

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی برق و رباتیک

پایان نامه کارشناسی ارشد مهندسی سیستم های الکترونیک دیجیتال

طراحی فیلتر FRM میان گذر و پیاده سازی آن روی FPGA

نگارنده: مهسا مقیمی

استاد راهنما

دکتر علی سلیمانی ایوری

شهریور ۱۳۹۵

تقدیم به آنکه دانش از اوست

تقدیم به آنان که در راه دانش و معرفت جان فرسودند

و

تقدیم به همسر م که در تمامی مراحل پشتیبان و یاور من بود

سپاسگذاری

منت خدای را عزوجل که طاعتش موجب قرمت و بشکر اندرش مزین نعمت. هر نفسی که فرو میرود مهر حیات است و چون بر میاید

منفح ذات. پس در هر نفسی دو نعمت موجود است و بر هر نعمتی شکر واجب

از دست و زبان که بر آید

کز عمده شکرش بر آید

از زحمات بی دریغ استاد ارجمند جناب آقای دکتر علی سلیمانی ایوری کمال تشکر و قدردانی را دارم که رهنمودهای جنابشان موجب به هدف رسیدن این تحقیق بوده است.

از تمامی اساتید گروه الکترونیک دانشگاه صنعتی شاهرود نیز برای زحمات ایشان در طول دوره تحصیل سپاسگذاری کرده و تلاش ایشان را برای افزایش علم و فرهیختگی در جامعه ارج می نهیم.

و نیز جا دارد از حمایتها و همراهیهای دلسوزانه خانواده ام که در طول مدت انجام پروژه کاستیهای مرا تحمل نمودند، تشکر و قدردانی کنم.

مهسا مقیمی

شهریور ۱۳۹۵

تعهدنامه

اینجانب مهسا مقیمی دانشجوی دوره کارشناسی ارشد رشته مهندسی سیستم های الکترونیک دیجیتال دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه طراحی فیلتر FRM میان گذر و پیاده سازی آن روی FPGA تحت راهنمایی دکتر علی سلیمانی ایوری متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام «دانشگاه صنعتی شاهرود» و یا «**Shahrood University of Technology**» به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

فیلترهای دیجیتال با پاسخ ضربه با طول محدود از جمله سیستم‌های زمان‌گسسته هستند که کاربردهای فراوانی در پردازش گفتار، پردازش تصویر، سیستم‌های کنترلی و ... دارند. از مشخصه‌های مهم این فیلترها تضمین پایداری و داشتن فاز خطی می‌باشد. در طراحی اینگونه فیلترها با چالش طراحی و پیاده‌سازی سخت‌افزاری فیلترها با تعداد ضرایب زیادتر مواجه می‌شویم. زیرا افزایش تعداد ضرایب فیلتر پیچیدگی و اشغال حافظه‌ی بیشتر را به همراه دارد. از دیگر مشکلات طراحی فیلترهای دیجیتال مشکل سطح خطای چند وجهی و کمینه‌های محلی می‌باشد.

در این پایان‌نامه طراحی فیلتر میان‌گذر با باند گذر 0.1π که از روش مستقیم و توسط الگوریتم تکاملی قادر به طراحی آن نبوده ایم با تکنیک ماسک‌گذاری روی پاسخ فرکانسی^۱ (FRM) محقق گردید. با استفاده از الگوریتم‌های تکاملی می‌توان بر چالش سطح خطای چند وجهی و کمینه‌های محلی چیره شد. در این تحقیق با ارائه روشی مبتنی بر ترکیب دو الگوریتم تکاملی بهینه‌سازی توده ذرات^۲ (PSO) و الگوریتم فرهنگی (Cultural Algorithm) طراحی فیلتر صورت می‌پذیرد که از مزایای این روش می‌توان به تضعیف ۱۰٪ بیشتر در باند توقف و همگرایی سریع‌تر الگوریتم اشاره کرد. در ادامه با استفاده از نرم-افزار system generator امکان پیاده‌سازی سخت‌افزاری این فیلتر بر روی FPGA^۳ محقق گردید.

واژه‌های کلیدی:

فیلتر FIR، فیلتر FRM، الگوریتم تکاملی بهینه‌سازی توده ذرات، الگوریتم فرهنگی، FPGA

¹ Frequency Response Masking

² Particle Swarm Optimization

³ Field Programmable Gate Array

۱ فصل اول مقدمه	۱
۱.۱ پیشگفتار	۲
۲.۱ سابقه پژوهش	۳
۱.۲.۱ تکنیک SFFM-FRM	۳
۲.۲.۱ استفاده از ساختار پیش فیلتر متعادل کننده	۵
۳.۲.۱ الگوریتم ژنتیک	۶
۴.۲.۱ طراحی فیلتر توسط روش مجموع مربعات وزن دار شده (WLS)	۷
۵.۲.۱ طراحی فیلتر با استفاده از الگوریتم لانه ی زنبوری	۷
۶.۲.۱ کاهش حجم حافظه برای پیاده سازی یک فیلتر FIR توسط الگوریتم CSD	۹
۳.۱ ساختار پایان نامه	۱۱
۲ فصل دوم تکنیک های طراحی فیلتر	۱۳
۱.۲ انواع فیلتر	۱۴
۱.۱.۲ فیلتر آنالوگ	۱۴
۲.۱.۲ فیلتر دیجیتال	۱۵
۱.۲.۱.۲ فیلترهای دیجیتال با پاسخ ضربه طول محدود (FIR)	۱۵
۲.۲.۱.۲ فیلترهای دیجیتال با پاسخ ضربه طول نا محدود (IIR)	۱۶
۲.۲ روشهای طراحی فیلتر	۱۶
۱.۲.۲ روشهای نظری طراحی فیلترهای IIR زمان گسسته به کمک فیلترهای زمان پیوسته	۱۷
۱.۱.۲.۲ طراحی فیلتر به وسیله تغییر ناپذیری ضربه	۱۸
۲.۱.۲.۲ تبدیل دو خطی	۲۰
۲.۲.۲ طراحی فیلتر FIR توسط پنجره گذاری	۲۱
۳.۲.۲ الگوریتم های تکاملی	۲۴
۴.۲.۲ تکنیک ماسک گذاری روی پاسخ فرکانسی	۲۴
۱.۴.۲.۲ مقدمات ریاضی فیلتر FRM	۲۵
۲.۴.۲.۲ فیلتر FRM با پهنای باند دلخواه	۲۶
۳.۴.۲.۲ فیلتر FRM با ساختار چند طبقه:	۳۰
۳ فصل سوم الگوریتم های تکاملی	۳۳
۱.۳ الگوریتم بهینه سازی توده ذرات	۳۴
۱.۱.۳ مزایای الگوریتمهای PSO	۳۵
۱.۱.۱.۳ بهره مندی از حافظه	۳۵
۲.۱.۱.۳ همکاری و اشتراک گذاری اطلاعات بین ذرات	۳۵
۳.۱.۱.۳ سرعت همگرایی بالا	۳۵
۴.۱.۱.۳ راحتی پیاده سازی و اجرا	۳۵
۲.۱.۳ معایب الگوریتم PSO	۳۶

۳۶	همگرایی سریع و گرفتار شدن در بهینه محلی.....	۱.۲.۱.۳
۳۶	کاهش تنوع جمعیت.....	۲.۲.۱.۳
۳۷	بروزرسانی موقعیت ذرات در الگوریتم PSO.....	۳.۱.۳
۳۹	تابع شایستگی (برازندگی).....	۴.۱.۳
۴۰	تابع خطا (تابع هدف) در طراحی فیلتر.....	۵.۱.۳
۴۱	الگوریتم فرهنگی.....	۲.۳
۴۲	فضای جمعیت.....	۱.۲.۳
۴۲	فضای باور.....	۲.۲.۳
۴۳	دانش موقعیتی.....	۱.۲.۲.۳
۴۵	استفاده از دانش معیار هم برای تعیین جهت و هم برای میزان حرکت.....	۳.۴.۲.۳
۴۷	فصل چهارم الگوریتم پیشنهادی.....	۴
۴۸	ترکیب الگوریتم های بهینه سازی توده ذرات و فرهنگی.....	۱.۴
۵۰	الگوریتم پیشنهادی.....	۲.۴
۵۲	تعیین مشخصات فیلتر میان گذر.....	۱.۲.۴
۵۲	تعیین مشخصات زیرفیلترها ساختار FRM.....	۲.۲.۴
۵۴	طراحی زیرفیلترها توسط الگوریتم تکاملی.....	۳.۲.۴
۵۴	طراحی فیلتر FIR توسط الگوریتم PSO.....	۱.۳.۲.۴
۵۵	طراحی فیلتر FIR توسط الگوریتم ترکیبی.....	۲.۳.۲.۴
۵۸	طراحی فیلتر FRM.....	۴.۲.۴
۵۸	پیاده سازی روی FPGA.....	۵.۲.۴
۶۱	فصل پنجم شبیه سازی.....	۵
۶۲	طراحی ضرایب فیلتر توسط الگوریتم های بهینه سازی.....	۱.۵
۶۲	تأثیر پهنای باندگذر در طراحی فیلتر.....	۱.۱.۵
۶۷	تأثیر مرتبهی فیلتر در طراحی فیلتر.....	۲.۱.۵
۶۷	مرتبه N=۳۰.....	۱.۲.۱.۵
۶۹	مرتبه N=۴۶.....	۲.۲.۱.۵
۷۰	مرتبه N=۶۰.....	۳.۲.۱.۵
۷۲	طراحی فیلتر باندگذر تیز FRM با الگوریتم های تکاملی.....	۲.۵
۷۶	شبیه سازی فیلتر میانگذر تیز FRM با FPGA.....	۳.۵
۸۱	فصل ششم جمع بندی و نتیجه گیری.....	۶
۸۳	منابع و مراجع.....	۸۳

فهرست شکل‌ها

صفحه

شکل (۱-۱) نمودار بلوکی تکنیک SFFM-FRM [۲]	۴
شکل (۲-۱) پاسخ فرکانسی تکنیک SFFM-FRM [۲]	۴
شکل (۳-۱) ساختار بلوکی پیش‌فیلتر متعادل کننده [۳]	۶
شکل (۴-۱) مقایسه ضرب کننده ۸ بیتی ساده و CSD [۱۱]	۱۰
شکل (۱-۲) پاسخ ضربه فیلتر FIR	۱۵
شکل (۲-۲) نمونه برداری از سیگنال پیوسته [۱۲]	۱۸
شکل (۳-۲) پاسخ فرکانسی $W(e^{j\omega})$ [۱۲]	۲۲
شکل (۴-۲) پاسخ فرکانسی پنجره‌های مورد استفاده رایج [۱۲]	۲۴
شکل (۵-۲) بلوک دیاگرام $F_a(z)$ و مکمل آن [۱۳]	۲۶
شکل (۶-۲) ساختار بلوکی تکنیک ماسکگذاری روی پاسخ فرکانسی [۱۴]	۲۶
شکل (۷-۲) بلوک دیاگرام $F_a(z^M)$ و مکمل آن [۱۴]	۲۷
شکل (۸-۲) فیلتر FRM حالت اول [۱۴]	۲۸
شکل (۹-۲) فیلتر FRM حالت دوم [۱۴]	۲۹
شکل (۱۰-۲) فیلتر FRM سه طبقه [۱۶]	۳۱
شکل (۱-۴) روندنما ترکیب الگوریتم PSO و فرهنگی	۵۰
شکل (۲-۴) فلوجارت الگوریتم ترکیبی	۵۱
شکل (۳-۴) بلوک دیاگرام فیلتر میانگذر FRM	۵۳
شکل (۴-۴) طراحی فیلتر با استفاده از الگوریتم PSO	۵۶
شکل (۵-۴) طراحی فیلتر با استفاده از الگوریتم ترکیبی	۵۷
شکل (۱-۵) پاسخ فرکانسی فیلتر f_1 بر حسب dB (پهنای باند گذر 0.08π)	۶۳
شکل (۲-۵) پاسخ فرکانسی فیلتر f_2 بر حسب dB (پهنای باند گذر 0.16π)	۶۳
شکل (۳-۵) پاسخ فرکانسی f_1	۶۴
شکل (۴-۵) پاسخ فرکانسی ناحیهی توقف f_1	۶۴
شکل (۵-۵) پاسخ فرکانسی f_2	۶۴
شکل (۶-۵) پاسخ فرکانسی باند توقف f_2	۶۵
شکل (۷-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_1	۶۶
شکل (۸-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_2	۶۶
شکل (۹-۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۳۰	۶۷
شکل (۱۰-۵) پاسخ فرکانسی باند توقف فیلتر f_1 مرتبه ۳۰	۶۷
شکل (۱۱-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۳۰	۶۸
شکل (۱۲-۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۴۶	۶۹
شکل (۱۳-۵) پاسخ فرکانسی باند توقف فیلتر f_1 مرتبه ۴۶	۶۹
شکل (۱۴-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۴۶	۶۹

- شکل (۵-۱۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۶۰ ۷۰
- شکل (۵-۱۶) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۶۰ ۷۱
- شکل (۵-۱۷) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۶۰ ۷۱
- شکل (۵-۱۸) پاسخ فرکانسی $F_a(z^8)$ ۷۳
- شکل (۵-۱۹) پاسخ فرکانسی F_{Mc} ۷۳
- شکل (۵-۲۰) پاسخ فرکانسی F_{Ma} ۷۳
- شکل (۵-۲۱) پاسخ فرکانسی فیلتر میان گذر FRM ۷۴
- شکل (۵-۲۲) پاسخ فرکانسی فیلتر میان گذر بدون استفاده از تکنیک FRM ۷۶
- شکل (۵-۲۳) نحوه پیاده‌سازی فیلتر میان گذر FRM با system generator ۷۷
- شکل (۵-۲۴) پاسخ فرکانسی فیلتر میان گذر FRM در FPGA ۷۸
- شکل (۵-۲۵) پاسخ فرکانسی فیلتر میان گذر FRM در simulink ۷۹
- شکل (۵-۲۶) تفاضل پاسخ فرکانسی فیلتر میان گذر FRM در FPGA و simulink ۷۹
- شکل (۵-۲۶) حجم حافظه اشغال شده فیلتر می‌انگذر FRM در FPGA ۸۰

۵	جدول (۱-۱) نتایج شبیه سازی تکنیک SFFM-FRM [۲].....
۸	جدول (۲-۱) مقدار خطا فیلتر پایین گذر با روش لانه زنبوری [۹].....
۵۴	جدول (۱-۴) مشخصات زیرفیلترهای FRM میان گذر مطلوب.....
۶۳	جدول (۱-۵) ماکزیمم مقدار پاسخ فرکانسی فیلتر در باند توقف.....
۶۵	جدول (۲-۵) حداکثر خطا باند توقف f_1 و f_2
۶۶	جدول (۳-۵) کمترین خطا.....
۶۸	جدول (۴-۵) فیلتر f_1 مرتبه ۳۰.....
۷۰	جدول (۵-۵) فیلتر f_1 مرتبه ۴۶.....
۷۱	جدول (۶-۵) فیلتر f_1 مرتبه ۶۰.....
۷۲	جدول (۷-۵) زیرفیلترهای فیلتر میانگذر.....
۷۴	جدول (۸-۵) طراحی فیلتر میانگذر FRM توسط الگوریتم PSO.....
۷۵	جدول (۹-۵) طراحی فیلتر میانگذر FRM توسط الگوریتم WPSO.....
۷۵	جدول (۱۰-۵) طراحی فیلتر میانگذر FRM توسط الگوریتم CAPSO.....
۷۸	جدول (۱۱-۵) بیشترین و کمترین مقدار باند عبور و توقف در FPGA.....
۷۹	جدول (۱۲-۵) بیشترین و کمترین مقدار باند عبور و توقف در simulink.....
۸۰	جدول (۱۳-۵) کمترین و بیشترین مقدار تفاضل پاسخ فرکانسی FPGA و simulink.....

