن‌ها با توجه به نوع ماشین القایی به شکل زیر انتخاب می‌گردند.

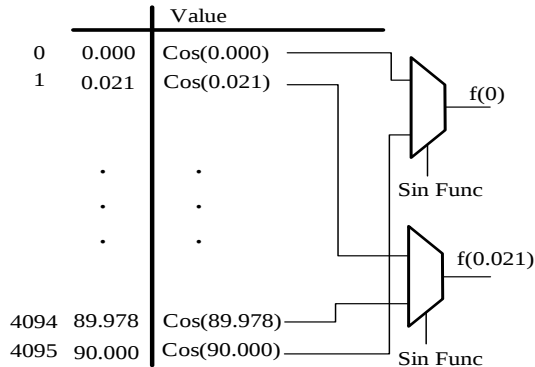
- مرجع ۱: سرعت صفر و محدوده عملکرد از همه مقادیر کمتر تا 0.25 سرعت سنکرون.
- مرجع ۲: سرعت نصف سرعت سنکرون و محدوده عملکرد از 0.25 تا 0.75 سرعت سنکرون.
- مرجع ۳: سرعت سنکرون و محدوده عملکرد از 0.75 تا 1.0 سرعت سنکرون به بالا.

در تقسیم بندی بالا مرجع اول جهت راه‌اندازی در نظر گرفته شده، مرجع دوم در زمان شتاب گیری و بارگذاری موتور کمک زیادی می‌کند و مرجع سوم نیز برای سرعت سنکرون در نظر گرفته شده است. وجود این مراجع و امکان انجام محاسبات مکرر تا ۱۰۰۰ بار در عین حال دسترسی به گام‌های زمانی کوتاه این امکان را فراهم می‌کند تا محاسبات ممیز شناور به دلیل تغییرات کم سرعت نسبت به گام‌های زمانی بسیار سریع با دقت مضاعف صورت پذیرند. شبیه‌سازی های انجام شده برای 10^9 گام زمانی متوالی بدون کوچکترین واگرایی گواه عملکرد صحیح این روش است.

۴-۲-۱- افزایش سرعت الگوریتم با تغییر شکل معادلات

با توجه به (۱۲) در هر گام زمانی تمامی مقادیر به جز مقدار ω که سرعت چرخش روتور در هر گام است، ثابت می‌باشد. لذا تنها درایه‌های دارای ω در هر گام زمانی تغییر می‌کنند. همانطور که از ماتریس مشخص است این متغیر در چهار ستون ماتریس وجود دارد لذا در هر گام زمانی باید یک ماتریس معکوس برای ماتریس D با ابعاد 4×4 را محاسبه نمود. اما با کمی دقت مشخص است که تمام این

بودن طراحی تأثیری ندارد. شکل (۴) نحوه ذخیره سازی مقادیر در حافظه را نمایش می‌دهد. در این شکل مقادیر $\cos \theta$ صفر تا ۹۰ درجه به صورت آینه‌ای به یک مالتی پلکسر متصل می‌شوند و با توجه به مقدار پایه کنترلی مقدار $\sin \theta$ یا $\cos \theta$ به خروجی منتقل می‌گردد. با این تکنیک مقدار حافظه مصرفی نصف می‌گردد.



شکل (۴): نحوه محاسبه توابع غیر خطی مثلثاتی

Block-RAM - ۳-۳-۴

حافظه سریع در این معماری ماتریس‌های A^{-1} متناظر با مراجع را ذخیره می‌کند. در این معماری سه ماتریس متناظر با هر کدام از مراجع ذخیره شده‌است. کنترلر شبیه‌ساز مشخص می‌کند که کدام ماتریس باید در خروجی حافظه در دسترس بقیه سیستم باشد.

UandVGenerator - ۴-۳-۴

در این قسمت با توجه به الگوریتم مشخص شده جهت حل معادلات ماشین ماتریس U دارای شکلی مشخص ولی با مقادیر نامعلوم است که در هر گام زمانی مشخص می‌گردند. اما ماتریس V به طور کامل مشخص شده‌است.

مقادیر K با توجه به اختلاف سرعت این گام زمانی با گام زمانی قبلی (در صورت ثابت بودن مرجع) و یا اختلاف سرعت این گام زمانی با سرعت مرجع مشخص شده توسط این بلوک به دست خواهد آمد.

D^{-1} Generator - ۵-۳-۴

این بلوک وظیفه تولید D^{-1} را بر عهده دارد. به این منظور D از (۱۵) بدست آمده سپس معکوس خواهد شد. با تمهیدات به کار گرفته شده ساینز D همواره یک ماتریس ۲ در ۲ خواهد بود.

SMW Calculation - ۶-۳-۴

این بلوک با در اختیار داشتن تمامی محاسبات قبلی از جمله V, U و D^{-1} طبق (۱۴) ماتریس A^{-1} متناظر با این گام زمانی را تولید خواهند نمود.

