

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده برق و رباتیک

گروه کنترل

عنوان رساله:

طراحی کنترل کننده بهینه PID مرتبه کسری با استفاده از الگوریتم ممتمیک

دانشجو:

سید یاشار موسوی

استاد راهنما

دکتر علیرضا الفی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

ماه و سال انتشار: شهریور ۱۳۹۳

تأییدیه هیات داوران

(برای پایان نامه)

یک نسخه اصل فرم مربوطه

تقدیم به

پدر فداکارم که همیشه از نصیحت هایش بهره مند بوده ام...

مادر دلسوزم که در ذره ذره لحظات زندگی
همراهم بوده ...

همسر عزیزم که با
مهربانی در کنارم بود ...

تشکر و قدردانی

از خداوند متعال سپاسگزارم که توفیق کسب علم و دانش را به من عطا فرموده تا بتوانم این مرحله از علم را با موفقیت به پایان برسانم. از خانواده عزیزم به خاطر محبت‌های بی دریغ که نسبت به من داشته و دارند، کمال تشکر و سپاس دارم. همچنین از استاد ارجمندم جناب آقای دکتر علیرضا الفی به خاطر راهنمایی‌های ارزشمندشان در کلیه مراحل انجام پایان نامه تقدیر و تشکر نموده، سلامتی و موفقیت روز افزون ایشان را از درگاه حق خواستارم.

چکیده

به دلیل اهمیت و کاربرد سیستم‌های مرتبه‌کسری از یکسو و وجود نامعینی‌ها از سوی دیگر، امروزه کنترل اینگونه سیستم‌ها از اهمیت ویژه‌ای برخوردار است. معمول‌ترین روش پیاده‌سازی و شبیه‌سازی توابع انتقال مرتبه‌کسری، تقریب آنها با توابع انتقال مرتبه صحیح است که در این پایان‌نامه انواع آنها مورد بررسی قرار می‌گیرد. سپس چند روش مهم در زمینه‌ی طراحی کنترل‌کننده‌های مرتبه‌کسری بیان می‌شود. توجه این پایان‌نامه به این نوع خاص از کنترل‌کننده‌ها معطوف شده است. بدین منظور از کنترل‌کننده‌ی PID مرتبه-کسری برای کنترل سه فرآیند مرتبه‌کسری، مرتبه صحیح و تاخیردار بهره برده خواهد شد که ضرایب آنها با استفاده از الگوریتم‌های تکاملی بدست می‌آیند. علاوه بر این، یک الگوریتم ممتیک جدید بر پایه الگوریتم بهینه‌سازی انبوه ذرات و با استفاده از مفهوم مشتقات مرتبه‌کسری پیشنهاد می‌شود. برای نشان‌دادن عملکرد الگوریتم پیشنهادی، مقایسه‌ای میان عملکرد آن و الگوریتم ممتیک پایه صورت گرفته و در ادامه کارایی الگوریتم پیشنهادی در کنترل سیستم‌های مورد بحث نشان داده می‌شود.

کلمات کلیدی:

حسابان کسری، کنترل‌کننده‌های مرتبه‌کسری، کنترل‌کننده PID مرتبه‌کسری، الگوریتم‌های بهینه‌سازی تکاملی، الگوریتم بهینه‌سازی ممتیک

فهرست مطالب

۱	فصل اول: حسابان کسری
۲	۱-۱ مقدمه
۲	۲-۱ تاریخچه‌ی حسابان کسری
۳	۳-۱ تعاریف به کار رفته در بحث حسابان کسری
۳	۱-۳-۱ تابع گاما
۳	۲-۳-۱ تابع میتاگ-لفلر
۴	۳-۳-۱ انتگرال مرتبه کسری کوشی
۴	۴-۳-۱ مشتق مرتبه کسری گرونوالد-لتنیکوف
۵	۵-۳-۱ مشتق مرتبه کسری ریمن-لیوویل
۵	۶-۳-۱ مشتق مرتبه کسری کاپوتو
۶	۴-۱ ویژگی‌های مشتق مرتبه کسری
۷	۵-۱ معادله دیفرانسیل با مرتبه کسری خطی
۸	۶-۱ تبدیل لاپلاس
۹	۷-۱ توابع تبدیل مرتبه کسری
۱۰	۱-۷-۱ نمایش فضای حالت مرتبه کسری
۱۳	۲-۷-۱ مشاهده پذیری، کنترل پذیری و پایداری سیستم‌های مرتبه کسری
۱۴	۸-۱ اهداف و روند ارائه پروژه
۱۵	فصل دوم: الگوریتم‌های بهینه‌سازی تکاملی
۱۶	۱-۲ مقدمه
۱۶	۲-۲ بهینه‌سازی تکاملی

۱۷	۲-۳ الگوریتم‌های بهینه‌سازی تکاملی
۱۸	۲-۳-۱ الگوریتم زنبورها
۲۰	۲-۳-۲ الگوریتم کرم شب تاب
۲۲	۲-۳-۳ الگوریتم بهینه‌سازی انبوه ذرات
۲۴	۲-۳-۴ الگوریتم تکامل تفاضلی
۲۵	۲-۳-۵ الگوریتم رقابت استعماری
۲۷	۲-۳-۶ الگوریتم ممتیک (الگوریتم پایه)
۲۸	۲-۳-۶ الگوریتم ممتیک (الگوریتم پیشنهادی)
۳۱	فصل سوم: طراحی کنترل‌کننده‌های مرتبه‌کسری
۳۲	۳-۱ مقدمه
۳۲	۳-۲ روش‌های تقریب سیستم‌های مرتبه‌کسری
۳۳	۳-۲-۱ تقریب کرون
۳۴	۳-۲-۲ تقریب کارلسون
۳۵	۳-۲-۳ تقریب مائسودا
۳۷	۳-۲-۴ تقریب گرونوالد-لتنیکوف
۳۸	۳-۳ روش‌های متداول طراحی کنترل‌کننده‌های مرتبه‌کسری
۳۸	۳-۳-۱ نسل اول کنترل‌کننده‌های کرون
۳۹	۳-۳-۲ نسل دوم کنترل‌کننده‌های کرون
۴۰	۳-۳-۳ نسل سوم کنترل‌کننده‌های کرون
۴۰	۳-۳-۳-۱ منحنی‌های بهره حلقه بسته ثابت
۴۱	۳-۳-۳-۲ منحنی‌های ضرایب میرایی تقریباً ثابت
۴۲	۳-۳-۳-۳ هدف کنترل‌کننده نسل سوم کرون

۴۴	۴-۳-۳ کنترل کننده‌های PID مرتبه کسری
۴۵	۱-۴-۳-۳ طراحی FOPID با کنترل مدل داخلی
۴۷	۲-۴-۳-۳ طراحی FOPID با می نیمم سازی
۴۸	۳-۴-۳-۳ طراحی FOPID با استفاده از قواعد تنظیمی
۴۹	۱-۳-۴-۳-۳ گروه اول قواعد تنظیمی
۵۰	۲-۳-۴-۳-۳ گروه دوم قواعد تنظیمی
۵۱	۵-۳-۳ روش طراحی کنترل کننده FOPID بهینه در این پایان نامه

فصل چهارم: طراحی کنترل کننده بهینه PID مرتبه کسری با استفاده از الگوریتم-

۵۳	های تکاملی
۵۴	۱-۴ مقدمه
۵۴	۲-۴ مقایسه عملکرد الگوریتم‌های ممیک پایه و پیشنهادی
۶۰	۳-۴ مقایسه عملکرد PID کلاسیک و PID مرتبه کسری
۶۲	۴-۴ مقایسه عملکرد الگوریتم‌های تکاملی در طراحی کنترل کننده PID مرتبه کسری
۸۱	فصل پنجم: نتیجه گیری و پیشنهادات
۸۲	۱-۵ نتیجه گیری
۸۳	۲-۵ پیشنهادات
۸۴	مراجع

فهرست شکل‌ها

- شکل (۱-۲): نحوه رقص زنبورها برای نشان دادن محل سایت غذایی ۱۸
- شکل (۱-۳): نمودار نیکولز با منحنی‌های بهره و ضرایب میرایی ثابت ۴۱
- شکل (۲-۳): هدف نسل سوم کنترل‌کننده کرون ۴۳
- شکل (۳-۳): بلوک دیاگرام کنترل مدل داخلی (چپ) و معادل آن (راست) ۴۵
- شکل (۱-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۵۷
- معيار I
- شکل (۲-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۵۸
- معيار II
- شکل (۳-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۵۸
- معيار III
- شکل (۴-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۵۹
- معيار IV
- شکل (۵-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۵۹
- معيار V
- شکل (۶-۴): مقایسه عملکرد الگوریتم‌های ممیتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع ۶۰
- معيار VI
- شکل (۷-۴): پاسخ پله فرآیند کنترل نشده و فرآیند کنترل شده با استفاده از کنترل‌کننده‌های ۶۱
- مختلف
- شکل (۸-۴): پاسخ پله فرآیند کنترل نشده I و کنترل شده با استفاده از کنترل‌کننده‌های بهینه ۷۵
- شده توسط الگوریتم‌ها
- شکل (۹-۴): پاسخ پله فرآیند کنترل نشده II و کنترل شده با استفاده از کنترل‌کننده‌های بهینه

- ۷۵ شده توسط الگوریتم‌ها
- شکل (۴-۱۰): پاسخ پله فرآیند کنترل نشده III و کنترل شده با استفاده از کنترل‌کننده‌های
- ۷۶ بهینه شده توسط الگوریتم‌ها
- شکل (۴-۱۱): روند تغییر پارامترهای طراحی توسط الگوریتم ممیتیک پیشنهادی
- ۷۷ شکل (۴-۱۲): روند تغییر پارامترهای طراحی توسط الگوریتم بهینه‌سازی انبوه ذرات
- ۷۷ شکل (۴-۱۳): روند تغییر پارامترهای طراحی توسط الگوریتم کرم شب‌تاب
- ۷۸ شکل (۴-۱۴): روند تغییر پارامترهای طراحی توسط الگوریتم رقابت استعماری
- ۷۸ شکل (۴-۱۵): روند تغییر پارامترهای طراحی توسط الگوریتم ممیتیک پایه
- ۷۹ شکل (۴-۱۶): روند تغییر پارامترهای طراحی توسط الگوریتم تکامل تفاضلی
- ۷۹ شکل (۴-۱۷): روند تغییر پارامترهای طراحی توسط الگوریتم زنبورها

فهرست جدول‌ها

- جدول (۳-۱): پارامترها برای گروه اول قواعد تنظیمی ۴۹
- جدول (۳-۲): پارامترها برای گروه دوم قواعد تنظیمی ۵۰
- جدول (۴-۱): توابع معیار استفاده شده در مقایسه الگوریتم‌های ممیتیک پایه و پیشنهادی ۵۵
- جدول (۴-۲): توابع معیار استفاده شده در مقایسه الگوریتم‌های ممیتیک پایه و پیشنهادی ۵۶
- جدول (۴-۳): نمایش ضرایب کنترل‌کننده‌ها و مقایسه حداکثر فراجهدش آنها ۶۱
- جدول (۴-۴): نمایش فرآیندهای تحت کنترل، مطلوبات طراحی و تنظیمات الگوریتم‌ها ۶۴
- جدول (۴-۵): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰ تکرار برای فرآیند اول ۶۵
- جدول (۴-۶): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار برای فرآیند اول ۶۶
- جدول (۴-۷): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار برای فرآیند اول ۶۷
- جدول (۴-۸): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰ تکرار برای فرآیند دوم ۶۸
- جدول (۴-۹): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار برای فرآیند دوم ۶۹
- جدول (۴-۱۰): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار برای فرآیند دوم ۷۰
- جدول (۴-۱۱): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰ تکرار برای فرآیند سوم ۷۱
- جدول (۴-۱۲): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار برای فرآیند سوم ۷۲
- جدول (۴-۱۳): پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار برای فرآیند سوم ۷۳
- جدول (۴-۱۴): پاسخ پله‌ی فرآیندهای کنترل‌شده با کنترل‌کننده‌ی مرتبه‌کسری بهینه‌شده ۷۴
- توسط الگوریتم‌های تکاملی

فصل اول :

حسابان کسری

در این فصل ابتدا تاریخچه‌ای از حسابان کسری در بخش (۲-۱) ارائه خواهد شد. سپس به بررسی تعاریف به کار رفته در حسابان مرتبه کسری در بخش (۳-۱) می‌پردازیم. ویژگی‌های مشتق مرتبه-کسری و معادله‌ی دیفرانسیل با مرتبه‌ی کسری خطی به ترتیب در بخش‌های (۴-۱) و (۵-۱) بررسی خواهند شد. در بخش (۶-۱) به بررسی تبدیل لاپلاس خواهیم پرداخت. توابع تبدیل مرتبه کسری، کنترل پذیری، رویت پذیری و پایداری سیستم‌های مرتبه کسری نیز در بخش (۷-۱) بررسی خواهند شد. در نهایت در بخش (۸-۱) به ارائه اهداف و روند ارائه پروژه خواهیم پرداخت. در ادامه در فصل دوم الگوریتم‌های تکاملی بررسی خواهند شد و الگوریتم پیشنهادی معرفی خواهد شد. فصل سوم به بررسی روش‌های طراحی کنترل کننده مرتبه کسری پرداخته خواهد شد. در نهایت در فصل چهارم به ارائه شبیه سازی ها و نتایج پرداخته خواهد شد.

۲-۱ تاریخچه‌ی حسابان کسری

در سال ۱۶۹۵ لایبنیز^۱ طی نامه‌ای به هوپیتال^۲ سوالی را مطرح کرد مبنی بر اینکه: " آیا می‌توان مشتق با مرتبه‌ی صحیح $\frac{d^ny(t)}{dx^n}$ را به مرتبه غیر صحیح تعمیم داد؟ ". هوپیتال نامه‌ی لایبنیز را با سوالی دیگر پاسخ داد: " در صورتی که $n = \frac{1}{2}$ باشد، آنگاه چه می‌شود؟ ". لایبنیز در تاریخ ۳۰ سپتامبر ۱۶۹۵ نامه را بدین گونه پاسخ داد: " یک پارادوکس واضح، که البته زمانی نتایج مفید استفاده از آن مشخص خواهد شد ". سوالی که لایبنیز در مورد مشتق مرتبه کسری مطرح نمود مبنای مطالعات بسیاری در طی ۳۰۰ سال گذشته قرار گرفته است. ریاضی دانان بسیاری در طی سال‌ها در این زمینه تلاش نموده‌اند که از بین آنها می‌توان گفت لیوویل^۳، ریمن^۱ و ویل^۲ بیشترین

^۱ Leibniz

^۲ Hopital

^۳ Liouville

نقش را در شکوفایی و گسترش حسابان مرتبه کسری ایفا کرده‌اند [۱]. پایه‌ی حسابان مرتبه کسری به قرن هفدهم برمی‌گردد، زمانی که لایبنیز و نیوتن^۳ حسابان را ابداع کردند. اما تئوری مشتقات مرتبه کسری به صورت گسترده در قرن نوزدهم ایجاد گردید. جهت اطلاعات بیشتر به [۲-۴] مراجعه شود. در گسترش حسابان مرتبه کسری تعاریف مشتق و انتگرال مرتبه کسری بسیاری بوجود آمدند. برخی از این تعاریف به طور مستقیم از حسابان مرتبه صحیح بدست آمده‌اند. در این بین سه تعریف ریمن-لیوویل، گرونوالد-لتنیکوف^۴ و کاپوتو^۵، که شامل رابطه انتگرال کوشی^۶ هم می‌باشند، از همه پرکاربردتر هستند. در ابتدا تعاریف را ارائه کرده و سپس به بررسی خواص آنها می‌پردازیم.

۳-۱ تعاریف به کار رفته در بحث حسابان کسری

۱-۳-۱ تابع گاما^۷

تابع گاما یکی از توابع مهم در بحث حسابان کسری است که به صورت بسط فاکتوریال^۸ برای تمام اعداد حقیقی می‌باشد. رابطه زیر بیانگر این تابع می‌باشد:

$$\Gamma(z) = \int_0^{\infty} e^{-u} u^{z-1} du, \quad \forall z \in \mathbb{R} \quad (1-1)$$

شایان ذکر است که تابع گاما برای مقادیر صحیح غیر مثبت به سمت بینهایت میل می‌کند.

۲-۳-۱ تابع میتاگ-لفلر^۹

^۱ Riemann

^۲ Weyl

^۳ Newton

^۴ Grunwald-Letnikov

^۵ Caputo

^۶ Cauchy

^۷ Gamma

^۸ Factorial

^۹ Mittag-Leffler

تابع میتاگ-لفلر یکی از پرکاربردترین توابع در حسابان کسری می‌باشد. فرم عمومی این تابع به صورت رابطه (۲-۱) تعریف می‌شود [۲]:

$$E\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha > 0 \quad (2-1)$$

تابع میتاگ-لفلر بسط تابع نمایی می‌باشد. گاهی از این تابع به صورت دو آرگومانی استفاده می‌شود. تابع میتاگ-لفلر با دو آرگومان به صورت رابطه (۳-۱) تعریف می‌شود:

$$E\alpha, \beta(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha > 0, \beta > 0 \quad (3-1)$$

در این پایان نامه هرگاه صحبت از تابع میتاگ-لفلر بدون هیچ پیشوندی می‌شود، منظور از آن، تابع با یک آرگومان می‌باشد. در بخش بعد تعاریف متداول و پرکاربرد در مشتقات مرتبه کسری مورد بررسی قرار خواهد گرفت.

۳-۳-۱ انتگرال مرتبه کسری کوشی

این تعریف همان بسط عمومی انتگرال مرتبه صحیح کوشی می‌باشد و به صورت رابطه (۴-۱) تعریف می‌شود:

$$\mathcal{D}^n f(t) = \frac{\Gamma(n+1)}{2\pi j} \oint \frac{f(\tau)}{(\tau-t)^{n+1}} d\tau \quad (4-1)$$

در این رابطه انتگرال روی سطح منحنی است که $f(t)$ را احاطه کرده است.

۴-۳-۱ مشتق مرتبه کسری گرونوالد-لتنیکوف

این تعریف به صورت زیر بیان می‌شود:

$$\mathcal{D}^n f(t) = \frac{d^n f(t)}{dt^n} = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^n (-1)^k \binom{n}{k} f(t - kh) \quad (5-1)$$

که در رابطه‌ی بالا خواهیم داشت:

$$\binom{n}{k} = \frac{n(n-1)(n-2) \dots (n-k+1)}{k!} \quad (6-1)$$

رابطه‌ی (۵-۱) را می‌توان به صورت زیر تعریف کرد [۱]:

$$\mathfrak{D}^n = \begin{cases} \frac{d^n}{dt^n} & , Re\{n\} > 0 \\ 1 & , Re\{n\} = 0 \\ \int_0^t (d\tau)^{-n} & , Re\{n\} < 0 \end{cases} \quad (7-1)$$

۵-۳-۱ مشتق مرتبه کسری ریمن-لیوویل

انتگرال مرتبه کسری به صورت زیر تعریف می‌شود:

$$\mathfrak{D}^n f(t) = \frac{1}{\Gamma(n)} \int_0^t (t-\tau)^{n-1} f(\tau) d\tau \quad (8-1)$$

که در رابطه‌ی بالا $0 < n < 1$ و t حد بالای انتگرال‌گیری [۴] می‌باشد. تعریف ریمن-لیوویل پرکاربردترین تعریف در حسابان مرتبه کسری می‌باشد. این تعریف را می‌توان به صورت مشتق مرتبه-کسری به صورت زیر نوشت:

$$\mathfrak{D}^n f(t) = \frac{d^m}{dt^m} [\mathfrak{D}^{-(m-n)} f(t)] = \frac{1}{\Gamma(m-n)} \frac{d^m}{dt^m} \left[\int_0^t \frac{f(\tau)}{(t-\tau)^{n-m+1}} d\tau \right] \quad (9-1)$$

که در رابطه بالا $m-1 < n \leq m$ می‌باشد.

۶-۳-۱ مشتق مرتبه کسری کاپوتو

$$\mathfrak{D}^n f(t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \frac{f^{(m+1)}(\tau)}{(t-\tau)^\gamma} d\tau \quad (9-1)$$

که در رابطه‌ی بالا $n = m + \gamma$ ، m عدد صحیح و $0 < \gamma \leq 1$ می‌باشند. به طور مشابه انتگرال مرتبه کسری کاپوتو به صورت زیر تعریف می‌شود:

$$\mathfrak{D}^n f(t) = \frac{1}{\Gamma(-n)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1+n}} d\tau, \quad n < 0 \quad (10-1)$$

می‌توان نشان داد [۳] که برای گروهی از توابع حقیقی، تعاریف مشتقات مرتبه‌کسری گرونوالد-لتنیکوف و ریمن-لیوویل با هم برابر می‌باشند.

۴-۱ ویژگی‌های مشتق مرتبه‌کسری

مشتق مرتبه‌کسری ویژگی‌های زیر را دارا می‌باشد [۵]:

- در صورتی که $f(t)$ یک تابع تحلیلی باشد، مشتق مرتبه‌کسری $\mathcal{D}^n f(t)$ نیز تحلیلی خواهد بود.
- در صورتی که $m \in \mathbb{Z}^+$ و $n = m$ باشد، مشتق مرتبه‌کسری $\mathcal{D}^n f(t)$ همان مشتق مرتبه صحیح $f^{(m)}(t)$ می‌باشد.
- مشتق مرتبه‌کسری خطی می‌باشد، به عبارتی به ازای a و b ثابت داریم:

$$\begin{aligned} \mathcal{D}^n [af(t) + bg(t)] & \quad (11-1) \\ & = a\mathcal{D}^n f(t) + b\mathcal{D}^n g(t) \end{aligned}$$

- عملگرهای مشتق مرتبه‌کسری از قوانین جابجایی‌پذیری و جمع‌پذیری طبیعت می‌کنند.

$$\mathcal{D}^n [\mathcal{D}^m f(t)] = \mathcal{D}^m [\mathcal{D}^n f(t)] = \mathcal{D}^{n+m} f(t) \quad (12-1)$$

- مشتق مرتبه‌کسری با $n = 0$ یک عملگر شناسایی می‌باشد.

$$\mathcal{D}^0 f(t) = f(t) \quad (13-1)$$

- در صورتی که $f(t)$ و $g(t)$ توابعی پیوسته در بازه $[0, t]$ باشند، قانون لایبنیز برای مشتقات مرتبه‌کسری به صورت زیر نتیجه می‌شود:

$$\mathcal{D}^n (f(t)g(t)) = \sum_{k=0}^{\infty} \binom{n}{k} f^{(k)}(t) \mathcal{D}^{n-k} g(t) \quad (14-1)$$

این قانون جهت توابعی که از ضرب دو تابع، که یکی چند جمله‌ای و دیگری تابعی با مشتق مرتبه-کسری شناخته شده است، مفید می‌باشد. اثبات این قانون در [۵] آمده است.

۵-۱ معادله دیفرانسیل با مرتبه‌ی کسری خطی

معادلات دیفرانسیل یک سیستم دینامیکی پیوسته‌زمانی مرتبه‌کسری خطی را می‌توان به صورت زیر نوشت [۶]:

$$H(\mathcal{D}^{\alpha_0 \alpha_1 \alpha_0 \dots \alpha_m})(y_1, y_2, \dots, y_l) = G(\mathcal{D}^{\beta_0 \beta_1 \beta_2 \dots \beta_n})(u_1, u_2, \dots, u_k) \quad (15-1)$$

که در آن u_i و y_i توابعی از زمان و $H(\cdot)$ و $G(\cdot)$ قوانین ترکیب عملگر مشتق مرتبه‌کسری می‌باشند. برای سیستم خطی تغییرناپذیر با زمان (LTI) یک متغیره معادله‌ی (۱۵-۱) به صورت زیر بازنویسی می‌شود:

$$\begin{aligned} H(\mathcal{D}^{\alpha_0 \alpha_1 \alpha_0 \dots \alpha_m})y(t) \\ = G(\mathcal{D}^{\beta_0 \beta_1 \beta_2 \dots \beta_n})u(t) \end{aligned} \quad (16-1)$$

که در آن $H(\cdot)$ و $G(\cdot)$ به صورت زیر تعریف می‌شوند:

$$H(\mathcal{D}^{\alpha_0 \alpha_1 \alpha_0 \dots \alpha_m}) = \sum_{k=0}^m a_k \mathcal{D}^{\alpha_k}, \quad a_k \in \mathbb{R} \quad (17-1)$$

$$G(\mathcal{D}^{\beta_0 \beta_1 \beta_2 \dots \beta_n}) = \sum_{k=0}^n b_k \mathcal{D}^{\beta_k}, \quad b_k \in \mathbb{R} \quad (18-1)$$

به وضوح می‌توان نشان داد که:

$$\begin{aligned} a_n \mathcal{D}^{\alpha_n} y(t) + a_{n-1} \mathcal{D}^{\alpha_{n-1}} y(t) + \dots + a_0 \mathcal{D}^{\alpha_0} y(t) = b_m \mathcal{D}^{\beta_m} u(t) \\ + b_{m-1} \mathcal{D}^{\beta_{m-1}} u(t) + \dots + b_0 \mathcal{D}^{\beta_0} u(t) \end{aligned}$$

(۱۹-۱)

که در آن $\begin{cases} a_k (k = 0, 1, \dots, n) \\ b_k (k = 0, 1, \dots, m) \end{cases}$ ضرایب ثابت معادله دیفرانسیل بوده و $\begin{cases} \alpha_k (k = 0, 1, \dots, n) \\ \beta_k (k = 0, 1, \dots, m) \end{cases}$

نیز مقادیر حقیقی می‌باشند. بدون از دست دادن کلیت مساله می‌توان فرض کرد که

$$\begin{cases} \beta_m > \beta_{m-1} > \dots > \beta_0 > 0 \\ \alpha_n > \alpha_{n-1} > \dots > \alpha_0 > 0 \end{cases}$$

می‌توان معادله دیفرانسیل فوق را به صورت تحلیلی حل نمود، جهت این کار می‌توان از تابع میتاگ-

لفر که تابعی بر حسب پارامترهای معادله دیفرانسیل می‌باشد، استفاده کرد [۶].

اگر در معادله‌ی (۱۹-۱) تمام مرتبه‌های مشتق مضرب صحیح از یک مرتبه‌ی خاص (α) باشند،

یعنی $\alpha_k, \beta_k = k\alpha$, $\alpha \in \mathbb{R}^+$ سیستم مرتبه-نسبی^۱ بوده و معادله‌ی (۱۹-۱) به صورت زیر

بازنویسی خواهد شد:

$$\sum_{k=0}^n a_k \mathcal{D}^{k\alpha} y(t) = \sum_{k=0}^m b_k \mathcal{D}^{k\alpha} u(t) \quad (20-1)$$

در صورتی که در معادله‌ی بالا $\alpha = \frac{1}{l}$ و $l \in \mathbb{R}^+$ باشد، سیستم از مرتبه گویا^۲ خواهد بود.

۱-۶ تبدیل لاپلاس

تبدیل لاپلاس یکی از ابزارهای اساسی در مهندسی کنترل می‌باشد. رابطه تبدیل لاپلاس برای

مشتق مرتبه کسری طبق تعریف لتنیکوف که در زیر آمده است:

$$\mathcal{D}^n f(t) = \frac{1}{\Gamma(m-n)} \frac{d^n}{dt^n} \int_0^t \frac{f(\tau)}{(t-\tau)^{n-m+1}} d\tau, \quad (m-1 < n \leq m) \quad (21-1)$$

به صورت زیر می‌باشد [۷]:

^۱ Commensurate Order

^۲ Rational-order

$$L\{\mathcal{D}^n f(t)\} = \begin{cases} \int_0^\infty e^{-st} \mathcal{D}^n f(t) dt = s^n F(s) - \sum_{k=0}^{n-1} s^k \mathcal{D}^{n-k-1} f(t_0), & m-1 < n < \infty, n > 0 \\ s^n F(s) & , n \leq 0 \end{cases} \quad \begin{matrix} ۱) \\ ۲) \end{matrix}$$

که در آن $F(s) = L\{f(t)\}$ تبدیل لاپلاس تابع $f(t)$ می‌باشد. بعنوان مثال تبدیل لاپلاس زیر را در نظر بگیرید:

$$G(s) = s^v, \quad v \neq 0 \quad (۲۳-۱)$$

با استفاده از تبدیل لاپلاس فوق می‌توان پاسخ ضربه و پله تابع انتقال فوق را بدست آورد. پاسخ ضربه تابع $G(s)$ به صورت زیر می‌باشد:

$$g(t) = \frac{t^{-v-1}}{\Gamma(-v)}, \quad t > 0 \quad (۲۴-۱)$$

در صورتی که $v > 0$ باشد، جهت بدست آوردن (۲۴-۱) باید فرض کنیم در رابطه‌ی (۲۲-۱) شرایط اولیه صفر هستند [۷]. پاسخ پله تابع $G(s)$ به صورت زیر می‌باشد:

$$r(t) = \frac{t^{-v}}{\Gamma(-v+1)}, \quad t > 0 \quad (۲۵-۱)$$

معادله‌ی فوق به صورت زیر بدست می‌آید:

$$L^{-1}\{G(s)L\{1\}\} = L^{-1}\left\{G(s) * \frac{1}{s}\right\} = L^{-1}\{s^{v-1}\} = \frac{t^{-v}}{\Gamma(-v+1)} \quad (۲۶-۱)$$

همچنین پاسخ فرکانسی تابع انتقال $G(s)$ را می‌توان به صورت زیر بدست آورد:

$$|G(j\omega)| = |j^v \omega^v| = |\omega^v| = \omega^v \rightarrow 20 \log |G(j\omega)| = 20v \log \omega \quad (۲۷-۱)$$

$$\begin{aligned} \arg[G(j\omega)] &= \arg[j^v \omega^v] = \arg[j^v] \\ &= v \frac{\pi}{2} \end{aligned} \quad (۲۸-۱)$$

بنابراین با اعمال تبدیل لاپلاس به معادله‌ی (۱۹-۱) با شرایط اولیه صفر، می‌توان نمایش ورودی-خروجی سیستم مرتبه کسری را به صورت زیر بدست آورد [۷,۶]:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta m} u(t) + b_{m-1} s^{\beta(m-1)} u(t) + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha n} y(t) + a_{n-1} s^{\alpha(n-1)} y(t) + \dots + a_0 s^{\alpha_0}} \quad (29-1)$$

$$= \frac{\sum_{k=0}^m b_k (s^\beta)^k}{\sum_{k=0}^n a_k (s^\alpha)^k}$$

بنابراین همانطور که مشاهده می‌شود، یک سیستم مرتبه کسری در حوزه لاپلاس دارای تابع تبدیلی از مرتبه‌ی غیرگویا می‌باشد.