فصل اول

مقدمه

۱.۱ پیش‌گفتار

امروزه پردازش سیگنال‌های دیجیتال از پرکاربردترین و سودمندترین علوم می‌باشد زیرا سیگنال‌های دیجیتال در صنعت با حجم وسیعی مورد استفاده قرار می‌گیرد. امکان آزمون و شبیه‌سازی در این فرآیند بسیار کلیدی است و از آنجایی که سرعت، دقت و کارایی برای ما اهمیت دارد ارائه روشی که شبیه‌سازی را با وضوح و سرعت بیشتری برای ما فراهم آورد می‌تواند بسیار مفید باشد و هزینه‌ها را کاهش دهد.

هر چقدر پهنای باندگذر فیلتر کوچک‌تر باشد، فیلتر قابلیت تفکیک‌پذیری بهتری بین باندگذر و باندتوقف پیدا می‌کند ولی در عوض فیلتر نیازمند تعداد ضرایب بیشتری جهت پیاده‌سازی خواهد بود که این امر باعث افزایش تعداد عملیات ضرب می‌گردد و کاهش سرعت را در پی دارد. محققان همیشه به دنبال روش‌هایی برای افزایش دقت و کاهش عملیات محاسباتی هستند. تکنیک ماسک‌گذاری روی پاسخ فرکانسی به لحاظ محاسباتی روش کارآمدی برای طراحی فیلترهای FIR^1 با فاز خطی و باند انتقال بسیار باریک می‌باشد. در این تکنیک به کمک فیلتر تیز لبه‌باند^۲ پهنای باند انتقال را کاهش می‌دهد و به کمک فیلترهای مکمل تأخیردهنده تحقق فیلتر با پهنای باند دلخواه را ممکن می‌سازد. با این روش می‌توان فیلتری با ناحیه‌ی گذر باریک و تعداد ضرایب کم‌تر طراحی کرد.

در این پایان‌نامه ابتدا از سه الگوریتم تکاملی در طراحی فیلتر پایین‌گذر استفاده کرده‌ایم. مزایا و معایب هر کدام از آنها را با یکدیگر مقایسه می‌کنیم و در ادامه به تأثیر مرتبه‌ی فیلتر و پهنای باندگذر بر روی پارامترهایی همچون اندازه‌ی فیلتر در باند توقف و حداکثر خطای باند توقف (δ_p) پرداخته می‌شود. در مرحله‌ی بعد ضرایب یک فیلتر میان‌گذر FRM^3 را به وسیله‌ی الگوریتم‌های تکاملی بدست

¹ Finite Impulsive Responsive

² band-edge

³ Frequency Response Masking

می‌آوریم. در نهایت با نرم افزار system generator که متعلق به شرکت Xilinx است فیلتر میان‌گذر FRM را بر روی FPGA پیاده‌سازی می‌کنیم. بلوک‌های system generator باعث شده‌است که پیاده‌سازی مدل به صورت سخت‌افزاری بر روی FPGA به سادگی امکان پذیر باشد.

۲.۱ سابقه پژوهش

فعالیت‌هایی که در این پایان‌نامه صورت گرفته است را می‌توان به صورت عمده در سه زمینه مورد بررسی قرار داد:

۱- بررسی در حوزه‌ی تکنیک ماسک‌گذاری روی پاسخ فرکانسی

۲- بررسی الگوریتم تکاملی در طراحی فیلتر FIR

۳- پیاده‌سازی فیلتر FRM توسط نرم‌افزار system generator

در این بخش به بررسی کارهایی که دیگر محققان در هر یک از زمینه‌ها انجام داده‌اند، خواهیم پرداخت.

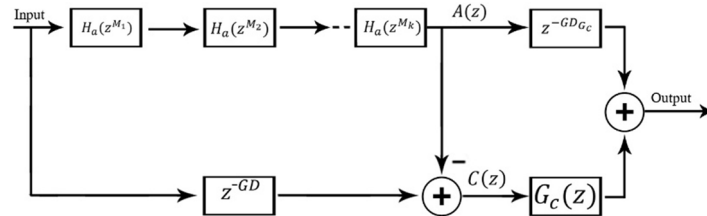
۱.۲.۱ تکنیک^۴ SFFM-FRM

با استفاده از تکنیک^۵ SFFM می‌توان به جای استفاده از فیلتر ماسک‌گذار از خود فیلتر برای حذف ناحیه‌ی عبور نامطلوب استفاده کرد [۱]. ساختار SFFM-FRM توسط Chun, Yong Lian, Zhu Yang در سال ۲۰۰۶ مطرح گردید [۲]. در ساختار اصلی تکنیک FRM که در فصل آینده به تفصیل در مورد آن صحبت خواهد شد، از دو فیلتر ماسک‌گذار برای طراحی فیلتر مطلوب استفاده

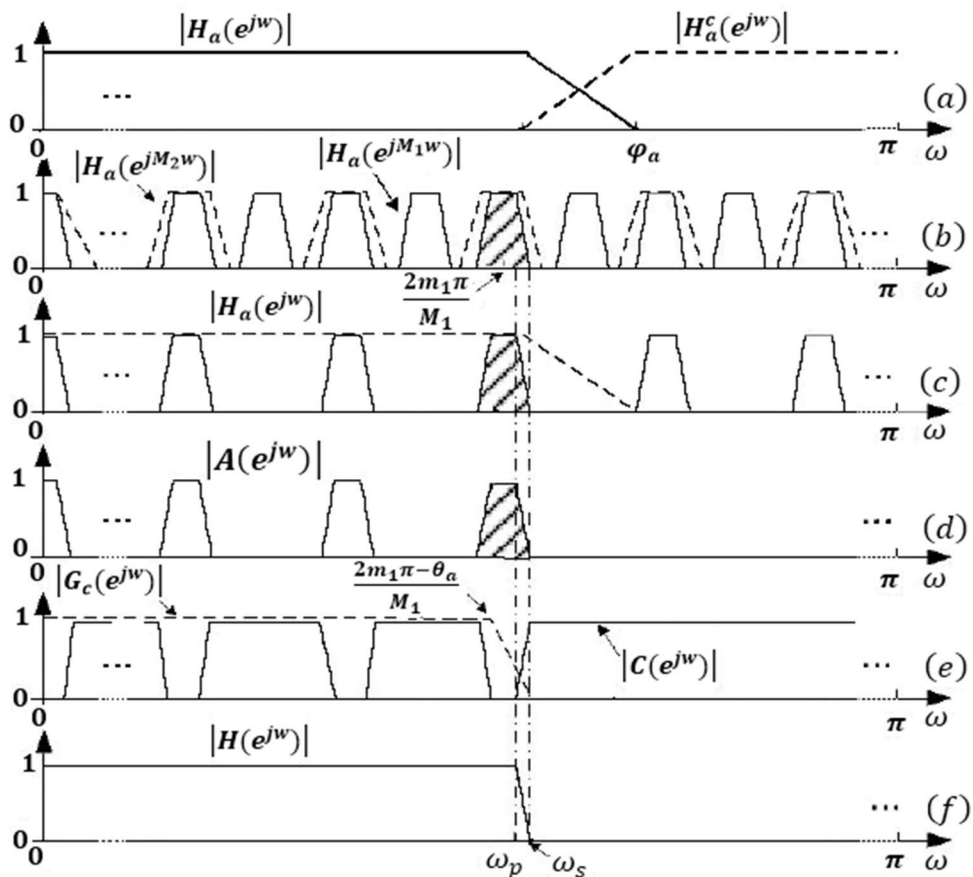
^۴ Single Filter Frequency Masking-Frequency Response Masking

^۵ Single Filter Frequency Masking

می‌گردد. مزیت تکنیک SFFM-FRM نسبت به ساختار اصلی FRM این است که فیلترهای ماسک-گذار از ۲ به ۱ تقلیل می‌یابند. شکل (۱-۱) ساختار این روش را نشان می‌دهد.



شکل (۱-۱) نمودار بلوکی تکنیک SFFM-FRM [۲].



شکل (۲-۱) پاسخ فرکانسی تکنیک SFFM-FRM [۲].

فیلتر لبه‌باند توسط سری کردن چندین فیلتر داخلی تولید می‌شود. در این روش $C(z)$ مکمل فیلتر $A(z)$ است و $G_c(z)$ فیلتر ماسک‌گذار برای $C(z)$ می‌باشد. پاسخ فرکانسی بلوک‌های تشکیل دهنده‌ی

این تکنیک در شکل (۲-۱) قابل مشاهده است. برای فیلتری با مشخصات $\omega_s = 0.602\pi$ و $\omega_p = 0.6\pi$ نتایج شبیه‌سازی در جدول (۱-۱) آورده شده است [۲].

جدول (۱-۱) نتایج شبیه‌سازی تکنیک SFFM-FRM [۲].

Design	Multipliers	Adders	Group delays
Conventional	925	1850	925
1-stage FRM	130	256	1000
SFFM-FRM	92	177	1105

۲.۲.۱ استفاده از ساختار پیش‌فیلتر متعادل کننده^۶

در این روش از یک ساختار FRM تعدیل یافته استفاده می‌شود [۳]. در ساختار FRM، طول فیلتر لبه‌باند با افزایش M (نرخ نمونه‌برداری) کاهش می‌یابد ولی همزمان طول فیلترهای ماسک‌گذار افزایش می‌یابد. برای کاهش پیچیدگی کل ساختار از یک پیش‌فیلتر متعادل کننده استفاده می‌کنیم که به واسطه آن می‌توان نرخ نمونه‌برداری را افزایش داد در حالیکه فیلترهای ماسک‌گذار تغییر نمی‌کنند.

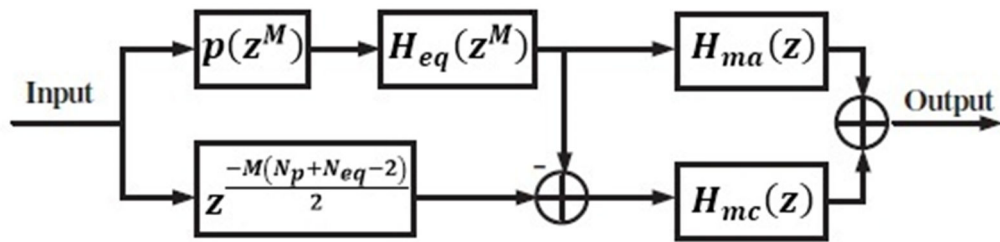
فیلتر $p_1(z) = 1 + z^{-1}$ (با طول زوج) که پاسخ فرکانسی آن را $\cos(\frac{\omega}{2})$ در نظر می‌گیریم. فیلتر $p_2(z)$ (با طول زوج) به گونه‌ای طراحی می‌شود که حاصل $p_1(z) + p_2(z)$ فیلتری باشد با باند عبوری شبیه $p_1(z)$ و باند توقف از $[\pi, 2\pi]$.

در حالت کلی‌تر رابطه‌ی (۱-۱) برقرار است.

$$p(z^M) = p_1(z^M) + p_2(z^M) \quad M \text{ عدد زوج} \quad (۱-۱)$$

شکل (۳-۱) ساختار بلوکی یک پیش‌فیلتر متعادل کننده را نشان می‌دهد.

^۶ Prefilter Equalizer



شکل (۳-۱) ساختار بلوکی پیش‌فیلتر متعادل کننده [۳].

برای طراحی فیلتری با مشخصات $\omega_s = 0.301\pi$ و $\omega_p = 0.3\pi$ به وسیله این روش ۲۰٪ از ضرایب کل فیلتر کاهش می‌یابد.

۳.۲.۱ الگوریتم ژنتیک

الگوریتم ژنتیک یک روش بهینه‌سازی عددی است که بر اساس اصول داروین و با الهام از ژنتیک طبیعی کار می‌کند. رقابت میان موجودات زنده برای تصاحب منبع مشترک محدود، باعث می‌شود که کارآمدترین افراد در این رقابت پیروز شوند و در انتقال ژن‌ها به نسل‌های بعدی نقش مؤثری داشته باشند. نتیجه‌ی این امر، برازندگی بیشتر افراد هر نسل و نیز سازگاری بیشتر افراد در نسل‌های بعدی با شرایط حاکم بر محیط اطراف خواهد بود [۴]. این روش مناسب‌ترین رشته‌ها را از میان اطلاعات تصادفی سازمان‌دهی شده با روش جستجوی انسانی انتخاب می‌کند، در هر نسل یک گروه جدید رشته‌ها با استفاده از بهترین قسمت‌های دنباله‌های قبلی و بخش جدید اتفافی برای رسیدن به یک جواب مناسب به وجود می‌آید [۵].

۴.۲.۱ طراحی فیلتر توسط روش مجموع مربعات وزن دار شده (WLS)^۷

در این روش ضرایب فیلتر با به حداقل رساندن پاسخ خطا ($E(\omega)$) در جهت کمترین مربعات خطا، محاسبه می‌گردد [۶]. به حداقل رساندن خطا در جهت کمترین مربعات خطا بوسیله یک تابع وزن دار معین $W(\omega)$ صورت می‌پذیرد. در باندگذر $W(\omega)$ برابر یک و در باند توقف برابر $\frac{\delta_p}{\delta_s}$ می‌باشد. پاسخ فرکانسی فیلتر ایده‌آل را با $D(\omega)$ و پاسخ فرکانسی فیلتر مطلوب را با $F(\omega)$ نشان می‌دهیم. رابطه‌ی (۲-۱) پاسخ خطا را محاسبه می‌کند.

$$E(\omega) = W(\omega)|D(\omega) - F(\omega)| \quad (۲-۱)$$

تحقیقات زیادی روی این روش انجام شده است و اغلب فیلترها را می‌توان توسط آن طراحی کرد. اما در این روش به دلیل رسیدن به محاسبات ماتریسی و گرفتن معکوس ماتریس دارای نوعی پیچیدگی محاسباتی خواهد بود و صرفاً بهترین روش طراحی محسوب نمی‌گردد.

