

پایدارسازی و کنترل سیستم قدرت با استفاده از الگوریتم‌های فراابتکاری

مهدی صبری*

گروه برق، واحد بین‌المللی جلفا، دانشگاه آزاد اسلامی، جلفا، ایران

تاریخ دریافت مقاله: ۱۳۹۶/۰۵/۱۴ تاریخ پذیرش نهایی: ۱۳۹۶/۰۹/۲۶

چکیده

الگوریتم جست‌وجوی کوکو (CS) در این مقاله برای نصب بهینه‌ی پایدارساز سیستم قدرت (PSS) طراحی شده در یک سیستم قدرت چندماشینه معرفی شده است. مشکل تنظیم پارامتر PSS به‌عنوان یک مسئله‌ی بهینه‌سازی که توسط الگوریتم CS حل شده، فرموله شده است. یک مقدار ویژه براساس تابع هدف شامل نسبت میرایی و فاکتور میرایی از حالت‌های الکترومکانیکی میراشونده‌ی کُند برای مشکل طراحی PSS در نظر گرفته شده است. عملکرد CS که براساس PSSها (CSPSS) با الگوریتم ژنتیک (GA) ارائه شده، بر پایه‌ی PSS (GAPSS) و PSSها (CPSS) متعارف در شرایط عملیاتی و اختلالات مختلف مقایسه شده و نتایج حاصل از CSPSS از طریق تجزیه و تحلیل در حوزه‌ی زمان توسعه داده شده و مقادیر ویژه و شاخص‌های عملکردی تأیید شده است. همچنین اثر الگوریتم پیشنهادی در ارائه‌ی ویژگی‌های میرایی خوب، به تأیید رسیده است.

واژه‌های کلیدی

انتگرال قدر مطلق مقدار خطا (IAE)، انتگرال قدر مطلق خطای ضرب‌شده در زمان (ITAE)، پایدارساز سیستم قدرت (PSS)، پایدارساز سیستم قدرت مرسوم (CPSS)، پایدارساز سیستم قدرت بر مبنای الگوریتم ژنتیک (GAPSS)، پایدارساز سیستم قدرت بر مبنای جست‌وجوی کوکو (CSPSS).

۱. مقدمه

سیستم‌های قدرت ممکن است نوسانات پایدار را تجربه کنند. این نوسانات ممکن است به قطع برق جزئی یا کلی بینجامد (Rogers, 2000). یکی از انواع معروف نوسانات، حالت‌های بین منطقه‌ای است که از نوسانی بودن محوطه‌ی یک نسل با توجه به مناطق دیگر نتیجه می‌شود. نوع دیگر، نوسانی بودن ژنراتورها در یک منطقه است که در برابر یکدیگر قرار دارند و به‌عنوان حالت محلی شناخته شده است (Eremia & Shahidehpour, 2013; Machowski et al., 2009). پایدارساز سیستم‌های قدرت (PSS) برای ارائه‌ی یک سیگنال کنترل کمکی به سیستم تحریک به‌منظور کاهش هر دو نوع از نوسانات استفاده می‌شود (Machowski et al., 2009).

در چند سال گذشته، روش‌های هوش مصنوعی (AI) برای حل مشکلات مرتبط با طراحی PSS بررسی شده است. تحقیقات چندی درباره‌ی شبکه‌ی عصبی مصنوعی (ANN) برای طراحی PSS صورت گرفته است (Kumar et al., 2001; El-Razaz & El-hameed, 2001; Segal et al., 2004). این شبکه عملکرد سیستم را افزایش می‌دهد، اما مشکل اصلی این کنترل‌کننده، زمان آموزش طولانی و انتخاب تعداد لایه‌ها و سلول‌های عصبی در هر لایه است. یکی دیگر از روش‌های کنترل منطق فازی (FLC) توجه زیادی را در برنامه‌ی کنترل جلب کرده است (Megala & Rajan, 2014; Rout & Panda, 2010). نیازی به یک مدل مناسب بر مبنای محور S نخواهد بود و می‌توان آن را براساس اطلاعات زبانی به‌دست آمده از دانش قبلی از سیستم کنترل طراحی کرد؛ با این حال، نتایجی بهتر از کنترل متعارف به‌دست می‌آید. دریافت سیگنال‌های مؤثر در هنگام طراحی FLC کاری سخت و اجتناب‌ناپذیر است. روش‌های قوی مانند H_{∞} (Bica & Dulau, 2015)، H_2 (Peng et al., 2011) و μ - سنتز (Yue & Schluter, 2003) نیز برای طراحی PSS استفاده شده است. این روش‌ها تکراری‌اند و عدم قطعیت سیستم باید در یک شکل خاص انجام گیرد. از دیگر سو، ترتیب تثبیت‌کننده‌ها به پایداری براساس محور s است و ساختار پیچیده از جمله تثبیت‌کننده را افزایش و کاربرد آنها را کاهش می‌دهد. در تحقیقی، روش تغییر قطب برای طراحی PSS نشان داده شده (El-Sherbiny et al., 2003)، اما این روش پیچیدگی الگوریتم محاسباتی و مشکل ذخیره‌سازی حافظه داشته و فرض در دسترس بودن کامل را دارد.

به‌تازگی، الگوریتم‌های بهینه‌سازی به مشکل طراحی PSS افزوده شده‌اند. روش شبیه‌سازی آیلینگ (SA) در تحقیقی (Abdio, 2000) برای پارامترهای تنظیم PSS ارائه شده، اما ممکن است با گرفتن بهینه‌ی محلی شکست بخورد. یکی دیگر از روش‌ها مانند جست‌وجوی ممنوع (TS) در تحقیق دیگری (Abdio, 1999) (Abdel-Magid & 1999) برای طراحی PSS معرفی شده است. به‌رغم اثر ابزار بهینه‌سازی در طراحی

PSS، بهره‌وری با استفاده از توابع هدف افزایش و تعداد زیادی از پارامترهای بهینه‌سازی شده کاهش می‌یابد. همچنین این روش بسیار وقتگیر است. الگوریتم ژنتیک (GA) در (Abodo & Magid- Abdel, 2000) برای طراحی بهینه‌ی PSS توسعه یافته است و به زمان اجرای بسیار طولانی بسته به اندازه‌ی سیستم تحت مطالعه نیاز دارد. همچنین با تکرار بازنگری راه‌حل‌های بهینه را افزایش می‌دهد. بهینه‌سازی ازدحام ذرات (PSO) برای طراحی پارامترهای PSS بررسی شده است (Jalalivand & Khalkhali, 2011). در الگوریتم PSO به صورتی خوش‌بینانه است که سبب تنظیم مناسب سرعت و جهت آن می‌شود. علاوه بر این، الگوریتم نمی‌تواند مشکلات پراکندگی و بهینه‌سازی را برطرف کند و از همگرایی آهسته در مرحله‌ی جست‌وجوی انتخابی، توانایی جست‌وجوی محلی مناسبی ندارد که ممکن است به مشکل حداقل راه‌حل محلی منجر شود. یک الگوریتم محاسبات تکاملی به نسبت جدیدتر، به نام باکتری جست‌وجوگر (BF) در پژوهشی (Passino, 2002) نشان داده شده و به‌تازگی توسط محقق (Abd-Elazim & Ali, 2013) منتشر شده است. الگوریتم BF به جهت جست‌وجوی تصادفی بستگی دارد که ممکن است به تأخیر در رسیدن به راه حل جهانی بینجامد.

یک الگوریتم فوق‌ابتکاری جدید شناخته شده به‌عنوان الگوریتم CS، براساس زندگی یک خانواده‌ی پرنده است که برای طراحی بهینه‌ی پارامترهای PSS بررسی شده است. مسئله‌ی طراحی PSS، یک عمل بهینه‌سازی با استفاده از الگوریتم CS است. تثبیت‌کننده‌ها برای تغییر همه‌ی حالت‌های الکترومکانیکی به یک منطقه‌ی از پیش تعریف شده براساس محور S تنظیم شده‌اند تا ثبات نسبی را اعلام کنند. اعتبارسنجی CSPSS روی یک سیستم قدرت چندماشین در حالت‌های عملیاتی متعدد در مقایسه با GAPSS و CPSS که از طریق تجزیه و تحلیل در حوزه‌ی زمان، مقادیر ویژه و شاخص‌های عملکردی اعمال می‌شود، توسعه داده شده است. نتایج ارزیابی اطمینان می‌دهد که الگوریتم پیشنهادی، عملکردی قوی را برای شدت نوسانات در حالت‌های مختلف بارگذاری و اختلالات فراهم می‌کند.