Initializer - ۷-۳-۴

این بلوک در اصل وظیفه آماده‌سازی ماتریس b را با توجه به (۱۳) برعهده دارد. مقدار سرعت گام زمانی قبل و همچنین جریان‌های

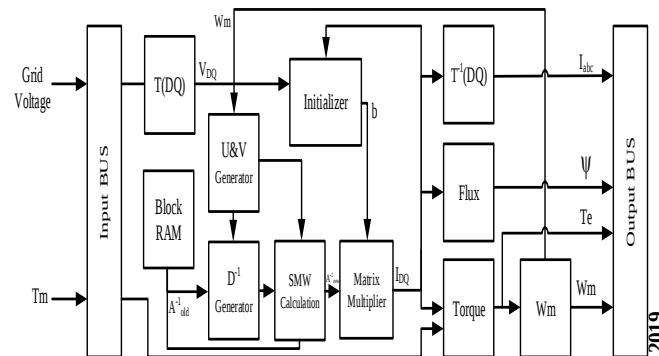
۴. با داشتن تمام پارامترهای لازم، ماتریس معکوس جدید متناظر با این گام زمانی توسط (۱۴) محاسبه می‌گردد.
۵. با به دست آمدن ماتریس معکوس و مشخص شدن جریان‌ها بقیه پارامترهای ماشین محاسبه خواهند شد.

۳-۴ معماری سخت‌افزار

شکل (۳) معماری سخت‌افزار نوینی که به طور خاص برای محاسبه بهینه الگوریتم شرم-موریسون اصلاح شده طراحی گردیده را نمایش می‌دهد. وظایف هریک از بلوک‌های این معماری در ادامه توضیح داده شده‌است.

Output Bus و Input Bus - ۱-۳-۴

این دو باس در اصل مسیرهای ارتباطی پایه‌های ورودی و خروجی FPGA است. در این شبیه‌ساز ولتاژ هر لحظه شبکه به صورت ۳ فاز دریافت شده که بیانگر دامنه و فاز هر لحظه از ولتاژ شبکه است. همچنین گشتاور مکانیکی وارد بر روتور نیز به عنوان ورودی دیگر دریافت می‌گردد.



شکل (۳): سخت افزار پیشنهادی برای شبیه ساز بلادرنگ

در باس خروجی، جریان‌های سیم پیچی استاتور و روتور، شار پیوندی، سرعت و همچنین گشتاور الکتریکی در دسترس هستند.

$T(DQ)$ - ۲-۳-۴

این بلوک وظیفه دارد ولتاژهای دریافت شده از شبکه را به پنجره DQ ببرد. برای این منظور در آن یک جدول قرار داده شده‌است تا بتواند نسبت‌های مثلثاتی را با دقت مناسب محاسبه کند. این جدول بازه صفر تا ۹۰ درجه را به ۴۰۹۶ قسمت تقسیم کرده و مقادیر لازم $\cos \theta$ در آن قرار گرفته‌اند. این مقادیر با توجه به نزدیک ترین زاویه در محاسبه تبدیل DQ استفاده می‌شوند. سپس یک ضرب ماتریسه‌صورت کاملاً موازی انجام می‌گیرد که این عمل با توجه به ممیز شناور بودن اعداد جمعا ۳ پالس ساعت به طول می‌انجامد. البته این عمل با توجه به مستقل بودن نسبت به بقیه محاسبات به صورت موازی انجام شده و در مدت زمان کل محاسبه به دلیل خط لوله‌ای

وابستگی‌های موجود در ذات مسئله و حداکثر موازی سازی زمان‌بندی بهینه شکل (۶) به دست آمده‌است.

۴-۵- نتایج پیاده سازی بر روی FPGA

شبیه‌ساز پیشنهادی جهت ارزیابی و مقایسه نتایج بر روی FPGA پیاده‌سازی شد. به این منظور از FPGA خانواده Virtex6; نوع LX240T بر روی برد ML605 استفاده شد. اعداد با ساختار متمیز شناور و ۳۲ بیتی طبق استاندارد IEEE-754 جهت انجام محاسبات با دقت هر چه تمام پیاده‌سازی شده اند [۱۷][۱۸]. جدول منابع مصرفی و منابع موجود در جدول (۲) آمده‌است.

جدول (۲): منابع مصرفی شبیه‌ساز بلادرنگ

منابع مصرفی	پیاده سازی شبیه‌ساز بلادرنگ	کل منابع موجود
تعداد ثبات	۳۷۶۷۱	۳۰۱۴۴۰
تعداد LUT	۹۰۷۹۲	۱۵۷۲۰۰
حافظه (کیلوبیت)	۱۳۰	۱۴۹۷۶
DSP	۳۲۲	۷۶۸
فرکانس پالس ساعت (مگاهرتز)	۲۰۰	-
دوره پالس ساعت (نانو ثانیه)	۵	-
تعداد پالس ساعت در یک گام زمانی	۲۲	-
گام زمانی (نانو ثانیه)	۱۱۰	-

۵- بررسی صحت عملکرد شبیه‌ساز بلادرنگ

به منظور بررسی و ارزیابی مدل پیشنهادی ماشین القایسه فاز با روتور قفس سنجایی با مشخصات موجود در جدول (۳) مورد شبیه‌سازی قرار گرفت. برای تحقیق صحت رفتار شبیه‌ساز از نرم افزار PSCAD استفاده شده‌است.