۷-۱ توابع تبدیل مرتبه کسری

رفتار سیستم‌های دینامیکی بسیاری را می‌توان با استفاده از معادلات مشتقی شامل مشتقات مرتبه-کسری مدل کرد. با اعمال تبدیل لاپلاس به چنین معادلاتی و با فرض شرایط اولیه‌ی صفر می‌توان توابع تبدیل با مرتبه‌ی غیر صحیح را ایجاد نمود. از آنجا که در مطالعات مهندسی اعداد تا محدوده‌ی خاصی در محاسبات اعمال می‌شوند، به توان‌های غیر صحیح این چنینی مرتبه کسری گفته می‌شود. تمام سیستم‌های دینامیکی را می‌توان به فرم زیر نوشت [۸،۶]:

$$F(s) = \frac{\sum_{i=1}^M b_i s^{q_i}}{\sum_{j=1}^N a_j s^{p_j}}, \quad M, N \in \mathbb{N}_0 \wedge a_j, b_i, p_j, q_i \in \mathbb{R} \quad (30-1)$$

می‌توان گفت که تمام مولفه‌های s مضر بی از یک مقدار معین هستند:

$$F(s) = \frac{\sum_{i=0}^M b_i s^{iq}}{\sum_{j=0}^N a_j s^{jq}}, \quad M, N \in \mathbb{N}_0 \wedge q, a_j, b_i \in \mathbb{R} \quad (31-1)$$

توابع تبدیلی مانند (۳۱-۱) را نسبی می‌نامند و به هر مقدار q که باعث نسبی شدن تابع تبدیل به فرم (۳۱-۱) شود، مرتبه‌ی نسبی گفته می‌شود. عموماً q معکوس یک عدد طبیعی می‌باشد.

$$F(s) = \frac{\sum_{i=0}^M b_i s^{i/Q}}{\sum_{j=0}^N a_j s^{j/Q}}, \quad M, N \in \mathbb{N}_0 \wedge Q \in \mathbb{N} \wedge a_j, b_i \in \mathbb{R} \quad (32-1)$$

باید توجه داشت که در عمل تمام توابع تبدیل موجود به این شکل نیستند. توابع تبدیل مرتبه صحیح عمومی حالت خاصی از توابع تبدیل مرتبه کسری (۳۲-۱) هستند، زمانی که $Q = 1$ باشد.

۷-۱-۱ نمایش فضای حالت مرتبه کسری

برای سیستم‌های مرتبه‌کسری با مرتبه‌ی نسبی گویا می‌توان نمایش فضای حالتی به فرم زیر بیان کرد:

$$\begin{aligned} \mathcal{D}^{\frac{1}{Q}}x &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (33-1)$$

که در آن u ورودی‌ها، x حالت‌ها و y خروجی‌های سیستم هستند. نمایش فضای حالت مرتبه-صحیح عمومی حالت خاصی از نمایش فضای حالت مرتبه‌کسری (33-1) هستند، زمانی که $Q = 1$ باشد. همانند نمایش فضای حالت مرتبه صحیح، نمایش فضای حالت مرتبه‌کسری نیز یکتا نمی‌باشد. به عبارت دیگر نمایش‌های فضای حالت متعددی برای یک تابع تبدیل مرتبه‌کسری وجود دارند.

قضیه ۱-۱: نمایش فضای حالت (33-1) متناظر با ماتریس تابع تبدیل زیر

$$y = \left[C(s^{\frac{1}{Q}}I - A)^{-1}B + D \right] u \quad (34-1)$$

نشان می‌دهد که اگر به تساوی اول تبدیل لاپلاس اعمال شود، هیچ شرط اولیه غیر صفری ظاهر نخواهد شد [۸].

اثبات: می‌توان نوشت:

$$s^{\frac{1}{Q}}x = Ax + Bu \Leftrightarrow \left(s^{\frac{1}{Q}}I - A \right) x = Bu \Rightarrow x = \left(s^{\frac{1}{Q}}I - A \right)^{-1}Bu \quad (35-1)$$

که با جایگذاری در تساوی دوم رابطه‌ی (33-1)، رابطه‌ی (34-1) به دست خواهد آمد [۵، ۶].

باید به این نکته توجه داشت که تنها در صورتی می‌توان یک تابع تبدیل را از نمایش فضای حالت بدست آورد که شرایط اولیه مجاز داشته باشد. حال در صورتی که از تعریف ریمن-لیوویل مشتقات مرتبه‌کسری استفاده شده باشد، این شرایط اولیه ممکن است شامل مشتقات مرتبه‌کسری تابع باشند. لذا عموماً از تعاریف کاپوتو و گرونوالد-لتنیکوف استفاده می‌شود [۵].

قضیه ۱-۲: تابع تبدیل زیر را در نظر بگیرید:

$$F(s) = \frac{\sum_{i=0}^M b_i s^{i/Q}}{s^{N/Q} + \sum_{i=0}^N a_i s^{i/Q}}, \quad N, Q \in \mathbb{N} \wedge a_i, b_i \in \mathbb{R} \quad (36-1)$$

تابع فوق نمایش فضای حالتی به فرم زیر خواهد داشت:

$$\mathbb{D}^{\frac{1}{Q}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} -a_{N-1} & -a_{N-2} & -a_{N-3} & \dots & -a_1 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u \quad (37-1)$$

$$y = [b_{N-1} - a_{N-1}b_N \mid b_{N-2} - a_{N-2}b_N \mid \dots \mid b_0 - a_0b_N] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} + b_N u$$

به فرم نمایش فضای حالت فوق کانونیکال کنترل پذیر گفته می شود و شرایط اعمال قضیه (۱-۱) را دارد.

قضیه ۳-۱: تابع تبدیل (۳۶-۱) نمایش فضای حالتی به فرم زیر خواهد داشت:

$$\mathbb{D}^{\frac{1}{Q}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & 0 & \dots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & -a_{N-2} \\ 0 & 0 & 0 & \dots & 1 & -a_{N-3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \begin{bmatrix} b_0 - a_0b_N \\ b_1 - a_1b_N \\ b_2 - a_2b_N \\ \vdots \\ b_{N-2} - a_{N-2}b_N \\ b_{N-1} - a_{N-1}b_N \end{bmatrix} u$$

$$y = [0 \mid 0 \mid 0 \mid \dots \mid 0 \mid 1] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} + b_N u \quad (38-1)$$

به فرم نمایش فضای حالت فوق کانونیکال مشاهده پذیر گفته می شود و شرایط اعمال قضیه (۱-۱) را دارد.

به فرم نمایش فضای حالت فوق کانونیکال جردن گفته می‌شود و شرایط اعمال قضیه (۱-۱) را دارد. در صورتی که قطب‌های دیگری از هر مرتبه‌ای وجود داشته باشند، نحوه نمایش همانند قطب p_1 خواهد بود.

۲-۷-۱ مشاهده پذیری، کنترل پذیری و پایداری سیستم‌های مرتبه کسری

دو قضیه زیر را به روشی مشابه مرتبه صحیح می‌توان اثبات کرد [۶].

قضیه ۱-۶: سیستم مرتبه کسری (۳۳-۱) مشاهده پذیر است، اگر و تنها اگر ماتریس مشاهده پذیری زیر دارای مرتبه‌ی^۱ کامل باشد.

$$\varphi_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \quad (۴۳-۱)$$

که در آن N تعداد حالت‌ها می‌باشد.

قضیه ۱-۷: سیستم مرتبه کسری (۳۳-۱) کنترل پذیر است، اگر و تنها اگر ماتریس کنترل پذیری زیر دارای مرتبه‌ی کامل باشد.

$$\varphi_o = [B \quad AB \quad \dots \quad A^{N-1}B] \quad (۴۴-۱)$$

که در آن N تعداد حالت‌ها می‌باشد.

قضیه ۱-۸ (قضیه متیگنون [۸,۹]): سیستم (۳۲-۱) پایدار است، اگر و تنها اگر تمام ریشه‌های مختلط آن شرط زیر را برآورده کنند.

$$|\arg(r)| > \frac{\pi}{2Q} \quad (۴۵-۱)$$

شرط پایداری برای سیستم‌های مرتبه صحیح همانند سیستم‌های مرتبه کسری با $Q = 1$ می‌باشد.

^۱ Rank

۸-۱ اهداف و روند ارائه پروژه

در این پایان نامه ابتدا در فصل دوم به بررسی الگوریتم‌های بهینه‌سازی تکاملی پرداخته و الگوریتم بهینه‌سازی پیشنهادی را مورد بررسی قرار خواهیم داد. در ادامه در فصل سوم چند روش از روش‌های تقریب سیستم‌های مرتبه‌کسری و کنترل‌کننده^۱ PID مرتبه‌کسری را معرفی خواهیم کرد و روش‌های متداول طراحی آن را مورد بررسی قرار خواهیم داد. سپس، یک روش طراحی برای کنترل‌کننده PID مرتبه‌کسری توسط الگوریتم‌های بهینه‌سازی ارائه خواهیم کرد. بر همین اساس، در فصل چهارم یک تابع معیار مناسب که هم شرایط پاسخ زمانی و هم شرایط پاسخ فرکانسی را برآورده می‌کند، ارائه می‌گردد. کارایی الگوریتم‌های بهینه‌سازی تکاملی در طراحی کنترل‌کننده PID مرتبه‌کسری برای سه فرآیند مرتبه صحیح، مرتبه‌کسری و تاخیردار نشان داده شده است و برتری الگوریتم پیشنهادی نشان داده خواهد شد.

^۱ تناسبی-مشتقی-انتهایی

فصل دوم :

الگوریتم‌های بهینه‌سازی تکاملی

در این فصل ابتدا به تعریف بهینه‌سازی تکاملی در بخش (۲-۲) می‌پردازیم. سپس در بخش (۳-۲) مروری خواهیم داشت بر چند روش از روش‌های (الگوریتم‌های) بهینه‌سازی تکاملی و در نهایت در زیربخش (۷-۳-۲) به بررسی الگوریتم پیشنهادی خواهیم پرداخت.

۲-۲ بهینه‌سازی تکاملی

بهینه‌سازی فرآیندی است که جهت بهبود پاسخ از آن استفاده می‌شود. بهینه‌سازی از یافتن بهترین پاسخ برای یک مساله صحبت می‌کند. لفظ بهترین به طور ضمنی بیان می‌کند که بیش از یک پاسخ برای مساله مورد نظر وجود دارد، که البته تمام پاسخ‌ها دارای ارزش یکسان نیستند. بهینه‌سازی شامل دو عنصر می‌نیمم و ماکزیمم می‌باشد. زمانی می‌گوییم عنصری می‌نیمم (یا ماکزیمم) است که مقادیر بدست آمده برای آن، یک سری شرایط را ارضاء کنند. فرآیند بهینه‌سازی، فرآیندی تکراری است که تا زمانی که شرایط خاتمه برقرار نشده باشد ادامه می‌یابد. شرایط خاتمه می‌توانند تعداد تکرار الگوریتم در هر اجرا یا رسیدن به پاسخ دلخواه باشند [۱۰]. دو روش جستجوی مجزا جهت حل مسائل بهینه‌سازی وجود دارد که عبارتند از: جستجوی محلی^۱ و جستجوی سراسری^۲. در روش جستجوی محلی، الگوریتم بهینه‌ترین پاسخ را از بین پاسخ‌های بدست آمده جستجو می‌کند، حال آنکه در روش جستجوی سراسری الگوریتم در کل فضای جستجو به دنبال پاسخ‌های بهینه می‌گردد. به مجموعه‌ی تمام پاسخ‌های ممکن فضای جستجو^۳ گفته می‌شود. هر دو روش جستجو مزایا و معایبی دارند که در ادامه به بررسی آنها خواهیم پرداخت.

^۱ Local Search

^۲ Global Search

^۳ Search Space

الگوریتم‌های جستجوی سراسری، مانند الگوریتم‌های تکاملی^۱، الگوریتم‌هایی بر پایه جستجوی تصادفی هستند، که از تکامل عناصر طبیعت الهام گرفته شده‌اند، مانند: الگوریتم ژنتیک^۲ [۱۱]، الگوریتم بهینه‌سازی انبوه ذرات^۳ [۱۳، ۱۲]، الگوریتم زنبورها^۴ [۱۴] و الگوریتم کرم شب‌تاب^۵ [۱۵]. این الگوریتم‌ها عملگرهای خود را جهت یافتن پاسخ بهینه‌ی سراسری در تمام فضای جستجو پخش می‌کنند. منظور از بهینه‌ترین پاسخ (در مسائل بهینه‌سازی)، کوچکترین مقداری می‌باشد که تابع هدف (تابع هزینه)^۶ به خود اختصاص می‌دهد. تابع هدف همان تابع تحت بهینه‌سازی می‌باشد. در طرف مقابل، الگوریتم‌های جستجوی محلی، الگوریتم‌هایی هستند که به‌صورت جهت دار تنها به بخشی از فضای جستجو یا پاسخ‌های به‌دست آمده اعمال می‌شوند. زمانی که فضای جستجو دارای نقاط اکسترمم زیادی باشد الگوریتم‌های جستجوی سراسری با سرعت بیشتری پاسخ خواهند داد، اما پاسخ‌هایی که از جستجوی محلی بدست می‌آیند به مراتب دقیق‌تر و بهینه‌تر هستند، زیرا این امکان وجود دارد که الگوریتم‌های جستجوی سراسری در نقاط اکسترمم محلی گرفتار شوند [۱۶-۱۸].

۳-۲ الگوریتم‌های بهینه‌سازی تکاملی

در این بخش به بررسی چند روش از روش‌های بهینه‌سازی تکاملی خواهیم پرداخت. در بخش (۲-۱-۳) به بررسی الگوریتم زنبورها می‌پردازیم. در بخش (۲-۳-۲) الگوریتم کرم شب‌تاب ارائه خواهد شد. الگوریتم بهینه‌سازی انبوه ذرات را در بخش (۲-۳-۳) بررسی خواهیم کرد. الگوریتم‌های تکامل تفاضلی^۷ و رقابت استعماری^۸ نیز به ترتیب در بخش‌های (۲-۳-۴) و (۲-۳-۵) بررسی خواهند شد.

^۱ Evolutionary Algorithms

^۲ Genetic Algorithm

^۳ Particle Swarm Optimization (PSO)

^۴ Bees Algorithm (BA)

^۵ Firefly Algorithm (FA)

^۶ Objective Function (Cost Function)

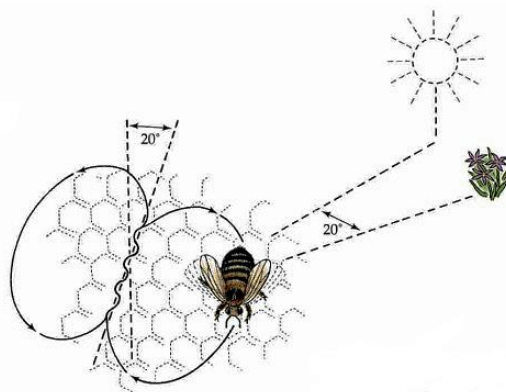
^۷ Differential Evolution Algorithm (DE)

^۸ Imperialistic Competitive Algorithm (ICA)

در بخش (۲-۳-۶) به بررسی الگوریتم ممتیک^۱ پایه پرداخته و در نهایت در بخش (۲-۳-۷) به بررسی الگوریتم ممتیک پیشنهادی خواهیم پرداخت.

۲-۳-۱ الگوریتم زنبورها

تاکنون مطالعات بسیاری در زمینه رفتارهای زنبورهای عسل صورت گرفته است، و لذا عضوی نسبتاً جدید در خانواده الگوریتم‌های تکاملی به نام الگوریتم زنبورها (کلونی زنبورها)، از بازسازی بخشی از رفتار و طبیعت زنبورهای عسل و اضافه کردن ویژگی‌هایی به آن بدست آمده است [۱۹]. این الگوریتم به طور کلی از دو رفتار زنبورها الهام گرفته است: نحوه جفت‌گیری و نحوه کاوش غذا. یکی از مهمترین مکانیزم‌هایی که بر نحوه کاوش غذای زنبورهای عسل اثر می‌گذارد، رقص چرخشی^۲ آنها می‌باشد. ابتدا تعدادی زنبور بعنوان زنبورهای پیشاهنگ مامور جستجوی منابع غذایی جدید اطراف کندو می‌شوند. زنبورهایی که در امر جستجوی غذا موفق بوده‌اند (زنبورهای پیشاهنگ موفق)، به کندو بازگشته و در محلی خاص شروع به انجام رقص چرخشی می‌کنند. با انجام این رقص اطلاعات خود در زمینه میزان، جهت و فاصله سایت‌های غذایی را با دیگر زنبورهای کلونی به اشتراک می‌گذارند. هرچه طول قسمت پیچشی رقص بیشتر باشد، به معنی غنی‌تر بودن آن سایت غذایی می‌باشد. مبنای نمایش جهت، زاویه نسبت به خورشید و مبنای نمایش فاصله، مدت زمان بخش چرخشی رقص می‌باشد [۲۰، ۲۱]. شکل (۲-۱) نمایشی از رقص زنبورها را نشان می‌دهد.



شکل (۲-۱). نحوه رقص زنبورها برای نشان دادن محل سایت غذایی

^۱ Memetic Algorithm (MA)

^۲ Waggle Dance

این روش، روشی بسیار موفق برای زنبورهای پیشاهنگ در هدایت زنبورهای کلونی می‌باشد. لذا زنبورها با سرعت و دقت بسیاری قادر به کشف سایت‌های غذایی و جمع آوری غذا خواهند بود. سپس زنبورهای آموزش دیده به سایت‌های مورد نظر رفته و در آنجا به جستجو می‌پردازند. زنبورهایی که بهترین محل‌ها را بیابند به کندو بازگشته و دیگر زنبورها را آموزش می‌دهند. دیگر زنبورها نیز در اطراف سایت به جستجوی سایت‌های غذایی جدید می‌پردازند. تاکنون از این الگوریتم در حل بسیاری از مسائل بهینه‌سازی استفاده شده است که برخی از آنها عبارتند از: آموزش شبکه‌های عصبی جهت تشخیص الگو [۲۲]، بهینه‌سازی سرویس اینترنت [۲۳]، بهینه‌سازی توابع پیوسته [۲۴] و مسائل داده کاوی [۲۵]. شبه کد الگوریتم زنبورها به صورت زیر می‌باشد.

۱. تعیین و مقداردهی اولیه اعضا (افراد) جمعیت
۲. بدست آوردن تابع هزینه برای هر فرد از جمعیت در موقعیت خود
۳. تا زمانی که شرایط توقف برقرار نشده است:
۴. انتخاب زنبورهای ممتاز (دارای بهترین تابع هزینه) بعنوان زنبورهای پیشاهنگ
۵. جستجوی سایت‌های غذایی توسط زنبورهای پیشاهنگ و انتخاب بهترین سایت‌ها
۶. هدایت زنبورهای دیگر به بهترین سایت‌های یافت شده
۷. بدست آوردن تابع هزینه برای افراد جمعیت و انتخاب بهترین آنها در هر سایت برای هدایت دیگران
۸. گماردن زنبورهای باقیمانده جهت جستجوی سایت‌های جدید
۹. پایان الگوریتم

الگوریتم زنبورها مزایا و معایبی دارد. از مزایای این الگوریتم می‌توان به قدرت و دقت قابل قبول آن در عمل جستجو و یافتن پاسخ‌های مناسب و جدید اشاره نمود. اما از عیوب اصلی این الگوریتم سرعت کم پاسخگویی و همچنین گرفتاری الگوریتم در می‌نیم‌های محلی می‌باشد [۲۳، ۱۴].

۲-۳-۲ الگوریتم کرم شب تاب

الگوریتم کرم شب تاب برای اولین بار توسط یانگ^۱ [۲۵] معرفی و بر اساس سه قانون زیر تعریف گشت.

- کرم‌های شب تاب به دیگر کرم‌های شب تاب، صرف نظر از جنسیت آنها، جذب می‌شود.
- جذابیت نسبی است، یعنی کرم‌های شب تاب با نور کمتر به سمت کرم‌های شب تاب با نور بیشتر جذب می‌شوند. در صورتی که هیچ کرم دیگری از یک کرم نورانی تر نباشد، این کرم به صورت تصادفی حرکت خواهد کرد.
- کرم‌های شب تاب با پخش کردن هرچه بیشتر نور طعمه را به سمت خود می‌کشند و در نهایت قربانی را با دیگر کرم‌ها به اشتراک می‌گذارند.

الگوریتم کرم شب تاب الگوریتمی تکاملی بر پایه‌ی جمعیت می‌باشد، که از رفتار کرم‌های شب تاب در جستجوی غذا و هوش جمعیتی آنها الهام گرفته شده است. در طبیعت کرم‌های شب تاب به صورت تصادفی حرکت می‌کنند و هرکدام که طعمه بهتری پیدا کند از خود نور بیشتری ساطع کرده، دیگران را به سمت خود جذب می‌کند. هر چه فاصله دو کرم از یکدیگر بیشتر باشد، درصد جذب آنها به یکدیگر کمتر می‌شود. به عبارتی فاصله با سرعت و میزان جذب نسبت عکس دارد. این الگوریتم از دو بخش اساسی تشکیل شده است:

الف) تغییرات شدت نور

میزان شدت نور به مقدار تابع هزینه بستگی دارد [۲۶]. لذا در مسائل می‌نیم‌سازی (ماکزیمم-سازی)، کرم شب تاب با نور زیادتر (کمتر)، کرم‌های شب تاب با نور کمتر (زیادتر) را به خود جذب می‌کند. فرض کنید n تعداد کرم‌ها، x_i موقعیت ذره i و $f(x_i)$ تابع هزینه باشد. لذا میزان درخشندگی هر کرم از برابر با مقدار تابع هزینه‌ی آن خواهد بود.

^۱ Yang

$$I_i = f(x_i), \quad 1 \leq i \leq n \quad (1-2)$$

(ب) حرکت به سمت کرم شب تاب با نور بیشتر

هر کرم دارای یک مشخصه جذب می باشد، که نشان می دهد کرم تا چه اندازه قوی است. این مشخصه مقداری نسبی بوده و با تغییر فاصله بین دو کرم i و j تغییر می کند. تابع جذب از رابطه زیر به دست می آید.

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (2-2)$$

که در آن β_0 میزان جذب به ازای $r = 0$ و γ ضریب جذب نور می باشد. حرکت کرم i با موقعیت x_i به سمت کرم j با موقعیت x_j (با نور بیشتر) از رابطه زیر بدست می آید.

$$x_i(t+1) = x_i(t) + \beta_r(x_j - x_i) \quad (3-2)$$

شبه کد الگوریتم کرم شب تاب به صورت زیر می باشد.

۱. تعیین و مقداردهی اولیه کرم ها (افراد) جمعیت
۲. بدست آوردن تابع هزینه برای هر فرد از جمعیت در موقعیت خود
۳. تا زمانی که شرایط توقف برقرار نشده است:
۴. تخصیص تصادفی شدت نور به هر فرد از جمعیت
۵. تعیین بهترین (پرنورترین) فرد در جمعیت بواسطه محاسبه توابع هزینه کل افراد جمعیت
۶. حرکت دیگر افراد جمعیت به سمت بهترین فرد و به روز رسانی شدت نور با توجه به آن
۷. پایان الگوریتم

عملکرد الگوریتم کرم شب تاب به طور کامل به تعداد جمعیت اولیه، تابع جذب و ضریب جذب وابسته است. هرچه ضریب جذب نور بیشتر باشد، سرعت جذب افراد جمعیت به سمت پرنورترین فرد بیشتر می شود. این الگوریتم دقت خوبی در یافتن پاسخ های جدید دارد، اما نه تنها دارای سرعت

همگرایی بسیار پایین می‌باشد، بلکه احتمال گرفتار شدن این الگوریتم در بهینه‌های محلی نیز وجود دارد [۲۶].

۳-۳-۲ الگوریتم بهینه‌سازی انبوه ذرات

الگوریتم بهینه‌سازی انبوه ذرات اولین بار توسط ابرهارت^۱ و کندی^۲ [۲۷] به عنوان یک مدل حرکت موضعی گروهی از حیوانات ارائه شد. این الگوریتم بر روی مسائل گوناگونی چون کمینه کردن توابع، نگاشت غیرخطی، آموزش شبکه‌های عصبی، وارونگی شبکه‌های عصبی و داده کاوی به خوبی اعمال شده است. این الگوریتم شامل انبوهی از ذرات می‌باشد که هر کدام می‌توانند پاسخی مطلوب برای مساله تحت بهینه‌سازی باشند. الگوریتم به طور مکرر موقعیت هر ذره را با محاسبه سرعت ذره و اعمال آن به موقعیت ذره به‌هنگام می‌کند. اگر $x_i(t)$ موقعیت ذره i ام در زمان t باشد، موقعیت ذره در هر زمان برابر خواهد بود با:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (۴-۲)$$

برای هر ذره i سرعت شامل سه مولفه می‌باشد :

- مولفه شناختی، که از بهترین موقعیتی که ذره تا کنون تجربه کرده است نتیجه می‌شود.
- مولفه اجتماعی، که از موقعیت بهترین ذره‌ای که ذره i از آن آگاه است نتیجه می‌شود.
- مولفه اینرسی، که از سرعت قبلی ذره نتیجه می‌شود.

هر ذره بهترین موقعیت خود را با محاسبه مولفه شناختی حفظ می‌کند. مولفه اجتماعی بر اساس موقعیت بهترین ذره که در همسایگی ذرات وجود دارد محاسبه می‌شود. همسایگی به ساختاری اجتماعی گفته می‌شود که در آن ذرات با یکدیگر در ارتباط باشند.

^۱ Eberhart

^۲ Kennedy

ساختار الگوریتم به گونه‌ای است که در آن تمام ذرات با یکدیگر کاملاً ارتباط دارند. در هر چرخه زمانی تمام ذرات بوسیله مولفه اجتماعی که از موقعیت بهترین ذره در کل ذرات نتیجه شده است، بهنگام می‌شوند. این مولفه را $\hat{y}(t)$ می‌نامیم. سرعت را به صورت زیر محاسبه می‌کنیم:

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_{1,j} [y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j} [\hat{y}_j(t) - x_{i,j}(t)] \quad (5-2)$$

که در آن $v_{i,j}(t)$ سرعت ذره i در بُعد j در زمان t می‌باشد. ثابت‌های ω ، c_1 و c_2 به ترتیب وزن اینرسی، ضریب آموزش شخصی و ضریب آموزش سراسری هستند که جهت تنظیم مولفه‌های شناختی و اجتماعی مورد استفاده قرار می‌گیرند. بردارهای $r_{1,j}(t) \sim U(0,1)$ و $r_{2,j}(t) \sim U(0,1)$ عوامل تصادفی الگوریتم هستند. با فرض می‌نیم‌سازی، بهترین موقعیت ذره i ام که با \hat{y}_i نشان داده می‌شود، به صورت زیر بدست می‌آید:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) \leq f(y_i(t)) \end{cases} \quad (6-2)$$

که در آن f تابع هدف می‌باشد. شبه کد الگوریتم بهینه‌سازی انبوه ذرات به صورت زیر می‌باشد.

۱. تعیین و مقداردهی اولیه موقعیت و سرعت ذرات
۲. بدست آوردن تابع هزینه برای هر ذره و تعیین بهترین ذره (با کوچکترین مقدار تابع هزینه)
۳. تا زمانی که شرایط توقف برقرار نشده است:
۴. به روز رسانی سرعت و موقعیت ذرات با توجه به موقعیت بهترین ذره
۵. بدست آوردن تابع هزینه تمام ذرات و تعیین بهترین ذره
۶. بروز رسانی بهینه سراسری
۷. پایان الگوریتم

عملکرد الگوریتم بهینه‌سازی انبوه ذرات به طور کامل به وزن اینرسی، ضریب آموزش شخصی و ضریب آموزش سراسری وابسته است [۱۲]. این الگوریتم از دقت و سرعت بالایی در یافتن بهینه‌ترین پاسخ برخوردار است و لذا برای مسائل بهینه‌سازی در فضای چند بعدی بسیار خوب عمل می‌کند [۱۳]. از آن جهت که امکان گرفتار شدن این الگوریتم در بهینه‌های محلی وجود دارد، نیاز به استفاده از یک بخش جستجوی محلی در آن دیده می‌شود.

۲-۳-۴ الگوریتم تکامل تفاضلی

الگوریتم تکامل تفاضلی اولین بار توسط استورن^۱ و پرایس^۲ [۲۸] معرفی شد. در این الگوریتم مقدار هر متغیر با یک عدد صحیح معرفی می‌شود. به دلیل سادگی و عدم نیاز به محاسبات طولانی، این الگوریتم در بسیاری از مسائل بهینه‌سازی عددی مورد استفاده قرار گرفته می‌شود. الگوریتم با مقدار دهی اولیه جمعیتی از افراد شروع کرده، روی رفتار اجتماعی آنها در جامعه کار کرده و با استفاده از اطلاعات فاصله و مسیر افراد بهینه سراسری را می‌یابد. در هر نسل عملیات جهش^۳ و تقاطع^۴ روی افراد جمعیت اعمال شده، جمعیت جدید تولید خواهند شد. در هر مرحله از انتخاب، تمام افراد جمعیت شانس برابری در انتخاب شدن دارند، به عبارتی معیار انتخاب افراد جدید مقدار تابع هزینه نمی‌باشد. در این الگوریتم هر فرد جدید با والدین خود مقایسه می‌شود، و فردی که دارای تابع هزینه کمتری باشد پیروز رقابت خواهد بود. شبه کد الگوریتم تکامل تفاضلی به صورت زیر می‌باشد.