۲. فرمولاسیون مسئله

۲-۱. مدل سیستم قدرت

یک سیستم قدرت به‌طور کلی می‌تواند با مجموعه‌ای از معادلات دیفرانسیل غیرخطی فرموله شود:

$$\dot{X} = f(X, U) \quad (1)$$

که در آن $X = [\delta, \omega, E_q', E_{fd}, V_f]^T$ و U سیگنال‌های خروجی PSS است.

در مسئله‌ی تنظیم PSS، مدل‌های افزایشی خطی در اطراف یک نقطه‌ی تعادل معمولاً استفاده می‌شود. بنابراین، معادله‌ی حالت یک سیستم قدرت با تعداد n ماشین و m PSS را می‌توان به شکل زیر تشکیل داد:

$$\dot{X} = (AX + BU) \quad (2)$$

که در آن A یک ماتریس $5n \times 5n$ و معادل $\partial f / \partial X$ است. در حالی که B یک ماتریس $5n \times m$ و معادل $\partial f / \partial U$ است. هم A و هم B در یک نقطه‌ی خاص عامل، برآورد شده است. X یک بردار حالت $5n \times 1$ و U یک بردار ورودی $m \times 1$ متر است.

۲-۲. ساختار کنترل PSS

ساختار CPSS با توجه به سهولت تنظیم آنلاین مطلوب است. تجزیه و تحلیل جامع از اثرات پارامترهای مختلف CPSS در عملکرد دینامیکی کلی سیستم قدرت در (Kundur et al., 1989) بررسی شده است. این نشان می‌دهد که انتخاب مناسب پارامترهای CPSS عملکرد رضایت‌بخش در طول اختلالات سیستم را نتیجه می‌دهد. ساختار چندمین PSS برابر است با:

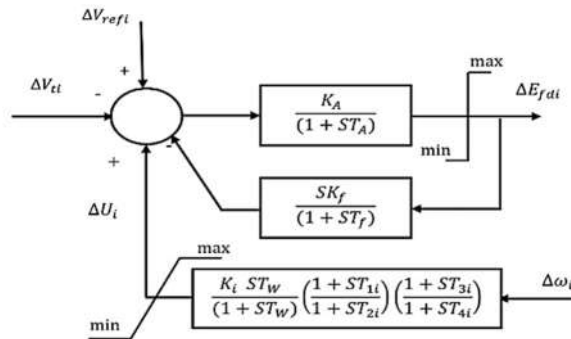
$$\Delta U_i = K_i \frac{ST_W}{1 + ST_W} \left[\frac{1 + ST_{ii}}{1 + ST_{vi}} \frac{1 + ST_{vi}}{1 + ST_{vi}} \right] \Delta \omega_i \quad (3)$$

این ساختار شامل یک افزایش، یک فیلتر شست‌وشو، و جبران پویا و محدود مطابق شکل ۱ است. سیگنال خروجی ΔU_i به‌عنوان یک سیگنال ورودی کمکی تغذیه می‌شود و سیستم تحریک را تنظیم می‌کند. سیگنال ورودی $\Delta \omega_i$ انحراف در سرعت سنکرون است. برای جبران تأخیر فاز بین تحریک و گشتاور الکتریکی دو مدار پیش‌فاز - پس‌فاز استفاده می‌شود. محدودکننده برای جلوگیری سیگنال خروجی از PSS از رسیدن سیستم تحریک به فوق اشباع گنجانده شده است (Eremia & Shahidehpour, 2013). در این مقاله، مقدار ثابت زمان شست‌وشو (T_W) در ۱۰ ثانیه نگه داشته می‌شود، مقدار ثابت زمان T_{2i} و T_{4i} در یک مقدار معقول ۰/۰۵ ثابت شده است. دستاورد تثبیت‌کننده‌ی K_i و ثابت زمان T_{3i} و T_{1i} باید از لحاظ عددی مشخص شوند.

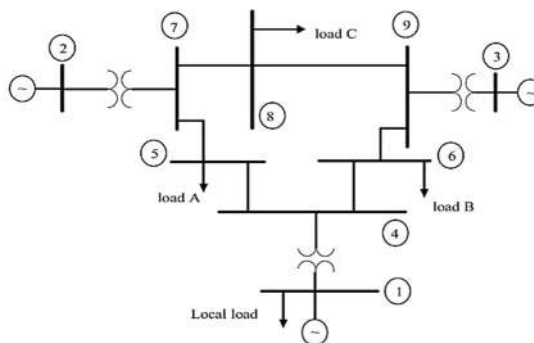
۲-۳. سیستم تحت مطالعه

دستگاه سه‌ماشینه، نه‌باسه در شکل ۲ نشان داده شده است. اطلاعات سیستم با جزئیات در منابع (Fouad & Anderson, 1977) نشان داده شده است. سه وضعیت عملیاتی مختلف به‌عنوان بار سبک،

نرمال و سنگین برای نشان دادن برتری الگوریتم پیشنهادی در طراحی قوی PSS در نظر گرفته شده و ژنراتور و سطح بارگذاری در جدول ۱ برای این حالت‌های بارگذاری داده شده است.



شکل ۱. بلوک دیاگرام PSS نام با سیستم تحریک



شکل ۲. سیستم تست سه‌ماشینه

جدول ۱. حالت‌های بارگذاری برای سیستم برحسب پریونیت (pu)

سنگین		نرمال		سبک		بار و ژنراتور	سیستم تست چندماشینه
Q	P	Q	P	Q	P		
۱/۸۱۴۳	۳/۵۷۳۰	۰/۶۲۰۵	۱/۷۱۶۴	۰/۲۲۳	۰/۹۶۴۹	ژنراتور ۱	ژنراتور
۰/۷۱۲۷	۲/۲۰	۰/۰۶۶۵	۱/۶۳۰	-۰/۱۹۳۳	۱/۰۰	ژنراتور ۲	
۰/۴۳۱۳	۱/۳۵	-۰/۱۰۸۶	۰/۸۵	-۰/۲۶۶۸	۰/۴۵	ژنراتور ۳	
۰/۹	۲/۰	۰/۵	۱/۲۵	۰/۳۵	۰/۷	بار A	بار
۰/۶	۱/۸	۰/۳	۰/۹	۰/۳	۰/۵	بار B	
۰/۶۵	۱/۶	۰/۳۵	۱/۰	۰/۲	۰/۶	بار C	
۰/۶۵	۱/۶	۰/۳۵	۱/۰	۰/۲	۰/۶	بار محلی ژنراتور ۱	

۳. روش‌های بهینه‌سازی

بررسی اجمالی الگوریتم جست‌وجوی کوکو، رفتار اصلاح گونه‌های کوکو و آیتم‌های اساسی از الگوریتم پیشنهادی در بخش زیر بررسی شده است.

۳-۱. رفتار اصلاح کوکو (CS)

CS یک الگوریتم جست‌وجوی فوق‌ابتکاری است که به‌تازگی توسط یانگ (Yang, 2010) معرفی شده است. این الگوریتم از الزام انگلی بودن نوزادان برخی از گونه‌های کوکوها که روی تخم خود در لانه‌ی پرندگان دیگر می‌خوابند (ممکن است از گونه‌های مختلف باشند) الهام گرفته شده است. در لحظه‌ی مناسب مرغ کوکو به لانه‌ی میزبان پرواز می‌کند، یک تخم را از لانه هل می‌دهد، یک تخم را باقی می‌گذارد و پرواز می‌کند. کل فرایند در حدود ۱۰ ثانیه طول می‌کشد. جنس ماده ممکن است به ۵۰ لانه در طول یک فصل بارگذاری مراجعه کند. پرندگان میزبان ممکن است تشخیص دهند که آن تخم خودشان نیست یا آنها را دور بریزند یا لانه را رها کنند و لانه‌ی جدیدی در جای دیگری درست کنند. این روند موجبات تکامل تخم کوکو را که ادعای شبیه بودن تخم پرندگان میزبان محلی را دارد، فراهم می‌آورد (Yang & Deb, 2010).