جدول (۳): مشخصات ماشین مورد آزمایش

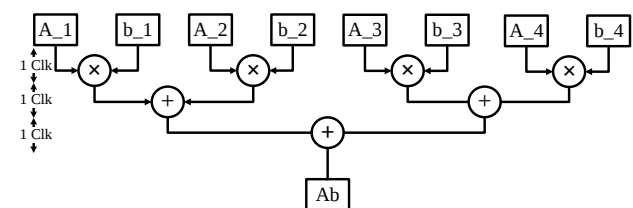
توان نامی	۳۷۳۰ VA	L_r	۰/۵۹۷۴ H
ولتاژ نامی	۷۴۶۰ V	L_M	۰/۲۰۳۷ H
فرکانس	۶۰ Hz	J	۰/۲ Kg m^2
R_s	۱/۵۱۱۵ Ω	B	۰/۵۷۵۲ Nms
L_s	۰/۵۹۷۴ H	P	۴
R_r	۱/۵۰۸۳ Ω	Δt	۱۱۰ ns

جهت بررسی صحت عملکرد و دقت و آزمایش را برای این ماشین طراحی کرده و نتایج حاصل از آن با نتایج حاصل از شبیه‌ساز برون خط مقایسه شده‌است.

خروجی ماتریس ضرب‌کننده اصلی و ولتاژهای حوزه DQ که برای این محاسبات نیاز هستند به این بلوک وارد، و خروجی به شکل ماتریس b برای انجام محاسبات تحویل بلوک ضرب‌کننده اصلی می‌گردد.

۴-۳-۸- Matrix Multiplier

این بلوک که یک ضرب‌کننده کاملاً موازی است محاسبات اصلی ماشین را در ۳ پالس ساعت انجام می‌دهد. در یک پالس ساعت تمام ضرب‌های ماتریسی انجام شده و در ۲ پالس ساعت بعدی چهار عدد حاصل به صورت کاهشی جمع می‌گردند. بلوک دیاگرام یک درایه این واحد در شکل (۵) آمده است.



شکل (۵): نحوه عملکرد بلوک ضرب‌کننده

۴-۳-۹- $T^{-1}(DQ)$

این بلوک همانند بلوک $T(DQ)$ بوده و وظیفه برگرداندن جریان‌های مدار از حوزه DQ به حوزه abc را دارد.

۴-۳-۱۰- Flux

این بلوک ضرب ماتریسی جهت محاسبه شارها را پس از محاسبه جریان‌ها انجام می‌دهد. این عملیات طبق (۸) انجام خواهد شد.

۴-۳-۱۱- Torque

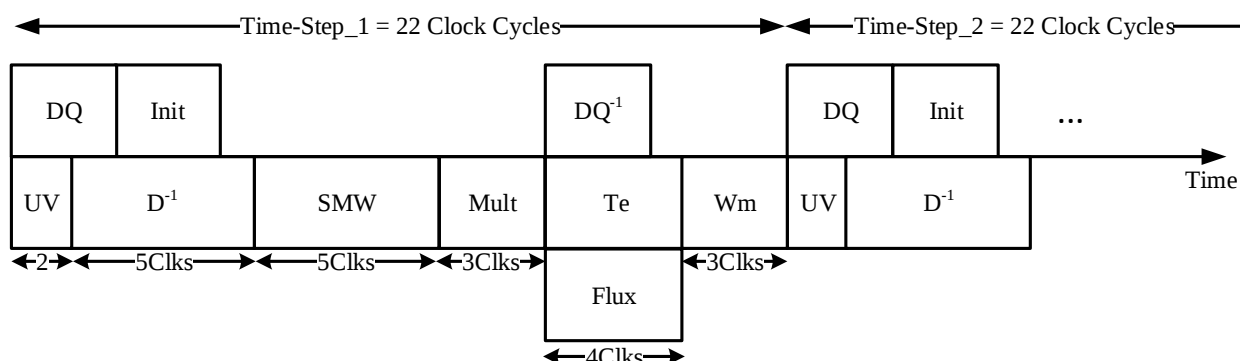
این بلوک نیز گشتاور را از روی جریان‌های مدار محاسبه خواهد کرد [۱۶]. اطلاعات خروجی این بلوک برعکس دو بلوک شار و جریان در حوزه abc برای محاسبات گام زمانی بعد ضروری است. لذا در موازی سازی محدودیت ایجاد خواهد کرد.