۱. تعیین و مقداردهی اولیه افراد جمعیت

۲. بدست آوردن تابع هزینه برای هر فرد از جمعیت و تعیین بهترین فرد (با بهترین تابع

هزینه)

^۱ Storn

^۲ Price

^۳ Mutation

^۴ Crossover

۳. تا زمانی که شرایط توقف برقرار نشده است:

۴. انجام عملیات جهش بر روی هر فرد از جمعیت و بدست آوردن بردار جهش^۱ آن

۵. انجام عملیات تقاطع بین هر فرد و بردار جهش خود، و بدست آوردن بردار آزمایش^۲ آن

۶. بدست آوردن تابع هزینه بردارهای آزمایش

۷. انجام عملیات انتخاب بین هر فرد و بردار آزمایش خود جهت تولید افراد جدید برای نسل

بعد

۸. بدست آوردن بهترین فرد در جمعیت (با بهترین تابع هزینه) و به روز رسانی تابع هزینه و

بهترین فرد

۹. پایان الگوریتم

موثرترین عوامل در عملکرد الگوریتم تکامل تفاضلی، تعداد جمعیت اولیه، فاکتور سنجش و ضریب تقاطع می‌باشند. هرچه عوامل فوق دقیق‌تر تنظیم شوند، سرعت همگرایی بیشتر خواهد بود [۲۸]. مقادیر کوچک فاکتور سنجش و ضریب تقاطع منجر به افزایش سرعت همگرایی و مقادیر بزرگ آنها منجر به افزایش پراکندگی جمعیت و در نتیجه جلوگیری از همگرایی زودرس خواهد شد. اگرچه این الگوریتم، یکی از سریع‌ترین الگوریتم‌های بهینه‌سازی تکاملی است، اما هیچ تضمینی مبنی بر یافتن پاسخ بهینه و همگرایی به بهینه‌ی سراسری ندارد، و لذا به راحتی در می‌نیم‌های محلی گرفتار می‌شود. این امر درصد اعتماد به الگوریتم برای یافتن پاسخ بهینه را گاهی تا حد صفر پایین می‌آورد [۱۸].

۲-۳-۵ الگوریتم رقابت استعماری

استعمارگری به معنی تحمیل قوانین و قدرت یک کشور به کشور دیگر می‌باشد، که از قرن هجدهم میلادی مبدل به یک رقابت بین کشورهای قدرتمند شده است. کشور قدرتمند را استعمارگر و کشور

^۱ Mutant Vector

^۲ Trial Vector

ضعیف‌تر را مستعمره می‌نامند. الگوریتم رقابت استعماری با الهام از اتفاقات واقعی طول تاریخ برای اولین بار توسط آتشیپز^۱ و لوکاس^۲ [۲۹] ارائه گشت. مشابه دیگر الگوریتم‌های بهینه‌سازی، این الگوریتم نیز با جمعیتی اولیه شروع به کار می‌کند. هر عضو از این جمعیت یک کشور خوانده می‌شود. برخی از بهترین کشورها (با تابع هزینه‌ی بهتر) بعنوان استعمارگر، و دیگر کشورهای باقی مانده بعنوان مستعمره انتخاب خواهند شد. هر کشور استعمارگر، بسته به میزان قدرت خود تعدادی از کشورهای مستعمره را تحت سلطه‌ی خود خواهند آورد. به مجموعه استعمارگر و مستعمره‌هایش، امپراتوری گفته می‌شود. قدرت یک امپراتوری به میزان قدرت کشور استعمارگر و میانگین قدرت کشورهای مستعمره‌ی آن بستگی دارد. زمانی که امپراتوری‌ها شکل گرفتند، کشورهای مستعمره به سمت کشور استعمارگر (مرکز امپراتوری) شروع به حرکت می‌کنند، که این عمل باعث ایجاد بهبود در زمینه‌های اجتماعی- سیاسی آنها می‌شود. سپس میان امپراتوری‌ها جنگ خواهد شد، و هر امپراتوری که نتواند در این جنگ به قدرت خود بیفزاید یا حداقل از کم شدن قدرتش جلوگیری کند، سقوط کرده حذف خواهد شد. این رقابت تا آنجا ادامه می‌یابد که تمام امپراتوری‌های ضعیف حذف شده و در نهایت تنها یک امپراتوری در دنیا باقی بماند [۲۹]. شبه کد الگوریتم رقابت استعماری به صورت زیر می‌باشد.

۱. تعیین جمعیت اولیه کشورها
۲. بدست آوردن تابع هزینه برای تمام کشورها و تعیین بهترین کشور (با بهترین تابع هزینه) بعنوان استعمارگر و دیگر کشورها بعنوان مستعمره
۳. تا زمانی که شرایط توقف برقرار نشده است:
۴. حرکت کشورها به سمت امپراتوری خود و در نتیجه بهبود آنها
۵. در صورتی که در مسیر حرکت، کشوری از تابع هزینه بهتری نسبت به استعمارگر برخوردار شد، به جای آن می‌نشیند

^۱ Atashpaz

^۲ Lucas

۶. محاسبه تابع هزینه کلی تمام امپراتوری‌ها

۷. آغاز جنگ و جداسازی ضعیف‌ترین کشور از ضعیف‌ترین امپراتوری و انتقال آن به قوی‌ترین

امپراتوری که بیشترین احتمال جذب آن را دارد

۸. در صورتی که تمام کشورهای یک امپراتوری جدا شده باشند، آن امپراتوری حذف خواهد

شد

۹. پایان الگوریتم

الگوریتم رقابت استعماری، الگوریتمی با سرعت همگرایی بالا می‌باشد، اما همانند الگوریتم‌های بهینه‌سازی نشات گرفته از طبیعت، امکان گرفتار شدنش در بهینه‌های محلی وجود دارد [۲۹].

همانطور که در بخش‌های گذشته بررسی شد، تقریباً تمام الگوریتم‌های بهینه‌سازی تکاملی با مشکل همگرایی زودرس و گرفتار شدن در نقاط بهینه محلی روبرو هستند. لذا به جهت رفع این مشکل از الگوریتم‌های ممتیک، که از ترکیب الگوریتم‌های تکاملی با یک بخش جستجوی محلی بدست می‌آیند، استفاده می‌شود.

۲-۳-۶ الگوریتم ممتیک (الگوریتم پایه)

در سال‌های اخیر الگوریتمی جدید جهت حل مسائل بهینه‌سازی پیچیده بوجود آمده است به نام الگوریتم ممتیک، که نسبت به دیگر الگوریتم‌های بهینه‌سازی تکاملی پاسخ‌دهی بهتر و دقیق‌تری دارد [۳۰، ۳۱]. منظور از مسائل پیچیده آن دسته از مسائلی هستند که الگوریتم‌های جستجوی سراسری در یافتن پاسخ‌های بهینه آنها به خوبی عمل نمی‌کنند. الگوریتم‌های ممتیک آن دسته از الگوریتم‌های تکاملی هستند که جهت بهبود پاسخ با یک بخش جستجوی محلی ترکیب شده‌اند. الگوریتم‌های ممتیک در حل بسیاری از مسائل بهینه‌سازی از قبیل بهینه‌سازی ترکیبی [۳۲]، بهینه‌سازی توابع متغیر [۳۳] و بهینه‌سازی چند منظوره [۳۴] عملکرد بسیار خوبی از خود نشان داده‌اند. در مطالعات مختلف روش‌های ترکیب الگوریتم‌های تکاملی با جستجوی محلی، نام‌های

گوناگونی را به خود اختصاص داده‌اند که از جمله می‌توان به الگوریتم ژنتیک مرکب^۱، الگوریتم تکاملی بالدوینیان^۲ و الگوریتم تکاملی لمرکیان^۳ اشاره نمود. ماسکتو^۴ [۳۵] از لفظ الگوریتم ممتیک جهت پوشش تمام الگوریتم‌های مبتنی بر جستجوی تکاملی که با یک الگوریتم جستجوی محلی ترکیب شده‌اند، استفاده نمود. در این پایان نامه در بخش (۲-۳-۷) یک الگوریتم ممتیک جدید بر پایه الگوریتم بهینه‌سازی انبوه ذرات ارائه خواهد شد، که در بخش جستجوی محلی آن از جستجوی انبوه ذرات با سرعت مرتبه‌کسری [۳۶] استفاده شده است. لذا در این بخش به بررسی نحوه عملکرد الگوریتم ممتیک اولیه که بر پایه‌ی بهینه‌سازی انبوه ذرات می‌باشد، خواهیم پرداخت. شبه کد الگوریتم ممتیک پایه به صورت زیر می‌باشد.

۱. تعیین و مقداردهی اولیه موقعیت و سرعت ذرات
۲. بدست آوردن تابع هزینه برای هر ذره و تعیین بهترین ذرات (با کوچکترین مقدار تابع هزینه)
۳. انتخاب ذرات برگزیده و اعمال جستجوی محلی روی آنها (بروز رسانی سرعت و در نتیجه موقعیت آنها با استفاده از رابطه (۲-۵))
۴. تا زمانی که شرایط توقف برقرار نشده است:
۵. بررسی تابع هزینه ذرات و انتخاب بهترین ذره (دارای بهترین تابع هزینه)
۶. بروز رسانی سرعت و موقعیت تمام ذرات و بدست آوردن تابع هزینه آنها
۷. انتخاب ذرات برگزیده و اعمال جستجوی محلی روی آنها (بروز رسانی سرعت و در نتیجه موقعیت آنها با استفاده از رابطه‌ی (۲-۹))
۸. بررسی تابع هزینه ذرات و انتخاب بهترین ذره (دارای بهترین تابع هزینه)
۹. پایان الگوریتم

^۱ Hybrid Genetic Algorithm

^۲ Baldwinian Evolutionary Algorithm

^۳ Lamarckian Evolutionary Algorithm

^۴ Moscato

همانطور که در شبه کد بالا مشخص است، در هر مرحله تکرار الگوریتم چند ذره به عنوان بهترین ذرات انتخاب شده و عمل جستجوی محلی و در نتیجه بروز رسانی سرعت و موقعیت آنها انجام می-پذیرد. این کار باعث افزایش دقت الگوریتم در یافتن پاسخ‌های بهینه‌تر و عدم گرفتاری الگوریتم در بهینه‌های محلی می‌شود. اما نکته‌ی قابل تامل اینست که، در هر مرحله جستجوی محلی تنها موقعیت قبلی ذرات اثر داده می‌شود و به عبارتی ذره تمام موقعیت‌های پیشین خود (بجز موقعیت قبلی و بهترین موقعیت خود) را فراموش خواهد کرد. این مساله باعث کاهش دقت جستجوی محلی می‌شود. لذا برای بهبود امر جستجوی محلی، از جستجوی انبوه ذرات با سرعت مرتبه‌کسری استفاده می‌کنیم. با استفاده از این روش، موقعیت‌های پیشین ذرات در بروز رسانی سرعت و موقعیت اثر خواهند داشت. در بخش (۷-۳-۲) الگوریتم پیشنهادی توضیح داده خواهد شد.

۷-۳-۲ الگوریتم ممتیک (الگوریتم پیشنهادی)

در این بخش به بررسی الگوریتم ممتیک پیشنهادی که بر پایه الگوریتم بهینه‌سازی انبوه ذرات و با جستجوی محلی انبوه ذرات با سرعت مرتبه‌کسری می‌باشد، خواهیم پرداخت. استفاده از جستجوی محلی با سرعت مرتبه‌کسری، الگوریتم را قادر می‌سازد تا خطرات بیشتری را حفظ کند، و لذا با دقت بیشتری عملیات جستجو را انجام دهد. در صورتی که رابطه (۵-۲) را به صورت ساده شده زیر بازنویسی کنیم، خواهیم داشت:

$$v_{t+1} = v_t + \varphi_1[b - x] + \varphi_2[g - x] \quad (۶-۲)$$

که در آن x ذره، $\varphi_{1,2}$ ضرایب سرعت، b بهترین موقعیت ذره و g بهترین موقعیت کل ذرات می-باشد. رابطه (۶-۲) را می‌توان به صورت زیر نوشت:

$$v_{t+1} - v_t = \varphi_1[b - x] + \varphi_2[g - x] \quad (۷-۲)$$

طرف سمت چپ تساوی بالا $(v_{t+1} - v_t)$ ، نسخه گسسته مشتق مرتبه‌کسری گرونوالد-لتنیکوف (رابطه (۵-۱)) با مرتبه $n = 1$ با فرض $h = 1$ می‌باشد :

$$\mathbb{D}^n [v_{t+1}] = \varphi_1[b - x] + \varphi_2[g - x] \quad (۸-۲)$$

مرتبه‌ی مشتق را می‌توان $0 \leq n \leq 1$ در نظر گرفت. لذا می‌توان حافظه هر ذره را تا آن میزان که لازم است نگه داشت. فرض کنید حافظه ذره تا ۳ حلقه قبل از موقعیت فعلی مد نظر باشد، بنابراین خواهیم داشت:

$$v_{t+1} - nv_t - \frac{1}{2}nv_{t-1} - \frac{1}{6}n(1-n)v_{t-2} - \frac{1}{24}n(1-n)(2-n)v_{t-3} \quad (۹-۲)$$

$$= \varphi_1[b - x] + \varphi_2[g - x]$$

با مرتب‌کردن رابطه (۹-۲) به فرم رابطه (۶-۲) خواهیم داشت:

$$v_{t+1} = nv_t + \frac{1}{2}nv_{t-1} + \frac{1}{6}n(1-n)v_{t-2} \quad (۱۰-۲)$$

$$+ \frac{1}{24}n(1-n)(2-n)v_{t-3}$$

$$+ \varphi_1[b - x] + \varphi_2[g - x]$$

لذا در هر مرحله از اجرای الگوریتم جستجوی محلی، سرعت ذرات انتخاب شده با رابطه‌ی (۱۰-۲) بروز خواهد شد. در این پایان نامه از ۵۰ ذره به عنوان جمعیت اولیه استفاده شده است، که در هر مرحله ۵ ذره بعنوان ذرات برگزیده (دارای تابع هزینه‌ی بهتر) جهت اعمال جستجوی محلی انتخاب می‌شوند. شبه کد الگوریتم ممتیک پیشنهادی به صورت زیر خواهد بود.

۱. تعیین و مقداردهی اولیه موقعیت و سرعت ذرات
۲. بدست آوردن تابع هزینه برای هر ذره و تعیین بهترین ذرات (با کوچکترین مقدار تابع هزینه)
۳. انتخاب ذرات برگزیده و اعمال جستجوی محلی روی آنها (بروز رسانی سرعت و در نتیجه موقعیت آنها با استفاده از رابطه (۱۰-۲))
۴. تا زمانی که شرایط توقف برقرار نشده است:
۵. بررسی تابع هزینه ذرات و انتخاب بهترین ذره (دارای بهترین تابع هزینه)
۶. به روز رسانی سرعت و موقعیت تمام ذرات و بدست آوردن تابع هزینه آنها

۷. انتخاب ذرات برگزیده و اعمال جستجوی محلی روی آنها (بروز رسانی سرعت و در نتیجه موقعیت آنها با استفاده از رابطه (۲-۱۰))

۸. بررسی تابع هزینه ذرات و انتخاب بهترین ذره (دارای بهترین تابع هزینه)

۹. پایان الگوریتم

همانطور که در فصل چهارم مشاهده خواهد شد، با بکارگیری جستجوی محلی انبوه ذرات با سرعت مرتبه کسری، الگوریتم ممتیک پیشنهادی از گرفتار شدن در بهینه‌های محلی و در نتیجه از همگرایی زودرس جلوگیری می‌کند. لذا کارایی و عملکرد مناسب الگوریتم پیشنهادی به خوبی مشاهده خواهد شد.

فصل سوم:

طراحی کنترل کننده‌های مرتبه کسری

در این فصل به بررسی کاربرد محاسبات مرتبه کسری در کنترل خواهیم پرداخت. در بخش ۳-۲ به بررسی چند روش از روش‌های پیوسته‌زمانی و گسسته‌زمانی تقریب مشتقات مرتبه کسری می‌پردازیم. در بخش ۳-۳ روش‌های متداول طراحی کنترل‌کننده‌های مرتبه کسری عنوان خواهد شد، و در نهایت در زیر بخش ۳-۳-۵ به بررسی روش طراحی کنترل‌کننده‌ی مرتبه کسری بهینه در این پایان نامه خواهیم پرداخت.

۳-۲ روش‌های تقریب سیستم‌های مرتبه کسری

توابع تبدیلی مانند (۱-۳۲) را نمی‌توان به راحتی جهت اهداف عملی و محاسباتی پیاده‌سازی کرد. شبیه‌سازی‌ها معمولاً توسط نرم‌افزارها انجام می‌شوند و کنترل‌کننده‌ها نیز توسط نرم‌افزارهای مشابه در آزمایشگاه‌ها طراحی و پیاده‌سازی می‌شوند. امروزه پیاده‌سازی سخت‌افزاری کنترل‌کننده‌ها با استفاده از المان‌های الکترونیکی، که قابلیت ایجاد توابع تبدیلی با توان‌های صحیح را فراهم می‌سازند، صورت می‌گیرد، اما هنوز به طور کامل نمی‌توان توابع تبدیل مرتبه کسری را با استفاده از المان‌های الکترونیکی ایجاد نمود. به همین دلیل یکی از مهمترین اهداف، یافتن تقریب‌های مرتبه-صحیح از توابع مرتبه کسری می‌باشد. بنابراین در زمان اجرای شبیه‌سازی یا اعمال کنترل‌کننده، تقریبی مرتبه صحیح از تابع مرتبه کسری اصلی با عملکردی تا حد ممکن نزدیک به تابع اصلی بجای آن عمل خواهد کرد [۳۷]. روش‌های بسیاری جهت یافتن تقریب‌های مرتبه صحیح از توابع مرتبه-کسری وجود دارند، اما صراحتاً نمی‌توان گفت که کدام روش از همه بهتر عمل می‌کند. زیرا عملکرد هر روش به شرایط تقریب بستگی دارد، شرایطی از قبیل مرتبه‌ی مشتق‌گیری، حد نهایی انتخاب مرتبه‌ی مشتق‌گیری و اینکه آیا پاسخ دقیق فرکانسی مد نظر است یا پاسخ دقیق زمانی. در این بخش به بررسی چند روش متداول تقریب سیستم‌های مرتبه کسری می‌پردازیم. تقریب‌های مرتبه-

کسری در هر دو حوزه S و Z ارائه شده‌اند. به تقریب در حوزه S تقریب پیوسته و به تقریب در حوزه Z تقریب گسسته می‌گویند.

۱-۲-۳ تقریب کرون^۱

روش کرون تقریبی پیوسته از تابع زیر بر اساس توزیع بازگشتی قطب‌ها و صفرها می‌باشد [۳۸].

$$F(s) = s^v, v \in \mathbb{R} \quad (۱-۳)$$

تابع (۱-۳) ساده‌ترین تابع تبدیل مرتبه‌کسری می‌باشد. این چنین توزیع صفرها و قطب‌ها امکان ایجاد یک تابع تبدیل با بهره و فاز نزدیک به (۱-۳) برای هر مقدار v را فراهم می‌نماید. ساختار تقریب \hat{F} به صورت زیر می‌باشد.

$$\hat{F}(s) = C_0 \prod_{n=1}^N \frac{1 + \frac{s}{\omega_{zn}}}{1 + \frac{s}{\omega_{pn}}} \quad (۲-۳)$$

که دارای N صفر و N قطب می‌باشد. کاملاً مشخص است که صفرها و قطب‌ها تنها در محدوده فرکانسی خاصی وجود خواهند داشت. در نتیجه در فرکانس‌های پایین و بالا شاهد رفتار شبه‌کسری نخواهیم بود. فرض کنید $[\omega_l; \omega_h]$ همان محدوده فرکانسی باشد که از آن صحبت می‌کنیم، بنابراین برای آن که بخواهیم N صفر و قطب را در این محدوده جایگذاری کنیم، فرکانس‌های آنها به صورت زیر خواهند بود.

در صورتی که با صفر آغاز کنیم:

$$\begin{aligned} \omega_{z1} &= \omega_l \sqrt{\eta} \\ \omega_{pn} &= \omega_{zn} \alpha, \quad n = 1, \dots, N \\ \omega_{zn+1} &= \omega_{pn} \eta, \quad n = 1, \dots, N-1 \\ \omega_h &= \omega_{pN} \sqrt{\eta} \end{aligned} \quad (۳-۳)$$

در صورتی که با قطب آغاز کنیم:

^۱ CRONE (Commande Robuste d'Ordre Non-Entier) Approximation

$$\omega_{p1} = \omega_l \sqrt{\eta}$$

$$\omega_{zn} = \omega_{pn} \alpha, \quad n = 1, \dots, N$$

$$\omega_{pn+1} = \omega_{zn} \eta, \quad n = 1, \dots, N-1 \quad (4-3)$$

$$\omega_h = \omega_{zN} \sqrt{\eta}$$

در واقع رابطه‌ی (۳-۳) منجر به یک بهره افزایشی و فاز مثبت می‌شود، که مطابق با $v > 0$ خواهد بود. درحالی‌که رابطه‌ی (۴-۳) منجر به یک بهره کاهش‌ی و فاز منفی مطابق با $v < 0$ می‌شود. از روابط (۳-۳) و (۴-۳) رابطه‌ی زیر حاصل می‌شود:

$$\omega_h = \omega_l (\alpha \eta)^N \Rightarrow \alpha \eta = \left(\frac{\omega_h}{\omega_l} \right)^{1/N} \quad (5-3)$$

با تجزیه رابطه‌ی (۵-۳) به روابط زیر می‌رسیم:

$$\alpha = (\alpha \eta)^{|v|} \Rightarrow \alpha \eta = \left(\frac{\omega_h}{\omega_l} \right)^{|v|/N} \quad (6-3)$$

$$\eta = \omega_l (\alpha \eta)^{1-|v|} \Rightarrow \alpha \eta = \left(\frac{\omega_h}{\omega_l} \right)^{(1-|v|)/N} \quad (7-3)$$

در روابط بالا علامت v به علت اینکه با توجه به استفاده از روابط بازگشتی (۳-۳) و (۴-۳) تغییر می‌کند، در نظر گرفته نشده است. جهت استفاده از این تقریب باید تعداد صفرها، قطب‌ها و محدوده فرکانسی $[\omega_l; \omega_h]$ معین باشد. مقادیر مختلف N (تعداد صفرها و قطب‌ها) نتایج زیر را به همراه خواهد داشت:

- مقادیر کم N منجر به تقریب‌هایی با نوسان (ریپل)^۱ قابل توجه در بهره و فاز خواهد شد.
- مقادیر زیاد N اگرچه منجر به تقریب‌هایی با بهره و فاز تقریباً ثابت می‌شود، اما از نقطه نظر محاسباتی بسیار پیچیده خواهد شد.

۳-۲-۲ تقریب کارلسون^۲

^۱ Ripple

^۲ Carlson Approximation

روش کارلسون تقریبی پیوسته [۳۹] از رابطه‌ی (۳-۱)، با دوباره نویسی آن به صورت تابع (۳-۸) و سپس حل این معادله برای تابع تبدیل $F(s)$ با استفاده از روش نیوتن [۴۰] ارائه می‌دهد. متأسفانه این روش تنها زمانی قابل پیاده‌سازی است که $1/v \in \mathbb{Z}$ باشد، به عبارت دیگر v تنها می‌تواند مقادیر $\dots, \pm \frac{1}{2}, \pm \frac{1}{3}, \pm \frac{1}{4}$ را به خود بگیرد. روش نیوتن دنباله‌ای از تقریب‌های $F_i(s)$, $i \in \mathbb{N}$ به صورت زیر می‌باشد:

$$F_i(s) = F_{i-1}(s) \frac{\left(\frac{1}{v} - 1\right) F_{i-1}^{\frac{1}{v}}(s) + \left(\frac{1}{v} + 1\right) s}{\left(\frac{1}{v} + 1\right) F_{i-1}^{\frac{1}{v}}(s) + \left(\frac{1}{v} - 1\right) s} \quad (۳-۸)$$

که جمله‌ی اول آن به صورت زیر می‌باشد:

$$F_0(s) = 1 \quad (۳-۹)$$

رابطه‌ی (۳-۸) به طور مشخص نشان می‌دهد که $1/v$ باید صحیح باشد، چرا که اگر $1/v$ صحیح نباشد توان‌های مرتبه‌کسری s در تقریب $F_i(s)$ ظاهر می‌شوند. در هر مرحله تکرار مقداری از نوسان بهره و فاز تقریب کاسته می‌شود و محدوده فرکانسی عریض‌تر می‌گردد. مراحل باید تا زمانی که نتایج رضایت بخش باشند تکرار شوند، اما باید به این نکته نیز توجه شود که تعداد صفرها و قطب‌ها به سرعت افزایش می‌یابد.

۳-۲-۳ تقریب ماتسودا^۱

روش ماتسودا تقریبی پیوسته از فرآیندهای مرتبه‌کسری، با شناسایی مدل با استفاده از بهره آن ارائه می‌دهد [۴۱]. فرض کنید می‌خواهیم با استفاده از روش ماتسودا تابع (۳-۱) را تقریب بزنیم. بهره تابع (۳-۱) به صورت زیر می‌باشد:

$$|F(j\omega)| = |j^v \omega^v| = |\omega^v| = \omega^v \quad (۳-۱۰)$$

^۱ Matsuda Approximation

باید مقدار بهره را در فرکانس‌های مختلف بدست آورد. تعداد فرکانس‌ها تعداد صفر و قطب‌ها را مشخص خواهد کرد. برای M صفر و M قطب، باید از $2M + 1$ فرکانس استفاده شود. بهتر اینست که تعداد فرکانس‌های مورد استفاده فرد باشد، در صورتی که تعداد فرکانس‌ها زوج باشد تعداد صفرها برابر خواهد بود با تعداد قطب‌ها به اضافه یک، و بنابراین مدل سره نخواهد بود. فرض کنید فرکانس‌های انتخاب شده $\omega_0, \omega_1, \omega_2, \dots, \omega_n$ باشند. توابع زیر تعریف خواهند شد:

$$\begin{aligned}
 d_0(\omega) &= |F(j\omega)| \\
 d_1(\omega) &= \frac{\omega - \omega_0}{d_0(\omega) - d_0(\omega_0)} \\
 d_2(\omega) &= \frac{\omega - \omega_1}{d_1(\omega) - d_1(\omega_1)} \\
 &\vdots \\
 d_k(\omega) &= \frac{\omega - \omega_{k-1}}{d_{k-1}(\omega) - d_{k-1}(\omega_{k-1})} \\
 &\vdots \\
 d_N(\omega) &= \frac{\omega - \omega_{N-1}}{d_{N-1}(\omega) - d_{N-1}(\omega_{N-1})} \quad (11-3)
 \end{aligned}$$

می‌توان ماتریسی $(n + 1) \times (n + 1)$ به قرار زیر نوشت:

$$D = \begin{bmatrix} d_0(\omega_0) & d_0(\omega_1) & d_0(\omega_2) & \dots & d_0(\omega_N) \\ 0 & d_1(\omega_1) & d_1(\omega_2) & \dots & d_1(\omega_N) \\ 0 & 0 & d_2(\omega_2) & \dots & d_2(\omega_N) \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_N(\omega_N) \end{bmatrix} \quad (12-3)$$

که ضرایب آن به صورت زیر بدست می‌آیند:

$$\alpha_k = D_{kk} = \begin{cases} |F(j\omega_0)|, & \text{if } k = 0 \\ \frac{\omega - \omega_{k-1}}{d_{k-1}(\omega) - d_{k-1}(\omega_{k-1})}, & \text{if } k = 1, 2, \dots, N \end{cases} \quad (13-3)$$

لذا تقریب مطلوب به صورت زیر خواهد بود:

$$\begin{aligned}
 \hat{F}(s) &= \alpha_0 + \frac{s - \omega_0}{\alpha_1 + \frac{s - \omega_1}{\alpha_2 + \frac{s - \omega_2}{\alpha_3 + \dots}}} = \alpha_0 + \frac{s - \omega_0}{\alpha_1 + \frac{s - \omega_1}{\alpha_2 + \frac{s - \omega_2}{\alpha_3 + \dots}}} + \dots \\
 &= \left[\alpha_0; \frac{s - \omega_{k-1}}{\alpha_k} \right]_{k=1}^{+\infty} \quad (14-3)
 \end{aligned}$$

مشابه تقریب کرون، این روش نیز تقریبی است که تنها در محدوده‌ی فرکانسی خاص $[\omega_0; \omega_N]$ کاربرد دارد را ارائه می‌دهد. اما مساله اساسی در این روش انتخاب فرکانس‌ها می‌باشد. برای آنکه تقریب به خوبی عمل کند می‌توان محدوده $[\omega_0; \omega_N]$ و N را ثابت قرار داده و سپس فرکانس‌ها را در این محدوده به صورت لگاریتمی توزیع کرد.

۴-۲-۳ تقریب گرونوالد - لتنیکوف

تقریب گسسته گرونوالد - لتنیکوف [۴۲] از آنجا نشات می‌گیرد که تابع (۱-۳) را با اعمال تبدیل لاپلاس می‌توان به معادله مشتقی مرتبه کسری زیر تبدیل کرد.

$$y(t) = \mathcal{D}^v u(t) \quad (۱۵-۳)$$

اگر در تعریف مشتق مرتبه کسری (۵-۱)، h را به زمان نمونه برداری T تغییر دهیم، خواهیم داشت:

$$\hat{F}(z^{-1}) = \frac{\sum_{k=0}^N (-1)^k \binom{v}{k} z^{-k}}{T^v} = \frac{1}{T^v} \sum_{k=0}^N \frac{(-1)^k \Gamma(v+1)}{\Gamma(k+1) \Gamma(v-k+1)} z^{-k} \quad (۱۶-۳)$$

در رابطه‌ی بالا در صورتی که v عددی غیر منفی باشد، k نیز عددی غیر منفی خواهد بود. حد بالای انتگرال که در اینجا N نامیده شده است، باید با کران‌های انتگرال تغییر کند.

$$N = \frac{t - c}{T} \quad (۱۷-۳)$$

اگر که $v < -1$ باشد، با افزایش k کسر سمت راست رابطه‌ی (۱۶-۳) کاهش می‌یابد. لذا با افزایش N عبارات بیشتری جهت جمع شدن به رابطه اضافه می‌شوند، به عبارت دیگر به حافظه بیشتری احتیاج خواهیم داشت و این امکان وجود دارد که تاثیر لحظات قبل از بین برود.