علاوه بر این، زمان خوابیدن روی تخم در برخی از گونه‌ها نیز شگفت‌انگیز است. کوکوهای انگلی اغلب لانه‌ای را انتخاب می‌کنند که پرند در آن فقط روی تخم خود می‌خوابد. به‌طور کلی، تخم کوکو کمی زودتر از تخم پرند میزبان بیرون می‌آید. این قضیه به‌طرز پیچیده‌ای با تمام فرزندان میزبان تفاوت دارد. جوجه‌ی این پرند بسیار بزرگ‌تر از جوجه‌های میزبان است و نیاز دارد تا انحصار مواد غذایی عرضه شده توسط پدر و مادر را در اختیار گیرد. این جوجه با فشار دادن دیگر تخم‌ها آنها را از لانه پایین می‌اندازد. اگر تخم پرند میزبان قبل از تخم کوکوها بیرون آید، جوجه‌ی کوکو، جوجه‌ی دیگر را به همین ترتیب از لانه به بیرون خواهد انداخت.

۳-۲. پروازهای باج‌گیرانه (Lévy flights)

استفاده از Lévy flights هم برای جست‌وجوی محلی و هم جهانی یک جزء مهم از CS است (Yang & Deb, 2009). روند Lévy flights یک فرایند تصادفی است که توسط یک سری از جهش‌های آنی توسط یک تابع چگالی احتمال که دارای یک قانون توان مشخص است انتخاب شده است. این فرایند نشان‌دهنده‌ی الگوی جست‌وجو به‌صورت تصادفی بهینه است و اغلب در طبیعت یافت می‌شود

(Bhargava et al., 2013). در هنگام تولید یک تخم جدید، یک Lévy flights انجام گرفته با شروع در محلی که یک تخم به صورت تصادفی انتخاب شده است، اگر مقدار تابع هدف در این مختصات جدید بهتر از تخم دیگری که به صورت تصادفی انتخاب شده باشد، پس از آن تخم به این موقعیت جدید تغییر مکان می‌دهد. یکی از مزایای استفاده از CS در مقابل الگوریتم‌های بهینه‌سازی دیگر این است که فقط یک پارامتر (بخشی از لانه که ترک شده است (Pa)) نیاز به تنظیم دارد. استفاده از Lévy flights به عنوان روش جست‌وجو بدان معناست که همزمان CS می‌تواند تمام بهینه‌ها را در یک فضای طراحی پیدا کند؛ این روش در مقایسه با دیگر الگوریتم‌ها عملکرد خوبی دارد. (Rajabioun, 2011)

۳-۳. اجرای جست‌وجوی کوکو (CS)

هر تخم در لانه یک راه‌حل را نشان می‌دهد و یک تخم کوکو یک راه‌حل جدید را معرفی می‌کند. هدف این است که از راه‌حل‌های جدید با پتانسیل بهتر به جای یک راه‌حل نه‌چندان خوب در لانه استفاده شود. هر لانه یک تخم برای موارد ساده دارد. الگوریتم می‌تواند به موارد پیچیده‌تر توسعه یابد که در آن هر لانه تخم‌های متعددی دارد که هر کدام نماینده‌ای از مجموعه‌ای از راه‌حل‌هاست. برای اعمال این جست‌وجو به عنوان یک الگوریتم بهینه‌سازی، سه قانون زیر ارائه شده‌اند (Yang & Deb, 2013; Abd-Elaziz & Ali, 2015): ۱. کوکوها لانه‌های تصادفی برای تخم‌گذاری تخم‌های خود را انتخاب می‌کنند. کوکو می‌تواند تنها روی یک تخم در یک زمان بخوابد؛ ۲. فرایند انتخاب دلایل اعمال می‌شود، به طوری که تنها بهترین تخم‌ها به نسل بعدی منتقل می‌شوند؛ ۳. تعداد لانه‌های میزبان در دسترس مشخص است و یک میزبان می‌تواند تخم خارجی با احتمال $P_a \in [0.1]$ را تشخیص دهد. پرنده‌ی میزبان در صورت تشخیص تخم کوکو، ممکن است آن را دور بیندازد یا لانه‌ی خود را برای امنیت کوکو رها کند. فرض آخر را می‌توان توسط یک عامل Pa از تعداد n لانه تقریب زد که توسط لانه‌های جدید تغییر داده شده‌اند. کیفیت یک راه‌حل نسبت به تابع هدف برای یک مسئله خیلی خوب است. شکل‌های دیگر تناسب را می‌توان در یک راه یکسان در تابع تناسب GA تعیین کرد. براساس این سه قانون، فلوچارت CS در شکل ۳ نشان داده شده و پارامترهای CS در ضمیمه آورده شده است. هنگام ایجاد یک راه‌حل جدید X_i^{t+1} برای کوکوی i ام، یک Lévy flights انجام می‌گیرد:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda) \quad (4)$$

که در آن $\alpha > 0$ اندازه‌ی گام است که باید با مقیاس‌های مشکل بهینه‌سازی متناسب باشد. محصول \oplus به معنای پیاده‌روی با ورود عاقلانه در فرایند ضرب است. اساساً Lévy flights مدل یک پیاده‌روی تصادفی است، در حالی که گام‌های تصادفی خود را از توزیع Lévy برای مراحل بزرگ کشیده است:

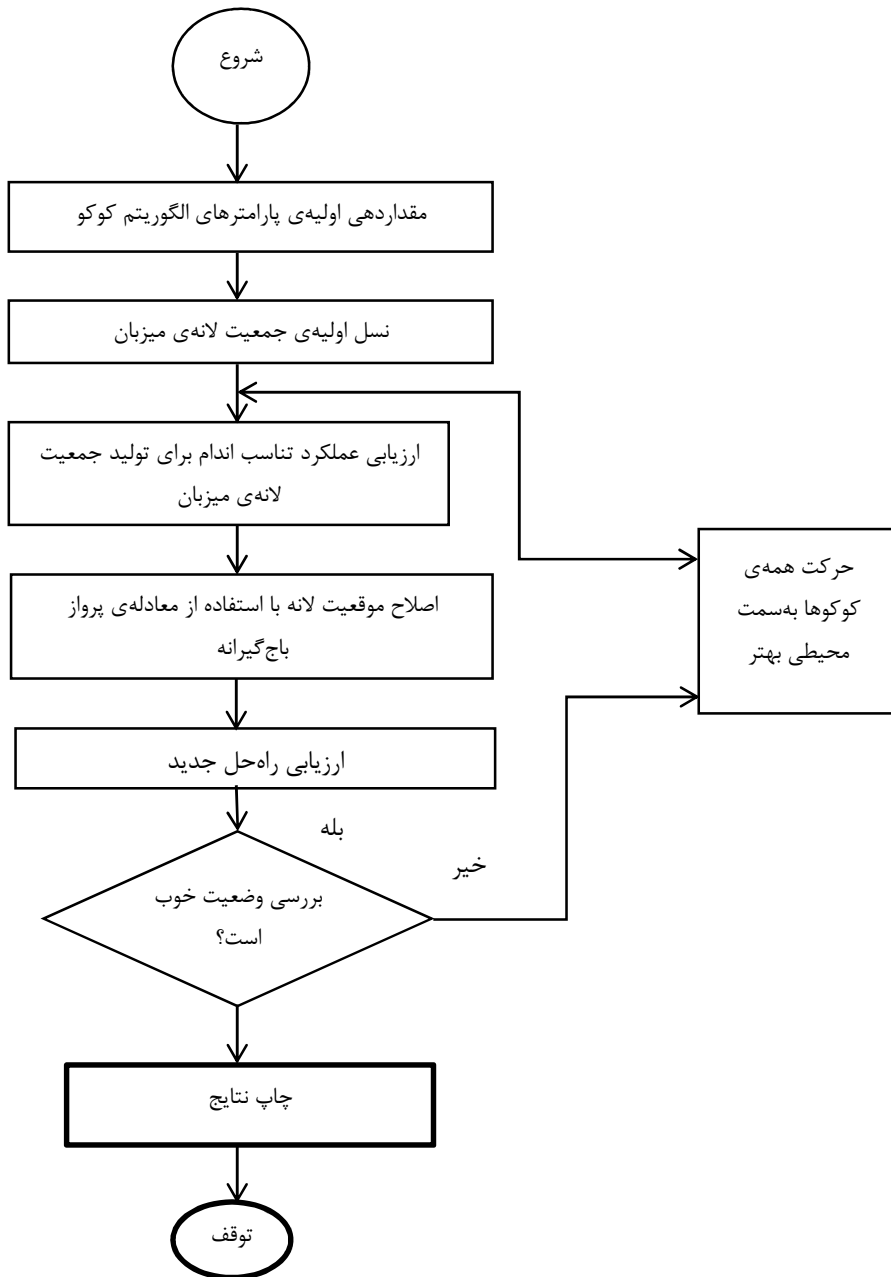
$$Lévy \sim u = t^{-\lambda}, (1 < \lambda < 3) \quad (5)$$