۵.۲.۱ طراحی فیلتر با استفاده از الگوریتم لانه‌ی زنبوری

الگوریتم کاوش زنبورهای عسل یک الگوریتم بهینه‌سازی الهام گرفته از تجمع زنبورهای عسل می‌باشد. این الگوریتم برای بهینه‌سازی توابع چند قله‌ای که تمامی قله‌های آن از یک حدی که یافت شده بالاتر می‌باشند، بسیار مفید خواهد بود [۷].

ابتدا n زنبور ایجاد و به طور تصادفی در فضای مسأله پخش می‌شوند. این ذرات بر حسب شایستگی مرتب می‌شوند. M نقطه بهتر انتخاب می‌شوند و اطراف آن یک کلونی با n زنبور تعریف می‌گردد. هر یک از M نقطه بدست آمده مرکز و نماینده‌ی یک محدوده است. هر محدوده را به یک کلونی می‌دهند تا بهترین میزان شایستگی در آن محدوده را به صورت مخفی بیابد [۸]. اگر تعداد کل

⁷ Weighted Least Square

زنبورها برابر h باشد، تعداد زنبورهای باقی مانده ($h - (M * n)$) به عنوان پیش‌آهنگ به صورت تصادفی مقداردهی می‌شوند. در هر تکرار الگوریتم n زنبور موجود در کلونی طبق الگوریتم‌های بهینه‌سازی دیگر مقدار بهینه‌ی خود را می‌یابند. پیش‌آهنگان نیز به تنهایی اطراف خود را می‌گردند. بعد از تعدادی تکرار یک تحلیل سراسری انجام می‌شود و بهترین میزان شایستگی گزارش شده توسط هر گروه و پیش‌آهنگان، همه ذخیره می‌شوند و کلونی‌ها طبق این نتایج برای تکرار بعدی تصمیم می‌گیرند [۸].

جدول (۲-۱) مقدار خطا در تکرارهای مختلف الگوریتم لانه‌ی زنبوری در طراحی یک فیلتر پایین‌گذر

$$\left\{ \begin{array}{l} \omega_p = 0.25 \\ \omega_s = 0.3 \\ M = 20 \end{array} \right. \text{ با مشخصات } \text{را نشان می‌دهد [۹].}$$

جدول (۲-۱) مقدار خطا فیلتر پایین‌گذر با روش لانه زنبوری [۹].

تکرار	میانگین مربعات خطا
۱۰	۳/۲۳۶۲
۴۰	۰/۲۲۵۱
۲۰۰	۰/۰۰۲۹

این الگوریتم صرفاً در محیط‌هایی که دارای یک بهینه‌ی سراسری و چندین بهینه‌ی محلی هستند، کاربرد دارد.

۶.۲.۱ کاهش حجم حافظه برای پیاده‌سازی یک فیلتر FIR توسط الگوریتم CSD^۸

این روش، یک روش کدگذاری اعداد باینری است که نتیجه‌ی حاصله کمترین اعداد غیر صفر را شامل می‌شود. بر خلاف اعداد باینری که با ۰ و ۱ نمایش داده می‌شوند، برای نشان دادن اعداد در این روش از (۱ و ۰ و -۱) استفاده می‌گردد [۱۰]. با این تکنیک یک ضرب‌کننده با پیچیدگی کمتر پیاده‌سازی می‌شود. زیرا پیچیدگی پیاده‌سازی ضرب‌کننده‌ها با امان‌های غیر صفر آنها تخمین زده می‌شود. این امان‌ها با استفاده از الگوریتم CSD به حداقل مقدار خود می‌رسند. رابطه‌ی (۳-۱) و (۴-۱) به ترتیب نمایان‌گر یک عدد W بیتی به فرم متمم ۲ و CSD می‌باشد. فرم CSD طبق مراحل زیر بدست می‌آید.

$$A = a'_{w-1} \cdot a'_{w-2} \dots a'_1 \cdot a'_0 \quad (۳-۱)$$

$$A = a_{w-1} \cdot a_{w-2} \dots a_1 \cdot a_0 \quad (۴-۱)$$

فرم CSD طبق مراحل زیر بدست می‌آید:

- $a'_{-1} = 0$;
- $\gamma_{-1} = 0$;
- $a'_w = a'_{w-1}$
- *for* ($i = 0$ to $W - 1$)

$$\theta_i = a'_i \oplus a'_{i-1} ;$$

$$\gamma_i = \overline{\gamma_{i-1}} \cdot \theta_i ;$$

$$a_i = (1 - 2a'_{i+1})\gamma_i ;$$

به عنوان مثال عمل ضرب $x * 57$ به دو طریق قابل انجام است:

⁸ Canonic Signed Digit

ضرب عادی:

$$(5-1)$$

$$(0110111)_2 * x$$

برای انجام این ضرب از ۴ جمع کننده استفاده می کنیم.

$$= x + x \ll 1 + x \ll 2 + x \ll 4 + x \ll 5 \quad (6-1)$$

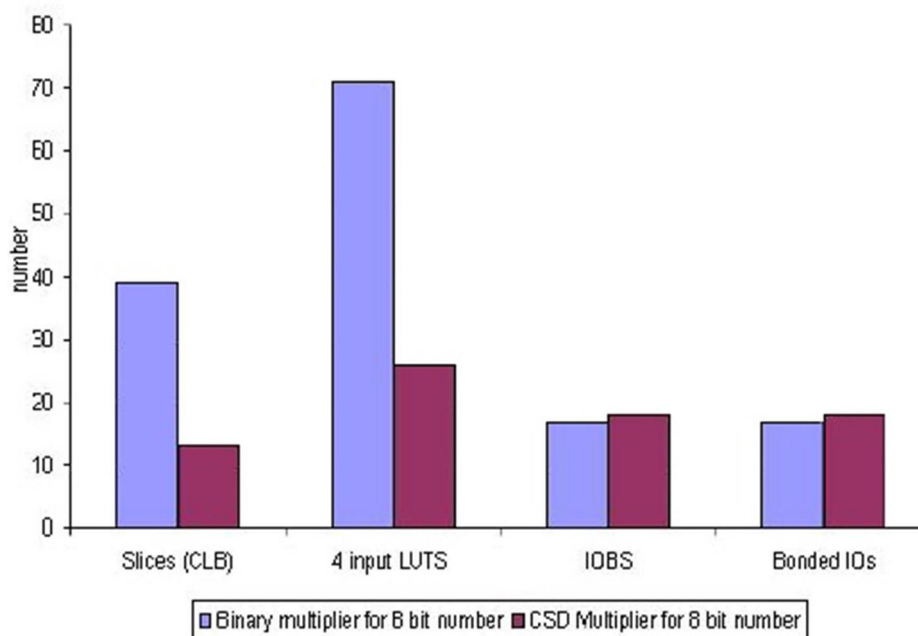
ولی اگر به فرم CSD درآید، عمل ضرب به ترتیب با روابط (۶-۱) و (۷-۱) انجام می گیرد:

$$(100 - 1001)_{CSD} \quad (7-1)$$

$$= x - x \ll 3 + x \ll 6 \quad (8-1)$$

شکل (۴-۱) یک ضرب کننده ۸ بیتی ساده و CSD را مقایسه می کند [۱۱]. همان طور که مشاهده

می کنیم محاسبات در حالت دوم به علت کاهش یافتن بیت های غیر صفر کمتر شده است.



شکل (۴-۱) مقایسه ضرب کننده ۸ بیتی ساده و CSD [۱۱].

۳.۱ ساختار پایان نامه

در فصل ۱ این پایان نامه مقدمه‌ای برای آشنایی با موضوع و خلاصه‌ای از تحقیقات انجام شده توسط محققان این حوزه ارائه شده است.

در فصل ۲ مرور مختصری بر انواع فیلترها و روش‌های نظری طراحی فیلترهای FIR و IIR^۹ خواهیم داشت. همچنین ساختار داخلی فیلتر FRM و روابط ریاضی حاکم بر آن مورد بررسی قرار می‌گیرد.

در فصل ۳ به معرفی دو الگوریتم تکاملی بهینه‌سازی دسته ذرات و الگوریتم فرهنگی پرداخته می‌شود.

در فصل ۴ الگوریتم ترکیبی برای طراحی فیلتر مطرح می‌شود و روش پیشنهادی و محاسبات لازم برای پیاده‌سازی یک فیلتر میان‌گذر FRM مورد بررسی قرار می‌گیرد.

در فصل ۵ تأثیر مرتبه و پهناى باندگذر بر روی پارامترهای اصلی فیلتر تحلیل می‌گردند و نتایج حاصل از شبیه‌سازی زیرفیلترهای FRM توسط نرم‌افزار MATLAB نشان داده می‌شوند و در نهایت فیلتر میان‌گذر FRM توسط نرم‌افزار system generator بر روی FPGA پیاده‌سازی می‌شود.

در فصل ۶ به نتیجه‌گیری و پیشنهادات لازم برای بهبود کار می‌پردازیم.

^۹ Infinite Impulse Response

فصل دوم

تکنیک‌های طراحی فیلتر

مقدمه

با داشتن مجموعه‌ای از مشخصات فیلتر می‌توان تابع تبدیل یک سیستم خطی زمان‌گسسته را به طور تقریبی طراحی کرد. طراحی فیلترهای IIR به طور ضمنی تقریب توسط تابع گویایی از Z را اقتضا می‌کند، درحالی‌که طراحی فیلترهای FIR تقریب چند جمله‌ای را ایجاب می‌کنند. در این فصل روش‌های نظری طراحی فیلترهای FIR و IIR مورد بحث قرار می‌گیرد. در ادامه تکنیکی برای طراحی فیلترهایی با باندگذر باریک که نیاز به ضرایب کمتری دارند مطرح می‌گردند.

۱.۲ انواع فیلتر

فیلتر سیستمی است که مشخصه‌ی فرکانسی ورودی را تحت تأثیر قرار می‌دهد. برای مثال فیلتر پایین‌گذر مؤلفه‌های بالای فرکانس سیگنال ورودی را حذف و مؤلفه‌های فرکانس پایین را عبور می‌دهد. فیلترهای فرکانس بالا مؤلفه‌های فرکانس پایین را حذف و فرکانس بالا را عبور می‌دهد. فیلتر میان‌گذر فیلتری است که محدوده‌ی خاص فرکانسی را عبور می‌دهد و مؤلفه‌های فرکانس‌های بالا و پایین محدوده را حذف می‌کند. فیلتر میان‌گذر فیلتری است که تمام باند به استثناء محدوده‌ی خاص خود را عبور می‌دهد. دو نوع فیلتر اصلی وجود دارد که عبارتند از: فیلتر دیجیتال و فیلتر آنالوگ.

۱.۱.۲ فیلتر آنالوگ

یک فیلتر آنالوگ از مدارهای الکترونیک آنالوگ ساخته شده‌است و از اجزایی همچون مقاومت، خازن و تقویت‌کننده‌های عملیاتی جهت ایجاد اثر فیلترینگ مورد نیاز استفاده می‌کند. این مدارات فیلتر به طور گسترده در کاربردها به عنوان کاهنده‌ی نویز، افزایشنده‌ی سیگنال ویدئو، یکنواخت‌کننده‌ی گرافیکی در سیستم‌های فرکانس بالا و بسیاری از نواحی دیگر استفاده می‌شوند. تکنیک‌های استاندارد ثابت شده‌ای جهت طراحی مدار فیلتر آنالوگ بر حسب نیاز وجود دارد.

۲.۱.۲ فیلتر دیجیتال

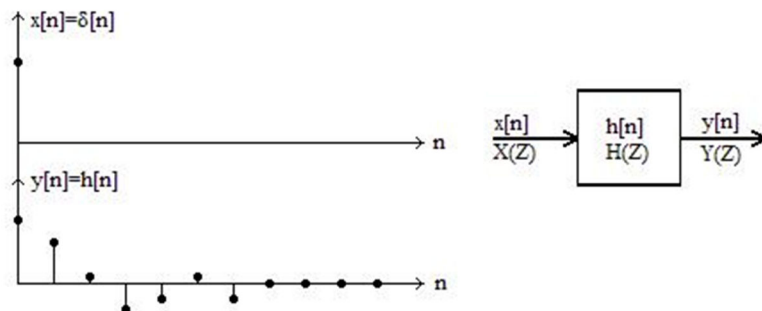
فیلترهای دیجیتال از مباحث پردازش سیگنال دیجیتال^۱ (DSP) می‌باشند. در واقع یک فیلتر دیجیتال یک تابع تبدیل دیجیتال است که می‌تواند خروجی را بر حسب مقدار کنونی ورودی و مقادیر لحظات قبل ورودی و خروجی محاسبه کند. در این نوع فیلترها عملیات ریاضی توسط پردازنده روی آن‌ها انجام می‌شود. گسسته بودن تابع تبدیل این فیلترها به ما امکان پیاده‌سازی انواع توابع تبدیل پیچیده‌ای را می‌دهد که با فیلترهای آنالوگ امکان پیاده‌سازی آن وجود ندارد.

فیلترهای دیجیتال را می‌توان از لحاظ پاسخ ضربه به دو دسته تقسیم نمود:

۱.۲.۱.۲ فیلترهای دیجیتال با پاسخ ضربه طول محدود (FIR)

FIR به معنی پاسخ ضربه محدود می‌باشد. به این معنی که اگر به ورودی این فیلترها تابع ضربه اعمال شود، خروجی پس از $n=N$ نمونه صفر می‌شود. یعنی خروجی در اثر ورودی تابع ضربه فقط برای مدت زمان محدودی مقادیر غیر صفر دارد.

از آنجایی که تابع تبدیل یک سیستم در واقع خروجی سیستم در اثر ورودی تابع ضربه می‌باشد، می‌توان نتیجه گرفت که تابع تبدیل یک فیلتر FIR فقط دارای N نمونه مقدار غیر صفر است.



شکل (۱-۲) پاسخ ضربه فیلتر FIR.

¹ Digital Signal Processing

همانگونه که از شکل (۱-۲) برمی آید در یک فیلتر FIR مقادیر خروجی در زمان های مختلف از رابطه-
ی (۱-۲) بدست می آید:

$$y[n] = h[n] * x[n] = \sum_{k=0}^N h[k]x[n-k] \quad (1-2)$$

۲.۲.۱.۲ فیلترهای دیجیتال با پاسخ ضربه طول نامحدود (IIR)

فیلتر IIR به معنای پاسخ ضربه نامحدود می باشد. به این معنی که اگر به ورودی این فیلتر تابع ضربه اعمال شود، خروجی هیچگاه از بین نمی رود. یعنی پس از مدت زمان محدود، خروجی به صفر میل نمی کند. پس می توان نتیجه گرفت تابع تبدیل این فیلترها (که همان خروجی در اثر ورودی تابع ضربه می باشد)، تعداد نامحدودی عنصر دارد. این فیلترها را نمی توان از کانولوشن ورودی در تابع تبدیل ساخت زیرا در هر لحظه برابر تعداد نامحدودی جمله است و این امر در یک سیستم واقعی دیجیتال غیرقابل پیاده سازی است. راه حل ساخت فیلترهای دیجیتال IIR، استفاده از روابط بازگشتی می باشد. بدین معنی که خروجی در هر لحظه علاوه بر اینکه به مقادیر ورودی در همان لحظه و لحظات قبل وابسته است، به مقادیر خروجی در لحظات قبل که قبلاً محاسبه و ذخیره شده است بستگی پیدا می کند.