قضیه ۱-۳ (اصل حافظه کوتاه)^۱:

^۱ Short Memory Principle

در صورتی که تابع F در بازه $[c; N]$ کراندار باشد و مقدار M بگونه‌ای که در رابطه زیر صدق کند وجود داشته باشد:

$$|F(\zeta)| < M, \quad \forall \zeta \in [c; N] \quad (18-3)$$

و در صورتی که تعداد عبارات جمع شونده برابر با L/h باشد و از تقریب مشتق مرتبه‌کسری گرونوالد-لتنیکوف استفاده کنیم، محدوده خطای تقریب به صورت زیر خواهد شد [۴۳]:

$$|\varepsilon| \leq \frac{ML^{-v}}{|\Gamma(1-v)|} \quad (19-3)$$

در رابطه‌ی بالا L طول بازه‌ی مشتق‌گیری (طول حافظه) است. زمانی که $N \gg c$ باشد، تعداد عبارات جمع شونده در رابطه‌ی (۱-۵) خیلی زیاد می‌شوند. بنابراین برای آنکه بخواهیم مقدار خطا از مقدار مشخصی $(|\varepsilon|)$ کوچکتر باشد، طول حافظه از رابطه زیر بدست خواهد آمد:

$$L \geq \sqrt[v]{\frac{M}{|\varepsilon\Gamma(1-v)|}} \quad (20-3)$$

قضیه (۱-۳) نشان می‌دهد که در صورتی که N افزایش یابد، ضرایب نزدیک به مقدار اولیه در تعریف گرونوالد-لتنیکوف تاثیر کمتری در پاسخ خواهند داشت. به عبارت دیگر در هر مرحله $[N-L; N]$ بروز شده و حافظه لحظات قبل حفظ خواهند شد.

۳-۳ روش‌های متداول طراحی کنترل‌کننده‌های مرتبه‌کسری

کنترل‌کننده‌های مرتبه‌کسری کنترل‌کننده‌هایی هستند که از مفاهیم مرتبه‌کسری در آنها استفاده شده است. اینگونه کنترل‌کننده‌ها را می‌توان به سیستم‌های مرتبه‌کسری و مرتبه صحیح اعمال کرد، بنابراین کنترل‌کننده‌های مرتبه‌کسری ممکن است به نحوی عمل کنند که کل سیستم کنترل شده رفتاری مرتبه‌کسری داشته باشد. در این بخش به بررسی چند روش متداول طراحی کنترل‌کننده-

های مرتبه‌کسری خواهیم پرداخت و در زیر بخش آخر روش طراحی کنترل‌کننده‌ی مرتبه‌کسری PID بهینه پیشنهاد شده در این پایان نامه مورد بررسی قرار خواهد گرفت.

۳-۳-۱ نسل اول کنترل‌کننده‌های کرون

اولین نسل از کنترل‌کننده‌های کرون به صورت زیر می‌باشد [۴۴]:

$$C(s) = s^v \quad (۲۱-۳)$$

هدف از استفاده‌ی این کنترل‌کننده افزایش حد فاز به مقدار مطلوب می‌باشد. در طراحی اولیه‌ی آن، رابطه‌ی (۵-۱) توسط روش تقریب کرون تقریب زده شده است، اما از روش‌های تقریب دیگر هم (که در زیر بخش ۳-۱) بررسی شدند می‌توان استفاده کرد. این نوع کنترل‌کننده زمانی که فرآیند تحت کنترل در محدوده‌ی فرکانسی اطراف فرکانس قطع بهره دارای فاز ثابت باشد، سودمند خواهد بود. در آن حالت سیستم حلقه بسته در برابر تغییرات بهره فرآیند مقاوم خواهد بود. لذا حتی اگر فرکانس قطع بهره هم تغییر کند، حد فاز فرآیند و کنترل‌کننده تغییر نخواهد کرد.

۳-۳-۲ نسل دوم کنترل‌کننده‌های کرون

در صورتی که فرآیند دارای فاز ثابت نباشد، نسل اول کنترل‌کننده‌های کرون در مقابل تغییرات بهره فرآیند مقاوم نخواهد بود. بنابراین باید از کنترل‌کننده‌ای استفاده نمود که فاز حلقه باز ثابت را تضمین کند [۴۴]. به عبارت دیگر، اگر کنترل‌کننده C و فرآیند کنترل نشده F باشد خواهیم داشت:

$$\arg[F(j\omega)C(j\omega)] = \arg[F(j\omega)] + \arg[C(j\omega)] = \varphi \quad (۲۲-۳)$$

معادله‌ی فوق تنها در محدوده‌ی فرکانسی زیر برقرار خواهد بود:

$$F(S)C(S) = s^{2\varphi/\pi} \quad (۲۳-۳)$$

نسل دوم کنترل‌کننده‌های کرون از حل معادله‌ی (۳-۲۳) با توجه به C و سپس شناسایی آن از رفتار فرکانسی‌اش بدست می‌آید. در طراحی اولیه‌ی آن از روش شناسایی کرون^۱ استفاده شده است [۴۸]. از رابطه‌ی (۳-۲۲) داریم:

$$\arg[C(j\omega)] = \varphi - \arg[F(j\omega)] \quad (۳-۲۴)$$

لذا روش شناسایی کرون، فرکانس صفرها و قطب‌ها را با هم پیدا می‌کند و در نهایت تعیین می‌کند که کدام صفر و کدام قطب باشد. بنابراین، این امکان وجود دارد که این فرکانس‌ها را بعنوان صفرها و قطب‌ها به نحوی توزیع کنیم که همه پایدار باشند. البته ذکر این نکته حائز اهمیت است که، در صورتی که کنترل‌کننده‌ی گسسته مطلوب باشد این مزیت (توزیع دلخواه فرکانس‌ها و در نتیجه داشتن صفرها و قطب‌های پایدار) وجود نخواهد داشت.

۳-۳-۳ نسل سوم کنترل‌کننده‌های کرون

نسل‌های اول و دوم کنترل‌کننده‌های کرون هدف مشترکی داشتند، و آن مقاوم بودن در مقابل تغییرات بهره فرآیند بود. اما عدم قطعیت‌های دیگر مدل مانند عدم جایابی صفر و قطب^۲ توسط دو نسل قبل پوشش داده نمی‌شد. لذا تلاش نسل سوم کنترل‌کننده‌ی کرون پوشش این عدم قطعیت می‌باشد [۴۷، ۴۵]. برای بررسی نحوه عملکرد نسل سوم کنترل‌کننده‌ی کرون ابتدا باید به دو گروه از منحنی‌های نیکولز توجه کرد.

۳-۳-۱ منحنی‌های بهره حلقه بسته ثابت

اگر پاسخ فرکانسی حلقه باز $G(j\omega)$ برای فرکانسی مشخص را به صورت زیر داشته باشیم:

$$\Psi \stackrel{\text{def}}{=} |G(j\omega)| \quad (۳-۲۵)$$

$$\Theta \stackrel{\text{def}}{=} \arg[G(j\omega)] \quad (۳-۲۶)$$

^۱ روش شناسایی کرون از حوزه بحث این پایان نامه خارج می‌باشد

^۲ Pole and zero misplacement

سیستم حلقه بسته به صورت زیر خواهد بود:

$$L(j\omega) = \frac{G(j\omega)}{1 + G(j\omega)} = \frac{\Psi e^{j\theta}}{1 + \Psi e^{j\theta}} \quad (27-3)$$

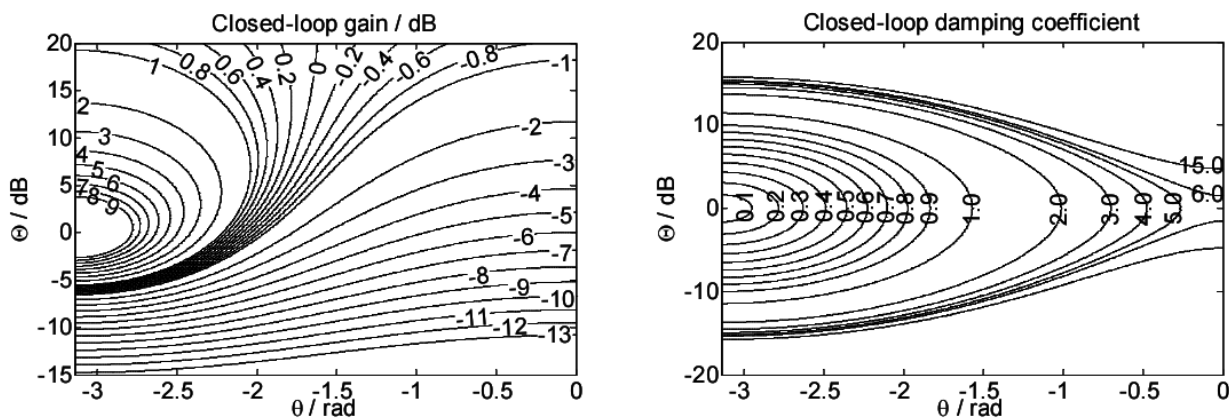
و بنابراین خواهیم داشت:

$$\begin{aligned} |L(j\omega)| &= \left| \frac{\Psi e^{j\theta}}{1 + \Psi e^{j\theta}} \right| = \frac{\Psi}{|1 + \Psi \cos\theta + j\Psi \sin\theta|} \\ &= \frac{\Psi}{\sqrt{(1 + \Psi \cos\theta)^2 + (\Psi \sin\theta)^2}} \\ &= \frac{\Psi}{\sqrt{1 + \Psi^2 \cos^2\theta + 2\Psi \cos\theta + \Psi^2 \sin^2\theta}} \\ &= \frac{\Psi}{\sqrt{1 + \Psi^2 + 2\Psi \cos\theta}} \end{aligned} \quad (28-3)$$

لذا بهره رزونانس^۱ (تشدید) حلقه بسته به صورت زیر خواهد بود.

$$|L(j\omega)| = \max_{\omega} \frac{G(j\omega)}{\sqrt{1 + |G(j\omega)|^2 - 2|G(j\omega)| \cos[\arg|G(j\omega)|]}} \quad (29-3)$$

منحنی های مقادیر ثابت $|L(j\omega)|$ در شکل (۱-۳) در نمودار نیکولز آورده شده است [۴۸، ۴۶].



شکل (۱-۳). نمودار نیکولز با منحنی های بهره و ضرایب میرایی ثابت

۳-۳-۲ منحنی های ضرایب میرایی تقریباً ثابت

ضریب میرایی تقریباً به صورت زیر تعریف می شود [۴۸، ۴۶]:

^۱ Resonance

$$\zeta \approx -\cos \frac{\pi^2}{2 \arccos \left(\frac{1 + \Psi^2 + 2\Psi \cos \theta}{2\Psi} - 1 \right)} \quad (30-3)$$

البته رابطه دقیق به مدل فرآیند بستگی خواهد داشت. منحنی‌های مقادیر ثابت ζ در نمودار نیکولز در شکل (۱-۳) آورده شده است.

۳-۳-۳-۳ هدف کنترل کننده نسل سوم کرون

هدف کنترل کننده‌ی نسل سوم کرون [۴۴]. این است که، اطمینان حاصل کند، بهره حلقه از یک مقدار مشخص بیشتر و یا ضریب میرایی از یک مقدار مشخص کمتر نشود، حتی اگر برخی از پارامترهای فرآیند در محدوده‌ی خاصی تغییر کنند. هر دو هدف فوق به این نتیجه منجر می‌شوند که نمودار نیکولز حلقه باز به نقاطی که بهره حلقه بسته زیاد و ضریب میرایی کم است نزدیک نشود. این نقاط، نقاط اطراف $(0 \text{ dB}; 2n\pi + \pi)$ هستند؛ $n \in \mathbb{Z}$. شکل (۲-۳) با یک مثال چگونگی اعمال این اهداف را نشان می‌دهد [۴۸]. مدل‌های زیر را در نظر بگیرید:

$$F_1(z^{-1}) = \frac{0.50666z^{-4} + 0.28261z^{-3}}{0.88642z^{-4} - 1.31608z^{-3} + 1.58939z^{-2} - 1.41833z^{-1} + 1} \quad (31-3)$$

$$F_2(z^{-1}) = \frac{0.18123z^{-4} + 0.10276z^{-3}}{0.89413z^{-4} - 1.84083z^{-3} + 2.20265z^{-2} - 1.99185z^{-1} + 1} \quad (32-3)$$

$$F_3(z^{-1}) = \frac{0.10407z^{-4} + 0.06408z^{-3}}{0.87129z^{-4} - 1.93353z^{-3} + 2.31962z^{-2} - 2.09679z^{-1} + 1} \quad (33-3)$$

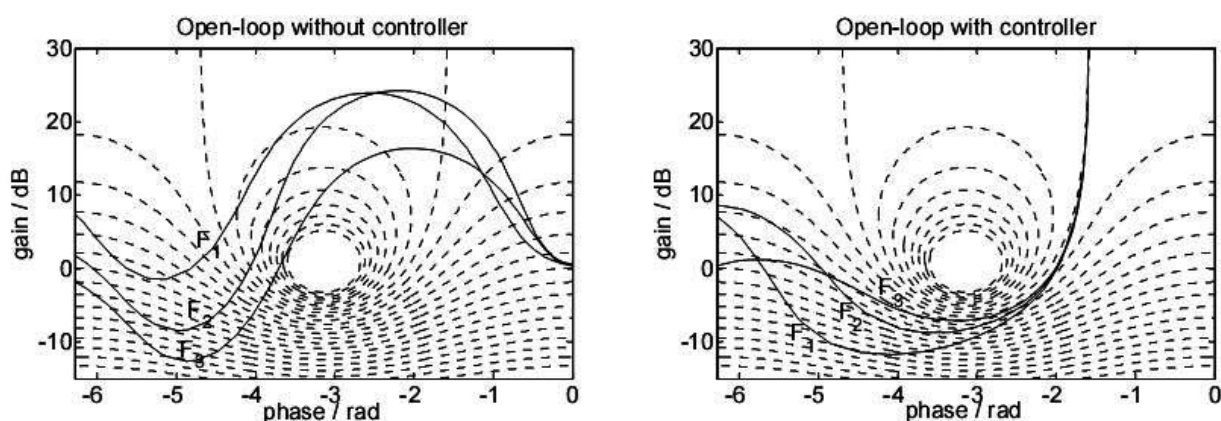
زمان نمونه برداری $T = 0.05$ ثانیه می‌باشد. همانگونه که در شکل (۲-۳) مشاهده می‌شود، نمودار نیکولز این مدل به نقاطی که بهره حلقه بسته زیاد است بسیار نزدیک می‌باشد. بنابراین کنترل-

کننده‌ای مورد نیاز خواهد بود که بتوانیم به‌واسطه آن اطمینان حاصل کنیم که بهره حلقه بسته هیچگاه از 1 dB بیشتر نمی‌شود. بنابراین خواهیم داشت:

$$C(z^{-1}) = \frac{N(z^{-1})}{D(z^{-1})} \quad (34-3)$$

$$\begin{aligned} N(z^{-1}) = & -0.1461419891z^{-7} + 0.6076390925z^{-6} \\ & - 0.5801126047z^{-5} - 0.8034704435z^{-4} \\ & + 1.60382672z^{-3} - 0.2081961813z^{-2} \\ & - 0.8704306516z^{-1} + 0.4105249593 \end{aligned} \quad (35-3)$$

$$\begin{aligned} D(z^{-1}) = & -0.0532773097z^{-7} + 0.1032290834z^{-6} \\ & - 0.5418714841z^{-5} + 0.5887788033z^{-4} \\ & + 1.324106971z^{-3} - 1.68052320z^{-2} \\ & - 0.7349137439z^{-1} + 1 \end{aligned} \quad (36-3)$$



شکل (۳-۲). هدف نسل سوم کنترل کننده کرون

فرض کنید تعداد n_z صفر و n_p قطب داشته باشیم و $S(0)$ ، ω_{zi} و ω_{pi} به ترتیب بهره مدل فضای حالت، صفرها و قطب‌های فرآیند باشند. ممکن است $S(0)$ ، ω_{zi} و ω_{pi} دستخوش تغییراتی محدود میان دو مقدار ماکزیمم و می‌نیمم شوند که به نوبت $\Delta S(0)$ ، $\Delta \omega_{zi}$ و $\Delta \omega_{pi}$ می‌نامیم. همانطور که مشخص است، در صورتی که تغییرات $\Delta S(0)$ ، $\Delta \omega_{zi}$ و $\Delta \omega_{pi}$ به تابع تبدیلی مانند $G(s)$ اعمال شوند، $|L(j\omega)|$ و ζ_{\min} نیز تغییر خواهند کرد. اگر این تغییرات را $\Delta|L(j\omega)|_{S(0)\min}$ و $\Delta|L(j\omega)|_{\omega_{pi}\max}$ ، $\Delta|L(j\omega)|_{\omega_{pi}\min}$ ، $\Delta|L(j\omega)|_{\omega_{zi}\max}$ ، $\Delta|L(j\omega)|_{\omega_{zi}\min}$ ، $\Delta|L(j\omega)|_{S(0)\max}$ و

همچنین $\Delta\zeta_{S(0)max}$, $\Delta\zeta_{S(0)min}$, $\Delta\zeta_{\omega_{zi}max}$, $\Delta\zeta_{\omega_{zi}min}$, $\Delta\zeta_{\omega_{pi}max}$, $\Delta\zeta_{\omega_{pi}min}$ بنامیم، هر دینامیک حلقه باز $G(s)$ متناظر با منحنی فاز متفاوت، تابع هزینه‌ای به فرم زیر خواهد داشت.

$$J(\Delta|L(j\omega)|) = (\Delta|L(j\omega)|_{S(0)min})^2 + (\Delta|L(j\omega)|_{S(0)max})^2 + \sum_{i=1}^{n_z} [(\Delta|L(j\omega)|_{\omega_{zi}min})^2 + (\Delta|L(j\omega)|_{\omega_{zi}max})^2] + \sum_{i=1}^{n_p} [(\Delta|L(j\omega)|_{\omega_{pi}min})^2 + (\Delta|L(j\omega)|_{\omega_{pi}max})^2] \quad (37-3)$$

$$J(\Delta\zeta) = (\Delta\zeta_{S(0)min})^2 + (\Delta\zeta_{S(0)max})^2 + \sum_{i=1}^{n_z} [(\Delta\zeta_{\omega_{zi}min})^2 + (\Delta\zeta_{\omega_{zi}max})^2] + \sum_{i=1}^{n_p} [(\Delta\zeta_{\omega_{pi}min})^2 + (\Delta\zeta_{\omega_{pi}max})^2] \quad (38-3)$$

و در نهایت منحنی متناظر با کمترین مقدار توابع بالا انتخاب خواهد شد. سپس کنترل کننده نهایی همانند نسل دوم کنترل کننده‌های کرون با استفاده از یکی از روش‌های شناسایی از روی رفتار فرکانسی آن بدست خواهد آمد.

۳-۳-۴ کنترل کننده‌های PID مرتبه کسری

کنترل کننده‌های PID کلاسیک (مرتبه صحیح)، کنترل کننده‌هایی هستند که خروجی آنها ترکیبی خطی از ورودی، انتگرال ورودی و مشتق ورودی است. کنترل کننده‌های PID مرتبه کسری تعمیم یافته‌ی PID کلاسیک هستند، بگونه‌ای که خروجی آنها ترکیبی خطی از ورودی، انتگرال مرتبه-کسری ورودی و مشتق مرتبه کسری ورودی می‌باشد [۴۹].

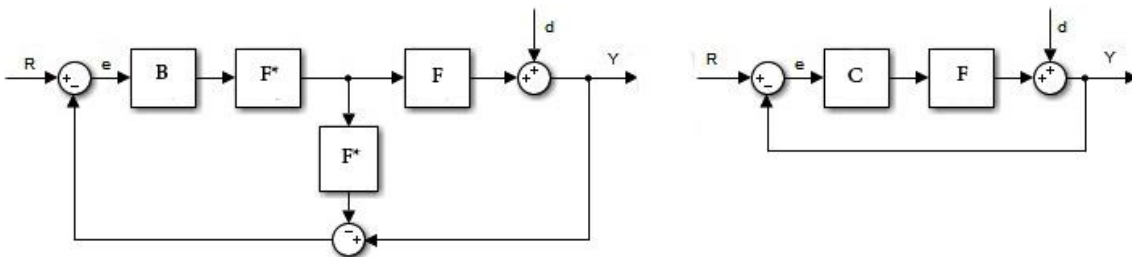
$$C(s) = P + \frac{I}{s^\lambda} + Ds^\mu \quad (39-3)$$

به همین دلیل کنترل کننده‌های PID مرتبه کسری را کنترل کننده‌های $PI^\lambda D^\mu$ یا FOPID می‌نامند. در صورتی که $\lambda = \mu = 1$ باشد، نتیجه همان کنترل کننده PID کلاسیک خواهد بود. لذا گاهی PID

کلاسیک را حالت خاصی از FOPID می‌نامند. در ادامه سه روش متداول طراحی کنترل‌کننده‌های FOPID بررسی خواهد شد.

۳-۳-۱ طراحی FOPID با کنترل مدل داخلی^۱

از روش کنترل مدل داخلی جهت طراحی کنترل‌کننده‌های PID یا FOPID استفاده می‌شود. این روش از مدل کنترلی شکل (۳-۳) چپ استفاده می‌کند [۵۰]، که در آن فرآیند تحت کنترل، F^* معکوس F و یا حداقل مدلی تا حد ممکن نزدیک به معکوس F ، F' مدلی از F و B فیلتر (در صورت لزوم) می‌باشند.



شکل (۳-۳). بلوک دیاگرام کنترل مدل داخلی (چپ) و معادل آن (راست)

در صورتی که F' دقیق باشد، خطای e برابر با اغتشاش d خواهد بود. همچنین در صورتی که F^* دقیقاً معکوس F و B نیز واحد باشد، عمل کنترل به خوبی صورت خواهد پذیرفت. اما از آنجا که هیچ یک از مدل‌ها دقیق نیستند، خطا دقیقاً معادل اغتشاش نخواهد بود. به همین دلیل است که معمولاً یک فیلتر پایین‌گذر (B) جهت کاهش اثرات خطاهای مدل‌سازی در فرکانس‌های بالا اضافه می‌کنند. مزیت دیگر وجود این فیلتر اینست که باعث می‌شود BF^* تحقق پذیر باشد. شکل (۳-۳) چپ معادل شکل (۳-۳) راست خواهد بود، اگر:

$$C(s) = \frac{B(s)F^*(s)}{1 - B(s)F^*(s)F'(s)} \quad (۴۰-۳)$$

^۱ Internal Model Control (IMC)

کنترل کننده‌ی $C(s)$ در حالت عمومی یک کنترل کننده PID یا FOPID نیست، اما اگر داشته باشیم:

$$F(s) = \frac{K}{1 + Ts^\mu} e^{-Ls} \quad (41-3)$$

و $F^*(s)$ ، $F'(s)$ و $B(s)$ به صورت زیر تعریف شوند:

$$F^*(s) = \frac{1 + Ts^\mu}{K} \quad (42-3)$$

$$F'(s) = \frac{K}{1 + Ts^\mu} (1 - Ls) \quad (43-3)$$

$$B(s) = \frac{1}{1 + T_B s} \quad (44-3)$$

لذا کنترل کننده‌ی (40-3) به صورت زیر خواهد بود:

$$C(s) = \frac{K}{s} + \frac{T}{s^{1-\mu}} \quad (45-3)$$

همانطور که مشاهده می‌شود، کنترل کننده فوق یک کنترل کننده FOPID با ترم تناسبی برابر صفر می‌باشد. در صورتی که مدل F' در رابطه‌ی (43-3) با یک تقریب بهبود یافته به فرم زیر جایجا گردد:

$$F'(s) = \frac{K}{1 + Ts^\mu} \frac{1 - \frac{Ls}{2}}{1 + \frac{Ls}{2}} \quad (46-3)$$

کنترل کننده‌ی (40-3) به صورت زیر خواهد بود:

$$C(s) = \frac{L}{2K(T+L)} + \frac{1}{K(T_B+L)s} + \frac{T}{K(T_B+L)s^{1-\mu}} + \frac{TL}{2K(T+L)} s^\mu \quad (47-3)$$

اگر یکی از دو قسمت انتگرالی قابل چشم پوشی باشند، کنترل کننده‌ی (47-3) یک کنترل کننده‌ی FOPID خواهد بود. مشخصاً در صورتی که $\mu \in \mathbb{Z}$ باشد، هر دو کنترل کننده‌ی (45-3) و (47-3) به PID کلاسیک تبدیل خواهند شد.

۳-۴-۲ طراحی FOPID با می‌نیمم‌سازی^۱

یکی از روش‌های طراحی کنترل‌کننده‌های FOPID یافتن پارامترهاست به طوری که شرایط زیر را ارضاء کنند [۵۱،۵۰]. در اینجا C کنترل‌کننده و G مدل فرآیند می‌باشد.

- فرکانس قطع بهره (ω_{cg}) را طوری می‌یابیم که:

$$|C(\omega_{cg})F(\omega_{cg})| = 0 \text{ dB} \quad (۴۸-۳)$$

- حد فاز (φ_m) را طوری می‌یابیم که:

$$-\pi + \varphi_m = \arg[C(\omega_{cg})F(\omega_{cg})] \quad (۴۹-۳)$$

- به منظور حذف نویز فرکانس بالا، تابع تبدیل حلقه بسته باید در فرکانس‌های بالا دامنه کمی داشته باشد. لذا در فرکانس‌های مشخصی (ω_h) دامنه باید کمتر از مقدار بهره مشخصی (H) باشد.

$$\left| \frac{C(\omega_h)F(\omega_h)}{1 + C(\omega_h)F(\omega_h)} \right| < H \quad (۵۰-۳)$$

- به منظور حذف اغتشاش خروجی و دنبال کردن مرجع، تابع حساسیت^۲ باید در فرکانس‌های پایین دامنه کمی داشته باشد. لذا در فرکانس‌های مشخصی (ω_l) دامنه باید کمتر از مقدار بهره مشخصی (N) باشد.

$$\left| \frac{1}{1 + C(\omega_l)F(\omega_l)} \right| < N \quad (۵۱-۳)$$

- به منظور مقاوم بودن تابع تبدیل حلقه بسته در مقابل تغییرات بهره فرآیند، فاز تابع تبدیل حلقه باز باید (تقریباً) در حوالی فرکانس قطع بهره ثابت باشد.

^۱ Minimization

^۲ Sensitivity Function

$$\frac{d}{d\omega} \arg[C(\omega)F(\omega)] = 0, \quad \omega = \omega_{cg} \quad (52-3)$$

از آنجا که پنج پارامتر جهت طراحی وجود دارند، پنج شرط طراحی داریم [53,52]. برای ارضای تمام شرایط فوق می‌توان از الگوریتم‌های بهینه‌سازی عددی استفاده نمود. یکی از این الگوریتم‌ها، الگوریتم سیمپلکس نلدر-مید^۱ می‌باشد [50]. علیرغم اینکه این الگوریتم در می‌نیم‌های محلی گرفتار می‌شود، در عمل الگوریتمی موثر بوده و پاسخ‌های نسبتاً مناسبی می‌دهد. البته پاسخ این الگوریتم کاملاً به تخمین اولیه پارامترها وابسته است.

۳-۳-۴-۳ طراحی FOPID با استفاده از قواعد تنظیمی^۲

امروزه از کنترل‌کننده‌های PID کلاسیک به دلیل سادگی، موثر بودن و فراوانی روش‌های طراحی استفاده‌های بسیاری می‌شود. تمام آن چیزی که روش‌های طراحی احتیاج دارند، پاسخ زمانی مشخص سیستم می‌باشد. از جمله‌ی این روش‌ها می‌توان به زیگلر-نیکولز^۳ [55]، کوهن-کون^۴ [56] و کاپا-تاو^۵ [57] اشاره نمود. اگرچه کنترل‌کننده‌هایی که با استفاده از این روش‌ها بدست می‌آیند بهینه نیستند، اما با طراحی مناسب می‌توان پاسخ‌های مطلوبی بدست آورد. در ادامه دو روش طراحی بررسی خواهد شد که به روش طراحی کنترل‌کننده‌ی PID کلاسیک زیگلر-نیکولز شباهت بسیاری دارد. همانند روش زیگلر-نیکولز فرآیند به‌صورتی در نظر گرفته می‌شود که پاسخ پله واحد آن S-شکل^۶ باشد و در صورتی که فرآیند چنین نباشد این روش‌ها کارایی مناسبی نخواهند داشت. ساده‌ترین فرآیند با چنین پاسخی به‌صورت زیر خواهد بود:

$$F(s) = \frac{K}{1 + sT} e^{-Ls} \quad (53-3)$$

^۱ Nelder-Mead Simplex Algorithm

^۲ Tuning Rules

^۳ Ziegler-Nichols

^۴ Cohen-Coon

^۵ Kappa-Tau

^۶ S-shaped

قواعد تنظیمی که در ادامه بحث خواهند شد، از اعمال روش می‌نیم‌سازی (۳-۳-۴-۲) به فرآیند (۳-۳-۵) به ازای مقادیر مختلف L, T و $K = 1$ ، و با اعمال روش کمترین مربعات^۱ جهت یافتن پارامترهای کنترل‌کننده‌ی FOPID بدست آمده‌اند.

۳-۳-۴-۱ گروه اول قواعد تنظیمی

گروه اول قواعد تنظیمی در جدول (۳-۱) آورده شده‌اند [۵۴]، و به صورت زیر خوانده می‌شود:

$$P = -0.0048 + 0.2664L + 0.4982T + 0.0232L^2 - 0.0720T^2 - 0.0348TL \quad (۳-۵۴)$$

و به همین ترتیب برای دیگر پارامترها به کار خواهد رفت. از این گروه قواعد تحت شرایط زیر استفاده می‌شود:

$$0.1 \leq T \leq 50 \wedge L \leq 2 \quad (۳-۵۵)$$

و زمانی که هدف از طراحی حصول مقادیر زیر باشد:

$$\omega_{cg} = 0.5 \text{ rad/s} \quad (۳-۵۶)$$

$$\varphi_m = \frac{2}{3} \text{ rad} \approx 38^\circ \quad (۳-۵۷)$$

$$\omega_h = 10 \text{ rad/s} \quad (۳-۵۸)$$

$$\omega_l = 0.1 \text{ rad/s} \quad (۳-۵۹)$$

$$H = -10 \text{ dB} \quad (۳-۶۰)$$

$$N = -20 \text{ dB} \quad (۳-۶۱)$$

ذکر این نکته حائز اهمیت است که، مقادیر فوق تقریبی خواهند بود.