این واریانس بی‌نهایت، یک معنای بی‌نهایت دارد. در اینجا قدم‌ها اساساً از فرایند پیاده‌روی تصادفی با قانون توزیع طول گام با یک دم سنگین تشکیل می‌شوند. برخی از راه‌حل‌های جدید باید توسط گام Lévy پیرامون بهترین راه‌حل‌های کنونی باشند تا این موضوع سرعت جست‌وجوی محلی را افزایش دهد. با این حال، بخش شایان توجهی از راه‌حل‌های جدید باید توسط میدان تصادفی دور تولید شوند و مکان‌ها باید به اندازه‌ی کافی از بهترین راه‌حل حاضر دور باشند. به کمک این روش، سیستم مطمئن خواهد شد که در بهینه‌ی محلی به دام نخواهد افتاد.

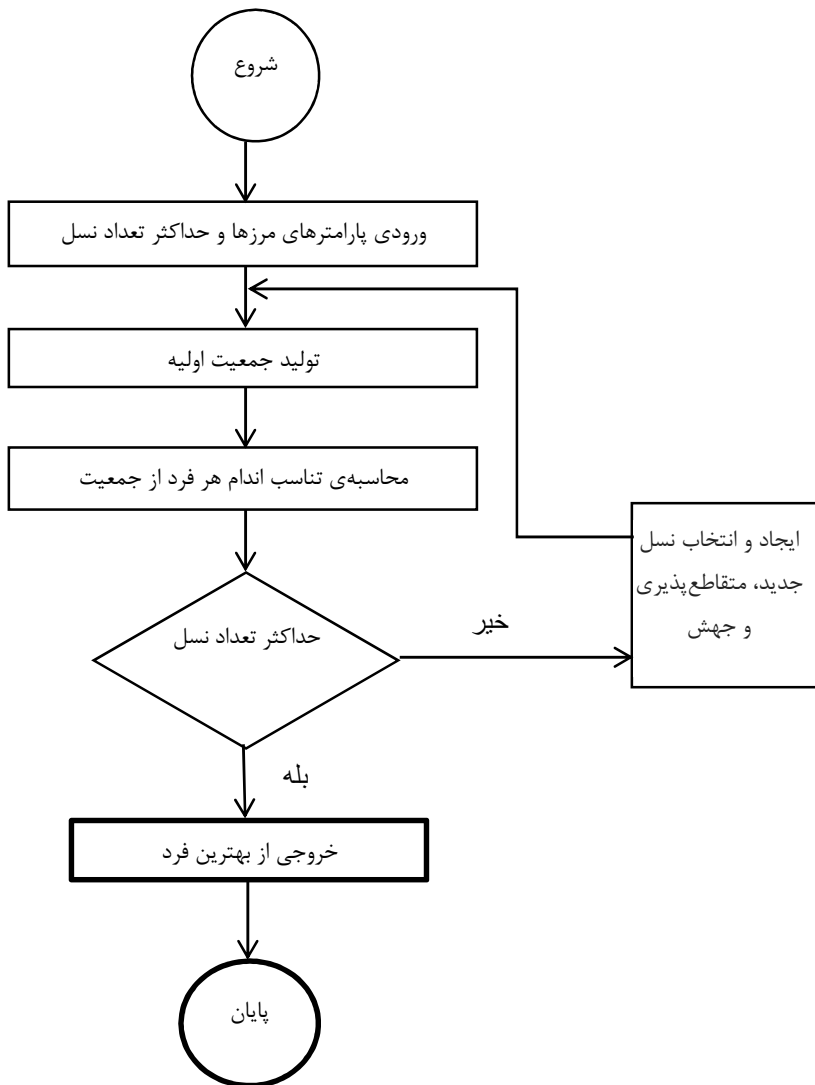
۳-۴. الگوریتم ژنتیک

حیوانات در قلمرو خود، با توجه به نقش «بقای اصلح» تکامل می‌یابند و همچنین تولیدمثل می‌کنند. در طبیعت، حیوانات به‌طور مداوم برای غذا، سرپناه و درمان فعالیت می‌کنند؛ بنابراین تنها اصلح زنده خواهد ماند و ضعیف‌ترین هلاک خواهد شد. این سازوکار از بین بردن بی‌فایده‌ها، قرن‌ها به‌خوبی کار کرده و به روشی خوب برای بهینه‌سازی تبدیل شده است. GA مانند یک روش بهینه‌سازی است و براساس سازوکار انتخاب طبیعی و ژنتیک طبیعی عمل می‌کند. روند جست‌وجوها مشابه تکامل طبیعی موجود بیولوژیکی است که در آن نسل موجودات، زاده شده و بزرگ می‌شوند تا زمانی که قادر به رشد باشند. دقیقاً مثل قلمرو حیوانات، تنها اصلح برای تولیدمثل زنده خواهد ماند، در حالی که ضعیف‌ترین حذف خواهد شد (Haupt & Haupt, 2004).

یک فلوچارت برای الگوریتم GA در شکل ۴ نشان داده شده است.



شکل ۳. فلوچارت الگوریتم بهینه‌سازی کوکو



شکل ۴. فلوچارت الگوریتم ژنتیک

چهار پارامتر اصلی بر عملکرد GA تأثیر می‌گذارد: اندازه‌ی جمعیت، تعداد نسل، نرخ متقاطع و نرخ جهش. اندازه‌ی جمعیت بزرگ‌تر و تعداد زیاد نسل‌ها، احتمال به‌دست آوردن راه‌حل بهینه‌ی نزدیک به راه‌حل جهانی را افزایش می‌دهد، اما به‌طور چشمگیری زمان پردازش را نیز افزایش می‌دهد. نرخ متقاطع در میان کروموزوم‌های پدر و مادر یک فرایند طبیعی مشترک است و به‌طور سنتی از ۰/۶ تا ۱ است. در متقاطع،

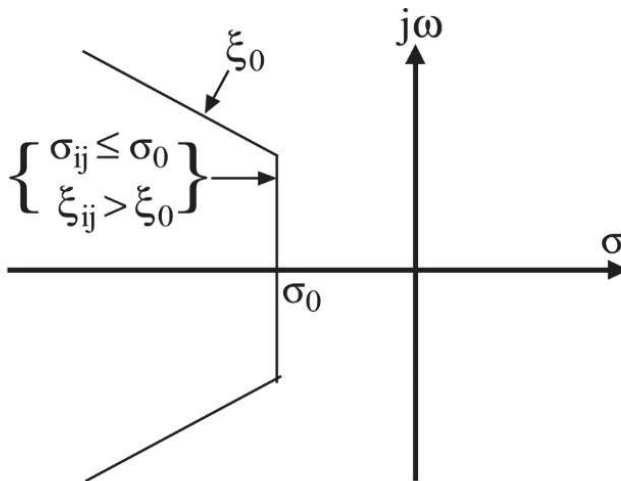
تبادل اطلاعات پدر و مادر، فرزند را تولید می‌کند. برخلاف متقاطع، جهش یک فرایند نادر و شبیه به تغییری ناگهانی در فرزندان است و می‌تواند به‌طور تصادفی با انتخاب یک کروموزوم از جمعیت و سپس تغییر خودسرانه‌ی برخی از اطلاعات آن انجام گیرد. مزیت جهش این است که به‌طور تصادفی مواد ژنتیکی جدید را در روند تکاملی معرفی می‌نماید. شاید از این طریق از رکود در اطراف حداقل محلی اجتناب شود. معمولاً نرخ جهش کوچک کمتر از ۰/۱ استفاده می‌شود (همان). پارامترهای GA در ضمیمه آورده شده است.