۴-۳-۱۲- W_m

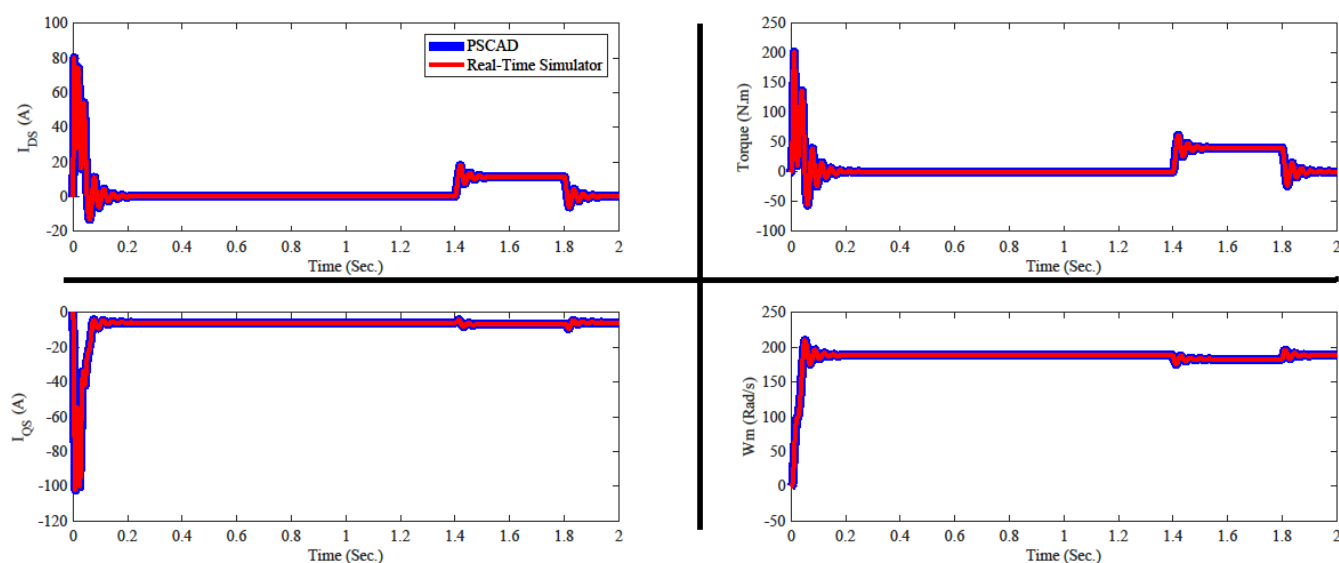
این بلوک طبق رابطه مکانیکی، سرعت در این گام زمانی را محاسبه می‌کند. خروجی این بلوک یکی دیگر از پارامترهای مورد نیاز بلوک Initializer می‌باشد.

۴-۴- نمودار زمان‌بندی عملکرد مدار

به منظور انجام کامل یک دور محاسبات ۲۲ پالس ساعت زمان مورد نیاز خواهد بود. بلوک‌های D^{-1} و SMW هر کدام پنج پالس ساعت، بلوک‌های شار و گشتاور هر کدام چهار پالس ساعت، بلوک تولیدکننده U و V دو پالس ساعت و باقی بلوک‌ها هر کدام به سه پالس ساعت جهت به پایان رساندن وظایف خود نیاز دارند. لذا با توجه به



شکل (۶): نمودار زمانبندی عملکرد سخت افزار



شکل (۷): نمودار نتایج حاصل از آزمایش راه اندازی و بارگذاری

۵-۱- شبیه سازی راه اندازی و بارگذاری ماشین

پیدااست نمودارهای شبیه سازی برون خط و بلادرنگ با دقت بسیار بالایی منطبق هستند.

شکل (۸) نیز نمودار گشتاور-سرعت ماشین را حین این آزمایش نشان می دهد. اهمیت این نمودار در رصد شرایط گذرای ماشین و فرایند رسیدن به حالت پایدار در نقطه کار آن می باشد. همان طور که از شکل پیدااست ماشین در ۲ نقطه به حالت ماندگار رسیده است. ابتدا در نقطه ای با گشتاور تقریباً صفر و سرعت سنکرون که این نقطه بیانگر شرایط بی باری پس از طی گذرای راه اندازی است. همچنین در نقطه ای با گشتاور صفر و سرعتی کمتر از سرعت سنکرون که نشان دهنده زمان بارگذاری ماشین می باشد.

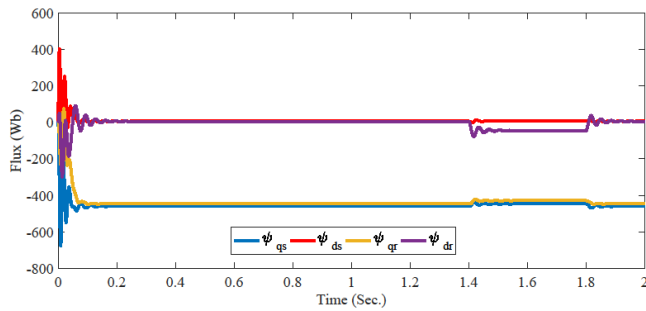
شکل (۹) جریان های استاتور را در فضای abc نشان می دهد به دلیل فشردگی این شکل، در شکل ۱۰ جریان های زمان بارگذاری به صورت بزرگ شده آورده شده اند.