جدول (۳-۱). پارامترها برای گروه اول قواعد تنظیمی

^۱ Least-Squares

	<i>Parameters to Use when $0.1 \leq T \leq 5$</i>					<i>Parameters to Use when $5 \leq T \leq 50$</i>				
	<i>P</i>	<i>I</i>	λ	<i>D</i>	μ	<i>P</i>	<i>I</i>	λ	<i>D</i>	μ
1	-0.0048	0.3254	1.5766	0.0662	0.8736	2.1187	-0.5201	1.0645	1.1421	1.2902
<i>L</i>	0.02664	0.2478	-0.2098	-0.2528	0.2746	-3.5207	2.6643	-0.3268	-1.3707	-0.5371
<i>T</i>	0.4989	0.1429	-0.1313	0.1081	0.1489	-0.1563	0.3453	-0.0229	0.0357	-0.0381
<i>L</i> ²	0.0232	-0.1330	0.0713	0.0702	-0.1557	1.5827	-1.0944	0.2018	0.5552	0.2208
<i>T</i> ²	-0.0720	0.0258	0.0016	0.0328	-0.0250	0.0025	0.0002	0.0003	-0.0002	0.0007
<i>LT</i>	-0.0348	-0.0171	0.0114	0.2202	-0.0323	0.1824	-0.1054	0.0028	0.2630	-0.0014

۲-۳-۴-۳-۳ گروه دوم قواعد تنظیمی

گروه دوم قواعد تنظیمی در جدول (۲-۳) آورده شده‌اند [۵۴]. از این گروه قواعد، تحت شرایط

$$0.1 \leq T \leq 50 \wedge L \leq 0.5 \quad (۶۲-۳)$$

و زمانی که هدف از طراحی، دستیابی به حصول مقادیر (تقریبی)

$$\omega_{cg} = 0.5 \text{ rad/s} \quad (۶۳-۳)$$

$$\varphi_m = 1 \text{ rad} \approx 57^\circ \quad (۶۴-۳)$$

$$\omega_h = 10 \text{ rad/s} \quad (۶۵-۳)$$

$$\omega_l = 0.01 \text{ rad/s} \quad (۶۶-۳)$$

$$H = -20 \text{ dB} \quad (۶۷-۳)$$

$$N = -20 \text{ dB} \quad (۶۸-۳)$$

باشد، استفاده می‌کنیم.

جدول (۲-۳). پارامترها برای گروه دوم قواعد تنظیمی

	<i>P</i>	<i>I</i>	λ	<i>D</i>	μ
1	-1.0574	0.6014	1.1851	0.8793	0.2778
<i>L</i>	24.5420	0.4025	-0.3464	-15.0846	-2.1522
<i>T</i>	0.3544	0.7921	-0.0492	-0.0771	0.0675
<i>L</i> ²	-46.7325	-0.4508	1.7317	28.0388	2.4387

T^2	-0.0021	0.0018	0.0006	0.0001	-0.0013
LT	-0.3106	-1.2050	0.0380	1.6711	0.0021

۳-۳-۵ روش طراحی کنترل کننده FOPID بهینه در این پایان نامه

همانطور که در بخش‌های گذشته اشاره شد، روش‌های طراحی کنترل کننده‌های مرتبه‌کسری معایب و کاستی‌هایی دارند. به عنوان مثال هدف کنترل کننده‌های کرون تنها غلبه بر عدم قطعیت‌ها می‌باشد و کنترل کننده‌های طراحی شده از این روش‌ها بهینه نخواهند بود. همچنین کنترل کننده‌های FOPID طراحی شده از روش می‌نیم‌سازی اگرچه بهینه هستند، اما به علت گرفتاری الگوریتم نلدر-مید در بهینه‌های محلی همیشه قابل اعتماد نیستند، و یا کنترل کننده‌های FOPID طراحی شده با استفاده از قواعد تنظیمی تنها برای گروه خاصی از فرآیندها قابل اجرا هستند و جهت استفاده از این روش باید فرآیند به صورتی در نظر گرفته شود که پاسخ پله واحد آن S -شکل باشد، و در صورتی که فرآیند چنین نباشد این روش‌ها کارایی مناسبی نخواهند داشت. لذا بنا به دلایل ذکر شده نیاز به روشی قابل اطمینان و کارآمد جهت طراحی کنترل کننده‌های FOPID برای انواع فرآیندهای مرتبه صحیح، مرتبه‌کسری و تاخیردار احساس می‌شود. روش طراحی کنترل کننده FOPID پیشنهادی در این پایان‌نامه، بر اساس روش طراحی مکان هندسی ریشه‌ها [۴۶] (روش ریشه‌های غالب) برای PID کلاسیک عمل می‌کند. همانند روش مکان هندسی کلاسیک، بر اساس ماکزیمم فراجش^۱ (M_P) و زمان صعود^۲ (T_r) مورد نظر، ضریب میرایی (ζ) و فرکانس طبیعی میرا نشده (ω_n) سیستم حلقه بسته طراحی خواهند شد. قطب‌های غالب به صورت زیر خواهند بود:

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -x \pm jy \quad (۶۹-۳)$$

در صورتی که تابع تبدیل حلقه بسته را به صورت زیر در نظر بگیریم:

^۱ Overshoot

^۲ Rise Time

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}, \quad G(s) = G_c(s)G_p(s) \quad (70-3)$$

که در آن $G_c(s)$ تابع تبدیل کنترل کننده و $G_p(s)$ تابع تبدیل فرآیند تحت کنترل و $H(s) = 1$ گین مسیر فیدبک می باشد. از معادله (70-3) معادله مشخصه سیستم به صورت زیر بدست خواهد آمد.

$$1 + G(s)H(s) = 0 \Rightarrow 1 + G_c(s)G_p(s).1 = 0 \quad (71-3)$$

قطب های غالب سیستم حلقه بسته، صفرهای معادله مشخصه (71-3) می باشند. لذا از معادله فوق خواهیم داشت:

$$1 + [K_p + K_i(-x + jy)^{-\lambda} + K_d(-x + jy)^\mu]G_p(-x + jy) = 0 \quad (72-3)$$

معادله (72-3) دارای پنج پارامتر مجهول $K_p, K_i, \lambda, K_d, \mu$ می باشد. در صورتی که R قسمت حقیقی، I قسمت موهومی و $\phi = \tan^{-1} \frac{I}{R}$ فاز معادله مشخصه (72-3) باشد، تابع هدف جهت طراحی کنترل کننده به صورت زیر خواهد بود.

$$J(K_p, K_i, \lambda, K_d, \mu) = |I|^2 + |R|^2 + |\phi|^2 \quad (73-3)$$

بنابراین هدف یافتن بهینه ترین مقدار پارامترهای مجهول $K_p, K_i, \lambda, K_d, \mu$ به گونه ایست که J می-نیم شود. در میان روش های بهینه سازی، الگوریتم های تکاملی به علت سرعت و دقت قابل قبول، جایگاه ویژه ای را به خود اختصاص داده اند. از آنجا که این روش طراحی بر اساس بهینه سازی عمل می کند، لذا در این پایان نامه از الگوریتم های بهینه سازی تکاملی معرفی شده در فصل دوم جهت یافتن مقدار بهینه پارامترهای کنترل کننده FOPID استفاده شده است.

فصل چهارم :

طراحی کنترل کننده بهینه PID مرتبه کسری با استفاده از

الگوریتم‌های تکاملی

در این فصل ابتدا در بخش ۲-۴ مقایسه‌ای میان عملکرد الگوریتم ممیتیک پیشنهادی و الگوریتم ممیتیک پایه در بهینه‌سازی چند تابع ریاضی خواهیم داشت. سپس عملکرد کنترل‌کننده‌های PID کلاسیک، PID مرتبه‌کسری (طراحی شده با روش می‌نیم‌سازی) و PID مرتبه‌کسری بهینه پیشنهادی (بهینه شده با الگوریتم ممیتیک پیشنهادی) در کنترل فرآیند مرتبه‌کسری در بخش ۳-۴ مقایسه خواهد شد. برای نشان دادن عملکرد کنترل‌کننده‌های PID مرتبه‌کسری در کنترل فرآیندهای گوناگون، آنها را بر روی سه فرآیند مرتبه‌کسری، مرتبه‌صحیح و تاخیردار اعمال خواهیم کرد. در بخش (۴-۴) به مقایسه عملکرد الگوریتم‌های تکاملی بررسی شده در فصل دوم در طراحی کنترل‌کننده‌های PID مرتبه‌کسری بهینه برای فرآیندهای مرتبه‌کسری، مرتبه صحیح و تاخیردار خواهیم پرداخت. تمام شبیه‌سازی‌های این فصل توسط نرم‌افزار متلب^۱ 64- R2013a (8.1.0.604) (win64) bit و با استفاده از رایانه‌ای با مشخصات زیر انجام گرفته است.

ASUS Laptop

Processor: Intel® Core™ i5 - 2450M CPU 2.50GHz

Installed Memory (RAM): 4.00 GB (3.78 usable)

System Type: 64-bit win7 Operating System

VGA: GeForce NVidia 610M 2G

۲-۴ مقایسه عملکرد الگوریتم‌های ممیتیک پایه و ممیتیک پیشنهادی

جهت بررسی عملکرد الگوریتم ممیتیک پیشنهادی، و مقایسه‌ی آن با الگوریتم ممیتیک پایه، الگوریتم‌ها را بر روی چند تابع معیار^۲ [۵۹] ریاضی اعمال کرده‌ایم. توابع معیار به ترتیب روزنبروک^۳،

^۱ MATLAB

^۲ Benchmark Function

^۳ Rosenbrock

راستریجین^۱، هیمبلبلو^۲، میشلویز^۳، گریوانک^۴ و اکلی^۵ می‌باشند. جدول (۱-۴) توابع ریاضی، بُعد توابع و محدوده‌ی فضای جستجوی آنها را نشان می‌دهد.

جدول (۱-۴). توابع معیار استفاده شده در مقایسه الگوریتم‌های ممتیک پایه و پیشنهادی

Fun	Name of Function	Mathematical Representation	Dimension (D)	Range Search of
I	Rosenbrock	$F_1(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	100	$(-30, 30)^D$
II	Rastrigin	$F_2(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$(-5.12, 5.12)^D$
III	Himmelblau	$F_3(x) = \frac{1}{n} \sum_{i=1}^D [x_i^4 - 1x_i^2 + 5x_i]$	100	$(-5, 5)^D$
IV	Michalewicz	$F_4(x) = - \sum_{i=1}^D [\sin(x_i) \sin^{20}(\frac{i \times x_i}{\pi})]$	100	$(0, \pi)^D$
V	Griewank	$F_5(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$(-600, 600)^D$
VI	Ackley	$F_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + \exp(1)$	30	$(-32, 32)^D$

جهت مقایسه‌ی عادلانه‌ی دو الگوریتم ممتیک پایه و پیشنهادی، پارامترهای هر دو الگوریتم را مشابه یکدیگر و بصورت زیر تعریف می‌کنیم.

- تعداد کل ذرات برابر با ۵۰
- وزن اینرسی برابر با ۰.۲
- ضریب آموزش شخصی و ضریب آموزش سراسری برابر با یکدیگر و برابر با ۱/۴۹۴
- تعداد تکرار الگوریتم در هر اجرا برابر با ۲۰۰
- تعداد کل اجرای هر الگوریتم برابر با ۵۰۰

^۱Rastrigin

^۲ Himmelblau

^۳ Michalewicz

^۴ Griewank

^۵ Ackley

مرتبه‌ی مشتق مرتبه‌کسری باید $0 < n < 1$ باشد، لذا $n = 0.5$ فرض می‌شود. پس از اجرای الگوریتم‌ها با شرایط فوق، نتایج به‌صورت جدول (۲-۴) بدست آمده‌اند. جهت مقایسه بهتر و دقیق‌تر الگوریتم‌ها، هر دو الگوریتم با ابعاد حداقل و حداکثر مورد بررسی قرار گرفته‌اند. جدول (۲-۴) شامل قسمت‌های زیر می‌باشد:

۱. مقادیر بهترین، بدترین و میانگین توابع هزینه‌ای که توسط الگوریتم‌ها بدست آمده است.

۲. مقدار انحراف معیار^۱ که از رابطه زیر محاسبه می‌شود، و n تعداد پارامترها می‌باشد.

$$S.D. = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}}, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (۱-۴)$$

۳. تعداد اجرای تابع^۲ که نشان دهنده تعداد تکرار تابع هزینه در طول یافتن پاسخ بهینه است.

جدول (۲-۴). نتایج بدست آمده از بهینه‌سازی توابع معیار توسط الگوریتم‌های ممتیک پیشنهادی و پایه

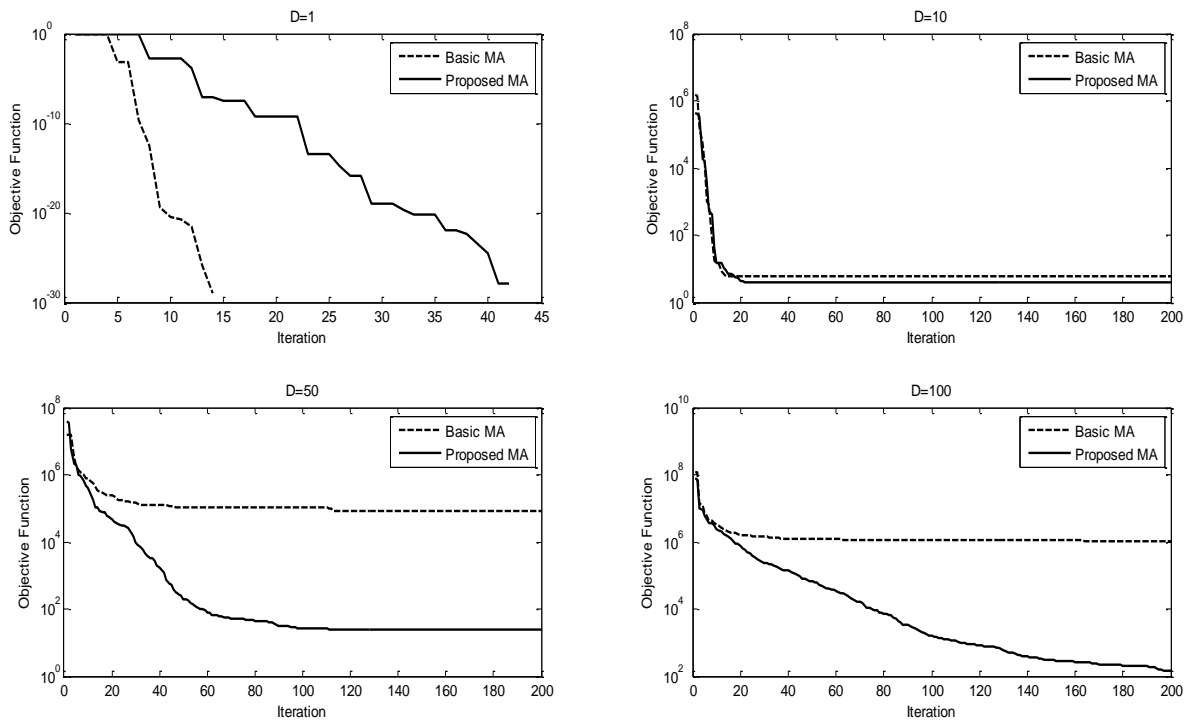
Benchmark Function		Dimension = 1		Dimension = Max	
		Proposed MA	Basic MA	Proposed MA	Basic MA
Rosenbrock	Best	0	0	38.3031	3.4008e+05
	Mean	0	0	84.6321	1.1574e+06
	Worst	0	0	239.0562	3.2006e+06
	S.D.	2.7113e-10	5.8542e-14	2.2940e+06	2.9861
	N.F.E.	512	1855	9434	14885
Rastrigin	Best	0	0	14.9244	18.8561
	Mean	0	0.0020	50.1677	58.6350
	Worst	0	0.9950	118.3997	147.1025
	S.D.	6.3906e-14	6.9925e-13	12.8950	13.1818
	N.F.E.	822	875	2744	3749
Himmelblau	Best	-39.1662	-39.1662	-3.4606e+03	-3.2154e+03
	Mean	-39.1662	-39.1662	-3.2704e+03	-2.7904e+03
	Worst	-39.1662	-39.1662	-2.7640e+03	-1.8982e+03
	S.D.	6.2879e-14	6.3869e-14	242.4230	4.0680
	N.F.E.	945	1052	8522	10360
Michalewicz	Best	-0.8013	-0.999	-77.7005	-64.4674
	Mean	-0.8013	-0.9957	-64.6206	-55.2547
	Worst	-0.8013	-0.9448	-52.8410	-45.5260
	S.D.	1.1683e-14	4.4983e-11	7.7491	5.0854
	N.F.E.	674	974	8907	11520
Griewank	Best	0	0	5.6645e-11	1.1290
	Mean	6.1161e-04	0.0016	0.0052	2.5149
	Worst	0.0099	0.0395	0.0691	6.5389
	S.D.	1.5413e-10	2.0854e-11	4.6904	4.0524
	N.F.E.	756	982	10640	14682
	Best	8.8818e-16	8.8818e-16	1.6462	4.6867
	Mean	8.8818e-16	8.8818e-16	4.9411	8.1550

^۱ Standard Deviation (S.D.)

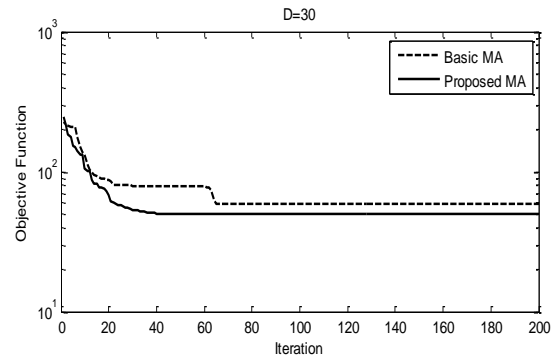
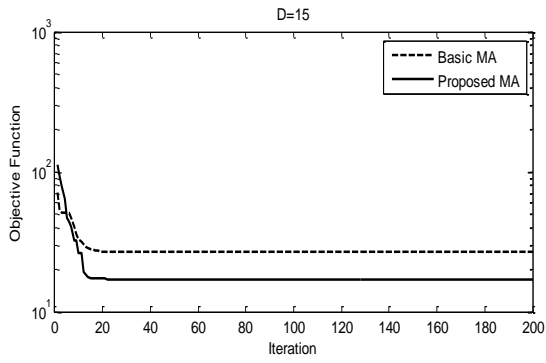
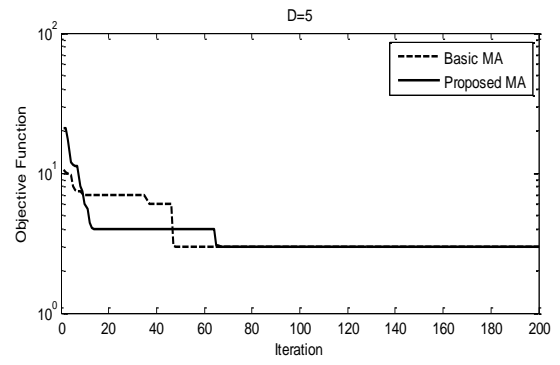
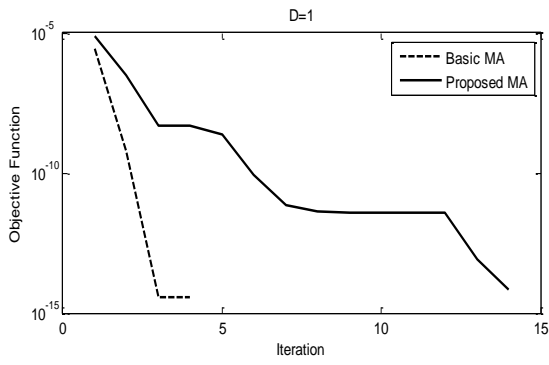
^۲ Number of Function Evaluation

Ackley	Worst	8.8818e-16	8.8818e-16	8.9518	11.1968
	S.D.	1.1632e-07	6.2770e-08	0.9440	0.5573
	N.F.E.	675	2475	6975	8511

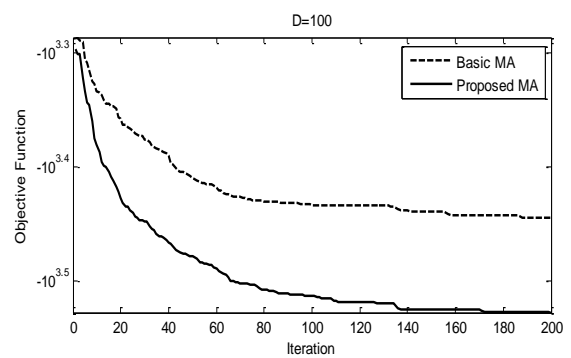
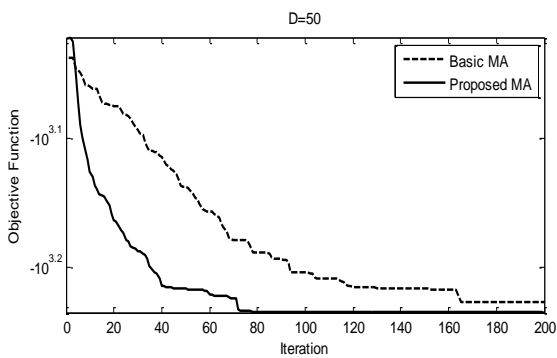
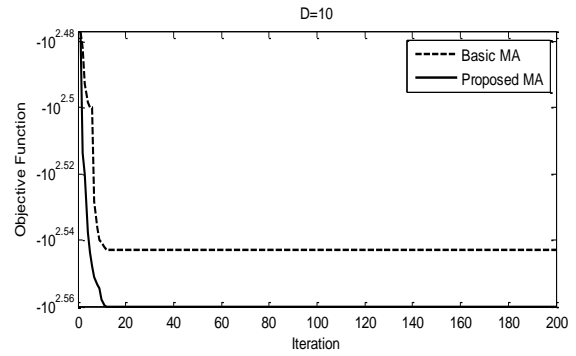
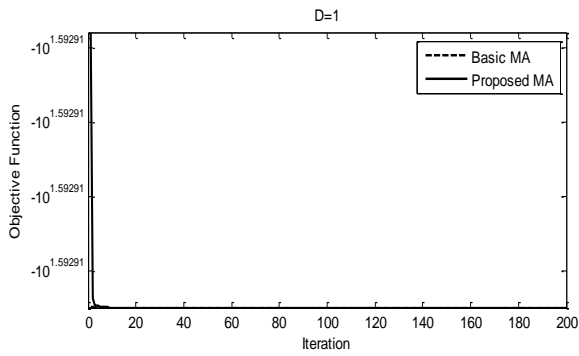
همانطور که در جدول (۴-۲) مشاهده می‌شود، الگوریتم ممتیک پیشنهادی در یافتن مقدار بهینه‌ی توابع معیار با بُعد یک، عملکردی برابر با الگوریتم ممتیک پایه از خود نشان داده است. اما با زیاد شدن ابعاد توابع معیار برتری الگوریتم پیشنهادی در یافتن مقادیر بهینه کاملاً مشهود خواهد بود. مشاهده می‌شود که الگوریتم ممتیک پیشنهادی سریع‌تر از الگوریتم ممتیک پایه پاسخ بهینه را می‌یابد، و در اکثر موارد میزان انحراف معیار الگوریتم پایه کمتر از الگوریتم پیشنهادی می‌باشد. در این بخش جهت مقایسه کامل‌تر دو الگوریتم، روند یافتن مقادیر بهینه‌ی توابع معیار را با ابعاد مختلف در شکل‌های (۴-۱) تا (۴-۶) بررسی خواهیم کرد.



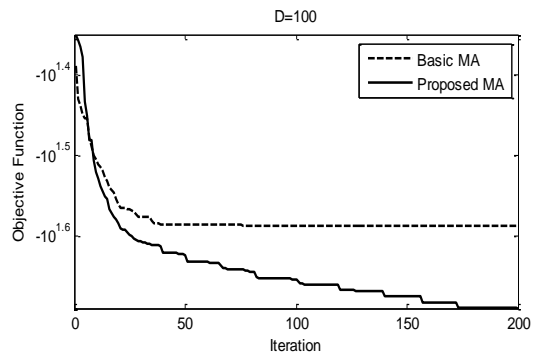
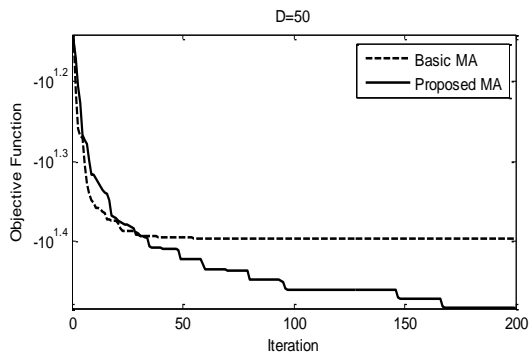
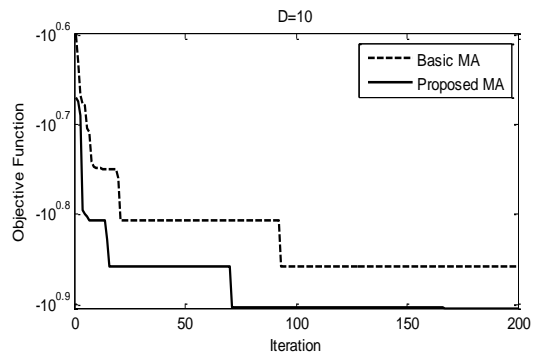
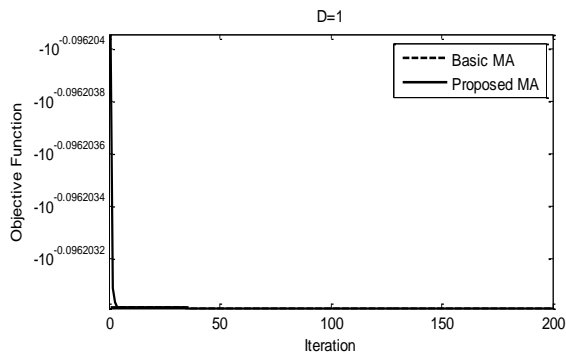
شکل (۴-۱). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار I



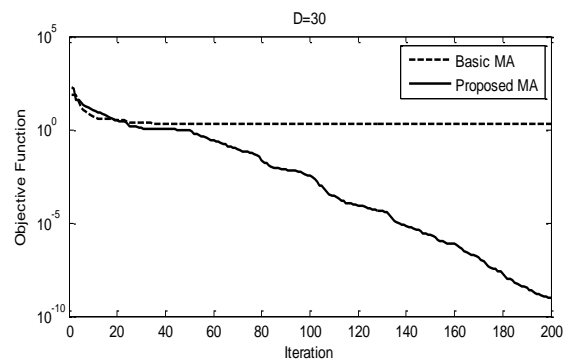
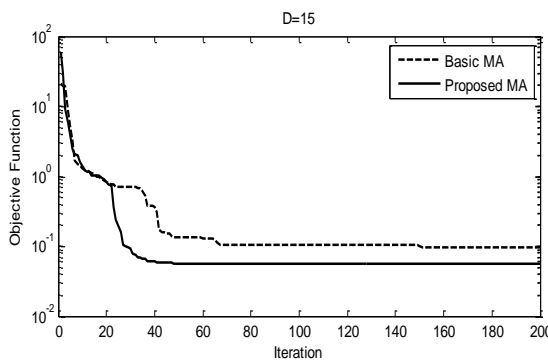
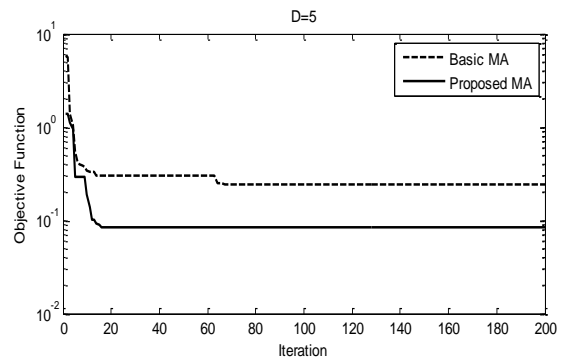
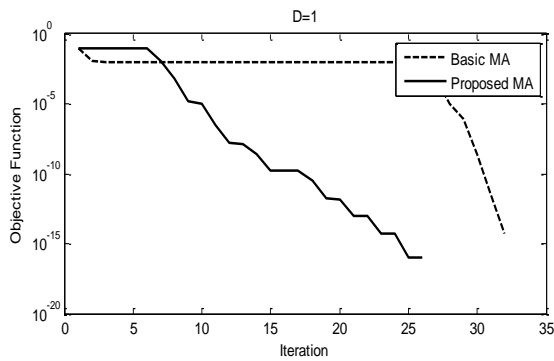
شکل (۴-۲). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار II



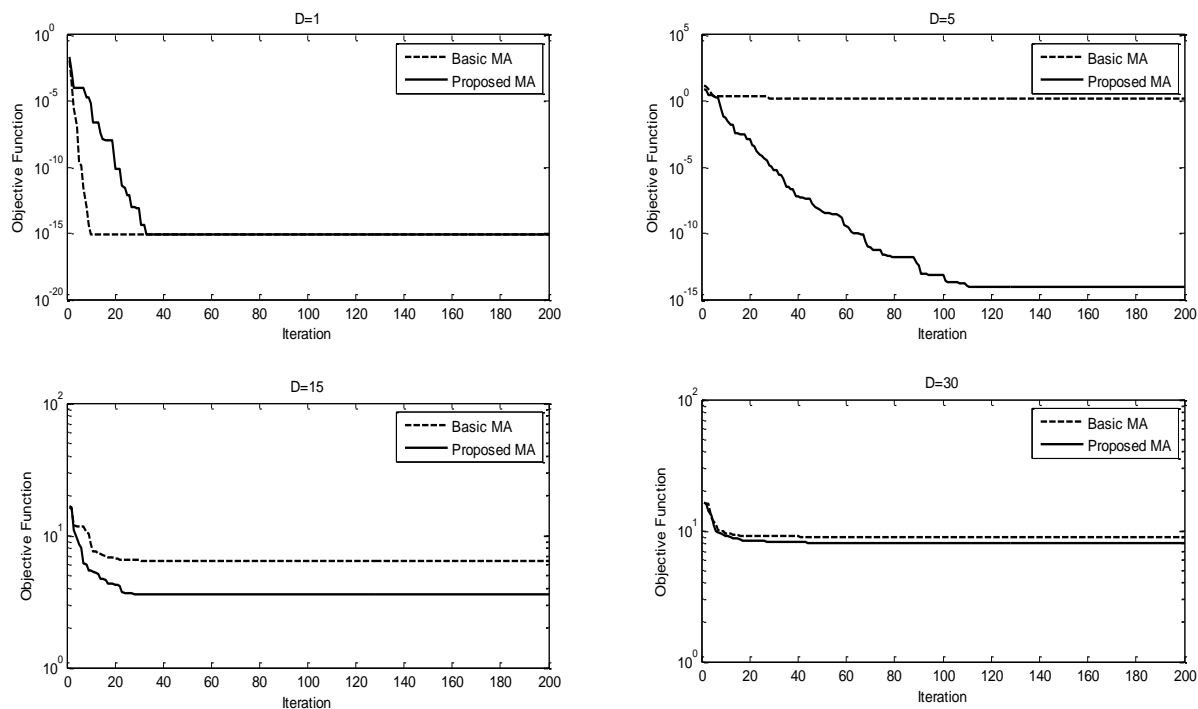
شکل (۴-۳). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار III



شکل (۴-۴). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار IV



شکل (۴-۵). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار V



شکل (۴-۶). مقایسه عملکرد الگوریتم‌های ممتیک پایه و پیشنهادی در یافتن مقدار بهینه تابع معیار VI همانطور که مشاهده شد، الگوریتم ممتیک پیشنهادی با بهره‌گیری از جستجوی محلی با سرعت مرتبه‌کسری، در حل مسائل بهینه‌سازی عددی عملکرد مناسبی از خود نشان می‌دهد. استفاده از سرعت مرتبه‌کسری در بخش جستجوی محلی الگوریتم پیشنهادی، باعث می‌شود که ذرات با حفظ حافظه‌ی بیشتر تغییر موقعیت دقیق‌تری داشته باشند، و لذا دقت الگوریتم در یافتن پاسخ بهینه به شکل قابل ملاحظه‌ای افزایش یابد.