۲.۲ روش های طراحی فیلتر

تکنیک های طراحی مناسب برای فیلترهای IIR و FIR با هم تفاوت دارند. در این فصل چند روش از تکنیک های طراحی برای هر دو نوع فیلتر مطرح می شود که از روال های فرم بسته ای که تنها مستلزم جایگذاری مشخصه های طراحی در فرمول های طراحی هستند تا تکنیک های الگوریتم های تکاملی که در آنها جواب از طریق یک روال تکراری بدست می آید را در برمی گیرد.

۱.۲.۲ روش های نظری طراحی فیلترهای IIR زمان گسسته به کمک فیلترهای زمان پیوسته

راهکار سنتی برای طراحی فیلترهای IIR زمان گسسته، مستلزم تبدیل یک فیلتر زمان پیوسته به یک فیلتر زمان گسسته با مشخصه های از پیش تعریف شده می باشد زیرا بسیاری از روش های مفید طراحی فیلترهای IIR زمان پیوسته، فرمول های طراحی شکل بسته نسبتاً ساده ای دارند. بنابراین، روش طراحی فیلتر IIR زمان گسسته بر پایه ی چنین فرمول های طراحی زمان پیوسته استاندارد از نظر انجام ساده تر می باشند. همچنین روش های تقریب استاندارد به خوبی برای فیلترهای IIR زمان پیوسته کار می کنند، ولی هنگامی که این روش ها مستقیماً در حالت IIR زمان گسسته بکار روند، منجر به فرمول های طراحی ساده با شکل بسته ای نمی شوند.

در این نوع طراحی مشخصه ها برای فیلتر زمان پیوسته توسط تبدیل مشخصه های فیلتر زمان گسسته مطلوب بدست می آید. سپس تابع سیستم $H_c(s)$ یا پاسخ ضربه $h_c(t)$ فیلتر زمان پیوسته توسط روش های تقریب استاندارد محاسبه شده است در نهایت با تبدیل $H_c(s)$ به $H(z)$ ، تابع تبدیل گسسته بدست می آید (محور موهومی صفحه s را بر روی دایره واحد صفحه z نگاشت می کنیم). برای تبدیل $H_c(s)$ به $H(z)$ شرط زیر باید رعایت گردد:

خواص اساسی پاسخ فرکانسی زمان پیوسته در پاسخ فرکانسی فیلتر زمان گسسته بدست آمده حفظ شود. یکی از این خواص، پایداری سیستم است. یک فیلتر زمان پیوسته باید به یک فیلتر زمان گسسته پایدار تبدیل شود.

دو تکنیک رایج در طراحی فیلتر IIR، بر مبنای تبدیل سیستم IIR زمان پیوسته به سیستم های IIR زمان گسسته، عبارتند از:

۱.۱.۲.۲ طراحی فیلتر به وسیله تغییر ناپذیری ضربه

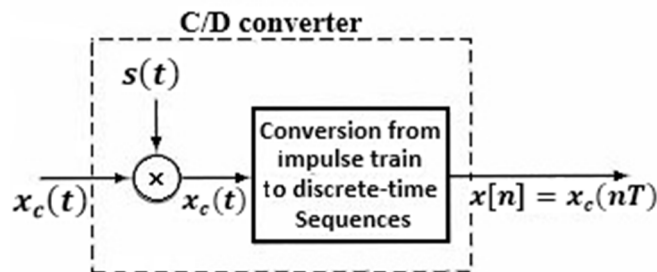
در این روش، تبدیل بر این اساس صورت می گیرد که هر دو سیستم پاسخ ضربه مشابه دارند.

یعنی پاسخ ضربه‌ی سیستم گسسته، گسسته پاسخ ضربه‌ی سیستم پیوسته می‌باشد [۱۲].

$$h[n] = T_d h_c(nT_d) \quad (۲-۲)$$

برای درک بهتر این مبحث شرح مختصری از نمونه‌برداری یک سیگنال لازم است. نمونه‌برداری از یک

سیگنال پیوسته $x_c(t)$ به صورت ساده با شکل (۲-۲) قابل نمایش است.



شکل (۲-۲) نمونه برداری از سیگنال پیوسته [۱۲].

برای بدست آوردن رابطه‌ی حوزه‌ی فرکانس بین ورودی و خروجی یک مبدل C/D ایده‌آل، ابتدا

تبدیل $x_c(t)$ به $x_s(t)$ را توسط مدولاسیون قطار ضربه متناوب زیر در نظر بگیریم :

$$S(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT) \quad (۳-۲)$$

$x_s(t)$ را با $x_c(t)$ مدوله می‌کنیم تا رابطه‌ی (۴-۲) بدست آید:

$$x_s(t) = x_c(t)S(t) = x_c(t) \sum_{n=-\infty}^{+\infty} \delta(t - nT) \quad (۴-۲)$$

طبق خاصیت غربالی تابع ضربه

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x_c(nT) \delta(t - nT) \quad (۵-۲)$$

$$x_s(t) \xrightarrow{\text{فوریه تبدیل}} X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) \quad (۶-۲)$$

$$\rightarrow X_s(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X_c(j(\Omega - k\Omega_s)) \quad (۷-۲)$$

رابطه‌ی (۷-۲) بیان می‌کند که تبدیل فوریه $x_s(t)$ متشکل از تکرارهای تبدیل فوریه $x_c(t)$ است. همان طور که می‌دانیم :

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-j\Omega T n} \quad (۸-۲)$$

و طبق رابطه‌ی (۲-۲) رابطه‌ی (۸-۲) به فرم رابطه‌ی (۹-۲) تغییر می‌کند

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \quad (۹-۲)$$

$$X_s(j\Omega) = X(e^{j\omega}) \stackrel{\omega=\Omega T}{\implies} X_s(j\Omega) = X(e^{j\Omega T}) \quad (۱۰-۲)$$

در نتیجه از معادلات (۷-۲) و (۱۰-۲) رابطه‌ی (۱۱-۲) بدست می‌آید:

$$X(e^{j\Omega T}) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X_c(j(\Omega - k\Omega_s)) \quad (۱۱-۲)$$

یا به طور معادل

$$X(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right) \quad (۱۲-۲)$$

طبق رابطه‌ی (۲-۲) و (۱۲-۲) خواهیم داشت:

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c\left(j\left(\frac{\omega}{T_d} + j\frac{2\pi k}{T_d}\right)\right) \quad (۱۳-۲)$$

اگر فیلتر زمان پیوسته باند محدود باشد به گونه‌ای که $H_c(j\Omega) = 0, |\Omega| \geq \pi/T_d$ آنگاه رابطه‌ی (۱۳-۲) به صورت رابطه‌ی (۱۴-۲) خلاصه می‌گردد:

$$H(e^{j\omega}) = H_c\left(j\frac{\omega}{T_d}\right) \quad |\omega| \leq \pi \quad (۱۴-۲)$$

یعنی پاسخ‌های فرکانسی زمان گسسته و زمان پیوسته بوسیله یک مقیاس‌دهی خطی محور فرکانسی $(\omega = \Omega T_d)$ به $|\omega| < \pi$ مرتبط می‌شوند. متأسفانه در عمل هیچ فیلتر زمان پیوسته باند محدودی وجود ندارد در نتیجه بین جملات متوالی در معادله‌ی (۱۳-۲) تداخل روی می‌دهد ولی اگر دامنه

فیلتر زمان پیوسته در فرکانس های بالا به سمت صفر میل کند، تداخل طیفی قابل صرف نظر خواهد بود.

از مزایای این نوع طراحی آن است که رابطه ی بین فرکانس زمان گسسته و فرکانس زمان پیوسته خطی می باشد. در نتیجه به استثناء تداخل طیفی شکل پاسخ فرکانسی حفظ می شود ولی متأسفانه این تکنیک برای فیلترهای باند محدود مناسب است و از این روش برای تبدیل فیلتر بالاگذر و میان-نگذر آنالوگ نمی توان استفاده کرد.

۲.۱.۲.۲ تبدیل دو خطی

در این تکنیک محور $-\infty < \Omega < \infty$ به $-\pi \leq \omega < +\pi$ نگاشت می شود. تبدیل بین متغیرهای فرکانسی زمان پیوسته و زمان گسسته باید غیرخطی باشد. بنابراین استفاده از این روش به موقعیت هایی محدود می شود که در آنها پیچش متناظر محور فرکانسی قابل قبول باشد. اگر $H_c(s)$ تابع سیستم زمان پیوسته و $H(z)$ تابع سیستم زمان گسسته را نشان دهد، تبدیل دو خطی متناظر با جایگذاری s توسط رابطه ی (۱۵-۲) می باشد.

$$s = \frac{2}{T_d} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \quad (15-2)$$

یعنی

$$H(z) = H_c \left[\frac{2}{T_d} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \right] \quad (16-2)$$

از رابطه ی (۱۵-۲) و (۱۶-۲) رابطه ی (۱۷-۲) بدست می آید:

$$z = \frac{1+(T_d/2)s}{1-(T_d/2)s} \quad (17-2)$$

و با جایگذاری $s = \sigma + j\Omega$ در معادله ی (۱۷-۲) رابطه ی (۱۸-۲) بدست می آید:

$$z = \frac{1+\sigma T_d/2+j\Omega T_d/2}{1-\sigma T_d/2-j\Omega T_d/2} \quad (18-2)$$

اگر $\sigma < 0$ آنگاه از معادله ی (۲-۱۸) برای هر مقدار Ω نتیجه می شود که $|z| < 1$. به طور مشابه اگر $\sigma > 0$ آنگاه $|z|$ برای تمامی مقادیر Ω بزرگ تر از یک می باشد. یعنی اگر یک قطب $H_c(s)$ در نیمه ی چپ صفحه ی s باشد، تصویر آن در صفحه ی z داخل دایره ی واحد خواهد بود. بنابراین فیلترهای زمان پیوسته پایدار و علی به فیلترهای زمان گسسته پایدار و علی نگاشت می شوند.

۲.۲.۲ طراحی فیلتر FIR توسط پنجره گذاری

روش های طراحی برای فیلترهای FIR مبتنی بر تقریب مستقیم پاسخ فرکانسی مطلوب سیستم زمان گسسته می باشد. ساده ترین روش طراحی فیلتر FIR روش پنجره نامیده شده است.

اگر پاسخ فرکانس فیلتر مطلوب ایده آل $H_d(e^{j\omega})$ و دنباله پاسخ ضربه ی متناظر آن را $h_d[n]$ بنامیم، روابط (۲-۱۹) و (۲-۲۰) بین آنها برقرار است:

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n} \quad (۲-۱۹)$$

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega \quad (۲-۲۰)$$

ساده ترین طریق برای دستیابی به یک فیلتر FIR علی از $h_d[n]$ ، تعریف یک سیستم جدید با پاسخ ضربه $h[n]$ است که با رابطه ی (۲-۲۱) نشان داده شده است:

$$h[n] = \begin{cases} h_d[n], & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (۲-۲۱)$$

در حالت کلی تر، می توانیم $h[n]$ را به صورت ضرب پاسخ ضربه مطلوب و یک پنجره ی طول محدود $w[n]$ نشان دهیم یعنی:

$$h[n] = h_d[n] w[n] \quad (۲-۲۲)$$

که این پنجره مستطیلی دارای مشخصات رابطه ی (۲-۲۳) است:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (۲-۲۳)$$

معادله‌ی (۲-۲۲) در حوزه‌ی فرکانس برابر است با:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \quad (2-24)$$

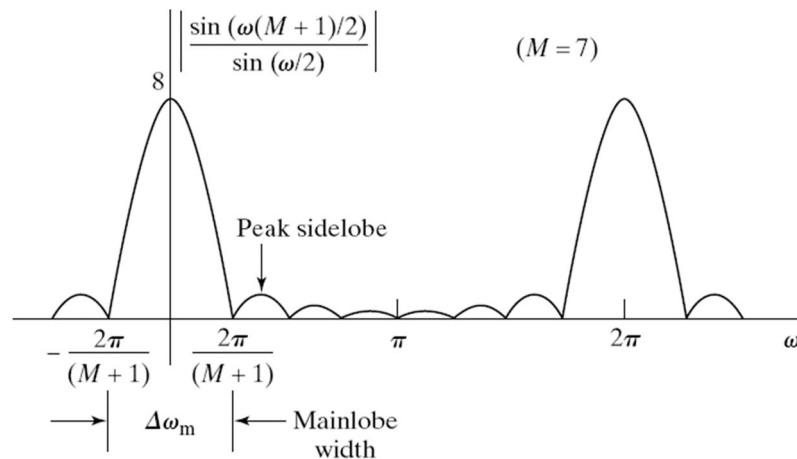
یعنی $H(e^{j\omega})$ کانولوشن متناوب پاسخ فرکانسی مطلوب ایده‌آل با تبدیل فوریه پنجره می‌باشد.

اگر $w[n]$ برای تمامی مقادیر n مساوی یک باشد (یعنی هیچگونه برشی ندهیم) $W(e^{j\omega})$ یک قطار ضربه متناوب با دوره تناوب 2π می‌باشد و بنابراین $H(e^{j\omega}) = H_d(e^{j\omega})$. در واقع اگر $w[n]$ به گونه‌ای انتخاب شود که $W(e^{j\omega})$ در یک باند باریک فرکانسی حول $\omega = 0$ متمرکز شود، آنگاه $H(e^{j\omega})$ مشابه $H_d(e^{j\omega})$ خواهد بود.

با توجه به رابطه‌ی (۲-۲۳) خواهیم داشت:

$$W(e^{j\omega}) = \sum_{n=0}^M e^{-j\omega n} = \frac{1-e^{-j\omega(M+1)}}{1-e^{-j\omega}} = e^{-j\omega M/2} \frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} \quad (2-25)$$

$W(e^{j\omega})$ برای پنجره مستطیلی دارای یک فاز خطی تعمیم یافته می‌باشد. با افزایش M عرض گلبرگ اصلی کاهش می‌یابد.



شکل (۲-۳) پاسخ فرکانسی $W(e^{j\omega})$ [۱۲].

پنجره‌های مورد استفاده رایج :

بعضی از پنجره‌های مورد استفاده رایج در شکل (۲-۵) نشان داده شده است [۱۲].