۴. تابع هدف

جهت رسیدن به میرایی بیشتر برای نوسانات فرکانس پایین، پارامترهای PSS ممکن است برای به حداقل رساندن فرمول D- شکل زیر تنظیم شود:

$$J_t = \sum_{j=1}^{np} \sum_{\sigma_{ij} \geq \sigma_0} (\sigma_{ij} - \sigma_0)^2 + \sum_{j=1}^{np} \sum_{\xi_{ij} \geq \xi_0} (\xi_{ij} - \xi_0)^2 \quad (6)$$

سیستم مقادیر ویژه‌ی حلقه‌ی بسته در بخش D شکل جا داده شده که با $\sigma_{ij} \leq \sigma_0$ و $\xi_{ij} > \xi_0$ همان‌طور که در شکل ۵ نشان داده شده، بسته شده است.



شکل ۵. بخش D- شکل در نیمه‌ی چپ محور S

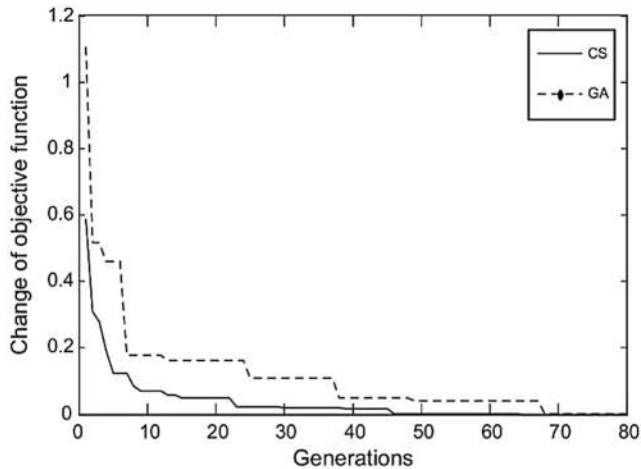
در این مقاله، σ_0 و ξ_0 انتخاب می‌شوند تا به ترتیب $0/5 -$ و $0/1$ باشند (Ali, 2014). محدودیت‌های معمولی از پارامترهای بهینه‌سازی برای K [۵۰-۱] و برای T_{1i} و T_{3i} [۱-۰/۰۶] هستند. در نهایت، به حداقل رساندن J_t به قرار زیر است:

$$\begin{aligned} K_i^{min} &\leq K_i \leq K_i^{max} \\ T_{li}^{min} &\leq T_{li} \leq T_{li}^{max} \\ T_{3i}^{min} &\leq T_{3i} \leq T_{3i}^{max} \end{aligned} \quad (7)$$

هدف از بهینه‌سازی، جست‌وجو برای مجموعه‌ی تنظیم بهینه‌ی پارامترهای PSSهاست که منعکس‌کننده‌ی حداقل زمان نشست و فراجهبش سیستم است. علاوه بر این، اهداف افزایش مربوط به ویژگی‌های نوسانات کم هستند، دستیابی به یک عملکرد خوب در شرایط عملیاتی مختلف و در نهایت طراحی یک کنترل سفارش پایین برای اجرای آسان است.

۵. نتایج و شبیه‌سازی

این قسمت، بخش شناخته شده‌ای از الگوریتم CS ارائه شده در طراحی PSS در مقایسه با PSS بهینه‌سازی شده با GA و CPSS را نشان می‌دهد (Ali, 2014). تغییر توابع هدف با دو الگوریتم بهینه‌سازی در شکل ۶ نشان داده شده است و بیش از تکرار، CS و GA را کاهش می‌دهد. مقدار نهایی تابع هدف برای هر دو الگوریتم $J_t = 0$ است و نشان می‌دهد تمام حالت‌ها به بخش D شکل مشخص شده در محور S منتقل شده‌اند و به تابع هدف پیشنهاد شده رسیده‌اند. علاوه بر این، CS با سرعت بیشتری (۴۶ نسل) در مقایسه با GA (۶۸ نسل) همگرا می‌شود. همچنین زمان محاسباتی (CPU) از هر دو الگوریتم براساس میانگین زمان CPU گرفته شده، مقایسه شده است تا راه‌حل را همگرا کند. میانگین CPU برای CS، $36/4$ و برای GA $48/6$ ثانیه است. روش پیشنهاد CS و GA از طریق نرم‌افزار Matlab برنامه‌ریزی شده است. زمان اشاره شده‌ی CPU متوسط ترکیبی از ۱۰ کد کامپیوتری است. مقادیر ویژه‌ی سیستم و نسبت میرایی از حالت مکانیکی با سه حالت بارگذاری مختلف در جدول ۲ آورده شده است.



شکل ۶. تغییرات توابع هدف

جدول ۲. مدل‌های مکانیکی و ζ در حالت‌های مختلف بارگذاری و کنترل (مقادیر پررنگ شده مدهای ناپایدار را نشان می‌دهد)

انواع بار	بدون تثبیت‌کننده	CPSS	GAPSS	CSPSS
بار سبک	$+0/01 \pm 0/0z, -0/12$	$-0/19 \pm 0/69z, 0/26$	$-1/06 \pm 0/7z, -0/83$	$-1/11 \pm 0/62z, -0/87$
	$-0/58 \pm 6/04z, 0/09$	$-2/25 \pm 4/15z, 0/49$	$-3/23 \pm 6/3z, 0/51$	$-6/28 \pm 6/49z, 0/69$
	$-0/96 \pm 7/77z, 0/12$	$-3/24 \pm 5/2z, 0/52$	$-3/49 \pm 8/1z, 0/4$	$-3/64 \pm 6/0z, 0/52$
بار نرمال	$+0/15 \pm 1/49z, -0/1$	$-2/41 \pm 0/75z, 0/3$	$-1/13 \pm 0/69z, 0/85$	$-1/17 \pm 0/69z, 0/86$
	$-0/35 \pm 8/14z, 0/04$	$-2/41 \pm 4/42z, 0/47$	$-4/23 \pm 7/0z, 0/52$	$-6/85 \pm 6/97z, 0/70$
	$-0/67 \pm 11/6z, 0/06$	$-3/32 \pm 5/34z, 0/52$	$-3/61 \pm 8/22z, 0/38$	$-0/96 \pm 7/77z, 0/12$
بار سنگین	$+0/32 \pm 1/6z, -0/19$	$-0/33 \pm 0/89z, 0/34$	$-1/71 \pm 0/72z, 0/85$	$-1/31 \pm 0/72z, 0/83$
	$+0/15 \pm 7/8z, -0/02$	$-1/96 \pm 4/32z, 0/41$	$-3/52 \pm 6/22z, 0/46$	$-7/99 \pm 5/42z, 0/83$
	$-0/36 \pm 12/2z, 0/03$	$-3/09 \pm 5/25z, 0/5$	$-3/79 \pm 8/91z, 0/39$	$-4/63 \pm 7/32z, 0/53$

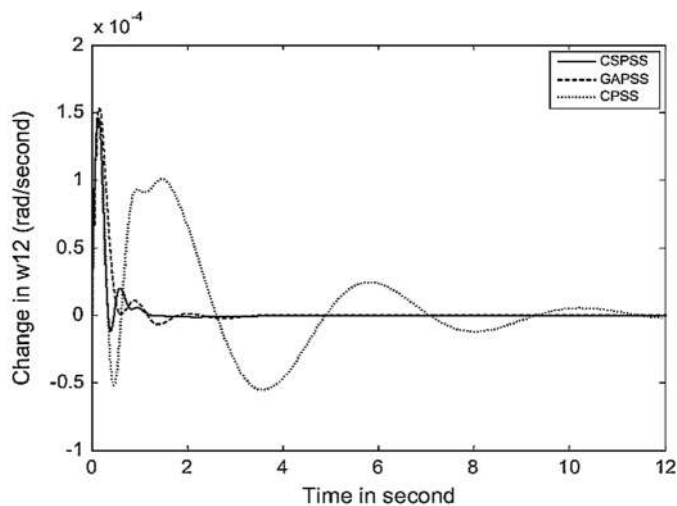
واضح است که حالت‌های الکترومکانیکی با CSPSS به سمت چپ از محور S منتقل شده و مقادیر عوامل میرایی به میزان چشمگیری برای بارهای سبک، نرمال و سنگین به ترتیب در حالت بارگذاری افزایش یافته است ($\sigma = -1/10, -1/17, -1/31$). همچنین، نسبت میرایی مربوط به کنترل‌کننده‌های CSPSS بزرگتر از آنهایی است که مربوط به GAPSS و CPSS هستند. از این رو، CSPSS عملکرد قوی به دست آمده از ویژگی‌های میرایی برتر از حالت الکترومکانیکی را فراهم می‌کند. جدول ۳ مقادیر پارامترهای PSSها را براساس تابع هدف ارائه شده با استفاده از CS، GA و روش متداول نشان می‌دهد.