هدف از این آزمایش بررسی دقت و صحت شبیه سازی حالت گذرا و دائمی ماشین القایی در زمان راه اندازی و بارگذاری می باشد. به این منظور خروجی های جریان های استاتور، سرعت، گشتاور الکتریکی از نرم افزار PSCAD به عنوان مرجع و شبیه ساز بلادرنگ پیشنهادی دریافت شدند.

در این آزمایش ماشین در ابتدا به صورت بی بار راه اندازی شده، پس از گذشت ۱/۴ ثانیه پس از طی گذرای راه اندازی و رسیدن ماشین به

سرعت سنکرون، گشتاور بار ۴۰ نیوتن متری به آن اعمال شده و بعد از ۰/۴ ثانیه از اعمال بار و رسیدن به نقطه کار جدید ماشین، گشتاور بار از ماشین برداشته و ماشین دوباره بی بار می گردد.

شکل (۷) شامل جریان های محوره های قائم و عمود در فضای DQ، همچنین سرعت و گشتاور تولیدی موتور در زمان شبیه سازی است. این نمودار ها با انجام آزمایش های یکسان در شبیه ساز پیشنهادی و نرم افزار مرجع جهت مقایسه به دست آمده اند. همان طور که از شکل ها



شکل (۱۲): شارهای ماشین در آزمایش راه اندازی و بار گذاری

میزان خطا و بررسی دقت آزمایشها در جدول (۴) آمده است. این خطاها به صورت P.U. گزارش شده اند.

جدول (۴): محاسبه میزان خطا برای شبیه ساز پیشنهادی

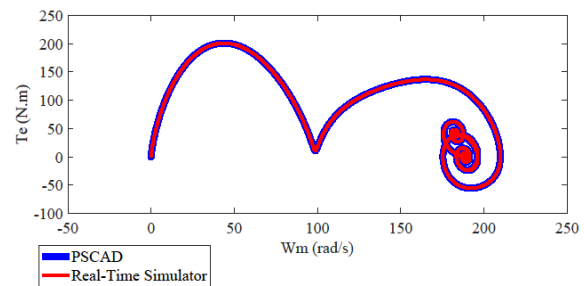
متغیر	خطای اندازه گیری شده
I_{DS}	۰/۰۰۰۲۵ P.U.
I_{QS}	۰/۰۰۰۲۹ P.U.
W_m	۰/۰۰۰۴۳ P.U.
T_e	۰/۰۰۰۳۹ P.U.

۵-۲- آزمایش تغییر ناگهانی فرکانس شبکه:

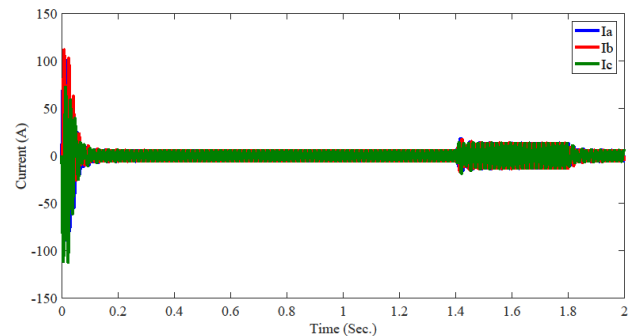
هدف از این آزمایش بررسی دقت و صحت شبیه سازی رویدادهای ناگهانی در حین عملکرد طبیعی ماشین می باشد. در این آزمایش ماشین ابتدا با فرکانس ۶۰ هرتز شروع به کار کرده و بازه نیم ثانیه ای راه اندازی را به طور کامل طی می کند. پس از نیم ثانیه فرکانس شبکه به طور ناگهانی از ۶۰ به ۵۰ هرتز کاهش پیدا می کند. سپس در لحظه ۱/۴ ثانیه گشتاور بار ۴۰ نیوتن متری به ماشین اعمال شده و بعد از ۰/۴ ثانیه این گشتاور بار حذف می شود و ماشین اکنون باید با فرکانس جدید به سرعت سنکرون برسد. به این منظور خروجی های گشتاور الکتریکی و سرعت در شکل (۱۳) گزارش شده اند.