۱- پنجره مستطیلی^۲:

$$w[n] = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (26-2)$$

۲- پنجره مثلثی^۳:

$$w[n] = \begin{cases} 2n/M, & 0 \leq n \leq M/2 \\ 2 - 2n/M, & M/2 < n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (27-2)$$

۳- پنجره هنینگ^۴:

$$w[n] = \begin{cases} 0.5 - 0.5 \cos(2\pi n/M), & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (28-2)$$

۴- پنجره همینگ^۵:

$$w[n] = \begin{cases} 0.54 - 0.46 \cos(2\pi n/M), & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (29-2)$$

۵- پنجره بلک من^۶:

$$w[n] = \begin{cases} 0.42 - 0.5 \cos(2\pi n/M) + 0.08 \cos(4\pi n/M), & 0 \leq n \leq M \\ 0, & \text{سایر مقادیر } n \end{cases} \quad (30-2)$$

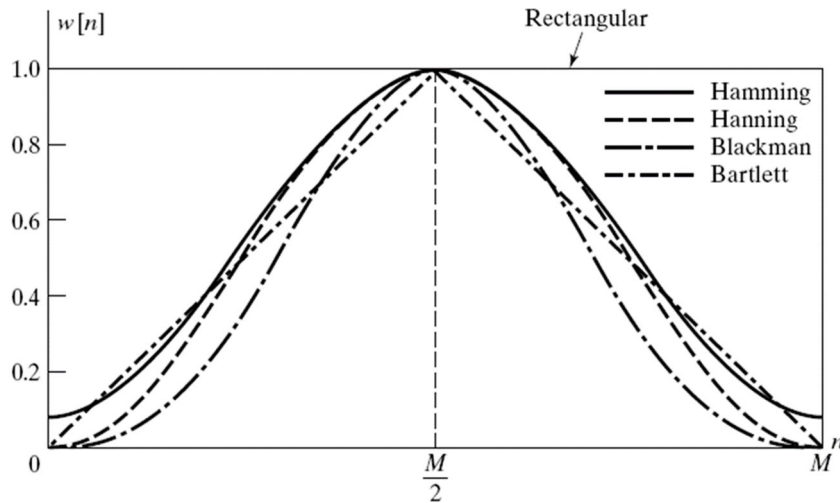
² Rectangular

³ Bartlett

⁴ Hanning

⁵ Hamming

⁶ Blackman



شکل (۲-۴) پاسخ فرکانسی پنجره‌های مورد استفاده رایج [۱۲].

این پنجره‌ها این خاصیت را دارند که تبدیل فوریه آنها حول $\omega = 0$ متمرکز می‌باشد و شکل تابع ساده‌ای دارند که به آنها اجازه می‌دهد به سادگی محاسبه شوند.

۳.۲.۲ الگوریتم‌های تکاملی

یکی از راه‌های طراحی فیلتر استفاده از الگوریتم‌های تکاملی است. الگوریتم‌های تکاملی فیلتر مطلوب را با بهره‌گیری از مجموعه‌ای از جواب و از طریق یک فرآیند تکراری بدست می‌آورند. در فصل‌های آینده توضیحات بیشتری در مورد الگوریتم‌های تکاملی و نحوه‌ی طراحی فیلتر توسط آنها ارائه می‌گردد.

۴.۲.۲ تکنیک ماسک‌گذاری روی پاسخ فرکانسی

فیلترهای ماسک‌گذاری پاسخ فرکانسی، تکنیکی برای طراحی فیلترهای تیز پایین‌گذر، بالاگذر و میان‌گذر با پهناهای باند عبور دلخواه هستند.

فیلترهای FIR با فاز خطی دارای قابلیت تضمین پایداری و عاری از اعوجاج فاز می باشند. مشکل این فیلترها پیچیدگی آنها برای پیاده سازی فیلترهای تیز می باشد. فیلترهای ساخته شده با استفاده از تکنیک ماسک گذاری پاسخ فرکانسی، ضرایب کمتری نسبت به فیلتر FIR مشابه دارند.

۱.۴.۲.۲ مقدمات ریاضی فیلتر FRM

دو فیلتر با فاز خطی $F_a(z)$ و $F_c(z)$ در صورتی مکمل یکدیگراند که پاسخ فرکانسی آنها در رابطه‌ی (۳۱-۲) صدق کند (برای هر ω).

$$|F_a(e^{j\omega}) + F_c(e^{j\omega})| = 1 \quad (31-2)$$

پاسخ فرکانسی یک فیلتر FIR با فاز خطی و طول N را می توان طبق رابطه‌ی (۳۲-۲) نمایش داد [۱۳، ۱۲].

$$F_a(e^{j\omega}) = e^{-j((N-1)/2)\omega} R(\omega) \quad (32-2)$$

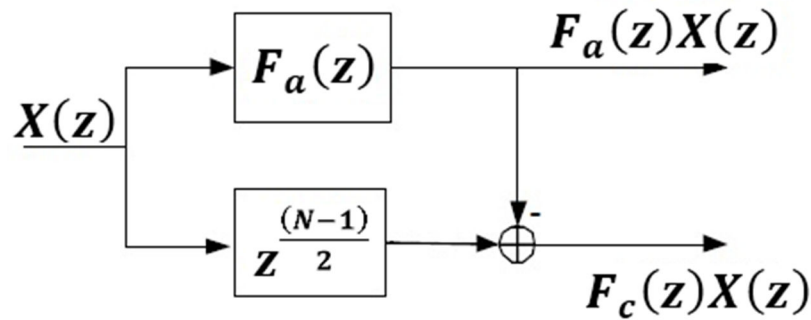
که $R(\omega)$ یک تابع مثلثاتی حقیقی و مثبت از ω می باشد. پاسخ فرکانسی مکمل $F_a(z)$ طبق رابطه‌ی (۳۳-۲) قابل محاسبه است.

$$F_c(e^{j\omega}) = e^{-j((N-1)/2)\omega} \{1 - R(\omega)\} \quad (33-2)$$

با توجه به رابطه‌ی (۳۲-۲) و (۳۳-۲) $F_c(z)$ برابر است با

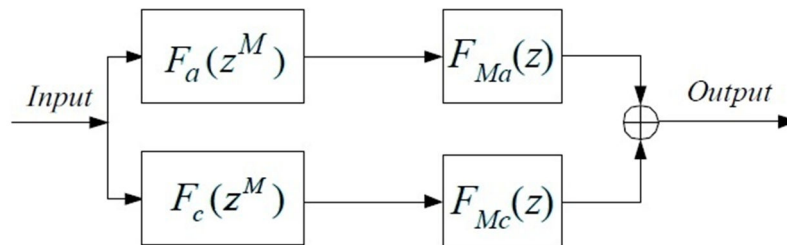
$$F_c(z) = z^{-((N-1)/2)} - F_a(z) \quad (34-2)$$

شکل (۶-۲) بلوک دیاگرام $F_a(z)$ و مکمل آن را نمایش می دهد.

شکل (۵-۲) بلوک دایگرام $F_a(z)$ و مکمل آن [۱۳].

۲.۴.۲.۲ فیلتر FRM با پهنای باند دلخواه

ساختار بلوکی این تکنیک به صورت شکل (۷-۲) نشان داده می‌شود [۱۴].



شکل (۶-۲) ساختار بلوکی تکنیک ماسک گذاری روی پاسخ فرکانسی [۱۴].

قوانین اولیه این تکنیک به شرح زیر است:

۱. در فیلتر فاز خطی $F_a(z)$ ، هر تأخیر با M تأخیر جایگزین می‌گردد.

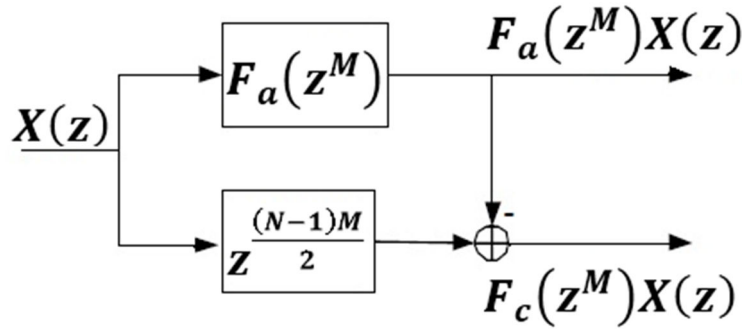
۲. نتیجه آن فیلتر متناوبی با باندگذر تیزتری حاصل می‌شود.

۳. نهایتاً با فیلترهای ماسک گذاری، باند توقف مطلوب را ایجاد می‌کنیم.

در شکل (۷-۲) همان $F_a(z)$ همان $F_a(z^M)$ است که هر تأخیر جایگزین M تأخیر شده است. $F_c(z^M)$

مکمل فیلتر $F_a(z^M)$ است که با توجه به رابطه (۲-۳۴) نحوه پیاده‌سازی این زوج مکمل در شکل

(۸-۲) قابل مشاهده است.



شکل (۷-۲) بلوک دیاگرام $F_a(z^M)$ و مکمل آن [۱۴]

طبق شکل (۷-۲)، اگر دو فیلتر ماسک گذار F_{Mc} و F_{Ma} به ترتیب برای دو فیلتر F_c و F_a استفاده شوند، فیلتر تیز با پهنای باند دلخواه قابل طراحی است.

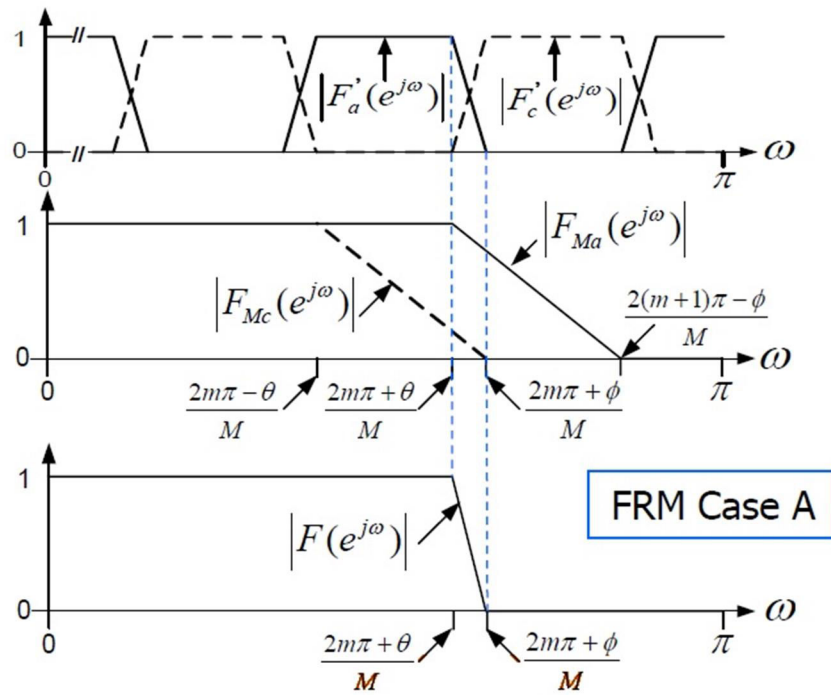
تبدیل Z شکل (۷-۲)، در رابطه (۳۵-۲) محاسبه شده است.

$$F(z) = F_a(z^M)F_{Ma}(z) + (z^{-\frac{N-1}{2}} - F_a(z^M))F_{Mc}(z) \quad (۳۵-۲)$$

بسته به فرکانس قطع باند عبور ω_p و فرکانس قطع باند توقف ω_s فیلتری که قصد طراحی آن را داریم، دو حالت برای فیلترهای ماسک گذار رخ می دهد [۱۴].

حالت اول: پاسخ فرکانسی فیلتری که قصد طراحی آن را داریم (F) در نزدیکی باند عبور توسط

F_a مشخص می گردد. پاسخ فرکانسی فیلترها در حالت اول توسط شکل (۹-۲) نمایش داده شده است.



شکل (۸-۲) فیلتر FRM حالت اول [۱۴].

روابط ریاضی زیرفیلترها در این حالت به شرح زیر می باشد:

$$\omega_p = \frac{2m\pi + \theta}{M} \quad (۳۶-۲)$$

$$\omega_s = \frac{2m\pi + \phi}{M} \quad (۳۷-۲)$$

$$\theta = \omega_p * M - 2m\pi \quad (۳۸-۲)$$

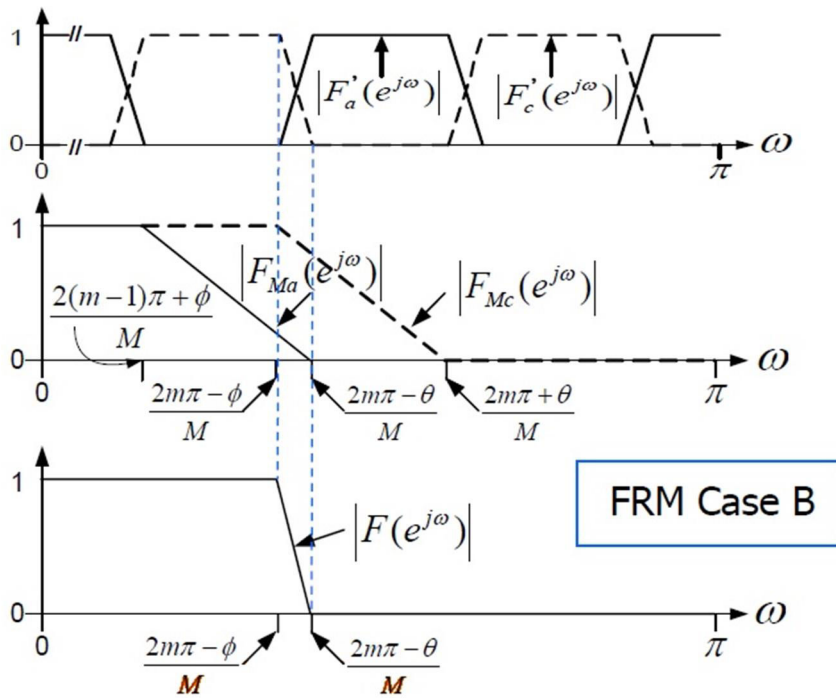
$$\phi = \omega_s * M - 2m\pi \quad (۳۹-۲)$$

$$F_{Ma}(z) = \begin{cases} w_{Ma,p} = w_p \\ w_{Ma,s} = \frac{2(m+1)\pi - \phi}{M} \end{cases} \quad (۴۰-۲)$$

$$F_{MC} = \begin{cases} W_{Mc,p} = \frac{2m\pi - \varphi}{M} \\ W_{Ms,s} = W_s \end{cases} \quad (41-2)$$

حالت دوم: پاسخ فرکانسی فیلتر مرود نظر (F) نزدیک باند عبور بوسیله F_c مشخص می گردد.

پاسخ فرکانسی فیلترها در حالت دوم در شکل (۲-۱۰) نمایش داده شده است.



شکل (۲-۹) فیلتر FRM حالت دوم [۱۴].