جدول ۳. پارامترهای PSS برای الگوریتم‌های مختلف

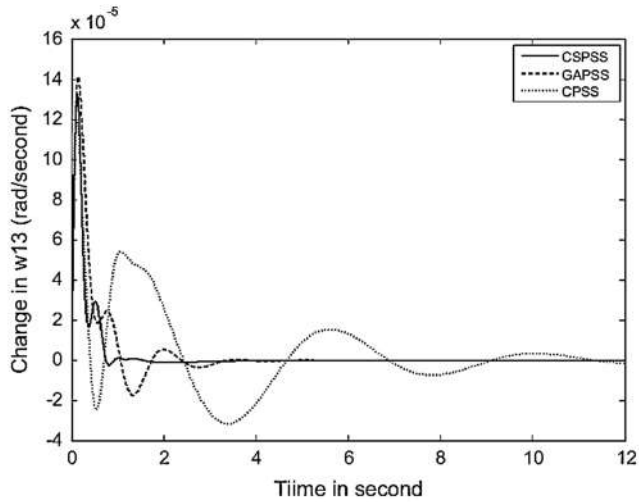
CPSS	GA	CS	پسها
$K=۱۴/۴۳۸۶$ $T_1=-۰/۲۶۵۲$ $T_3=-۰/۸۹۵۲$	$K=۲۵/۳۶۹۶$ $T_1=-۰/۴۶۸۴$ $T_3=-۰/۷۴۲۸$	$K=۴۴/۲۴۵۸$ $T_1=-۰/۴۲۴۳$ $T_3=-۰/۳۵۷۸$	PSS1
$K=۵/۱۶۵۹$ $T_1=-۰/۵۲۴۲$ $T_3=-۰/۲۰۲۲$	$K=۷/۸۱۸۷$ $T_1=-۰/۲۹۱۸$ $T_3=-۰/۱۴۲۹$	$K=۸/۲۴۵۱$ $T_1=-۰/۴۵۷۳$ $T_3=-۰/۱۷۲۴$	PSS2
$K=۸/۳۲۸۷$ $T_1=-۰/۵۸۷۱$ $T_3=-۰/۴۲۶۸$	$K=۳/۵۳۱۴$ $T_1=-۰/۵۱۲۱$ $T_3=-۰/۳۷۳۱$	$K=۴/۶۲۶۱$ $T_1=-۰/۲۳۵۸$ $T_3=-۰/۱۷۷۱$	PSS3

۵-۱. پاسخ در حالت بار طبیعی

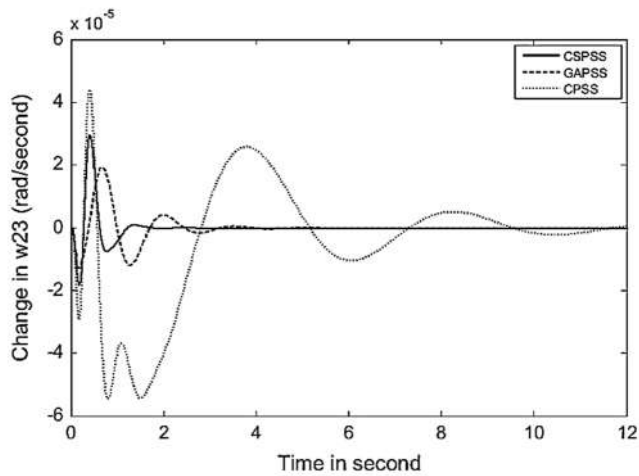
شکل‌های ۷ تا ۹، پاسخ‌های $\Delta\omega_{12}$ و $\Delta\omega_{13}$ را با توجه به ۲۰ درصد افزایش گشتاور مکانیکی ژنراتور ۱ در حالت بارگذاری طبیعی نشان می‌دهد. روشن است که سیستم با CSPSS ارائه شده بیش از GAPSS و CPSS تثبیت شده است.



شکل ۷. تغییرات $\Delta\omega_{12}$ در بار نرمال



شکل ۸. تغییرات $\Delta\omega_{13}$ در بار نرمال

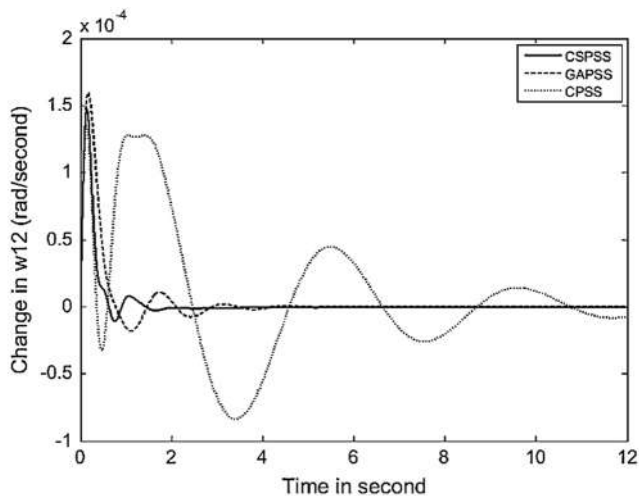


شکل ۹. تغییرات $\Delta\omega_{23}$ در بار نرمال

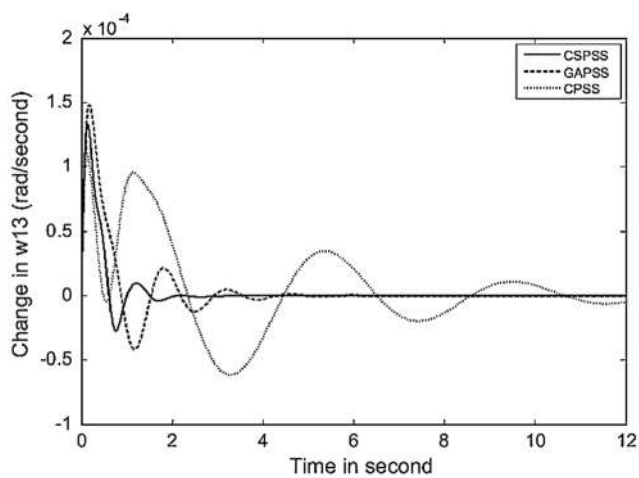
همچنین میانگین حل و فصل بار برای کاهش نوسانات سیستم حدود $1/0.5$ با CSPSS، $1/6$ ثانیه برای GAPSS و 7 با CPSS است. بنابراین طراحی کنترل کننده برای تأمین منابع کافی جهت میرایی نوسانات فرکانس پایین کافی است.

۲-۵. پاسخ در حالت بار سبک

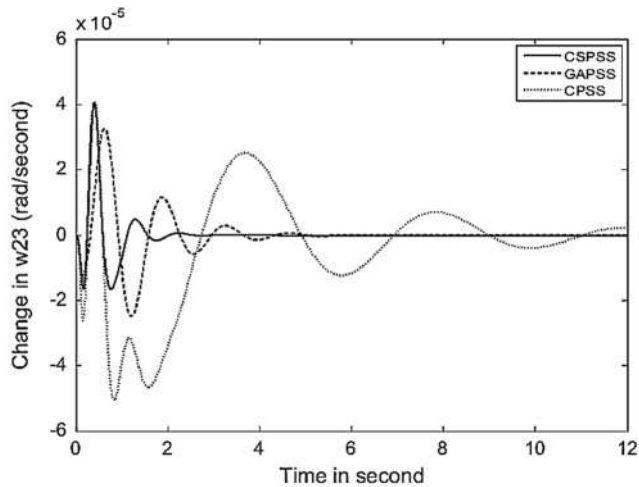
پاسخ‌های سیستم در حالت بارگذاری سبک با همان پارامترهای کنترل‌کننده در شکل‌های ۱۰ تا ۱۲ نشان داده شده است. واضح است از CSPSS پیشنهادی، ویژگی‌های میرایی خوبی برای حالت سیستم نوسانی فراهم می‌آورد و سیستم را به‌طور مکرر تثبیت می‌کند.