۶- مقایسه با شبیه سازهای موجود

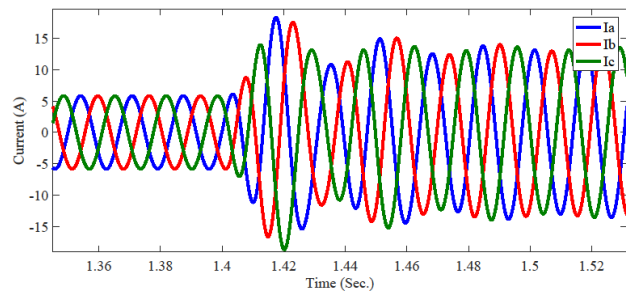
ماشین القایی با توجه به اهمیتی که در صنعت برق دارد همواره مورد توجه طراحان قرار گرفته است. در این قسمت به مقایسه اجمالی با کارهای مشابه در سال های اخیر پرداخته شده است. کارهای موجود از لحاظ سیستم محاسباتی، گام زمانی، منابع مصرفی و نوع FPGA در جدول (۵) مقایسه شده اند. تمامی این شبیه سازها از مدل سازی مشابه استفاده می کنند. همانطور که از نتایج پیداست شبیه ساز پیشنهادی دارای کوتاه ترین گام زمانی به همراه محاسبات ممیز شاور می باشد.



شکل (۸): نمودار گشتاور - سرعت آزمایش راه اندازی و بار گذاری

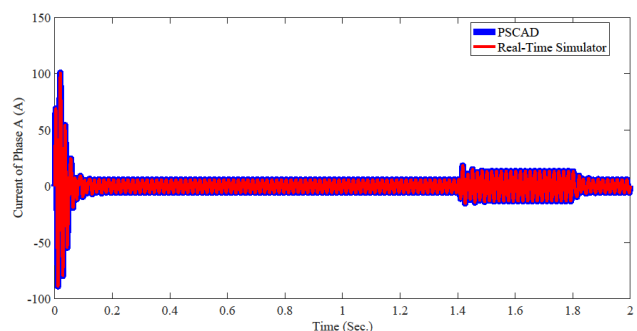


شکل (۹): نمودار جریان های سه فاز موتور در آزمایش راه اندازی و بار گذاری



شکل (۱۰): نمودار بزرگ شده جریان های سه فاز

شکل (۱۱) جهت بررسی دقت در محاسبه جریان ها آورده شده است. این نمودار جریان فاز اول استاتور را در شبیه ساز پیشنهادی و نرم افزار PSCAD مقایسه می کند.



شکل (۱۱): نمودار مقایسه جریان یک فاز شبیه ساز و مرجع

شکل (۱۲): شارهای محورهای مختلف روتور و استاتور را نمایش می دهد.

نتایج حاصل از مقالات در جدول (۵) ذکر شده اند. نتایج گزارش شده در جدول زیر نتایج تطبیق داده شده با مدل و دقت استفاده شده در این مقاله هستند، تا مقایسه منصفانه باشد.

البته مدلها و شبیه‌سازهای بسیار دیگری نیز برای ماشینهای الکتریکی و دیگر سیستمهای قدرت وجود دارند [۲۳-۲۵] که مخصوص شبیه‌سازی بلادرنگ نیستند و بنابراین مستقیماً قابل مقایسه با روش ارائه شده در این مقاله نمی‌باشند.

جدول (۵): مقایسه شبیه‌سازهای موجود

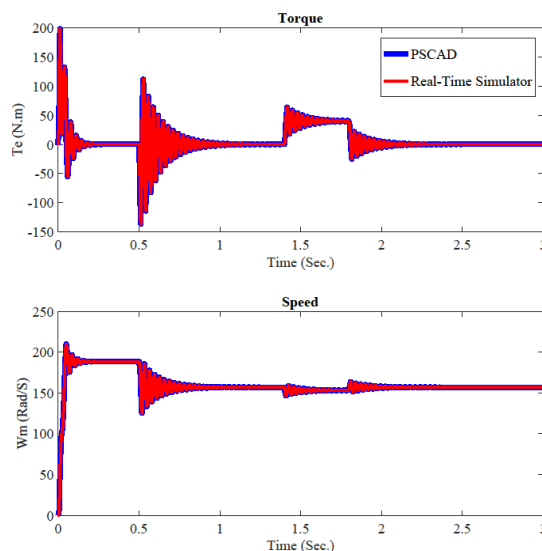
مرجع	خانواده FPGA	سیستم محاسباتی	گام زمانی (نانوثانیه)	متوسط منابع
[۱]	Virtex 7	ممیز ثابت	۲۳۰	٪۱۰
[۲]	Virtex 6	ممیز ثابت	۲۰۰	٪۳۵
[۳]	Virtex 5	ممیز شناور	۱۱۰۰۰	٪۸۰
[۱]	Virtex 7	ممیز شناور	۱۷۵۵	٪۱۵
[۱۹]	Virtex 6	ممیز شناور	۴۰ (اویلر پیشرو)	٪۲۵
این مقاله	Virtex 6	ممیز شناور	۱۱۰	٪۵۳