۳-۴ مقایسه عملکرد PID کلاسیک و PID مرتبه‌کسری

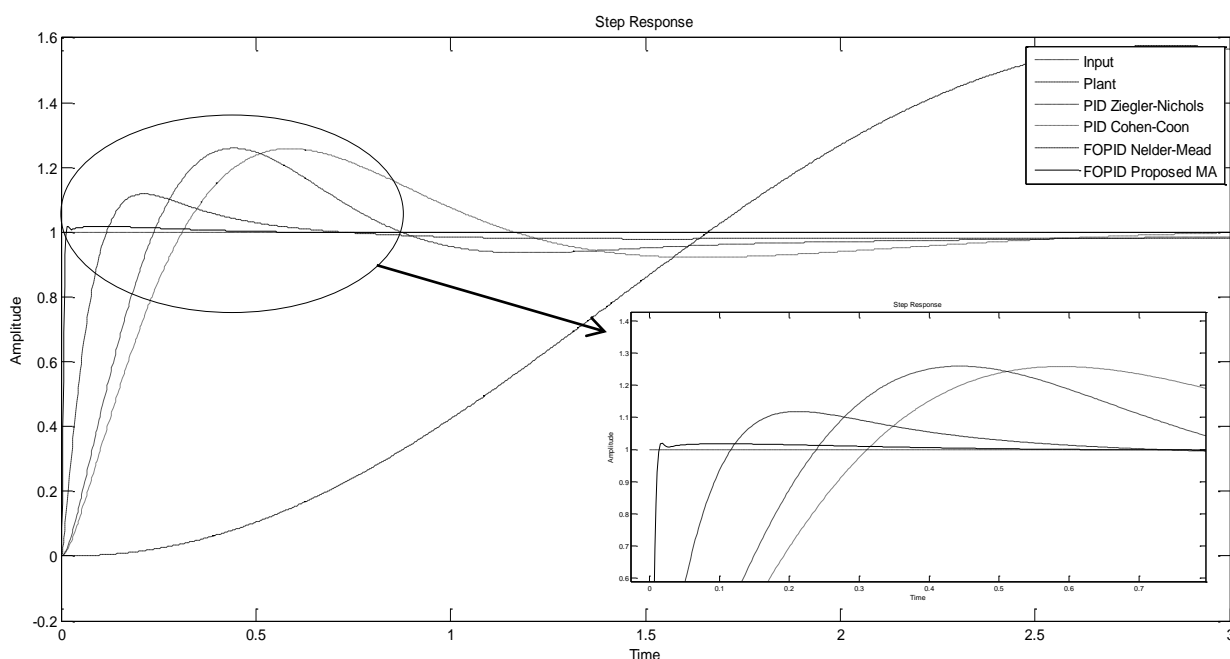
در این بخش جهت بررسی عملکرد کنترل‌کننده‌های PID کلاسیک و مرتبه‌کسری، به طراحی و اعمال آنها بر روی یک فرآیند مرتبه‌کسری خواهیم پرداخت. فرآیند مرتبه‌کسری تحت کنترل و کنترل‌کننده‌های طراحی شده برای آن در جدول (۴-۳) آورده شده‌اند.

جدول (۴-۳). نمایش ضرایب کنترل کننده‌ها و مقایسه حداکثر فراجهش آنها

Plant	Designed Controller	Controller Parameters					Maximum Overshoot (%)
		K_p	K_i	λ	K_d	μ	
$\frac{1}{0.8s^{2.2} + 0.9s^{0.5} + 1}$	PID Ziegler-Nichols	16.628102	9.442247	-	7.223016	-	25.92
	PID Cohen-Coon	8.965401	11.866824	-	5.382799	-	25.84
	FOPID Nelder-Mead	22.40669	31.21804	0.12833	20.32678	1.06563	11.80
	FOPID MA-Opt.	393.955	353.985	0.0122	117.849	1.2240	1.73

جهت طراحی و اعمال کنترل کننده‌های مرتبه صحیح، ابتدا فرآیند مرتبه‌کسری با یک سیستم مرتبه‌صحیح تقریب زده شده است. سپس کنترل کننده‌های PID توسط جعبه ابزار SISO Tool طراحی شده‌اند. کنترل کننده‌ی مرتبه‌کسری طراحی شده از روش می‌نیم‌سازی نیز با استفاده از روش سیمپلکس نلدر-مید و با استفاده از جعبه ابزار FOMCON در نرم‌افزار متلب طراحی شده است.

پس از اعمال ورودی پله و با فیدبک واحد طبق شکل (۴-۷)، عملکرد کنترل کننده‌ها در کنترل فرآیند به صورت زیر می‌باشد. در شکل زیر علاوه بر سیستم‌های کنترل شده با استفاده از کنترل کننده‌های طراحی شده، پاسخ پله فرآیند کنترل نشده نیز آورده شده است.



شکل (۴-۷). پاسخ پله فرآیند کنترل نشده و فرآیند کنترل شده با استفاده از کنترل کننده‌های مختلف

همانطور که از جدول (۳-۴) و شکل (۷-۴) مشخص است، کنترل کننده‌های FOPID در پاسخ دهی بهتر و سریعتر از کنترل کننده‌های PID کلاسیک عمل می‌کنند، چرا که دو پارامتر بیشتر از کنترل کننده‌های PID کلاسیک جهت تنظیم و طراحی دارند. این دو پارامتر همان مشتق و انتگرال مرتبه-کسری هستند. شکل (۷-۴) به خوبی نشان می‌دهد که کنترل کننده‌ی FOPID طراحی شده با الگوریتم ممیتیک پیشنهادی نسبت به کنترل کننده‌ی FOPID طراحی شده از روش می‌نیمم‌سازی، از زمان صعود، زمان نشست و فراجاهش بسیار کمتری برخوردار می‌باشد، که این امر نشان دهنده‌ی برتری روش طراحی پیشنهادی و استفاده شده، نسبت به دیگر روش‌های طراحی می‌باشد.

۴-۴ مقایسه عملکرد الگوریتم‌های تکاملی در طراحی کنترل کننده PID مرتبه-

کسری

در این بخش به بررسی و مقایسه عملکرد الگوریتم‌های تکاملی توضیح داده شده در فصل دوم و الگوریتم ممیتیک پیشنهادی در طراحی کنترل کننده PID مرتبه‌کسری خواهیم پرداخت. بدین منظور برای سه فرآیند مرتبه‌کسری، مرتبه‌صحیح و تاخیردار زیر، کنترل کننده‌های PID مرتبه‌کسری بهینه با استفاده از الگوریتم‌های تکاملی زنبورها، کرم شب‌تاب، انبوه ذرات، تکامل تفاضلی، رقابت استعماری، الگوریتم ممیتیک پایه و الگوریتم ممیتیک پیشنهادی طراحی خواهیم کرد. جدول (۴-۴) فرآیندهای تحت کنترل، مشخصات مطلوب و تنظیمات الگوریتم‌ها را نشان می‌دهد. اولین فرآیند، یک فرآیند مرتبه‌کسری نمونه می‌باشد که در مرجع [۵] معرفی شده است. فرآیند دوم که فرآیندی مرتبه صحیح است، یک سیستم راکتور زیستی [۵۸] می‌باشد. راکتورهای زیستی تجهیزاتی هستند که جهت کشت میکروب‌ها و در نتیجه بررسی تاثیر و عملکرد آنها روی مواد مورد استفاده قرار می‌گیرند. فرآیند سوم نیز فرآیندی تاخیردار می‌باشد که در مرجع [۴۳] معرفی شده است. مشخصات مطلوب در این طراحی‌ها، حداکثر فراجاهش برابر ۵٪ و حداکثر زمان صعود برابر ۳ ثانیه می‌باشد، که

در جدول مشخص شده‌اند. از آنجا که پنج پارامتر K_p, K_i, K_d, λ و μ جهت طراحی داریم، لذا تعداد متغیرهای تعریف شده برای هر الگوریتم ۵ متغیر می‌باشد. تعداد افراد جمعیت برای تمام الگوریتم‌ها ۵۰ نفر در نظر گرفته شده است و هر الگوریتم ۵۰ مرتبه اجرا شده و بهترین، بدترین و مقدار میانگین هر پارامتر بدست آورده شده‌اند. جهت نمایش دقیق عملکرد هر الگوریتم در بدست آوردن پارامترهای کنترل‌کننده، برای هر الگوریتم ۱۰، ۵۰ و ۱۰۰ تکرار در هر مرحله اجرا (شرط توقف الگوریتم) در نظر گرفته شده است. تنظیمات الگوریتم‌ها به صورت زیر در نظر گرفته شده است.

- **الگوریتم زنبورها:** تعداد زنبورهای پیشاهنگ ۱۵ عدد، شعاع همسایگی هر زنبور ۰/۱ فضای جستجو و نرخ تضعیف شعاع همسایگی ۰/۹۹ در نظر گرفته شده است.
- **الگوریتم کرم شب تاب:** ضریب جذب نور برابر ۱، ضریب جذب کرم‌ها ۲، ضریب جهش ۰/۲ نرخ تضعیف ضریب جهش ۰/۹۹ در نظر گرفته شده است.
- **الگوریتم بهینه‌سازی انبوه ذرات:** وزن اینرسی، ضریب آموزش شخصی و ضریب آموزش سراسری به ترتیب ۰/۷۲۹۸، ۱/۴۹۶۲ و ۱/۴۹۶۲ در نظر گرفته شده است.
- **الگوریتم تکامل تفاضلی:** فاکتور سنجش بین ۰/۱ تا ۰/۰۲، و احتمال تقاطع برابر ۰/۱ در نظر گرفته شده است.
- **الگوریتم رقابت استعماری:** تعداد کشورهای استعمارگر ۱۰ و احتمال انقلاب برابر ۰/۱ در نظر گرفته شده است.
- **الگوریتم ممتیک پایه:** وزن اینرسی، ضریب آموزش شخصی و ضریب آموزش سراسری به ترتیب ۰/۲، ۱/۴۹۴ و ۱/۴۹۴ در نظر گرفته شده است.
- **الگوریتم ممتیک پیشنهادی:** وزن اینرسی، ضریب آموزش شخصی و ضریب آموزش سراسری به ترتیب ۰/۲، ۱/۴۹۴ و ۱/۴۹۴ در نظر گرفته شده است. همچنین مرتبه‌ی مشتق مرتبه کسری برابر با ۰/۵ در نظر گرفته شده است.

محدوده فضای جستجو برای فرآیندها و همچنین تنظیمات الگوریتمها نیز به صورت زیر در نظر گرفته شده است.

فرآیند اول:

$$0 \leq K_p, K_i, K_d \leq 500 \quad \blacksquare$$

$$0 \leq \lambda, \mu \leq 2 \quad \blacksquare$$

• فرآیند دوم:

$$-50 \leq K_p, K_i, K_d \leq 50 \quad \blacksquare$$

$$0 \leq \lambda, \mu \leq 2 \quad \blacksquare$$

• فرآیند سوم:

$$0 \leq K_p, K_i, K_d \leq 5 \quad \blacksquare$$

$$0 \leq \lambda, \mu \leq 2 \quad \blacksquare$$

جدول (۴-۴). نمایش فرآیندهای تحت کنترل، مطلوبات طراحی و تنظیمات الگوریتمها

Plant Number	Process plant transfer function	Maximum overshoot (%)	Maximum Rise time (sec)	Parameter Settings for EAs			
				No. of Variables	Initial Population	No. of Iterations	Total Runs
I	$\frac{1}{0.8s^{2.2} + 0.9s^{0.5} + 1}$	10	0.3	5	50	10 50 100	50
II	$\frac{-1.39s^2 - 1.99s - 0.2577}{s^3 + 1.408s^2 + 0.656s + 0.101}$	10	0.3	5	50	10 50 100	50
III	$\frac{400}{s^2 + 50s} e^{-0.5s}$	10	0.3	5	50	10 50 100	50

نتایج بدست آمده از اعمال الگوریتم‌های ذکر شده در طراحی کنترل‌کننده‌ها، به منظور کنترل فرآیندهای جدول (۴-۴) در جدول‌های (۴-۴) تا (۱۴-۴) آورده شده‌اند.

جدول (۵-۴). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰ تکرار در هر اجرا برای فرآیند اول

Iteration Number : 10		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
I	Kp	Best	343.4051	320.4442	400	400	380.1162	365.1417	398.6546
		Mean	325.6174	345.8150	360.5684	383.6276	338.9045	324.1805	341.4922
		Worst	350.5249	362.0252	400	400	375.8866	295.3790	242.0841
		S.D.	0	1.5842e-05	1.6646	0.3738	0.8990	0	5.9918e-05
	Ki	Best	397.4980	351.8942	400	400	400	271.4608	329.1348
		Mean	319.8010	345.2257	335.0707	352.7167	327.5922	315.2664	327.4132
		Worst	273.0608	327.8952	297.5121	240	311.2180	334.5815	273.6442
		S.D.	0	0	0.5312	0	3.4270	0	5.9918e-05
	λ	Best	0.0127	0.0125	0.0120	0.0120	0.0120	0.0140	0.0129
		Mean	0.0151	0.0142	0.0130	0.0124	0.0145	0.0154	0.0148
		Worst	0.0158	0.0151	0.0193	0.0199	0.0149	0.0141	0.0120
		S.D.	0	3.8162e-03	4.9996e-04	1.8286e-18	1.1744e-04	0	1.8124e-05
	Kd	Best	121.6641	122.5210	120	120	120	135.9069	123.2168
		Mean	130.6846	128.5461	120.716	120.0096	121.6491	133.4814	124.3556
		Worst	138.5155	135.9942	120	120.012	139.1614	148.9983	127.3093
		S.D.	0	0	0.0128	0	0.2604	0	2.9959e-05
	μ	Best	1.2253	1.2375	1.2245	1.2245	1.2248	1.2212	1.2208
		Mean	1.2313	1.2499	1.2243	1.2240	1.2246	1.2716	1.2360
		Worst	1.2724	1.2516	1.2219	1.2202	1.2293	1.3312	1.3698
		S.D.	2.3406e-16	5.6152e-08	3.1051e-04	1.7005e-04	6.3435e-04	0	1.2151e-05
	O.F.	Best	1.5778e+06	1.5934e+06	1.6152e+06	1.6032e+06	1.6470e+06	1.7595e+06	1.6115e+06
		Mean	2.9584e+06	1.9750e+06	1.8820e+06	1.7232e+06	2.1996e+06	2.1446e+06	1.7974e+06
		Worst	4.0312e+06	2.4320e+06	2.4622e+06	2.3848e+06	3.2929e+06	2.7120e+06	2.4595e+06
	Time (sec)		0.254	0.247	0.689	1.406	0.213	0.0841	0.110

همانطور که از جدول (۴-۵) مشخص است، پس از ۱۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $1.5778e+06$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های FA, Basic MA, DE, BA, PSO و ICA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد.

جدول (۴-۶). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار در هر اجرا برای فرآیند اول

Iteration Number : 50		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
I	Kp	Best	299.4784	350.9510	400	400	400	355.4609	400.4764
		Mean	350.2932	372.6097	399.9997	400	399.6398	342.1126	393.9356
		Worst	294.2992	395.2780	400	400	399.9541	358.6771	264.9519
		S.D.	5.7421e-14	3.6421e-06	4.1479	0	4.6423	1.1817	3.6530
	Ki	Best	356.4143	376.5193	400	400	400	329.9795	387.8856
		Mean	323.5888	342.3891	399.9998	400	392.0636	324.2436	372.5527
		Worst	368.4920	350.4822	400	400	279.6183	383.2904	243.7528
		S.D.	0	1.9862e-05	0.5077	0.3460	4.7057	0.0494	10.7925
	λ	Best	0.0122	0.0120	0.0120	0.0120	0.0120	0.0127	0.0117
		Mean	0.0144	0.0135	0.0120	0.0120	0.0120	0.0143	0.0124
		Worst	0.0126	0.0146	0.0120	0.0120	0.0120	0.0140	0.0196
		S.D.	3.5047e-18	4.9510e-08	4.8762e-05	2.1701e-06	1.6627e-04	8.3642e-05	9.6625e-05
	Kd	Best	117.9850	120.5562	120	120	120	125.3030	118.4429
		Mean	123.5797	122.9530	120.0007	120	120.0146	126.6986	120.6411
		Worst	144.163	128.4175	120.0342	120	120.1252	126.6720	118.6887
		S.D.	0.4439	0.3162	0.0239	0	0.1205	0.3569	0.4757
	μ	Best	1.2248	1.2361	1.2245	1.2245	1.2245	1.2212	1.2215
		Mean	1.2253	1.2425	1.2246	1.2255	1.2244	1.2595	1.2344
		Worst	1.2248	1.2441	1.2247	11.2743	1.2227	1.3313	1.2082
		S.D.	5.6140e-04	3.8251e-04	1.8598e-04	1.7978e-04	3.7204e-04	5.3129e-05	2.1948e-04
	O.F.	Best	1.5301e+06	1.5812e+06	1.6032e+06	1.6032e+06	1.6032e+06	1.6011e+06	1.5850e+06
		Mean	1.7488e+06	1.6530e+06	1.6033e+06	1.6146e+06	1.6166e+06	1.7731e+06	1.4245e+06
		Worst	1.9833e+06	1.9654e+06	1.6047e+06	2.1724e+06	1.7444e+06	2.1078e+06	1.7538e+06
	Time (sec)		0.752	0.689	1.398	7.219	0.581	0.267	0.409

همانطور که از جدول فوق مشخص است، پس از ۵۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $1.5301e+06$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های Basic MA، DE و ICA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم‌های FA، BA و PSO نیز با مقداری یکسان در جایگاه بعد قرار گرفته‌اند. همانطور که مشخص است، الگوریتم FA در بهینه محلی گرفتار شده است، چرا که پاسخ آن برای ۱۰ و ۵۰ تکرار در هر مرحله اجرا یکسان می‌باشد. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد.

جدول (۴-۷). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار در هر اجرا برای فرآیند اول

Iteration Number : 100		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
I	Kp	Best	393.9550	395.7720	400	400	400	396.0709	400.9225
		Mean	342.2355	368.1892	400	400	400	358.3717	389.3161
		Worst	255.2316	270.5820	400	400	400	292.2961	243.0294
		S.D.	5.7130e-14	3.1124e-08	0.2162	0.0061	6.6488	2.2607	8.1938
	Ki	Best	353.9859	370.4120	400	400	400	299.1998	402.5145
		Mean	339.5296	352.9470	400	400	399.9998	343.5250	380.8306
		Worst	282.9075	320.2249	400	400	399.9883	383.0178	400.4538
		S.D.	1.1426e-13	5.9510e-06	1.3722	0	7.1707	3.0934	10.1811
	λ	Best	0.0122	0.0120	0.0120	0.0120	0.0120	0.0125	0.0117
		Mean	0.0144	0.0135	0.0120	0.0120	0.0120	0.0136	0.0125
		Worst	0.0151	0.0142	0.0120	0.120	0.0120	0.0156	0.0198
		S.D.	2.1159e-06	6.5294e-06	1.8389e-05	9.2776e-08	2.1811e-04	7.9014e-05	9.0066e-05
	Kd	Best	117.8497	119.8542	120	120	120	120.8268	118.7882
		Mean	122.0385	120.9814	120	120	120	124.2272	120.3004
		Worst	123.5634	122.8720	120	120	120	125.9110	121.0450
		S.D.	0.5909	0.2014	0.0057	0	0.1029	0.6784	0.2860
	μ	Best	1.2240	1.2120	1.2245	1.2245	1.2245	1.2210	1.2215
		Mean	1.2179	1.2350	1.2245	1.2245	1.2245	1.2445	1.2249
		Worst	1.2252	1.2344	1.2245	1.2245	1.2245	1.2818	1.2445
		S.D.	1.8572e-04	2.3597e-04	2.0060e-04	8.9017e-05	4.6317e-04	2.1197e-04	2.7742e-05
	O.F.	Best	1.5161e+06	1.5570e+06	1.6032e+06	1.6032e+06	1.6032e+06	1.5708e+06	1.5518e+06
		Mean	1.5764e+06	1.5875e+06	1.6032e+06	1.6032e+06	1.6032e+06	1.6017e+06	1.5753e+06
		Worst	1.8775e+06	1.7920e+06	1.6032e+06	1.6032e+06	1.6033e+06	1.8831e+06	1.8313e+06
	Time (sec)		1.416	1.398	7.106	21.724	1.104	0.497	0.735
N.F.E.		559889	559889	540151	517306	544739	534439	464939	

همانطور که از جدول فوق مشخص است، پس از ۱۰۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $1.5161e+06$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های DE، Basic MA و ICA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم‌های FA، BA و PSO نیز با مقداری یکسان در جایگاه بعد قرار گرفته‌اند. همانطور که مشاهده می‌شود، الگوریتم‌های FA، BA و PSO در بهینه محلی گرفتار شده‌اند، چرا که پاسخ آنها نسبت به جدول (۴-۶) تغییری نکرده است. مشخص است که سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد و همچنین کمترین تعداد اجرای تابع هدف توسط الگوریتم DE صورت گرفته است. کاملاً واضح است که استفاده از بخش جستجوی محلی انبوه ذرات با سرعت مرتبه‌کسری و ترکیب آن با الگوریتم بهینه‌سازی انبوه ذرات نه تنها از همگرایی زودرس و گرفتار شدن در بهینه‌های محلی جلوگیری کرده است، بلکه منجر به نیل به

بهینه‌ترین پاسخ نیز شده است. لذا جهت طراحی کنترل‌کننده FOPID بهینه برای فرآیند مرتبه-کسری، مناسب‌ترین الگوریتم، الگوریتم ممتیک پیشنهادی می‌باشد.

جدول (۴-۸). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰ تکرار در هر اجرا برای فرآیند دوم

Iteration Number : 10		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
II	Kp	Best	-28.4845	-28.6462	-28	-28	-28.0135	-28.2614	-28.0904
		Mean	-28.5649	-28.7422	-28.0031	-28.0017	-28.0031	-28.9413	-28.2964
		Worst	-28.7452	-28.7916	-28	-28	-28	-30.2729	-28.6823
		S.D.	0	0	0	0	0	0	7.4898e-15
	Ki	Best	-9.8925	-9.9522	-10	-10	-9.5532	-7.8365	-9.9193
		Mean	-8.9437	-9.9841	-9.5917	-9.8084	-8.9345	-8.7714	-9.5134
		Worst	-7.8213	-9.9720	-9.6908	-10	-8.0567	-8.4640	-8.8384
		S.D.	0	0	0.0078	0	0.0442	0	0.0122
	λ	Best	0.7359	0.7346	0.7333	0.7335	0.7334	0.7824	0.7335
		Mean	0.7878	0.7380	0.8399	0.7434	0.8024	0.8185	0.7683
		Worst	0.7285	0.7459	0.8331	0.7336	0.8834	0.7828	0.8828
		S.D.	0	0	1.9149e-04	2.5888e-04	5.6363e-04	0	0
	Kd	Best	0.8074	0.9251	0.9584	1	0.9687	0.9415	0.9997
		Mean	0.8527	0.9522	0.9333	0.9160	0.8646	0.8684	0.9072
		Worst	0.7921	0.9422	0.9133	0.7234	0.8297	0.9188	0.7960
		S.D.	0	0	0.0028	0	0.0042	0	0.0059
	μ	Best	0.1605	0.1611	0.1613	0.1610	0.1624	0.1640	0.1593
		Mean	0.1637	0.1639	0.1618	0.1625	0.1611	0.1618	0.1609
		Worst	0.1887	0.1722	0.2000	0.2000	0.1627	0.1599	0.1633
		S.D.	0	1.3351e-05	9.7596e-05	2.9257e-17	7.8087e-04	0	4.2562e-04
	O.F.	Best	8.7292e+05	8.7924e+05	8.7825e+05	8.7489e+05	8.8723e+05	9.5282e+05	8.8393e+05
		Mean	1.0070e+06	9.9561e+05	9.1613e+05	8.9058e+05	9.2328e+05	1.0121e+06	9.3010e+05
		Worst	1.1485e+06	1.1254e+06	9.7021e+05	9.4099e+05	9.5626e+05	1.0831e+06	9.9684e+05
	Time (sec)		0.339	0.295	1.429	3.053	0.247	0.177	0.218

همانطور که از جدول فوق مشخص است، پس از ۱۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $8.7292e+05$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های FA، BA، Basic MA، PSO، DE و ICA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد.

جدول (۴-۹). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار در هر اجرا برای فرآیند دوم

Iteration Number : 50		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
II	Kp	Best	-27.7752	-27.8914	-28	-28	-28	-28.0830	-27.5379
		Mean	-28.1770	-27.9961	-28	-28	-28.0007	-28.4458	-28.0772
		Worst	-28.4115	-28.1644	-28	-28	-28	-29.4067	-28.6789
		S.D.	0.0138	0.0264	9.6252e-05	0	0.0139	0.0073	0.0549
	Ki	Best	-9.0496	-9.8104	-10	-10	-10	-9.8996	-10.0373
		Mean	-8.9822	-9.2541	-10	-10	-9.9991	-9.4111	-9.9609
		Worst	-8.6435	-9.0151	-10	-10	-10	-9.5478	-10.538
		S.D.	1.7944e-15	7.6942e-09	0.0432	0	0.0484	0.0266	0.0991
	λ	Best	0.7337	0.7336	0.7335	0.7335	0.7335	0.7337	0.7337
		Mean	0.8023	0.7452	0.8152	0.7484	0.7973	0.7895	0.7425
		Worst	0.8334	0.8102	0.9327	0.8331	0.8831	0.7362	0.8332
		S.D.	1.1215e-15	4.2311e-06	1.5079e-04	1.4269e-04	3.8489e-04	0.0016	0.0010
	Kd	Best	0.9411	0.9458	1	1	1	0.9498	1.0038
		Mean	0.8560	0.9215	1	1	0.9895	0.9242	0.9956
		Worst	0.7749	0.8652	1	1	1	0.9887	1.0039
		S.D.	0	0	0.0109	0	0.0044	0.0058	0.0175
	μ	Best	0.1590	0.1602	0.1609	0.1609	0.1609	0.1606	0.1609
		Mean	0.1607	0.1608	0.1610	0.1609	0.1610	0.1617	0.1609
		Worst	0.1682	0.1612	0.1610	0.1610	0.1607	0.1610	0.1605
		S.D.	8.4112e-17	5.6415e-09	1.2701e-04	1.3809e-04	4.9550e-04	1.3189e-04	4.7602e-04
	O.F.	Best	8.5305e+05	8.6532e+05	8.7489e+05	8.7489e+05	8.7489e+05	8.8619e+05	8.7355e+05
		Mean	8.7145e+05	8.7924e+05	8.9678e+05	8.7897e+05	8.9295e+05	9.4023e+05	8.8389e+05
		Worst	8.8379e+05	8.9279e+05	9.2474e+05	9.0171e+05	9.1377e+05	9.9022e+05	8.4722e+05
	Time (sec)		1.204	1.152	3.991	8.702	0.852	0.577	0.803

همانطور که از جدول فوق مشخص است، پس از ۵۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $8.5305e+05$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز الگوریتم‌های Basic MA و DE بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم‌های FA، BA و PSO نیز با مقداری یکسان در جایگاه بعد قرار گرفته‌اند. همانطور که مشخص است، الگوریتم FA در بهینه محلی گرفتار شده است، چرا که پاسخ آن برای ۱۰ و ۵۰ تکرار در هر مرحله اجرا یکسان می‌باشد. به نظر می‌رسد که الگوریتم‌های BA و PSO نیز با مقادیری مشابه FA در بهینه محلی گرفتار شده باشند. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد.