شکل ۱۰. تغییرات $\Delta\omega_{12}$ در بار سبک



شکل ۱۱. تغییرات $\Delta\omega_{13}$ در بار سبک

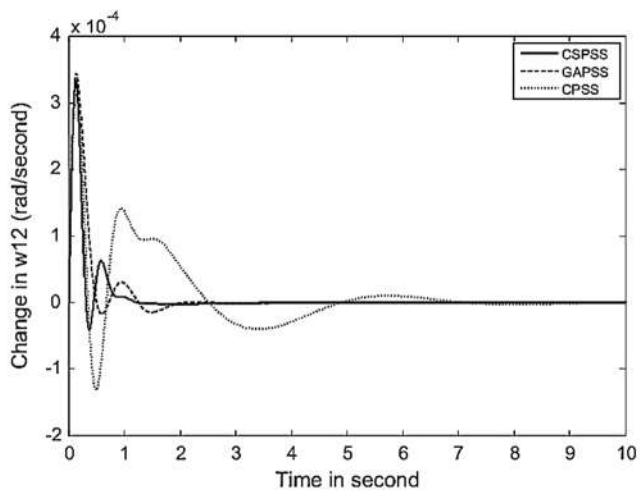


شکل ۱۲. تغییرات $\Delta\omega_{23}$ در بار سبک

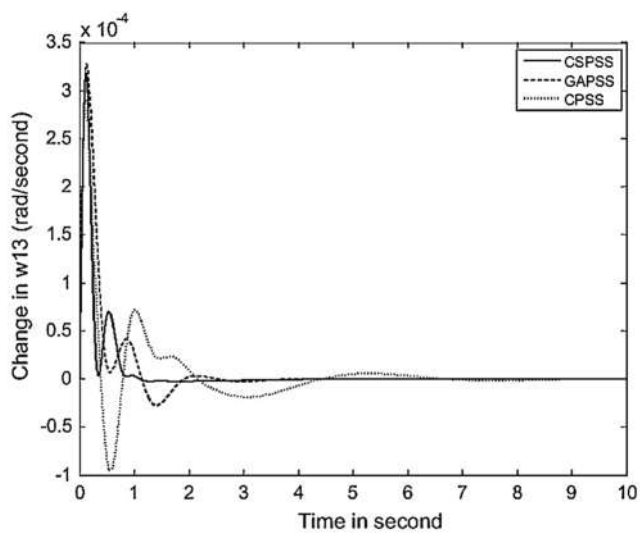
همچنین میانگین حل و فصل بار نوسانات $10/6$ ، $2/6$ ، $1/46$ برای T_s CSPSS، GAPSS و CPSS هستند. از این رو، CSPSS پیشنهادی، کنترل‌کننده‌ی GAPSS و CPSS را به‌طور مؤثر در کاهش نوسانات بهبود می‌بخشد و زمان نشست را کمتر می‌کند.

۳-۵. پاسخ در حالت بار سنگین

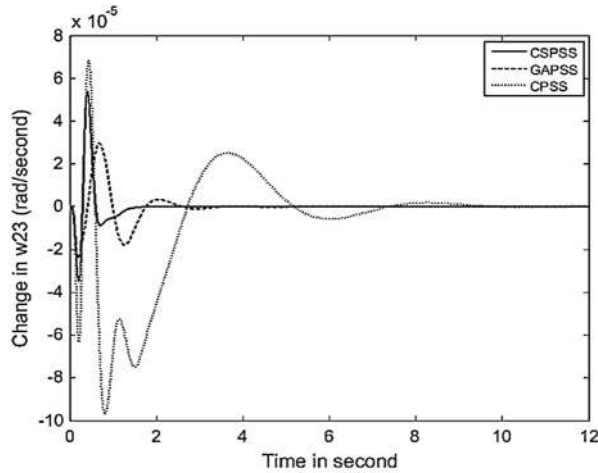
پاسخ سیستم در حالت بارگذاری سنگین در شکل‌های ۱۳ تا ۱۵ نشان داده شده است. برتری CSPSS در کاهش زمان نشست و میرایی نوسانات سیستم قدرت تأیید شده است. همچنین میانگین زمان‌های نشست از این نوسانات $5/3$ ، $1/43$ ، 1 به ترتیب برای CSPSS، GAPSS و CPSS است. از این رو، کنترل‌کننده‌ی CSPSS تا حد زیادی ویژگی‌های میرایی سیستم قدرت را افزایش می‌دهد. از طرفی، زمان نشست این کنترل‌کننده کوچک‌تر از آن است که در برخی منابع (Abd-Elazim & Ali, 2013 ; Ali, 2014) بیان شده است.



شکل ۱۳. تغییرات $\Delta\omega_{12}$ در بار سنگین



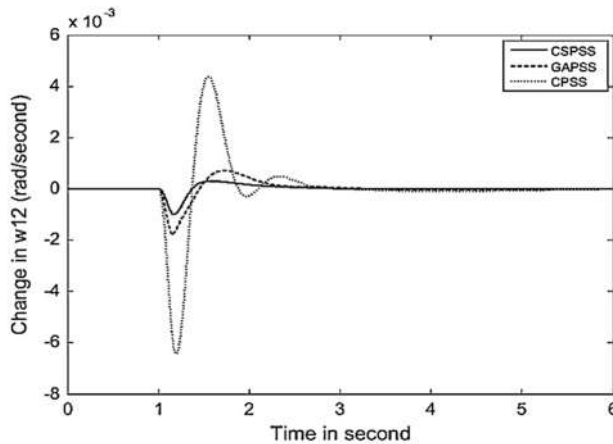
شکل ۱۴. تغییرات $\Delta\omega_{13}$ در بار سنگین



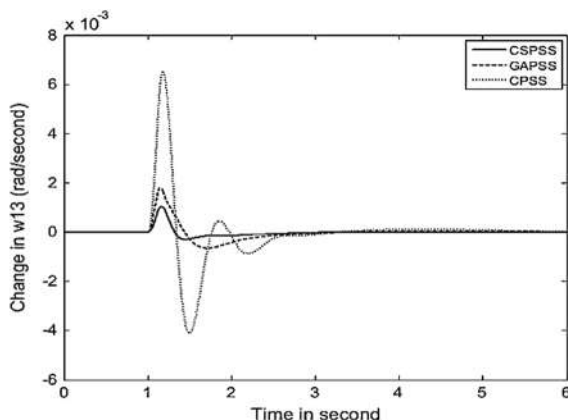
شکل ۱۵. تغییرات $\Delta\omega_{23}$ در بار سنگین

۴-۵. پاسخ در حالت اختلال شدید

اعتبار سنجی از عملکرد در حالت اختلال شدید با اعمال یک خطای سه فاز از مدت زمان شش چرخه در ۱ ثانیه در نزدیکی باس ۷، تأیید شده است. شکل‌های ۱۶ و ۱۷ پاسخ‌های A6012 و A6013 در حالت بارگذاری سنگین را نشان می‌دهد. از این شکل‌ها، می‌توان دید CS براساس PSSها با استفاده از تابع هدف ارائه شده، میرایی برتر و عملکرد قوی‌تری در مقایسه با روش‌های دیگر معرفی می‌کند.



شکل ۱۶. تغییرات $\Delta\omega_{12}$ در حالت اختلال شدید



شکل ۱۷. تغییرات $\Delta\omega_{13}$ در حالت اختلال شدید

۶. شاخص پایداری و عملکرد

استحکام کنترل‌کننده‌ی ارائه شده از طریق برخی از شاخص‌های عملکرد نشان داده شده است. از انتگرال قدر مطلق مقدار خطا (IAE)، و انتگرال قدر مطلق خطای ضرب‌شده در زمان (ITAE) استفاده می‌شود:

$$IAE = \int_0^{\infty} (|\Delta\omega_{12}| + |\Delta\omega_{23}| + |\Delta\omega_{13}|) dt \quad (۸)$$

$$ITAE = \int_0^{\infty} t(|\Delta\omega_{12}| + |\Delta\omega_{23}| + |\Delta\omega_{13}|) dt \quad (۹)$$

جالب توجه است که هر چه مقدار این شاخص کمتر باشد، پاسخ سیستم از نظر ویژگی‌های حوزه‌ی زمان بهتر است (Ali, 2015). از جدول ۴ روشن است که در مقایسه با کسانی که از BATPSS، GAPSS و CPSS (Ali, 2014) استفاده کرده‌اند، مقادیر اجرای این سیستم با CSPSS کوچک‌تر است. داده‌های جدول ۴ تأیید می‌کند که فراجبهش، زمان نشست و انحراف سرعت همه‌ی واحدها تا حد زیادی با استفاده از CS پیشنهادی براساس PSS‌های تنظیم شده کاهش یافته‌اند.