روابط ریاضی حالت دوم عبارت اند از:

$$\omega_p = \frac{2m\pi - \varphi}{M} \quad (42-2)$$

$$\omega_s = \frac{2m\pi - \theta}{M} \quad (43-2)$$

$$\theta = 2m\pi - W_s * M \quad (44-2)$$

$$\varphi = 2m\pi - W_p * M \quad (45-2)$$

$$F_{Ma}(z) = \begin{cases} W_{Ma,p} = \frac{2(m-1)\pi + \varphi}{M} \\ W_{Ma,s} = \omega_s \end{cases} \quad (46-2)$$

$$F_{MC} = \begin{cases} W_{Mc,p} = \omega_p \\ W_{Mc,s} = \frac{2m\pi + \theta}{M} \end{cases} \quad (47-2)$$

در نتیجه گیری مسأله موارد زیر باید در نظر گرفته شوند:

۱. ω_s و ω_p در مفروضات مسأله بوده و مقادیر φ, θ, M, m باید محاسبه گردد.

۲. M باید به گونه ای انتخاب گردد که پیچیدگی کل فیلتر به حداقل رسد.

۳. باندگذر فیلتر اولیه F_a برابر است با:

$$\varphi - \theta = M(\omega_s - \omega_p) \quad (48-2)$$

و مجموع باندگذر فیلترهای ماسک گذار $\frac{2\pi}{M}$ می باشد. در نتیجه با افزایش M پیچیدگی فیلتر اولیه

$F_a(z)$ کاهش می یابد ولی در عوض پیچیدگی فیلترهای ماسک گذار افزایش پیدا می کنند. بهترین M

برای کاهش پیچیدگی ساختار کلی مدار از رابطه ی (۴۹-۲) بدست می آید [۱۴]:

$$M_{opt} = \frac{1}{\sqrt{1.6(\omega_s T - \omega_c T)/\pi}} \quad (49-2)$$

ریپل باند عبور و توقف فیلترهای ماسک گذار ۱۰٪ تا ۱۵٪ کوچکتر از فیلتری است که قرار است

طراحی شود [۱۵].

۳.۴.۲.۲ فیلتر FRM با ساختار چند طبقه^۷:

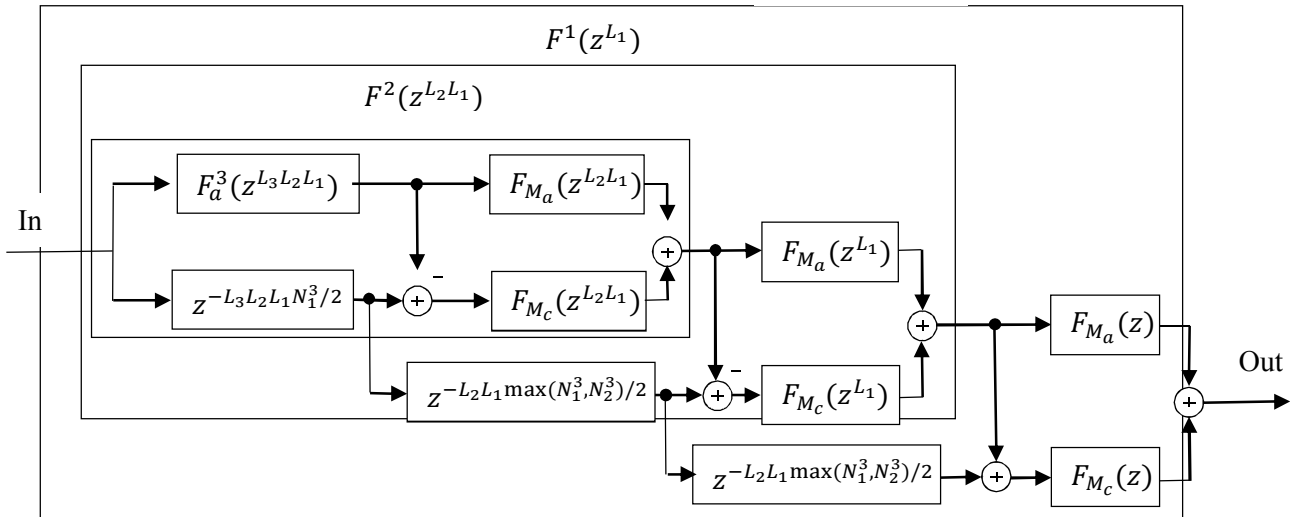
در ساختار اصلی فیلتر FRM اگر فیلتر اولیه $F_a(z)$ باندگذر تیزتری به سبب مرتبه ی بالا فیلتر

داشته باشد، از تکنیک FRM می توان بصورت بازگشتی و برای ایجاد یک ساختار چند طبقه استفاده

^۷ Multi-Stage FRM Filter

کرد. این روش باعث کاهش پیچیدگی کل فیلتر می شود. شکل (۲-۱۱) بلوک دیاگرام یک فیلتر FRM

$$H(z) = F^0(z)$$



شکل (۲-۱۰) فیلتر FRM سه طبقه [۱۶].

تبدیل z کلی سیستم $H(z)$ در یک FRM، K طبقه به صورت رابطه (۲-۵۱) می باشد [۱۶]:

$$H(z) = F^0(z) \quad (۲-۵۰)$$

$$F^{k-1}(z) = F_a^k(z^{L_k})F_{M_a}^k(z) + [z^{-\frac{L_k N_F^k}{2}} - F_a^k(z^{L_k})]F_{M_c}^k(z) \quad (۲-۵۱)$$

برای $k = 1, 2, \dots, K$ ، N_F^k مرتبهی فیلتر $F_a^k(z)$ (فیلتر لبه باند در هر طبقه) و $F_{M_a}^k(z)$ و $F_{M_c}^k(z)$

فیلتر ماسک گذار مرحله k ام است که به ترتیب از مرتبهی N_1^k و N_2^k می باشد. از نظر تئوری به این

نتیجه رسیده اند که تعداد طبقات بهینه که کمترین پیچیدگی ساختاری را در پی داشته باشد از

میزان لازم حداکثر خطای باند توقف مستقل می باشند و منحصرأً توسط پهنای باند گذر بدست می -

آیند. تعداد طبقات بهینه در شرط (۲-۵۲) صدق می کند [۱۶].

$$\beta_b(k) < \omega_s T - \omega_p T \leq \beta_b(k-1) \quad (۲-۵۲)$$

$$\beta_b(k) \cong \frac{1}{4} \left(\frac{k+1}{k+2} \right)^{(k+1)(k+2)} \quad (۲-۵۳)$$

در فیلتر FRM با k_{opt} طبقه (k_{opt} مقدار بهینه طبقات)، ضریب فشردگی در هر طبقه M_k تمایل دارند که با هم برابر شوند و مقدار آن از رابطه‌ی (۵۴-۲) محاسبه می‌گردد [۱۵].

$$\left(\frac{k_{opt+1}}{k_{opt}}\right)^{k_{opt}} \leq M_k \leq \left(\frac{k_{opt+2}}{k_{opt+1}}\right)^{k_{opt+1}} \quad (54-2)$$

در رابطه (۵۴-۲) اگر k بزرگ باشد حد پایین و حد بالا به یک عدد میل می‌کنند.

فصل سوم

الگوریتم‌های تکاملی

مقدمه

الگوریتم‌های تکاملی با بهره‌گیری از جمعیتی از جواب‌ها و با استفاده از عملگرهای خود، سعی در رسیدن به جواب‌های بهینه دارند. همان‌طور که در فصل قبل اشاره شد می‌توان از الگوریتم‌های تکاملی در طراحی فیلتر بهره جست. الگوریتم بهینه‌سازی دسته ذرات یک الگوریتم بهینه‌سازی تکاملی است که جستجو در تمام فضای جواب‌ها را برای یافتن پاسخ بهینه ممکن می‌سازد. علی‌رغم موفقیت‌های خوبی که این الگوریتم بدست آورده است، تلاش‌هایی برای برطرف کردن برخی ضعف‌های آن صورت گرفته است. ترکیب این الگوریتم با الگوریتم فرهنگی نتایج بهتری در محاسبه ضرایب فیلتر حاصل می‌کند.

۱.۳ الگوریتم بهینه‌سازی توده ذرات^۱

روش‌های بهینه‌سازی شده در حالت کلی به دو دسته کلاسیک و تکاملی تقسیم می‌شوند که روش‌های بهینه‌سازی تکاملی، به منظور برطرف کردن برخی نقاط ضعف روش‌های بهینه‌سازی کلاسیک شکل گرفته‌اند. الگوریتم‌های تکاملی بر خلاف روش‌های کلاسیک، بجای استفاده از یک جواب، جمعیتی از جواب‌ها را در فرآیند خود استفاده می‌کنند و با استفاده از عملگرهای خود، سعی در رسیدن به جواب بهینه دارند. همچنین الگوریتم‌های تکاملی این قابلیت را دارند که با ترکیب برخی روش‌های سودمند جواب‌های بهینه‌ی چندگانه را در برخی مسائل خاص بدست آورد.

الگوریتم بهینه‌سازی دسته ذرات (PSO) یک الگوریتم بهینه‌سازی تکاملی می‌باشد که با الهام گرفتن از اصول حاکم بر رفتار گونه‌های اجتماعی موجود در طبیعت از قبیل گروه پرنندگان ابداع گردید [۱۷].

^۱ Particle Swarm Optimization

۱.۱.۳ مزایای الگوریتم‌های PSO

۱.۱.۱.۳ بهره‌مندی از حافظه

الگوریتم بهینه‌سازی دسته ذرات دارای حافظه است به گونه‌ای که دانش راه حل‌های خوب توسط همه‌ی ذرات حفظ می‌شود. به عبارت دیگر در این الگوریتم بهینه‌سازی دسته ذرات، هر ذره از اطلاعات گذشته خود سود می‌برد.

۲.۱.۱.۳ همکاری و اشتراک‌گذاری اطلاعات بین ذرات

در الگوریتم بهینه‌سازی دسته ذرات هر عضو جامعه موقعیت خود را با توجه به تجربیات شخصی تجربیات کل جامعه تغییر می‌دهد. اشتراک اجتماعی اطلاعات بین اعضای یک جامعه یک سری مزیت‌های تکاملی را در پی دارد و این فرضیه پایه و اساس الگوریتم بهینه‌سازی دسته ذرات و توسعه‌ی آن محسوب می‌شود.

۳.۱.۱.۳ سرعت همگرایی بالا

در الگوریتم بهینه‌سازی دسته ذرات، اعضای جامعه با هم ارتباط دارند و از طریق تبادل اطلاعات به حل مسئله می‌رسند و در نتیجه از سرعت همگرایی بالایی برخوردار می‌باشند.

۴.۱.۱.۳ راحتی پیاده‌سازی و اجرا

الگوریتم بهینه‌سازی دسته ذرات از الگوریتم‌های ژنتیک و بهینه‌سازی کلونی مورچه ساده‌تر می‌باشد. همچنین اندازه‌ی جمعیت الگوریتم بهینه‌سازی دسته ذرات کم است و همین امر مقاربتی اولیه جمعیت در این الگوریتم را ساده‌تر کرده‌است. این الگوریتم به آسانی قابل اجرا است و در حل بسیاری از مسائل بهینه‌سازی گسسته و غیرخطی پیوسته مورد استفاده قرار گرفته است. نکته‌ی جالب

اینکه این الگوریتم فقط عملگرهای اصلی ریاضی را به کار می‌برد و نتایج خوبی را در محیط‌های ثابت و نوپزی را فراهم می‌کند.

۲.۱.۳ معایب الگوریتم PSO

۱.۲.۱.۳ همگرایی سریع و گرفتار شدن در بهینه محلی

عمده‌ترین مسأله این الگوریتم همگرایی سریع می‌باشد. در این الگوریتم ذرات به تدریج در فضای جستجو در نزدیکی بهترین نقطه‌ی بهینه‌ی عمومی سیر می‌کنند و بقیه فضا را مورد کاوش قرار نمی‌دهند و مجبور می‌شوند به بهترین موقعیتی که تا کنون کشف کرده‌است همگرا شوند و تضمینی نیست که بهترین راه حل سراسری باشد. در طول فرآیند جستجو ممکن است برخی ذرات در بهینه محلی گرفتار شوند و در اکتشافات بعدی شرکت نمایند. این مسأله در بهینه‌سازی توابع مولتی با ابعاد بالا کاملاً رایج است. ذرات گیر افتاده در بهینه‌ی محلی موجب همگرایی زودرس و راه حل ضعیف می‌شوند.

۲.۲.۱.۳ کاهش تنوع جمعیت

در الگوریتم بهینه‌سازی دسته ذرات پس از مقداردهی اولیه به ذرات، ذرات در جمعیت، معمولاً در کل تکرار پی در پی فرآیند جستجو به بهینه‌ی محلی یا سراسری، در فضای جستجو همگرا می‌شوند و ذرات جمعیت تحت تأثیرپذیری و تبعیت از بهترین ذره کشف شده در کل جمعیت اطراف آن ذره جمع می‌شوند و یا به عبارتی منقبض می‌گردند، در نتیجه در این الگوریتم تنوع به مرور زمان کاهش می‌یابد. با این شرایط برای محیط‌های پویا که در آن با محیط دینامیک رو به رو هستیم و امکان تغییر بهینه سراسری وجود دارد، دیگر ذرات همگرا شده از توانایی جستجوی سراسری مناسبی جهت دنبال کردن نقطه‌ی بهینه برخوردار نخواهند بود.

۳.۱.۳ بروزرسانی موقعیت ذرات در الگوریتم PSO

اگر موقعیت ذره i ام در گام t را $x_i(t)$ و سرعت ذره i ام در گام t را $v_i(t)$ بنامیم، موقعیت ذره i -

ام در گام بعدی به صورت زیر محاسبه می‌شود [۱۸]:

$$x_i(t+1) = x_i(t) + v_i(t) \quad (۱-۳)$$

$$v_i(t+1) = \gamma[\omega \cdot v_i(t) + c_1 \cdot \varphi_1(pbest_i(t) - x_i(t)) + c_2 \cdot \varphi_2(gbest_i(t) - x_i(t))] \quad (۲-۳)$$

که در رابطه‌ی فوق $\omega, c_1, c_2, pbest, gbest$ به ترتیب وزن اینرسی، ضریب اعتماد به نفس، ضریب اعتماد به گروه، مؤلفه‌ی شناختی و مؤلفه‌ی جمعی هستند. در واقع در این روش، معادله‌ی سرعت ضامن حرکت ذرات به سمت ناحیه‌ی بهینه می‌باشد. معادله‌ی (۲-۳) معمولاً بر اساس سه عنصر اصلی ارائه می‌شود که عبارتند از:

- سرعت سکون
- مؤلفه‌ی شناختی ($pbest$)
- مؤلفه‌ی جمعی ($gbest$)

پارامتر رابطه‌ی (۲-۳) به شرح زیر تعریف می‌گردد [۱۹]:

$pbest_i(t)$: بردار بهترین مکانی که ذره i ام در طول تکرار الگوریتم تا لحظه‌ی t ام داشته است.