جدول (۴-۱۰). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار در هر اجرا برای فرآیند دوم

Iteration Number : 100		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
II	Kp	Best	-27.6888	-27.7632	-28	-28	-28	-28.0351	-27.4563
		Mean	-28.0542	-27.9981	-28	-28	-28	-28.2968	-28.0982
		Worst	-28.3925	-28.0156	-28	-28	-28	-28.7111	-28.9512
		S.D.	0.0604	0.0251	1.3994e-09	0	0.0052	0.0100	0.0153
	Ki	Best	-9.2725	-9.5890	-10	-10	-10	-9.8090	-9.9764
		Mean	-9.0326	-9.6166	-10	-10	-9.9982	-9.7644	-9.9588
		Worst	-9.4793	-9.7084	-10	-10	-10	-9.9120	-9.9352
		S.D.	1.4282e-14	4.6259e-06	0.0315	0	0.0928	0.0948	0.0842
	λ	Best	0.7334	0.7335	0.7335	0.7335	0.7335	0.7336	0.7833
		Mean	0.7943	0.8014	0.8162	0.7554	0.7913	0.7488	0.7434
		Worst	0.7836	0.8162	0.9327	0.7336	0.7834	0.7836	0.7335
		S.D.	3.1190e-04	1.3594e-04	1.3049e-04	9.0578e-05	2.7343e-04	5.1168e-04	2.3095e-04
	Kd	Best	0.8049	0.8591	1	1	1	0.9898	1.0026
		Mean	0.8678	0.9120	1	1	1	0.9762	0.9950
		Worst	0.8885	0.9328	1	1	1	0.9921	1.0038
		S.D.	2.2316e-16	3.5149e-06	0.0037	4.8840e-05	0.0110	0.0127	0.0124
	μ	Best	0.1603	0.1607	0.1609	0.1609	0.1609	0.1615	0.1609
		Mean	0.1599	0.1605	0.1610	0.1617	0.1618	0.1613	0.1608
		Worst	0.1523	0.1601	0.1610	0.2000	0.2000	0.1612	0.1610
		S.D.	2.5192e-05	1.5840e-04	1.7891e-04	8.2091e-05	1.7515e-04	5.0366e-04	4.1712e-04
	O.F.	Best	8.4210e+05	8.5158e+05	8.7489e+05	8.7489e+05	8.7489e+05	8.8151e+05	8.6288e+05
		Mean	8.6350e+05	8.7122e+05	8.9696e+05	8.8188e+05	8.9179e+05	9.0593e+05	8.8567e+05
		Worst	8.7236e+05	8.8692e+05	9.2474e+05	9.3573e+05	9.5029e+05	9.3943e+05	9.4020e+05
	Time (sec)		2.306	2.270	12.932	37.721	1.580	0.973	1.417
N.F.E.		565282	544284	525120	525612	549520	542150	474526	

همانطور که از جدول فوق مشخص است، پس از ۱۰۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $8.4210e+05$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های Basic MA، DE و ICA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم‌های FA، BA و PSO نیز با مقداری یکسان در جایگاه بعد قرار گرفته‌اند. همانطور که مشاهده می‌شود، الگوریتم‌های FA، BA و PSO در بهینه محلی گرفتار شده‌اند، چرا که پاسخ آنها نسبت به جدول (۴-۹) تغییری نکرده است. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد و همچنین کمترین تعداد اجرای تابع هدف توسط الگوریتم DE صورت گرفته است. کاملاً واضح است که استفاده از بخش جستجوی محلی انبوه ذرات با سرعت مرتبه‌کسری و ترکیب آن با الگوریتم بهینه‌سازی انبوه ذرات نه تنها از همگرایی زودرس و گرفتار شدن در بهینه‌های محلی جلوگیری کرده است، بلکه منجر به نیل به بهینه‌ترین پاسخ نیز

شده است. لذا جهت طراحی کنترل کننده FOPID بهینه برای فرآیند مرتبه صحیح مناسبترین الگوریتم، الگوریتم ممتیک پیشنهادی می باشد.

جدول (۴-۱۱). پارامترهای یافت شده توسط الگوریتمهای تکاملی با ۱۰ تکرار در هر اجرا برای فرآیند سوم

Iteration Number : 10		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
III	Kp	Best	1.0194	1.1201	1	1	1	1	0.9955
		Mean	1.1088	1.1306	1	1	1.0028	1.0020	1.0357
		Worst	1.4213	1.1280	1	1	1	1	1.1212
		S.D.	0	0	0	0	0	0	5.9150e-05
	Ki	Best	0.4665	0.5011	0.4002	0.4000	0.4302	0.4000	0.4060
		Mean	0.5307	0.5128	0.4100	0.4008	0.4726	0.4194	0.4280
		Worst	0.4957	0.5952	0.4120	0.4000	0.6208	0.6751	0.4112
		S.D.	0	0	5.8225e-16	5.8514e-17	0.0037	5.5769e-04	9.5251e-04
	λ	Best	0.9658	0.9506	0.4015	0.4170	0.6165	0.4664	0.6113
		Mean	0.7264	0.9109	0.5232	0.6447	0.6946	0.6839	0.5822
		Worst	0.8207	0.8922	0.8532	1	0.5554	0.5653	0.5239
		S.D.	0	0	0.0011	0	4.6409e-04	0.0012	0.0115
	Kd	Best	0.5280	0.5694	0.4009	0.4000	0.4890	0.4352	0.4181
		Mean	0.6294	0.5952	0.5211	0.5271	0.6003	0.5418	0.4977
		Worst	0.7786	0.5486	0.7532	0.8632	0.6570	0.4299	0.4430
		S.D.	0	0	0.0010	1.0742e-04	0.0080	0.0017	0.0143
	μ	Best	0.4639	0.4820	0.4411	0.4672	0.4669	0.4672	0.4651
		Mean	0.3871	0.4201	0.4519	0.4641	0.3914	0.3973	0.4290
		Worst	0.3677	0.3956	0.4599	0.4671	0.3158	0.4670	0.3642
		S.D.	0	0	1.5122e-04	1.8322e-04	3.0850e-04	1.3837e-04	1.1495e-05
	O.F.	Best	3.5052e+07	3.6122e+07	3.9511e+07	3.6706e+07	3.9007e+07	3.7045e+07	3.8598e+07
		Mean	3.8903e+07	3.7966e+07	3.9857e+07	3.8076e+07	4.6334e+07	4.2318e+07	4.3279e+07
		Worst	4.2976e+07	4.1633e+07	4.1128e+07	4.0594e+07	5.9876e+07	4.9189e+07	5.1633e+07
	Time (sec)		0.154	0.134	0.961	1.458	0.116	0.079	0.165

همانطور که از جدول فوق مشخص است، پس از ۱۰ تکرار در هر مرحله اجرای الگوریتمها، بهینه ترین مقدار تابع هدف به میزان $8.5052e+07$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتمهای Basic MA، FA، ICA، DE، PSO و BA بهینه ترین مقادیر تابع هدف را بدست آورده اند. سریع ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می باشد. مشاهده می شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتمها کمتر می باشد.

جدول (۴-۱۲). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۵۰ تکرار در هر اجرا برای فرآیند سوم

Iteration Number : 50		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
III	Kp	Best	1.0154	1.0186	1	1	1	1	0.9634
		Mean	1.0698	1.0420	1	1	1.0001	1	1.0060
		Worst	1.1842	1.1620	1	1	1	1	1.0883
		S.D.	4.7195e-04	4.0036e-04	1.2256e-16	0	0.0021	5.4013e-05	0.0071
	Ki	Best	0.4034	0.4102	0.4000	0.4000	0.4000	0.4000	0.4119
		Mean	0.4634	0.4522	0.4001	0.4000	0.4001	0.4000	0.4096
		Worst	0.4665	0.5490	0.4003	0.4000	0.4000	0.4000	0.4326
		S.D.	0.0016	0.0026	1.5581e-14	1.6822e-16	0.0087	0.0013	0.0018
	λ	Best	0.4225	0.4206	0.4171	0.4171	0.5672	0.5167	0.4172
		Mean	0.6166	0.5520	0.6142	0.5988	0.6871	0.6910	0.4469
		Worst	0.8172	0.5952	0.5830	0.4171	0.7657	0.9151	0.4171
		S.D.	0.0012	0.0166	0	0	0.0020	8.9898e-04	0.0023
	Kd	Best	0.4718	0.4825	0.4000	0.4000	0.4000	0.4000	0.4032
		Mean	0.5746	0.4911	0.4003	0.4000	0.4342	0.4000	0.4060
		Worst	0.4911	0.4350	0.4004	0.4000	0.4000	0.4000	0.4070
		S.D.	2.4638e-04	1.3369e-04	1.9964e-18	1.6822e-16	0.0068	0.0059	0.0171
	μ	Best	0.4176	0.4720	0.4521	0.4672	0.4670	0.4672	0.4670
		Mean	0.4278	0.4811	0.4722	0.4612	0.3974	0.3885	0.4562
		Worst	0.3719	0.5120	0.3892	0.2679	0.2181	0.2679	0.4672
		S.D.	1.6616e-04	2.7403e-04	5.2316e-05	9.6892e-05	4.6236e-04	1.8880e-04	5.1435e-04
	O.F.	Best	3.4571e+07	3.5155e+07	3.4808e+07	3.6706e+07	3.7367e+07	3.7164e+07	3.5084e+07
		Mean	3.7298e+07	3.7266e+07	3.6584e+07	3.7524e+07	4.0154e+07	4.0283e+07	3.7974e+07
		Worst	3.9530e+07	3.9852e+07	3.8772e+07	4.2732e+07	4.5290e+07	4.4415e+07	4.3277e+07
	Time (sec)		0.617		3.821	7.790	0.417	0.272	0.385

همانطور که از جدول فوق مشخص است، پس از ۵۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه‌ترین مقدار تابع هدف به میزان $3.4571e+07$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز الگوریتم‌های BA، DE و Basic MA بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم FA نیز با مقداری برابر با جدول (۴-۱۱) در جایگاه بعد قرار گرفته است. همانطور که مشخص است، الگوریتم FA در بهینه محلی گرفتار شده است، چرا که پاسخ آن برای ۱۰ و ۵۰ تکرار در هر مرحله اجرا یکسان می‌باشد. الگوریتم‌های ICA و PSO نیز در آخر قرار گرفته‌اند. مشاهده می‌شود که بر خلاف دو سیستم پیشین، الگوریتم BA از بهینه محلی فرار کرده است. سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد.

جدول (۴-۱۳). پارامترهای یافت شده توسط الگوریتم‌های تکاملی با ۱۰۰ تکرار در هر اجرا برای فرآیند سوم

Iteration Number : 100		Algorithms Used							
Plant	Coefficients	Proposed MA	Basic MA	BA	FA	PSO	ICA	DE	
III	Kp	Best	1.0084	1.0152	1	1	1	1	0.9696
		Mean	1.0453	1.0345	1	1	1	1	1.0046
		Worst	1.0426	1.0884	1	1	1	1	1.0571
		S.D.	0.0075	0.0086	0	0	0.0021	1.9523e-05	0.0064
	Ki	Best	0.4021	0.4088	0.4000	0.4000	0.4000	0.4000	0.3928
		Mean	0.4310	0.4096	0.4000	0.4000	0.4000	0.4000	0.4026
		Worst	0.5118	0.5055	0.4000	0.4000	0.4000	0.4000	0.4135
		S.D.	0.0033	1.1456e-6	6.5953e-18	7.8107e-16	0.0051	0.0011	0.0057
	λ	Best	0.4192	0.4181	0.4171	0.4171	0.4670	0.4669	0.4171
		Mean	0.5038	0.4892	0.4719	0.5719	0.6631	0.6542	0.4460
		Worst	0.5191	0.5120	0.5171	0.4171	0.6661	0.6661	0.4170
		S.D.	0.0202	0.0275	0	0	0.0027	2.2757e-04	0.0217
	Kd	Best	0.4115	0.4088	0.4000	0.4000	0.4000	0.4000	0.4018
		Mean	0.4890	0.4355	0.4001	0.4000	0.4047	0.4000	0.4067
		Worst	0.5371	0.4941	0.4005	0.4000	0.4000	0.4000	0.4019
		S.D.	0.0035	1.3251e-04	5.6327e-18	7.8107e-16	0.0121	0.0060	0.0133
	μ	Best	0.4670	0.4670	0.4671	0.4672	0.4671	0.4672	0.4671
		Mean	0.4576	0.4415	0.4211	0.4582	0.4094	0.3765	0.4592
		Worst	0.4168	0.4322	0.3852	0.3178	0.2679	0.2181	0.4674
		S.D.	0.0011	1.6651e-04	6.3267e-06	9.4265e-05	1.8058e-04	1.3599e-04	3.8447e-04
	O.F.	Best	3.3384e+07	3.3812e+07	3.4213e+07	3.6706e+07	3.6965e+07	3.6947e+07	3.4726e+07
		Mean	3.6867e+07	3.4579e+07	3.6412e+07	3.7559e+07	3.9552e+07	4.0538e+07	3.7520e+07
		Worst	3.9118e+07	3.5667e+07	3.7566e+07	4.1426e+07	4.3796e+07	4.4998e+07	4.0597e+07
	Time (sec)		1.179	1.015	9.124	17.936	0.761	0.523	0.730
N.F.E.		589562	594221	536452	520289	534628	523642	422305	

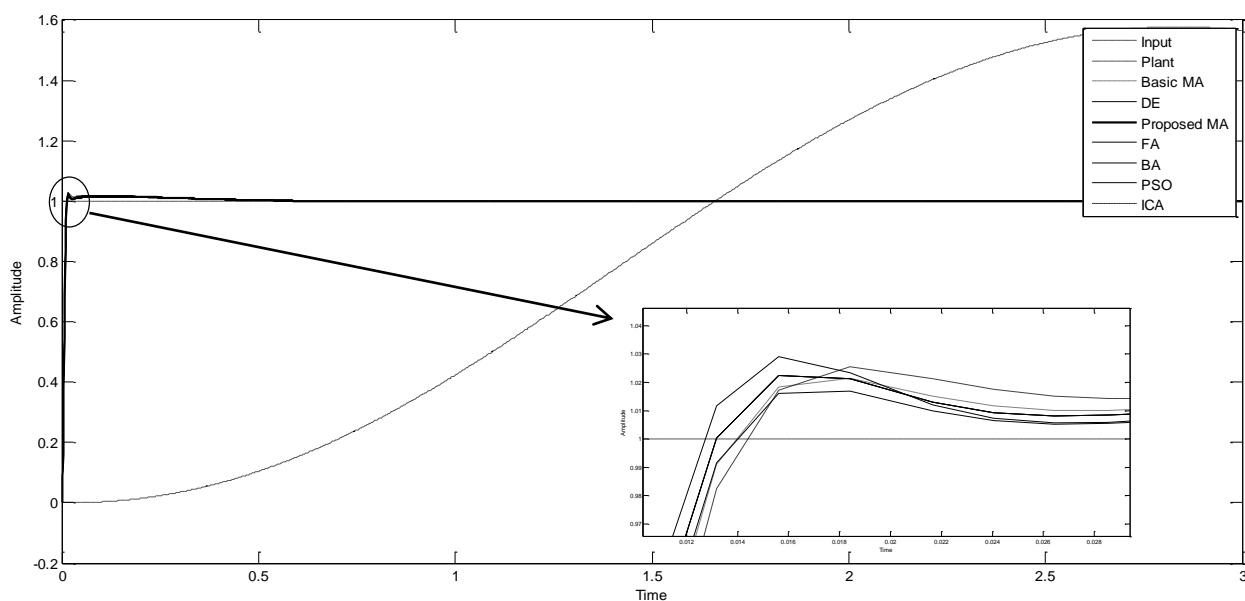
همانطور که از جدول فوق مشخص است، پس از ۱۰۰ تکرار در هر مرحله اجرای الگوریتم‌ها، بهینه ترین مقدار تابع هدف به میزان $3.3384e+07$ توسط الگوریتم ممتیک پیشنهادی بدست آمده است. پس از آن نیز به ترتیب الگوریتم‌های Basic MA، BA و DE بهینه‌ترین مقادیر تابع هدف را بدست آورده‌اند. الگوریتم FA، نیز همانند جدول (۴-۱۲) با مقداری یکسان در بهینه محلی گرفتار شده است. الگوریتم ICA نیز در جایگاه بعد قرار دارد. مشاهده می‌شود که الگوریتم PSO در بهینه محلی گرفتار نشده است (بر خلاف دو فرآیند قبل). سریع‌ترین الگوریتم در یافتن پاسخ ICA و کندترین الگوریتم FA می‌باشد. مشاهده می‌شود که میزان انحراف معیار الگوریتم ممتیک پیشنهادی از دیگر الگوریتم‌ها کمتر می‌باشد و همچنین کمترین تعداد اجرای تابع هدف توسط الگوریتم DE صورت گرفته است. کاملاً واضح است که استفاده از بخش جستجوی محلی انبوه ذرات با سرعت مرتبه‌کسری و ترکیب آن با الگوریتم بهینه‌سازی انبوه ذرات نه تنها از همگرایی زودرس و گرفتار شدن در بهینه‌های محلی جلوگیری کرده است، بلکه منجر به نیل به بهینه‌ترین پاسخ نیز شده است. لذا جهت طراحی کنترل‌کننده FOPID بهینه برای فرآیند تاخیردار مناسب‌ترین

الگوریتم، الگوریتم ممتیک پیشنهادی می‌باشد. نتایج بدست آمده از الگوریتم‌های تکاملی (به ازای ۱۰۰ تکرار در هر مرحله اجرا) در طراحی کنترل‌کننده بهینه، و با اعمال ورودی پله واحد شبیه‌سازی شده و نتایج به صورت جدول (۴-۱۴) بدست آمده‌اند.

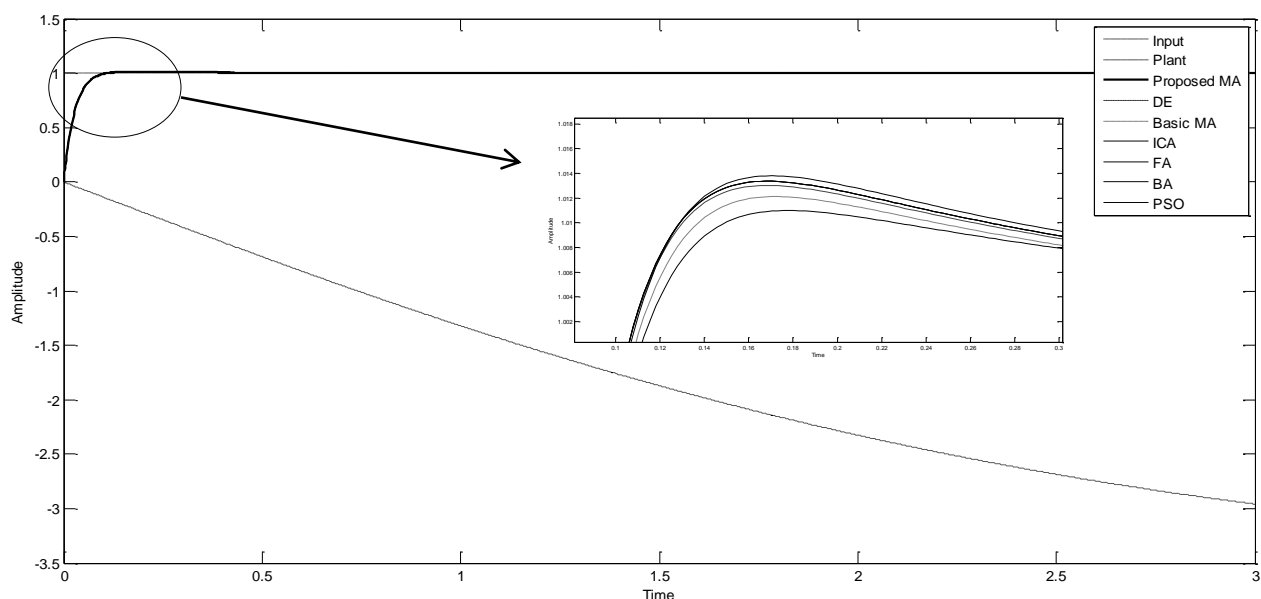
جدول (۴-۱۴). پاسخ پله فرآیندهای کنترل شده با کنترل‌کننده‌ی مرتبه‌کسری بهینه‌شده توسط الگوریتم‌های تکاملی

Plant	Algorithm Used	Unit Step Response		
		OverShoot (%)	Rise Time (sec)	Steady State Error
I	Proposed MA	1.732	0.0033	0.00120
	Basic MA	2.015	0.0061	0.00122
	BA	2.241	0.0077	0.00126
	FA	2.241	0.0077	0.00126
	PSO	2.241	0.0077	0.00126
	ICA	2.229	0.0074	0.00124
	DE	2.146	0.0068	0.00122
II	Proposed MA	1.142	0.0455	2.4876e-03
	Basic MA	1.191	0.0486	2.5103e-03
	BA	1.338	0.0548	2.5271e-03
	FA	1.338	0.0548	2.5271e-03
	PSO	1.338	0.0548	2.5271e-03
	ICA	1.323	0.0540	2.5509e-03
	DE	1.313	0.0542	2.5158e-03
III	Proposed MA	0.344	0.1445	0.001247
	Basic MA	0.412	0.1568	0.001289
	BA	0.422	0.1591	0.001291
	FA	0.452	0.1641	0.001331
	PSO	0.716	0.1705	0.001793
	ICA	0.645	0.1674	0.001649
	DE	0.441	0.1607	0.001312

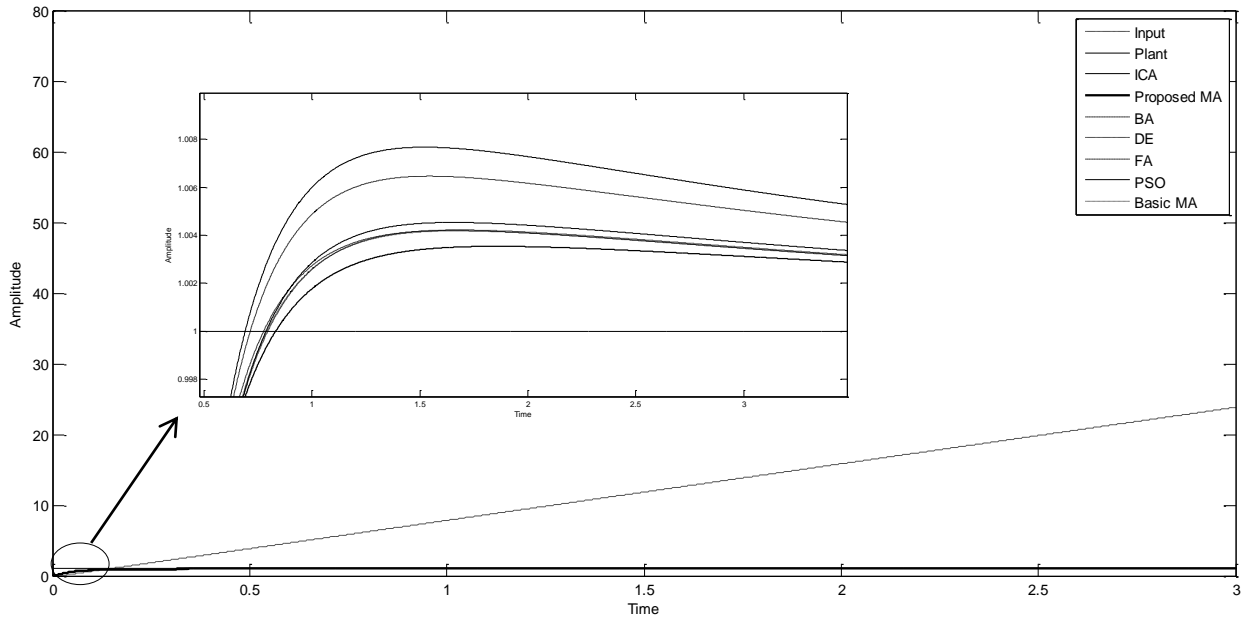
همانطور که از جدول فوق مشخص است، الگوریتم ممتیک پیشنهادی برای هر سه فرآیند بهترین کنترل کننده را طراحی کرده است. کنترل کننده ذکر شده منجر به رسیدن به بیشترین سرعت، کمترین فرآیند و کمترین خطای حالت ماندگار می شود. طبق بررسی ها و مقایسات انجام شده در قسمت های قبل بهترین عملکرد را به ترتیب الگوریتم های *ICA*, *DE*, *Basic MA*, *Proposed MA*, *FA*, *BA*, *PSO* و *ICA* از خود نشان دادند. پس از اعمال کنترل کننده ها به فرآیندها، با اعمال ورودی پله واحد نتایج به صورت زیر بدست می آیند.



شکل (۴-۸). پاسخ پله فرآیند کنترل نشده I و کنترل شده با استفاده از کنترل کننده های بهینه شده توسط الگوریتمها

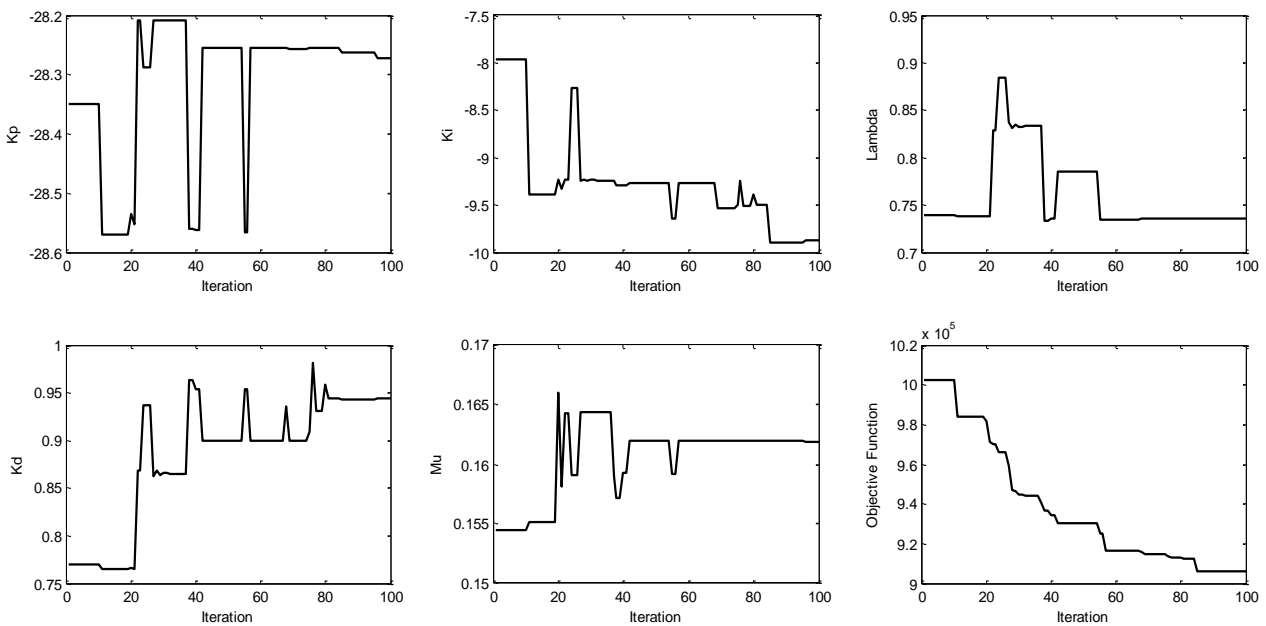


شکل (۴-۹). پاسخ پله فرآیند کنترل نشده II و کنترل شده با استفاده از کنترل کننده های بهینه شده توسط الگوریتمها

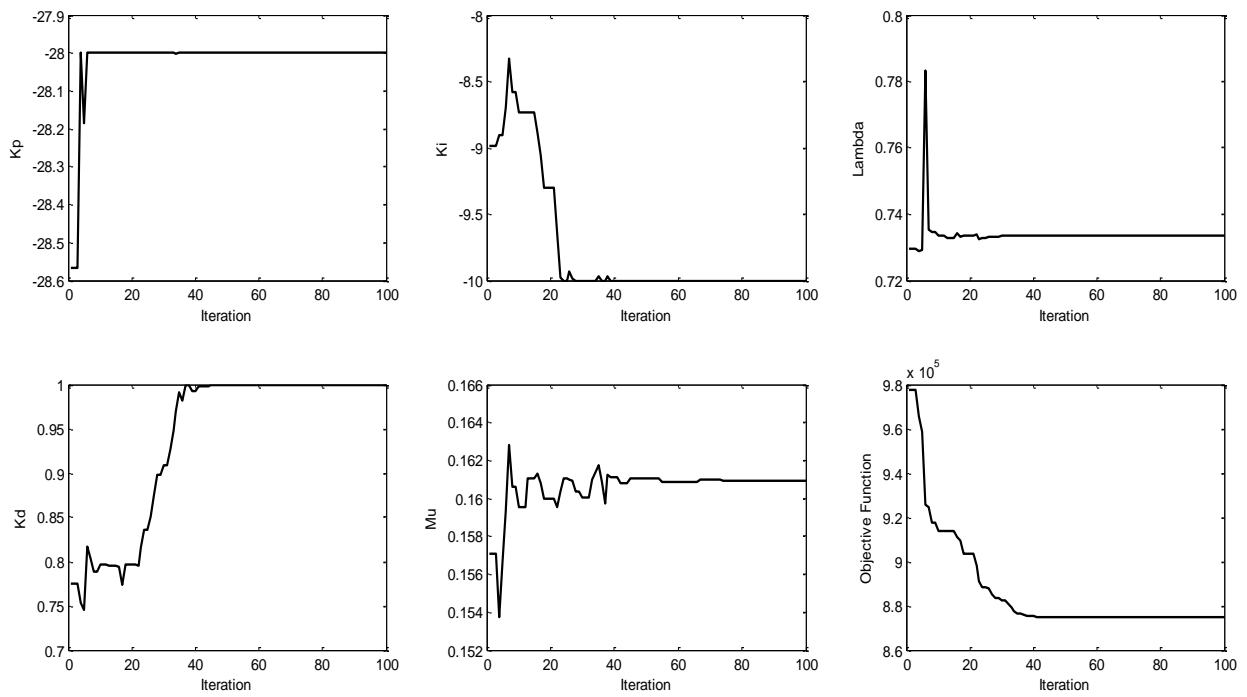


شکل (۴-۱۰). پاسخ پله فرآیند کنترل نشده III و کنترل شده با استفاده از کنترل کننده‌های بهینه شده توسط الگوریتم‌ها

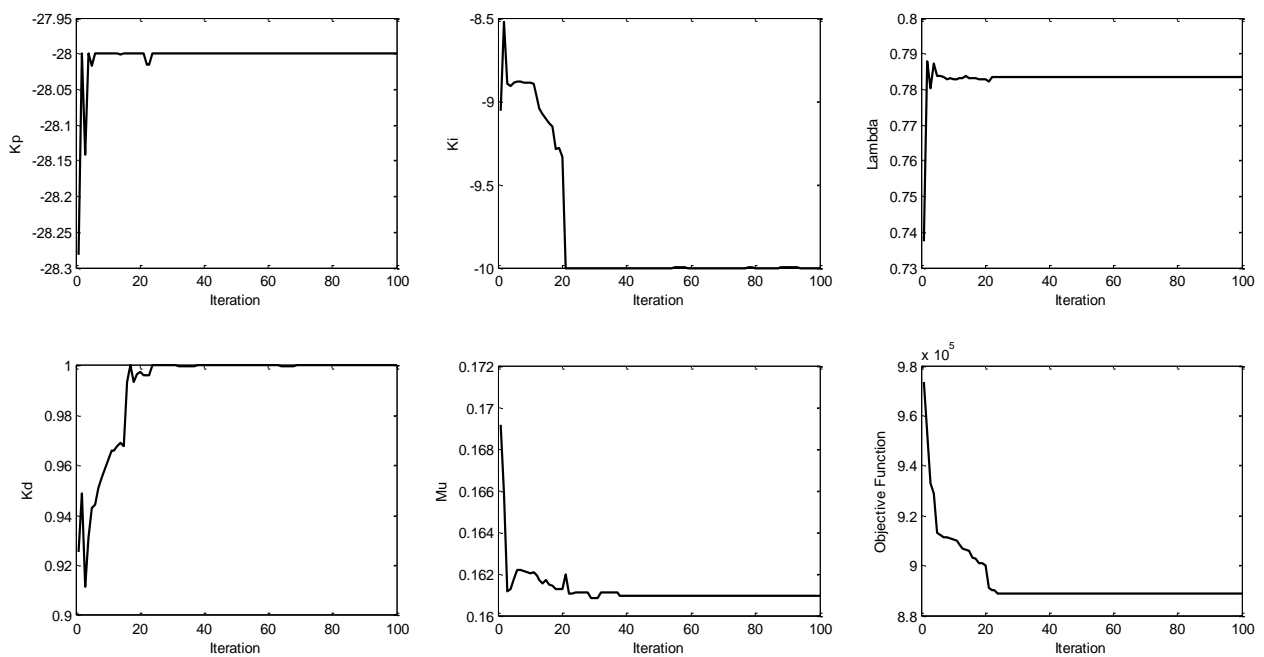
همانطور که در جدول (۴-۱۴) ذکر شده بود، در شکل‌های (۴-۸) تا (۴-۱۰) مشاهده می‌شود که الگوریتم ممتیک پیشنهادی مطلوب‌ترین پاسخ را ارائه می‌کند. روند تغییر پارامترهای طراحی توسط الگوریتم‌ها برای ۱۰۰ مرحله اجرا، جهت طراحی کنترل کننده‌ی بهینه برای فرآیند II، در شکل‌های (۴-۱۱) تا (۴-۱۷) مشاهده می‌شود. لازم به ذکر است که شکل‌های (۴-۱۱) تا (۴-۱۷) صرفاً جهت نمایش نحوه عملکرد الگوریتم‌ها آورده شده‌اند، و نمایانگر بهینه‌ترین حالت نمی‌باشند.