۷- نتیجه‌گیری

در این پژوهش با پیشنهاد روش جدیدی جهت حل معادلات دیفرانسیل و ماتریس معکوس و همچنین طراحی سخت‌افزاری نوین جهت پیاده‌سازی مدل، دستیابی به شبیه‌ساز بلادرنگ با قابلیت استفاده به صورت سخت‌افزار در حلقه فراهم آمد. این شبیه‌ساز با بهره‌گیری از سیستم عددی ممیز شناور علاوه بر دقت مضاعف محاسبات، سریع‌ترین شبیه‌ساز ارائه شده تاکنون می‌باشد. به طور خلاصه می‌توان مراحل طراحی و پیاده‌سازی یک شبیه‌ساز بلادرنگ برای ماشین الکتریکی را در مدل‌سازی مناسب، گسسته‌سازی معادلات، حل کردن سریع معادلات و طراحی سخت‌افزار خلاصه کرد. پس از پیاده‌سازی، ارزیابی و صحت‌سنجی عملکرد شبیه‌ساز نیز ضروری می‌باشد.

مراجع

- [1] N. R. Tavana and V. Dinavahi, "A general framework for fpga-based real-time emulation of electrical machines for hil applications," IEEE Transactions on Industrial Electronics, vol. 62, no. 4, pp. 2041–2053, April 2015.
- [2] M. Matar and R. Irvani, "Massively parallel implementation of ac machine models for fpga-based real-time simulation of electromagnetic transients," IEEE Transactions on Power Delivery, vol. 26, no. 2, pp. 830–840, April 2011.
- [3] L. Wang, J. Jatskevich, V. Dinavahi, H. W. Dommel, J. A. Martinez, K. Strunz, M. Rioual, G. W. Chang, and R. Irvani, "Methods of interfacing rotating machine models in transient simulation programs," IEEE Transactions on Power Delivery, vol. 25, no. 2, pp. 891–903, April 2010.



شکل (۱۳): نمودار سرعت و گشتاور در آزمایش تغییر فرکانس شبکه

در [۲۲، ۲۱] روش شرمین برای مبدل‌های الکترونیک قدرت مورد استفاده قرار گرفته است که به طور کل متفاوت از این مقاله است و روابط ریاضی و پیاده‌سازی سخت‌افزاری کاملاً متفاوتی را می‌طلبد.

در [۱] یک مجموعه کلی برای شبیه‌سازهای بلادرنگ ماشین‌های الکتریکی تدوین شده است. در این مجموعه مراحل مدل‌سازی و گسسته‌سازی مشابه همین مقاله پیگیری شده‌اند. وجه تمایز اصلی این شبیه‌ساز با شبیه‌ساز پیشنهادی نحوه حل معادلات دیفرانسیل (معادلات حالت) سیستم است. در آن مقاله نویسندگان معادلات حالت را پس از گسسته‌سازی به روش اویلر پسرو به صورت ممیز ثابت و ممیز شناور با تکنیک‌های متداول حل کرده‌اند. بررسی تاثیر ممیز ثابت و ممیز شناور بر دقت معادلات و سرعت شبیه‌ساز از نقاط قوت آن مقاله است. مهمترین دستاورد شبیه‌ساز ارائه شده در این مقاله نسبت به مقاله مورد نظر روش حل نوین معادلات دیفرانسیل است که باعث شده گام زمانی ۱۰۶ برابر در حالت ممیز شناور از [۱] که سریع‌ترین شبیه‌ساز تا قبل از این مقاله است، کوتاه‌تر باشد. جالب توجه است که شبیه‌ساز ارائه شده در حالت ممیز شناور حتی از [۱] در حالت ممیز ثابت هم سریع‌تر است. FPGA استفاده شده در مقاله حاضر نسبت به FPGA استفاده شده در [۱] نیز بسیار ارزان‌تر و ساده‌تر است.

در [۲] نویسندگان تمرکز ویژه‌ای بر روی نحوه اتصال سیستم‌های قدرت از مجله مبدل‌های الکترونیک قدرت به ماشین‌های AC داشته‌اند. به این منظور به شبیه‌سازی ماشین الکتریکی و مبدل‌های قدرت پرداخته‌اند. محاسبات انجام شده در این مقاله با روش ممیز ثابت بوده و از مدل ماشین القایی و ماشین سنکرون مغناطیس دائم برای شبیه‌سازی استفاده شده است. در [۳] تلاش شده تا روشی جهت تعامل ماشین‌های گردان با برنامه‌های تحلیل حالت گذرای مغناطیسی (EMTP) ارائه گردد، در یکی از روش‌های بحث شده، نویسندگان پیشنهاد کرده‌اند که محاسبات مرتبط به ماشین و مبدل‌های قدرت به دلیل امکان پردازش موازی و سخت‌افزار در حلقه، در محیط FPGA انجام گردد و سپس نتایج حاصل از آن به سیستم مادر منتقل شود.

دومین کنفرانس بین المللی پژوهش‌های دانش بنیان در مهندسی کامپیوتر و فناوری اطلاعات، تهران.