$gbest$: بردار بهترین مکانی که ذرات تا لحظه‌ی t ام داشته‌اند.

φ_1, φ_2 : اعداد تصادفی که میزان وابستگی سرعت ذره را به $pbest_i(t)$ و $gbest$ مشخص می‌کنند.

C_1, C_2 : اعداد ثابتی هستند که آنها را به ترتیب ضرایب اعتماد به نفس و اعتماد به گروه می‌نامند. C_1, C_2 در مؤلفه‌های شناختی و اجتماعی سرعت ذره تأثیر می‌گذارند و نقش بسیار زیادی در راندمان ذره دارند. اگر $C_1, C_2 = 0$ باشد، ذرات فقط با سرعت خاصی بدون هدف در فضا حرکت می‌نمایند و آنقدر حرکت می‌کنند تا به مرز فضای جستجو برسند. اگر $C_1 > 1, C_2 = 0$ باشد، ذرات فقط به تجربه‌ی فردی خود توجه می‌نمایند. اگر $C_1 = 0, C_2 > 0$ باشد، ذرات فقط به بهترین فرد گروه توجه می‌کنند. معمولاً در بسیاری از موارد C_1, C_2 را ثابت در نظر می‌گیرند. در بعضی موارد سعی شده است پارامترها را به صورت خطی تغییر دهند با این تفاوت که C_1 را کاهش و C_2 را افزایش می‌دهند. این استراتژی بر روی جستجوی عمومی ذرات در فضای جستجو در ابتدای الگوریتم تأکید دارد و در انتهای الگوریتم به جستجوی محلی ذرات اهمیت می‌دهد و ذرات به سمت بهترین موقعیت ملاقات شده گروه جذب می‌شوند. پارامترها به صورت معادله‌ی (۳-۳) به روز می‌گردند [۲۰].

$$\begin{cases} c_2(t) = (c_{2,max} - c_{2,min}) \frac{t}{n_t} + c_{2,max} \\ c_1(t) = (c_{1,max} - c_{1,min}) \frac{t}{n_t} + c_{1,max} \end{cases} \quad (3-3)$$

$$\text{where } \begin{cases} c_{1,max} = c_{2,max} = 2.5 \\ c_{1,min} = c_{2,min} = 0.5 \end{cases}$$

ω (وزن اینرسی): وزن اینرسی است که میزان وابستگی سرعت ذره در هر گام را نسبت به گام قبلی تعیین می‌کند. جهت تنظیم این پارامتر روش‌های زیادی ارائه شده است که در ادامه به آنها می‌پردازیم.

- مقدار تصادفی

وزن اینرسی را می‌توان به صورت به طور تصادفی مقداردهی نمود. یک نوع مقدار دهی استفاده از توزیع گوسین می‌باشد. $\omega \sim N(0.72, \sigma)$ است که σ را به گونه‌ای انتخاب می‌نمایند که ω از ۱ بزرگتر نباشد.

- کاهش خطی وزن اینرسی

وزن اینرسی را در ابتدای الگوریتم مقدار زیادی (معمولاً ۰/۹) در نظر می‌گیرند و سپس به مرور زمان آن را کاهش می‌دهند (معمولاً ۰/۴). معادله این تغییر خطی در رابطه‌ی (۳-۴) نمایش داده شده‌است [۲۱].

$$\omega(t) = (\omega(0) - \omega(n_t)) \frac{(n_t - t)}{n_t} + \omega(n_t) \quad (۳-۴)$$

n_t ماکزیمم تعداد تکرارهای الگوریتم می‌باشد. $\omega(0)$ وزن اینرسی اولیه است. $\omega(n_t)$ مقدار نهایی وزن اینرسی می‌باشد و $\omega(t)$ وزن اینرسی در تکرار t ام الگوریتم می‌باشد.

- کاهش غیر خطی وزن اینرسی

در این حالت وزن اینرسی به طور غیر خطی کاهش می‌یابد. این امر باعث می‌شود که ذره در فضای جستجو بیشتر جستجوی عمومی داشته باشد. با توجه به نوع مسأله تابع غیرخطی متفاوتی مطرح می‌گردد.

۴.۱.۳ تابع شایستگی (برازندگی)

برای حل مسأله بهینه‌سازی، باید یک تابع شایستگی تعریف گردد. این تابع باید به ازای هر جواب (مجموعه‌ای از مقادیر ورودی) که توسط ذرات ارائه می‌شود یک خروجی تولید می‌کنند. به عبارت دیگر، این تابع شایستگی همان تابع هدف است و به گونه‌ای باید طراحی شود که هرچه جواب ارائه

شده به جواب بهینه نزدیک‌تر باشد، خروجی بزرگتری را تولید می‌کند که نشان‌دهنده‌ی شایستگی آن ذره است. و به عبارتی هرچه تابع خطا کوچک‌تر باشد ذره از ارزش بیشتری برخوردار است.

۵.۱.۳ تابع خطا (تابع هدف) در طراحی فیلتر

از مهمترین پارامترهای این الگوریتم، تعریف تابع خطا می‌باشد. این تابع با توجه به مسأله به صورت‌های مختلفی طراحی می‌گردد. در ادامه تابع خطایی که در طراحی فیلتر کاربرد دارند را بررسی می‌کنیم.

تابع خطا Parks-McClellan طبق رابطه‌ی (۵-۳) تعریف می‌شود [۲۲].

$$E(\omega) = G(\omega)[H_d(e^{j\omega}) - H_i(e^{j\omega})] \quad (۵-۳)$$

و پارامترهای این تابع به صورت زیر تعریف می‌شوند:

$$- H_d(e^{j\omega}) : \text{پاسخ فرکانسی فیلتر مطلوب.}$$

$$- H_i(e^{j\omega}) : \text{پاسخ فرکانسی فیلتر ایده‌آل.}$$

- $G(\omega)$: تابع وزن دهی که جهت گردآوری وزن‌های مختلف برای خطاهای تقریبی در فرکانس‌های متفاوت استفاده می‌شود.

بزرگترین عیب این تابع خطا این است که نسبت حداکثر خطای باند عبور به حداکثر خطای باند توقف ($\frac{\delta_p}{\delta_s}$) ثابت است. در نتیجه تابع خطا به صورت زیر ارتقاء پیدا می‌کند.

$$J_1 = \max_{\omega \leq \omega_p} (|E(\omega)| - \delta_p) + \max_{\omega \geq \omega_s} (|E(\omega)| - \delta_s) \quad (۶-۳)$$

پارامترهای این تابع به صورت زیر تعریف می‌شوند:

$$- \delta_p, \delta_s : \text{حداکثر خطای باند توقف و باند عبور.}$$

– ω_p, ω_s : فرکانس نرمالیزه قطع باند توقف و فرکانس قطع باند عبور.

۲.۳ الگوریتم فرهنگی^۲

یک سیستم دوگانه وراثتی است، که دو فضای جستجو را ارائه می‌دهد. یکی فضای جمعیت که بر مبنای نظریه ژنتیکی داروین است و دیگری فضای باور که یک قسمت از فرهنگ را ارائه می‌کند که این مورد وجه تمایز بین الگوریتم ژنتیک با الگوریتم فرهنگی است. فضای اعتقادی بر روی اعضای جمعیت تأثیر می‌گذارد، اعضای جمعیت تأثیری را که گرفته‌اند کمی تغییر می‌دهند و به یک فرهنگ تغییر یافته تبدیل می‌کنند. این روند این قدر ادامه می‌یابد تا فرهنگ به بهترین فرهنگ و اعضای جمعیت به بهترین اعضای جمعیت همگرا شود [۲۳].

برای ارتباط دادن بین این دو فضا یک پروتکل ارتباطی تعریف می‌شود. یکی برای انتخاب گروهی از افراد تا فضای باور را شکل دهند و دیگری روشی برای تأثیر این فضای باور بر روی تولید افراد در فضای جمعیت است.

بطور کلی الگوریتم فرهنگی به صورت زیر عمل می‌کند:

در هر نسل ابتدا افراد مانند الگوریتم ژنتیکی در فضای جمعیت وارد می‌شوند و توسط تابع شایستگی ارزیابی می‌گردند. سپس توسط تابع پذیرش افرادی را که مناسب شکل دادن به فضای باور است را انتخاب می‌کند و تجربیات پذیرفته شده افراد برای ساختن و تغییر فضای باور بکار برده می‌شود (در اینجا فرهنگ شبیه‌سازی می‌شود). فرهنگ ایجاد شده در فضای باور، بر روی تکامل جمعیت در فضای جمعیت تأثیر می‌گذارد.

اجزای الگوریتم فرهنگی به صورت زیر می‌باشد:

² Cultural Algorithm

- فضای جمعیت
- فضای باور
- تابع پذیرش
- تابع تأثیر

که هر کدام در بخش‌های آینده شرح داده خواهد شد.

۱.۲.۳ فضای جمعیت

این فضا در واقع فضای اصلی جمعیت می‌باشد و با مقداردهی اولیه کار خود را شروع می‌کند و استخراج فرهنگ و ذخیره‌ی آن در فضای باور در این قسمت انجام می‌گیرد.

۲.۲.۳ فضای باور

در فضای باور، تجربیات عمومی شده‌ی افراد موفق از فضای جمعیتی بدست می‌آید و این تجارب در سراسر نسل‌های بعدی شکل گرفته و ذخیره می‌شوند. این تجارب بر تمامی نسل‌ها تأثیرگذار است و به نسل‌های آینده منتقل می‌گردد.

در واقع، این فضا برای هرس کردن فضای جمعیت مؤثر است. هر فرد یک ذره در فضای جستجو است که فضای باور برای دور ساختن افراد از ناحیه‌های نامطلوب و سوق دادن آنها به سمت ناحیه‌های امیدبخش و نزدیک به جواب بکار برده می‌شود.

دانش‌های مختلفی فضای باور را تشکیل می‌دهند که به شرح زیر است:

۱.۲.۲.۳ دانش موقعیتی

بهترین پاسخ تمام نسل‌ها و یا نسل جاری را به عنوان دانش موقعیتی در نظر می‌گیریم و طبق رابطه‌ی (۷-۳) به روز می‌گردد.

$$S^{t+1} = \begin{cases} p_{gd}^{t+1} & \text{if } f(p_{gd}^{t+1}) < f(s^t) \\ S^t & \text{Othe} \end{cases} \quad (7-3)$$

موقعیت

۲.۲.۲.۳ دانش معیار

دانش معیار مجموعه بازه‌های خوب و امیدبخش را که از مجموعه‌ای از ذرات خوب استخراج شده- است را برای هر بعد از مسئله نگهداری می‌کند این دانش طبق رابطه‌ی (۸-۳) معرفی می‌گردد.

$$Norm \quad \{X_i, \dots, X_D\} \quad (8-3)$$

در این جا، D معرف تعداد ابعاد مسأله است و هر X_i به صورت رابطه‌ی (۹-۳) تعریف می‌شود:

$$X_i = [l_i \quad u_i \quad L_i \quad U_i] \quad (9-3)$$

که l_i و u_i به ترتیب حد بالا و حد پایین بعد i ام می‌باشند، L_i و U_i مقدار تابع شایستگی در آن حدود می‌باشد [۲۴].

دانش معیار طبق روابط زیر به روز می‌شود :

$$l_i^{t+1} = \begin{cases} x_{j,i} & \text{if } x_{j,i} \leq l_i^t \quad \text{Or } f(\bar{x}_j) < L_i^t \\ l_i^t & \text{Othe} \end{cases} \quad (10-3)$$

$$u_i^{t+1} = \begin{cases} x_{k,i} & \text{if } x_{k,i} \geq u_i^t \quad \text{Or } f(\bar{x}_k) < U_i^t \\ u_i^t & \text{Othe} \end{cases} \quad (11-3)$$

$$L_i^{t+1} = \begin{cases} f(\bar{x}_j) & \text{if } x_{j,i} \leq l_i^t \quad \text{Or } f(\bar{x}_j) < L_i^t \\ L_i^t & \text{Othe} \end{cases} \quad (12-3)$$

$$U_i^{t+1} = \begin{cases} f(\bar{x}_k) & \text{if } x_{k,i} \geq u_i^t \text{ Or } f(\bar{x}_k) < U_i^t \\ U_i^t & \text{Othe} \end{cases} \quad (13-3)$$

طبق این دانش، فضای جستجو رفته رفته کوچکتر و به ناحیه‌های خوب نزدیک‌تر می‌شود.

با تشکیل دانش معیار و دانش موقعیت فضای باور ایجاد می‌گردد.

$$\beta(t) = (S(t), Norm(t)) \quad (14-3)$$

۳.۲.۳ تابع پذیرش

این تابع افراد شایسته را در هر نسل برای شکل‌دهی به فضای باور، انتخاب می‌کند.

۴.۲.۳ تابع تأثیر

باورها در فضای باور برای تغییر دادن افراد و نزدیک کردن آنها به باور سراسری (بهینه کل) به

کار برده می‌شوند که این تغییرات با استفاده از تابع تأثیر، تحقق می‌یابد.

روش‌های متفاوتی برای تغییر پاسخ‌ها وجود دارد که عبارت‌اند از [۲۴]:

۱.۴.۲.۳ صرفاً استفاده از دانش معیار

$$x_{ji}^{t+1} = x_{ji}^t + size(I_i) \cdot N(0,1) \quad (15-3)$$

که در رابطه‌ی (۱۵-۳) $size(I_i) = u_i - l_i$ است و $N(0,1)$ متغییر تصادفی با توزیع نرمال با

میانگین صفر و انحراف معیار ۱ می‌باشد.

۲.۴.۲.۳ استفاده از دانش موقعیت برای تعیین جهت و دانش معیار برای میزان حرکت

$$x_{ji}^{t+1} = \begin{cases} x_{ji}^t + |\text{size}(I_i) \cdot N(0,1)|, & \text{if } x_{ji}^t < s_i^t \\ x_{ji}^t - |\text{size}(I_i) \cdot N(0,1)|, & \text{if } x_{ji}^t > s_i^t \\ x_{ji}^t + \eta \cdot \text{size}(I_i) \cdot N(0,1), & \text{Othe} \end{cases} \quad (16-3)$$

η ضریب ثابت کوچکتر ۱

۳.۴.۲.۳ استفاده از دانش معیار هم برای تعیین جهت و هم برای میزان حرکت

$$x_{ji}^{t+1} = \begin{cases} x_{ji}^t + |\text{size}(I_i) \cdot N(0,1)|, & \text{if } x_{ji}^t < l_i^t \\ x_{ji}^t - |\text{size}(I_i) \cdot N(0,1)|, & \text{if } x_{ji}^t > u_i^t \\ x_{ji}^t + \eta \cdot \text{size}(I_i) \cdot N(0,1), & \text{Othe} \end{cases} \quad (17-3)$$

مراحل الگوریتم فرهنگی را به صورت زیر خلاصه می‌کنیم:

- ۱- تولید جمعیت اولیه و ارزیابی آنها.
- ۲- آماده سازی فضای اعتقادی.
- ۳- انتخاب اعضای جمعیت برای تأثیر بر فرهنگ.
- ۴- تغییر مؤلفه‌های فضای اعتقادی توسط اعضای منتخب.
- ۵- تولید پاسخ‌های جدید با تأثیر فرهنگ بر پاسخ‌های فعلی و ارزیابی آنها.
- ۶- بازگشت به مرحله ۳ در صورت برآورده نشدن شرایط خاتمه.