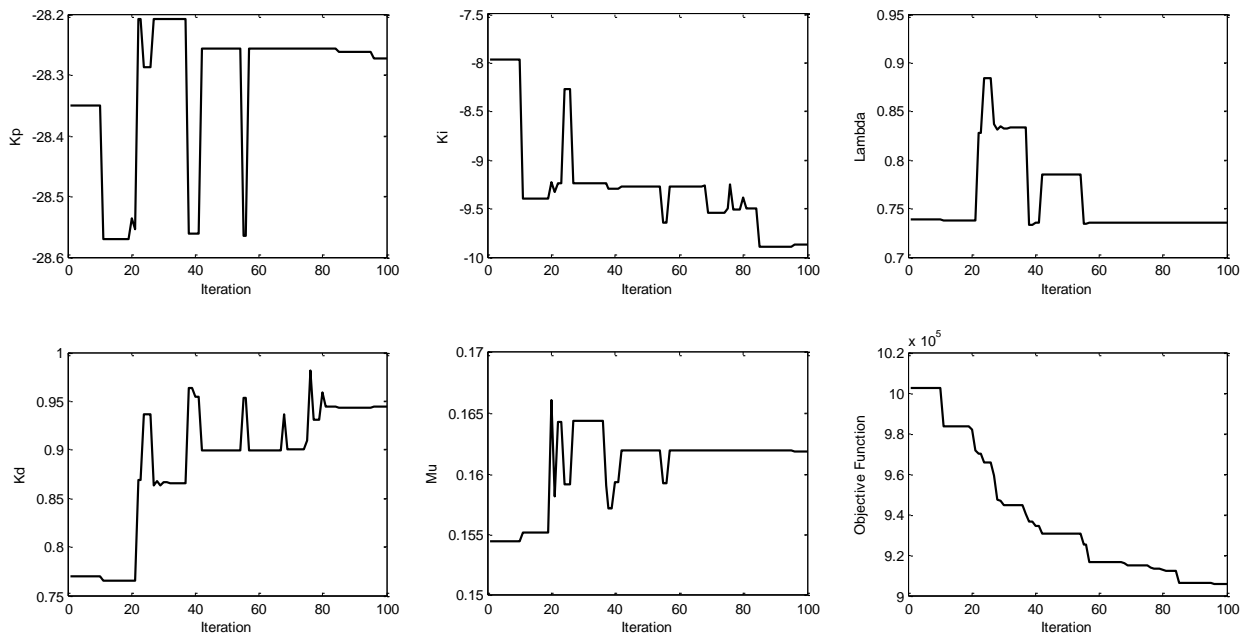
شکل (۴-۱۱). روند تغییر پارامترهای طراحی توسط الگوریتم ممتیک پیشنهادی



شکل (۴-۱۲). روند تغییر پارامترهای طراحی توسط الگوریتم بهینه‌سازی انبوه ذرات

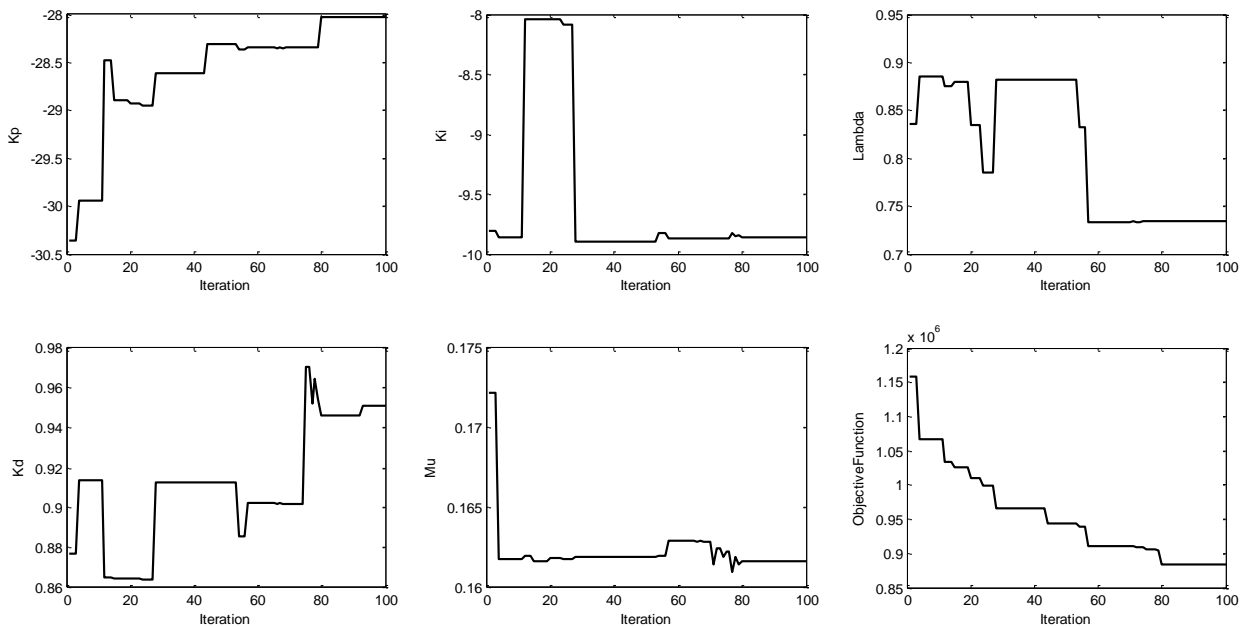


شکل (۴-۱۳). روند تغییر پارامترهای طراحی توسط الگوریتم کرم شب تاب

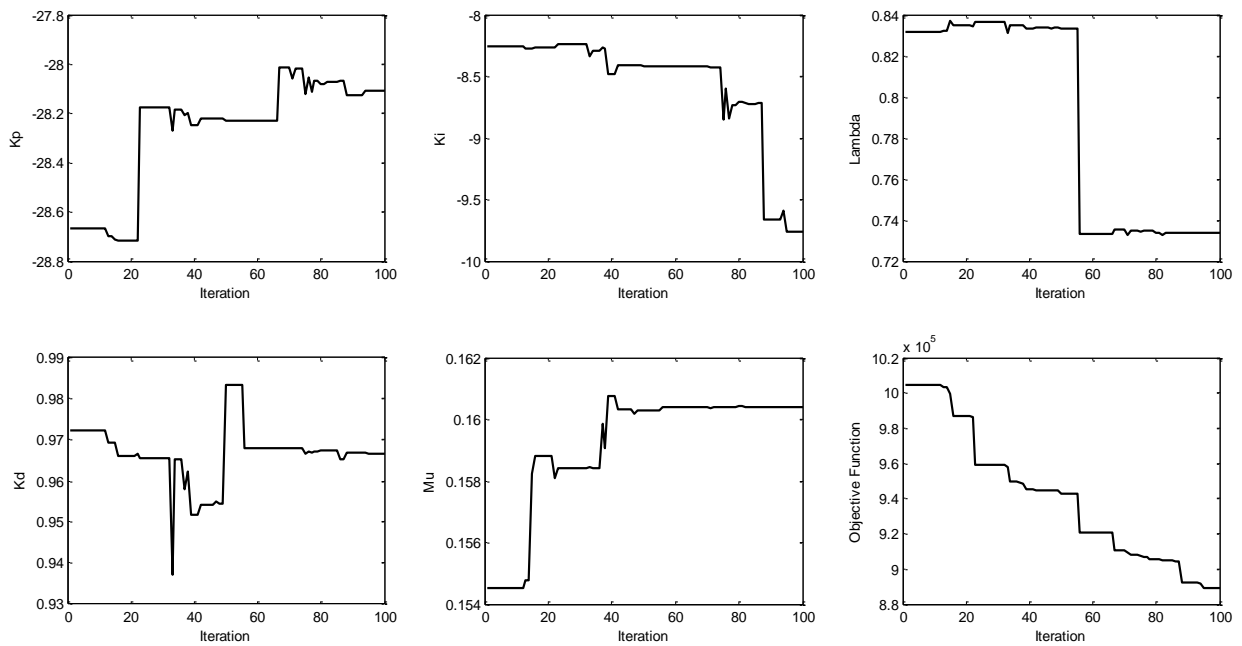


شکل (۴-۱۴). روند تغییر پارامترهای طراحی توسط الگوریتم رقابت استعماری

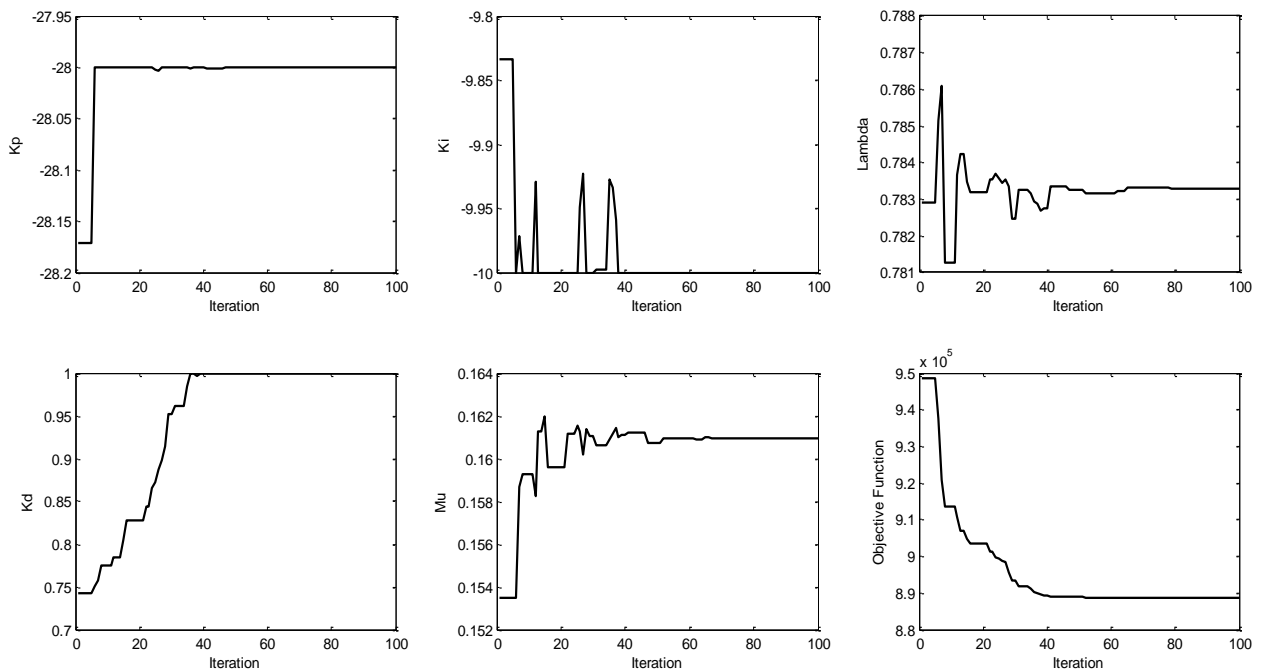
نحوه تغییر پارامترهای طراحی الگوریتم های بهینه سازی جستجوی سراسری و ممیک در شکل- های فوق مشاهده می شود. در برخی از الگوریتم ها با همگرایی زودرس پارامترها و در برخی الگوریتم ها همگرایی مناسب صورت گرفته است.



شکل (۴-۱۵). روند تغییر پارامترهای طراحی توسط الگوریتم ممیک پایه



شکل (۴-۱۶). روند تغییر پارامترهای طراحی توسط الگوریتم تکامل تفاضلی



شکل (۴-۱۷). روند تغییر پارامترهای طراحی توسط الگوریتم زنبورها

همانطور که مشاهده شد، بهترین کنترل کننده برای هر سه فرآیند تحت کنترل، توسط الگوریتم ممیک پیشنهادی طراحی شد. همانند بخش ۴-۲، الگوریتم ممیک پیشنهادی با بهره‌گیری از جستجوی محلی با سرعت مرتبه کسری، در یافتن پارامترهای بهینه کنترل کننده‌ها عملکرد مناسبی

از خود نشان می‌دهد. استفاده از سرعت مرتبه‌کسری در بخش جستجوی محلی الگوریتم پیشنهادی، باعث حفظ حافظه‌ی موقعیتی طولانی‌تر ذرات و در نتیجه دستیابی به پاسخ‌های دقیق‌تر می‌شود. لذا به‌واسطه‌ی بهره‌گیری از جستجوی محلی انبوه ذرات با سرعت مرتبه‌کسری در الگوریتم پیشنهادی، به سریع‌ترین پاسخ، کمترین فراجهدش و کمترین میزان خطای حالت ماندگار دست یافتیم.

فصل پنجم :

نتیجه گیری و پیشنهادات

۱-۵ نتیجه گیری

در این پایان نامه چند الگوریتم از الگوریتم‌های بهینه‌سازی تکاملی مورد بررسی و تحقیق قرار گرفته‌اند. یک الگوریتم ممتیک با جستجوی محلی انبوه ذرات و با سرعت مرتبه کسری جهت بهینه‌سازی پیشنهاد شده است و در ادامه روند پایان نامه از آن برای طراحی کنترل کننده FOPID بهینه استفاده شده است. در مقایسه‌ای که میان عملکرد الگوریتم ممتیک پیشنهادی و پایه در بهینه‌سازی چند تابع معیار ریاضی صورت گرفته شد، الگوریتم ممتیک پیشنهادی عملکرد بسیار مناسبی از خود نشان داد. چند روش از روش‌های طراحی کنترل کننده‌های مرتبه کسری مورد بررسی قرار گرفته شده، و با طراحی کنترل کننده‌های FOPID و PID مختلف برای یک فرآیند مرتبه کسری عملکرد مطلوب‌تر کنترل کننده‌های FOPID نسبت به PID نشان داده شده است. همچنین در مقایسه‌ای که میان کنترل کننده‌های FOPID مرسوم و کنترل کننده FOPID بهینه طراحی شده، صورت گرفته است، توانایی کنترل کننده FOPID بهینه به خوبی نشان داده شده است. در ادامه روند پایان نامه، مقایسه‌ای میان الگوریتم‌های بهینه‌سازی انبوه ذرات، کرم شبتاب، تکامل تفاضلی، زنبورها، رقابت استعماری، الگوریتم ممتیک پایه و الگوریتم ممتیک پیشنهادی، در طراحی کنترل کننده‌های FOPID بهینه جهت کنترل فرآیندهای مرتبه کسری، مرتبه صحیح و تاخیردار صورت گرفته است. نتایج شبیه‌سازی‌ها، نشان از برتری الگوریتم ممتیک پیشنهادی نسبت به دیگر الگوریتم‌های تکاملی دارد. اهم نتایج بدست آمده از این پایان نامه را می‌توان به صورت زیر بیان کرد:

- معرفی یک الگوریتم بهینه‌سازی جدید و کارآمد جهت بهینه‌سازی مسائل بهینه‌سازی مختلف.
- برتری بلا مانع کنترل کننده‌های FOPID نسبت به کنترل کننده‌های PID.
- برتری کنترل کننده FOPID بهینه استفاده شده نسبت به دیگر کنترل کننده‌های FOPID.
- برتری الگوریتم ممتیک پیشنهادی نسبت به دیگر الگوریتم‌های تکاملی مرسوم.

- توانایی کنترل‌کننده‌های FOPID در کنترل انواع فرآیندهای مرتبه‌کسری، مرتبه‌صحیح و تاخیردار.

۲-۵ پیشنهادات

سیستم‌های مرتبه‌کسری و کنترل‌کننده‌های مرتبه‌کسری از موضوعات جدید و بروز دنیای علم می‌باشند، که به دلیل توانایی بسیار بالای آنها در مدل‌سازی و کنترل انواع فرآیندهای صنعتی و غیر-صنعتی مورد توجه بسیاری از محققان و اندیشمندان قرار گرفته‌اند. الگوریتم‌های تکاملی نیز جایگاه ویژه‌ای در بهبود عملکرد سیستم‌ها و کنترل‌کننده‌ها به خود اختصاص داده‌اند. در راستای کمک به دیگر محققین پیشنهادات زیر ارائه می‌گردد.

- استفاده از مفهوم مشتقات مرتبه‌کسری در الگوریتم کرم شب‌تاب و بهبود عملکرد آن.
- استفاده از مفهوم مشتقات مرتبه‌کسری در طراحی کنترل‌کننده‌های PID فازی.
- استفاده از مفهوم مشتقات مرتبه‌کسری در بالا بردن دقت آموزش شبکه‌های عصبی.
- ترکیب الگوریتم زنبورها (به دلیل عملکرد بسیار خوب آن در جستجوی سراسری) با یک بخش جستجوی محلی (که از مفهوم مشتقات مرتبه‌کسری استفاده می‌کند).

1. Xue D. and Chen Y. (2007), "Linear feedback control, Analysis and design with MATLAB", Philadelphia.
2. Podlubny I. (1999), "Fractional differential equations", San Diego: Academic Press.
3. Oldham K. B. and Spanier J. (1974), "The Fractional Calculus", New York: Academic Press.
4. Miller K. S. and Ross B. (1993), "An Introduction to the Fractional Calculus and Fractional Differential Equations", New York: Wiley.
5. Podlubny I., Dorcak L. and Kostial I. (1997), "On fractional derivatives, fractional order dynamic systems and $PI^\lambda D^\mu$ -controller" in Proceedings of the 36th IEEE Dec. Cont. Conf. (DCC), pp. 4985-4990.
6. Monje A., Chen Y., Vinagre M., Dingyü X. and Feliu V. (2010), "Fractional order systems and controls fundamentals and applications", Springer.
7. Podlubny I. (1997), "The Laplace transform method for linear differential equations of the fractional order", in Proc. of the 9th International BERG Conference, Kosice, Slovak Republic.
8. Petras I. and Dorcak L. (1999), "The frequency method for stability investigation of fractional control systems" SACTA journal, vol. 2, no. 1-2, pp. 75-85.
9. Malti R. (2013), "A note on \mathcal{L}_p -norms of fractional systems", Automatica, Vol. 49, Iss. 9, pp. 2923-2927.
10. Pedro Mata D. and Valério O. (2005), "Fractional Robust System Control", Lisboa.
11. Gorges-Schleuter M. (1989), "An Asynchronous Parallel Genetic Optimization Strategy" in Proceedings of the Third International Conference on Genetic Algorithms, pp. 422-427, Mogan.
12. Brits R. (2002), "Niche Strategies for Particle Swarm Optimization", Pretoria.
13. Van der Bergh F. (2001), "An Analysis of Particle Swarm Optimizers", Pretoria.
14. Derelia T. and Dasb G. S. (2011), "A Hybrid Bees Algorithm for Solving Container Loading Problems", Applied Soft Computing, pp. 2854-2862.
15. Mani V., Senthilnath J. and Omkar S. N. (2011), "Swarm and Evolutionary Computation", pp. 164-171.
16. Goldberg D. E. (1999), "Optimizing Global-Local Hybrids", in GECCO, Vol. 1, pp. 220-228, San Francisco.
17. Denzinger J. (1999), "On Cooperation Between Evolutionary Algorithms and Other Search Paradigms", IEEE Press, pp. 2317-2324.
18. Birru H. K. (1998), "Effectiveness of Local Search Operators in Evolutionary Programming", Genetic Programming, pp. 753-761.
19. Yang X. S., Yang J. M., Alvarez J. R. (2005), "Engineering optimizations via nature-inspired virtual bee algorithms", Springer, Berlin.
20. Seeley T. D., Visscher P. K., Passino K. M. (2006), "Group decision making in honey bee swarms American Scientist, pp. 220-229.

21. Seeley T. D. (1995), "The Wisdom of the Hive", Harvard University Press, Cambridge.
22. Pham D. T., Koc E., Ghanbarzadeh A., Otri S. (2006), "Optimization of the weights of multi-layered perceptrons using the bees algorithm", in: Proceedings of the 5th International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey, pp. 38–46.
23. Nakrani S., Tovey C. (2007), From honeybees to Internet servers: biomimicry for distributed management of Internet hosting centers, *Bioinspiration and Biomimetic*, pp. 182–197.
24. Karaboga D., Basturk B. (2007), "A powerful and efficient algorithm for numerical function optimization: bee colony (BC) algorithm, *Journal of Global Optimization*, pp. 459–471.
25. Yang X.S. (2008), "Nature-Inspired Metaheuristic Algorithms", Luniver Press.
26. Tyler J. (2001), "Glow-worms", London.
27. Eberhart R. C., Kennedy J. (1995) , "A New Optimizer Called Particle Swarm", 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43.
28. Storn R. and Price K. (1997), "Differential Evolution—a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, pp. 341–359.
29. Atashpaz E., Lucas C. (2007), "Imperialist Competitive Algorithm: an algorithm for optimization inspired by imperialistic competition", in *Proceedings IEEE Congress on Evolutionary Computation*, pp. 4661–4667.
30. Radcliffe N. J., Sury P. D. (1994), "Formal Memetic Algorithms", University of Edinburg, Springer-Verlag.
31. Hodgson R. J. W. (2000), "Memetic Algorithm and the Molecular Geometry Optimization Problem", In *Proceeding of the 2000 Congress on Evolutionary Computation*, San Diego, pp. 625-632.
32. Back T. (1996), "Evolutionary Algorithms in Theory and Practice", Oxford University Press, Oxford, UK.
33. Krasnogor N., Smith J. (2001), "Emergence of profitable search strategies based on a simple inheritance mechanism", in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, Morgan Kaufmann, San Francisco, pp. 432-439.
34. Fogel L., Owens A., Walsh M. (1965), "Artificial intelligence through a simulation of evolution", in *Proceedings of Biophysics and Cybernetic Systems*, Spartan, Washington DC, pp. 131-156.
35. Moscato P. (1989), "On evolution, search, optimization, genetic algorithms and martial arts towards memetic algorithms", Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Caltech, Pasadena, California.
36. Solteiro Pires E. J, Tenreiro Machado J. A., de Moura Oliveira P.B., Boaventura Cunha J. (2010), "Particle swarm optimization with fractional-order velocity", *Nonlinear Dynamics*, Springer Science Business Media, pp. 295-301.

37. Podlubny I. (1994), "Fractional-order systems and fractional-order controllers", in UEF-03-94, The Academy of Sciences Institute of Experimental Physics, Kosice, Slovak Republic.
38. Oustaloup A. (1991), "La commande CRONE: commande robuste d'ordre non entiere", Herme`s, Paris.
39. Carlson G. (1964), "Approximation of Fractional Capacitors $(1/s)^{(1/n)}$ by a Regular Newton Process", Circuit theory, IEEE Transactions, pp. 210-213.
40. Liu J. (2012), "An interior point method based on continuous Newton's method for optimal power flow", Innovative smart grid technologies, IEEE, pp. 1-6.
41. Matsuda G. (1994), "Fractional Processes and Fractional-Order Signal Processing", pp. 134-141.
42. Vinagre B., Podlubny I., Hernandez A. and Feliu V. (2000), "Some approximations of fractional order operators used in control theory and applications", Fractional Calculus Application Analysis, vol. 3, pp. 231-248.
43. Petras I., Dorcak L., O'Leary P. and Podlubny I. (2000), "The modelling and analysis of fractional-order control systems in frequency domain" in Proceedings of ICCS, High Tatras, Slovak Republic, pp. 261 - 264.
44. Podlubny I. (1999), "Fractional order system and $PI^\lambda D^\mu$ -controllers," IEEE Trans. Auto. Cont., vol. 44, no. 1, pp. 208-214.
45. Petras I. (1999), "The fractional-order controllers: Methods for their synthesis and application", Journal of Electrical Engineering, vol. 50, no. 9-10, pp. 284-288.
46. Ogata K. (1997), "Modern Control Engineering", Prentice Hall: New Jersey.
47. Landau D. (1995), "Digital Control Systems", Seattle, USA.
48. Oustaloup A. (1995), "Numerical Solutions of fractional systems", Chicago, Illinois.
49. Häggglund T. and Åström K. (1996), "Automatic tuning of PID controllers", W.S. Levine (Ed.), The Control Handbook, CRC Press, Boca Raton.
50. Monje C. A., Vinagre B. M., Chen Y. Q., Feliu V., Lanusse P. and Sabatier J. (2004), "Proposals for fractional $PI^\lambda D^\mu$ tuning", Fractional Differentiation and its Applications, Bordeaux.
51. Monje C. A., Vinagre B. M., Chen Y. Q. and Feliu V. (2003), "On fractional PID controllers: some tuning rules for robustness to plant uncertainties", Kluwer Academic Publishers, printed in the Netherlands, pp. 1-18.
52. Zhao C., Xue D. and Chen Y. Q. (2005), "A fractional order PID tuning algorithm for a class of fractional order plants", in Proce. of the IEEE Inter. Conf. on Mech. & Auto., pp. 216-221, Canada.
53. Monje C. A., Vinagre B. M., Feliu V. and Chen Y. Q. (2006), "On auto-tuning of fractional order $PI D^\mu$ controllers", in Proceedings of the 2nd IFAC Workshop on Fractional Differentiation and its Applications, Porto, Portugal.
54. Valerio D. and Costa J. (2004), "Tuning rules for fractional PID controllers", Signal Processing Journal, Elsevier.
55. Ziegler J.G. and Nichols N. B. (1942), "Optimum settings for automatic controllers", Transactions of the ASME, pp. 759-768.
56. Cohen H. and Coon J. (1965), "Performance and gain and phase margins of well-known PID tuning formulas", IEEE Transactions, pp. 624-231.

57. Kappa L. and Tau H. (1968), "PI tuning rules for specified robust levels", IEEE Transactions, pp. 331-342.
58. Karad S. and Chatterji S. (2012), "Performance Analysis of Fractional Order PID Controller with the Conventional PID Controller for Bio-Reactor Control", International Journal of Scientific and Engineering Research, Vol. 3, Iss. 6.
59. Gao W. and Liu S. (2011), "Improved Artificial Bee Colony Algorithm for Global Optimization", Information Processing Letters, pp. 871-882.

Abstract:

Nowadays control of the fractional order systems has attracted considerable attentions due to their applications and the existence of uncertainties. The most popular method to implement and simulate fractional order transfer functions is to approximate them with integer order transfer functions, which is presented in this work. Some methods in designing fractional order PID controllers are presented then. This particular kind of controller is taken into attention. In this dissertation the method, tuning using evolutionary algorithms, is used to design fractional order PID controllers to control fractional order, integer order and time delayed plants. A novel Memetic Algorithm (MA) based on Particle Swarm Optimization (PSO) and by use of fractional order differentiations concept is proposed by then. To show the performance and efficiency of the proposed MA and the designing method used, a comparison between is presented.

Key Words:

Fractional calculus, Fractional order controllers, Fractional order PID controllers, Evolutionary algorithms, Memetic algorithm.



Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

**An Optimal Fractional-Order PID Controller Design Using
Memetic Algorithm**

Seyed Yashar Mousavi

Supervisor:

Dr Alireza Alfi

Date: 2014 September