جدول ۴. شاخص‌های عملکرد برای الگوریتم‌های مختلف

IAÉ*10 ⁻⁴				ITAÉ*10 ⁻⁴				انواع بار
CSPSS	BATPSS	GAPSS	CPSS	CSPSS	BATPSS	GAPSS	CPSS	
-۰/۲۹۱۶	-۰/۲۹۴۹	۱/۵۱۴۸	۴۲/۲	-۰/۰۵۴۱	-۰/۰۶۴۴	-۰/۴۲۶۴	۷/۳۹	بار سبک
-۰/۷۰۰۲	-۰/۷۰۳۶	۱/۸۴۵	۳۵/۵۶	-۰/۰۶۶۷	-۰/۰۶۷۶	-۰/۶۰۲۸	۱۶/۳	بار نرمال
-۰/۹۴۴۷	-۰/۹۴۹۸	۳/۵۰۶۷	۴۷/۹	-۰/۱۰۰۹	-۰/۱۰۰۹	۱/۱۷۳۶	۲۵/۷۶	بار سنگین

۷. نتیجه‌گیری

در این مقاله، الگوریتم CS برای طراحی بهینه‌ی پارامترهای PSSها ارائه شد. مشکل تنظیم پارامترهای PSSها به‌عنوان مسئله‌ی بهینه‌سازی فرموله شده و الگوریتم CS برای دستیابی به پارامترهای مطلوب استفاده شده است. یک مقدار ویژه براساس تابع هدف، منعکس‌کننده‌ی ترکیبی از فاکتور میرایی و رتبه‌ی میرایی برای شرایط عملیاتی مختلف بهینه شده است. نتایج شبیه‌سازی، استحکام و برتری کنترل‌کننده پیشنهادی در ارائه‌ی ویژگی میرایی خوب در نوسانات سیستم در دامنه‌ی گسترده‌ای از حالت‌های بارگذاری را نشان می‌دهد. علاوه بر این، CSPSS پیشنهادی نشان می‌دهد که از دیگر کنترل‌کننده‌های دارای شاخص‌های عملکرد متفاوت، مؤثرتر است. استفاده از الگوریتم پیشنهادی در سیستمی با مقیاس بزرگ، محدوده‌ی کاری این موضوع در آینده است.

منابع

1. Abdel-Magid YL, Abido MA. (2003), "Optimal multi-objective design of robust power system stabilizers using genetic algorithms," IEEE Trans Power Syst., 18 (3): 1125–32.
2. Abd-Elazim, S.M. & Ali, E.S. (2013), "A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design", Int J. electr. Power energy syst., 46(1):334–41.
3. Abd-Elaziz, A.Y. & Ali, E.S. (2015), "Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system", Int J. electric power energy syst., 73 C: 632–43.
4. Abido, M.A. (2000), "Simulated annealing based approach to PSS and FACTS based stabilizer tuning", Int J electric power energy syst., 22(4):247–58.
5. Abido, M.A. & Abdel-Magid, Y.L. (1999), "A tabu search based approach to power system stability enhancement via excitation and static phase shifter control", Int J electric power syst. Res., 52(2):133–43.
6. Ali, E.S. (2014), "Optimization of power system stabilizers using BAT search algorithm", Int J. electric power energy syst., 61(C): 683–90.
7. Ali, E.S. (2015), "Speed control of induction motor supplied by wind turbine via imperialist competitive algorithm", Energy, 89C:593–600.
8. Anderson, P.M. & Fouad, A.A. (1977), "Power system control and stability", Iowa: Iowa State University Press.
9. Bhargava, V.; Fateen, S.E.K. & Petriciolet, A.B. (2013), "Cuckoo search: A new nature inspired optimization method for phase equilibrium calculations", Fluid phase equilib, 337:191–200.
10. Dulau, M. & Bica, D. (2015), "Design of robust control for single machine infinite bus system", Procedia tech., 19: 657–64.
11. El-Razaz, Z.S. & El-Hameed, M.A. (2001), "An artificial neural network based power system stabilizer for multi-machine power system", Int. Arab J. Sci. Eng., 26 (1B):29–40.
12. El-Sherbiny, M.K.; Hasan, M.M.; El-Saady, G. & Yousef, A.M. (2003), "Optimal pole shifting for power system stabilization", Int. J. electric power syst. Res., 66(3): 253–8.
13. Eremia, M. & Shahidehpour, M. (2013), "Handbook of electrical power system dynamics: Modeling, stability, and control", 1st edition, Wiley-Blackwell.
14. Haupt, R.L. & Haupt, S.E. (2004), "Practical genetic algorithms", 2nd ed. John-Wiley & Sons Inc.
15. Jalilvand, A. & Khalkhali, R.A.E. (2011), "Optimal design of PID power system stabilizer in multimachine power system using PSO and genetic algorithm", Int rev. electric eng., 6(2):907–12.
16. Kumar, J.; Kumar, P.P.; Mahesh, A. & Shrivastava, A. (2001), "Power system stabilizer based on artificial neural network", in: international conference on power and energy systems, Chennai, p. 1–6.

17. Kundur, P.; Klein, M.; Rogers, G.J. & Zywno, M.S. (1989), "Application of power system stabilizers for enhancement of overall system stability", IEEE trans. power syst., 4(2):614–26.
18. Machowski, J.; Bialek, J. & Bumby, J. (2009), "Power system dynamics stability and control", 2nd edition, John –Wiley, New York.
19. Megala, M. & Rajan, C.C.A. (2014), "Design of fuzzy logic power system stabilizer in multimachine power system", Adv. Intell. Syst. Comput., 247:49–57.
20. Passino, K.M. (2002), "Biomimicry of bacterial foraging for distributed optimization and control," IEEE contr. Syst. Mag., 22(3):52–67.
21. Peng, Y.; Zhu, Q.M. & Nouri, H. (2011), "Robust H2 power system stabilizer design using LMI techniques", In: proceedings of international conference on modeling, identification and control (ICMIC), p. 405–10.
22. Rajabioun, R. (2011), "Cuckoo optimization algorithm", Applied soft computing, 11:5508–18.
23. Rogers, G. (2000), "Power system oscillations", The Springer international series in engineering and computer science.
24. Rout, K.C. & Panda, P.C. (2010), "Power system dynamic stability enhancement of SMIB using fuzzy logic based power system stabilizer", In: power electronics and instrumentation engineering communications in computer and information science, Vol. 102; 2010. p. 10–4.
25. Segal, R.; Sharma, A. & Kothari, M.L. (2004), "A self-tuning power system stabilizer based on artificial neural network", Electric power energy syst., 26(6):423–30.
26. Yue, M. & Schlueter, R.A. (2003), "L-Synthesis power system stabilizer design using a bifurcation subsystem based methodology", IEEE trans. power syst., 18 (4):1497–506.
27. Yang, X.S. (2010), "Engineering optimization: An introduction with metaheuristic applications", John - Wiley & Sons.
28. Yang, X.S., Deb, S. (2010), "Engineering optimization by cuckoo search", Int J. math model numer optimization, 1(4):330–43.
29. Yang, X.S., Deb, S. (2009), "Cuckoo search via Lévy flights", In: proceedings of world congress on nature & biologically inspired computing", Springer, USA, p. 210–4.
30. Yang, X.S. & Deb, S. (2013), "Multiobjective cuckoo search for design optimization", comput. Oper. res., 40: 1616–24.

ضمیمه

- (الف) پارامترهای الگوریتم CS به شرح زیر است: حداکثر نسل = ۱۰۰؛ تعداد لانه = ۵۰؛ $PA = 0.25$.
- (ب) پارامترهای GA به شرح زیر است: حداکثر نسل = ۱۰۰؛ اندازه جمعیت = ۵۰؛ احتمال متقاطع = 0.75 ؛ احتمال جهش = 0.1 .