- [20] R. Rohrer and H. Nosrati, "A modified forward Euler approach to stable step response computations," in IEEE Transactions on Circuits and Systems, vol. 28, no. 3, pp. 180-186, March 1981.
- [21] A. Hadizadeh, M. Hashemi, M. Labbaf, M. Parniani, "A matrix-inversion technique for FPGA-based real-time EMT simulation of power converters," IEEE Transactions on Industrial Electronics (TIE), vol. 66, no. 2, February 2019.
- [22] Z. Huang and V. Dinavahi, "A fast and stable method for modeling generalized nonlinearities in power electronic circuit simulation and its real-time implementation," in IEEE Transactions on Power Electronics.
- [23] Khosravi M, Hossini Aliabadi M, Javadi S. Double-Fed Induction Generator Optimization to Improve the Performance of Variable Speed Wind Turbine Using Finite Element Method. Journal of Iranian Association of Electrical and Electronics Engineers. 2018; 15 (1):103-112
- [24] Tohidi S. Analysis of Steady State Performance of Brushless Doubly Fed machine. Journal of Iranian Association of Electrical and Electronics Engineers. 2017; 14 (1):119-125
- [25] Lesani H, Nasiri Gheydari Z, Tootoonchian F. A New Method for Dynamometry of Single Phase Induction Motors. Journal of Iranian Association of Electrical and Electronics Engineers. 2008; 5 (1):68-79
- [4] P. Krause, O. Wasynczuk, S. Sudhoff, and I. P. E. Society, Analysis of electric machinery and drive systems, ser. IEEE Press series on power engineering. IEEE Press, 2002.
- [5] N. Watson, J. Arrillaga, and Institution of Electrical Engineers, Power Systems Electromagnetic Transients Simulation, ser. Energy Engineering Series. Institution of Engineering and Technology, 2003.
- [6] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," IEEE Transactions on Industrial Electronics, vol. 59, no. 2, Feb 2012.
- [7] N. R. Tavana and V. Dinavahi, "Real-time nonlinear magnetic equivalent circuit model of induction machine on fpga for hardware-in-the-loop simulation," IEEE Transactions on Energy Conversion, vol. 31, no. 2, pp. 520-530, June 2016.
- [8] B. Jandaghi and V. Dinavahi, "Hardware-in-the-loop emulation of linear induction motor drive for maglev application," IEEE Transactions on Plasma Science, vol. 44, no. 4, pp. 679-686, April 2016.
- [9] V. Jalili-Marandi and V. Dinavahi, "SIMD-based large-scale transient stability simulation on the graphics processing unit," IEEE Transactions on Power Systems, vol. 25, no. 3, pp. 1589-1599, Aug 2010.
- [10] F. E. Fleming and C. S. Edrington, "Real-time emulation of switched reluctance machines via magnetic equivalent circuits," IEEE Transactions on Industrial Electronics, vol. 63, no. 6, pp. 3366-3376, June 2016.
- [11] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics," in Power and Energy Society General Meeting, 2011 IEEE, July 2011, pp. 1-11.
- [12] M. A. T. Montero, A. E. H. Garces, J. A. T. Ladino, and R. R. W. Collao, "Simulation and analysis of an axial flux induction machine," IEEE Latin America Transactions, vol. 15, no. 7, pp. 1263-1269, 2017.
- [13] J. R. Marti and J. Lin, "Suppression of numerical oscillations in the EMTP power systems," IEEE Transactions on Power Systems, vol. 4, no. 2, pp. 739-747, May 1989.
- [14] J. P. David, "Low latency and division free gauss-jordan solver in floating point arithmetic", Journal of Parallel and Distributed Computing, vol. 106, pp. 185 - 193, 2017.
- [15] W. W. Hager, "Updating the inverse of a matrix," SIAM Review, vol. 31, no. 2, pp. 221-239, 1989.
- [16] T. O. Bachir, C. Dufour, J. David, J. Bélanger, and J. Mahseredjian, "Reconfigurable floating-point engines for the real-time simulation of pems: A high-speed pmsm drive case study," Proceedings of Electrimacs (ELECTRIMACS-2011), Cergy-Pontoise, France, 2011.
- [17] B. N. Datta, Numerical Linear Algebra and Applications, Second Edition, 2nd ed. SIAM, 2010.
- [18] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," IEEE Transactions on Industrial Electronics, vol. 58, no. 6, pp. 2547-2555, June 2011.

زیر نویس‌ها

¹ Real-time simulator

² Hardware-in-the-loop (HIL)

³ Offline

⁴ General-Purpose Graphics Processing Unit

⁵ Central Processing Unit

⁶ (SMW) Sherman-Morrison-Woodbury

- [۱۹] علی هادی زاده، متین هاشمی، و مصطفی پرنیانی، ۱۳۹۶، طراحی و پیاده سازی سخت افزاری شبیه ساز بلادرنگ ماشین‌های الکتریکی بر بستر FPGA به منظور استفاده در سیستم‌های سخت افزار در حلقه،