=

فصل چهارم

الگوریتم پیشنهادی

مقدمه

برای دست یافتن به دقت بهتر از الگوریتم ترکیبی که از ترکیب دو الگوریتم PSO و الگوریتم فرهنگی بوجود آمده است، استفاده می‌کنیم. این الگوریتم ترکیبی علاوه بر آنکه مزیت الگوریتم PSO را دارد، از الگوریتم فرهنگی نیز برای افزایش تنوع جمعیت استفاده می‌کند. در ادامه الگوریتم پیشنهادی برای یک طراحی فیلتر میان‌گذر FRM مطلوب از ابتدا تا پیاده‌سازی آن بر روی FPGA مطرح می‌شود.

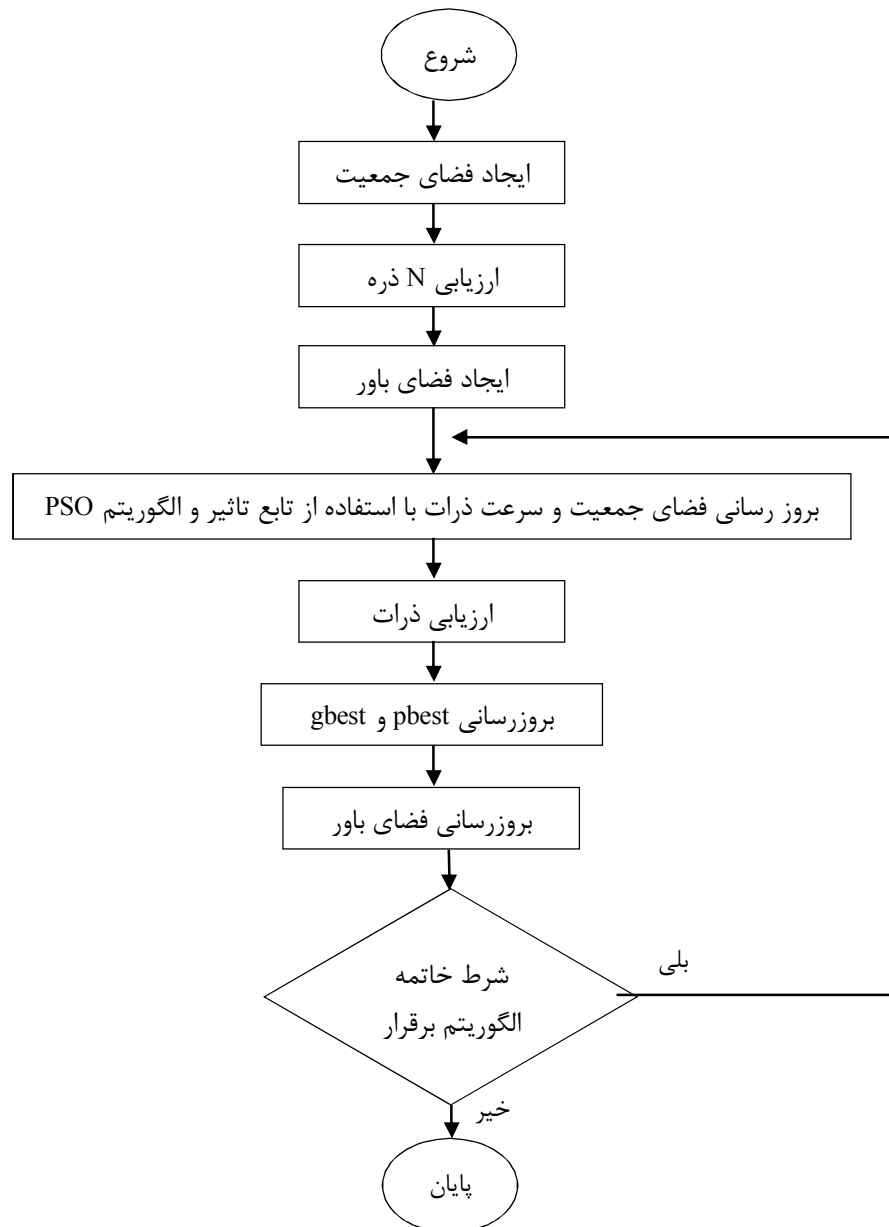
۱.۴ ترکیب الگوریتم‌های بهینه‌سازی توده ذرات و فرهنگی

الگوریتم حاصل از ترکیب الگوریتم PSO و الگوریتم فرهنگی علاوه بر آنکه مزیت الگوریتم PSO را دارد، از الگوریتم فرهنگی نیز برای افزایش تنوع جمعیت استفاده می‌کند، در نتیجه توانایی جستجوی عمومی افزایش پیدا می‌کند. در الگوریتم ترکیبی، فضای باور، جمعیت اولیه را هدایت می‌کند. تنوع جمعیت نیز با استفاده از این فضا افزایش می‌یابد. در نتیجه احتمال گیر افتادن در مینیمم محلی را کاهش می‌دهد. این دو الگوریتم با روش‌های مختلفی با هم ترکیب می‌شوند. نحوه ترکیب این دو الگوریتم ارتباط مستقیمی با نوع مسأله دارد. در [۲۵] از ترکیب این دو الگوریتم برای توزیع بار اقتصادی استفاده شده است. اما در این پایان‌نامه از الگوریتم ترکیبی برای طراحی فیلتر استفاده شده است و علاوه بر آن تابع خطا و وزن اینرسی الگوریتم PSO نسبت به [۲۵] متفاوت خواهد بود. با استفاده از دانش معیار تابع تأثیر موقعیت ذره‌ها را به سمت مکان مناسب تغییر می‌دهد. تابع تأثیر در ترکیب این الگوریتم به صورت رابطه‌ی (۱-۴) تعریف می‌شود [۲۵]:

$$x_{ji}^{t+1} = \begin{cases} x_{ji}^t + \text{abs}(v_{ji}^t) & \text{if } x_{ji} < l_j \\ x_{ji}^t - \text{abs}(v_{ji}^t) & \text{if } x_{ji} > u_j \\ x_{ji}^t + v_{ji}^t, & \text{Otherwise} \end{cases} \quad (1-4)$$

ساختار الگوریتم ترکیبی به صورت زیر تعریف می‌شود و روندنما آن در شکل (۱-۴) نشان داده شده-
است.

- ۱- تشکیل فضای جمعیت و فضای باور.
- ۲- ارزیابی فضای جمعیت توسط تابع شایستگی.
- ۳- بروزرسانی موقعیت و سرعت فضای جمعیت به وسیله‌ی تابع تأثیر رابطه (۱-۴) و رابطه‌ی سرعت (۲-۳).
- ۴- ارزیابی موقعیت جدید فضای جمعیت و بروزرسانی بهترین موقعیت محلی و سراسری.
- ۵- باور انتخاب اعضای جمعیت برای تأثیر بر فرهنگ و بروزرسانی فضای باور.
- ۶- بازگشت به مرحله ۲ در صورت برآورده نشدن شرایط خاتمه.

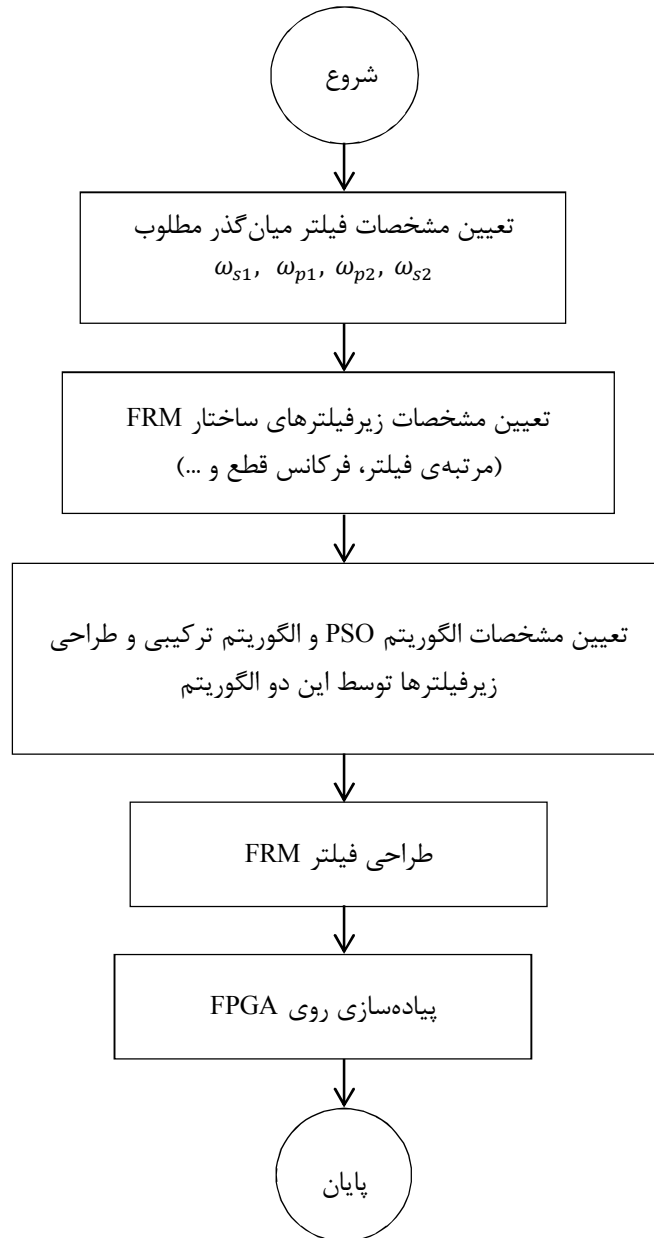


شکل (۴-۱) روندنما ترکیب الگوریتم PSO و فرهنگی.

۲.۴ الگوریتم پیشنهادی

روند طراحی فیلتر میان گذر FRM در این پایان نامه را می توان در روندنما شکل (۴-۲) نمایش

داد:



شکل (۴-۲) فلوچارت الگوریتم ترکیبی.

هر یک از مراحل فوق را در بخش‌های این فصل و نتایج حاصل از شبیه‌سازی را در فصل ۵ مورد

بررسی قرار می‌دهیم.

۱.۲.۴ تعیین مشخصات فیلتر میان‌گذر

در این پایان‌نامه طراحی یک فیلتر میان‌گذر با مشخصات:

$$\omega_{s1} = 0.29\pi, \quad \omega_{p1} = 0.3\pi, \quad \omega_{p2} = 0.7\pi, \quad \omega_{s2} = 0.71\pi \quad (۲-۴)$$

مد نظر می‌باشد.

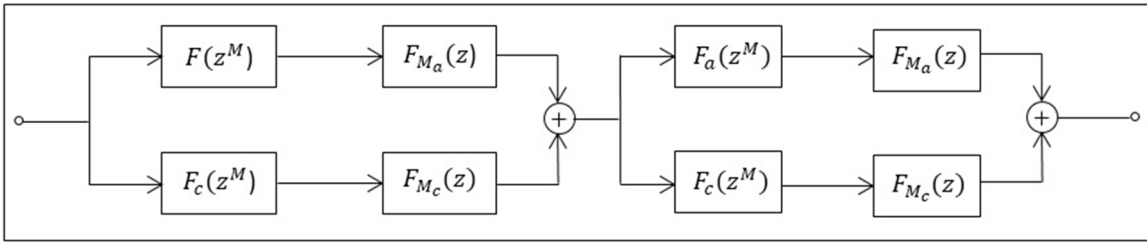
در [۲۶] فیلتری با مشخصات (۲-۴) و تکنیک FRM توسط الگوریتم ژنتیک طراحی شده است. انتخاب این مشخصات به این دلیل است که در نهایت بتوان مقایسه‌ای بین این روش و الگوریتم ژنتیک انجام داد.

۲.۲.۴ تعیین مشخصات زیرفیلترها ساختار FRM

همان‌طور که در فصل دوم اشاره شد، تکنیک FRM روشی برای ساخت فیلتری با باندگذر باریک و تعداد ضرایب فیلتر کم‌تر می‌باشد. ساختار مطرح شده در فصل دوم برای طراحی یک فیلتر پایین‌گذر می‌باشد. از آنجایی که حداکثر تضعیف فیلتر در باند توقف به مرتبه‌ی فیلتر و نوع الگوریتم تکاملی بستگی دارد، تعیین مرتبه‌ی دقیق زیرفیلترها با سعی و خطا بعد از طراحی آنها با الگوریتم تکاملی و مشاهده نتایج حاصل امکان‌پذیر می‌باشد. یکی از روش‌ها برای طراحی فیلتر میان‌گذر، متوالی کردن یک فیلتر پایین‌گذر با یک فیلتر بالاگذر می‌باشد.

$$F(z) = F_{lp}(z)F_{hp}(z) \quad (۳-۴)$$

بر این اساس طراحی فیلتر میان‌گذر با باند عبور باریک، طبق بلوک دیاگرام (۳-۴) صورت می‌پذیرد.



شکل (۴-۳) بلوک دیاگرام فیلتر میان گذر FRM

در نتیجه $F_{lp}(z)$ و $F_{hp}(z)$ از طریق روابط (۴-۴) و (۴-۵) محاسبه می گردد

$$F_{lp}(z) = F_{a_{lp}}(z^M)F_{M_{a_{lp}}}(z) + F_{c_{lp}}(z^M)F_{M_{c_{lp}}}(z) \quad (۴-۴)$$

$$F_{hp}(z) = F_{a_{hp}}(z^M)F_{M_{a_{hp}}}(z) + F_{c_{hp}}(z^M)F_{M_{c_{hp}}}(z) \quad (۴-۵)$$

یک روش دیگر برای طراحی فیلتر میان گذر با باندگذر باریک، استفاده از ساختار اصلی فیلتر FRM است با این تفاوت که بجای فیلترهای ماسک گذار پایین گذر از فیلترهای ماسک گذار میان گذر استفاده می کنیم [۲۶]. کاهش حجم محاسبات و فضای اشغال شده در روش دوم کاملاً مشهود است. این روش نسبت به روش اول بهتر خواهد بود زیرا از فیلترهای ماسک گذار کمتری استفاده می شود. در این پایان نامه از روش دوم استفاده می کنیم.

از مفروضات مسأله است. مقادیر φ, θ, M, m باید محاسبه گردند. ضریب فشردگی بهینه (M_{opt}) از رابطه ی (۲-۴۹) بدست می آید که با توجه به مقادیر $\omega_{p_2}, \omega_{s_2}$ برابر ۸ می گردد. فرکانس قطع باند عبور (θ) و فرکانس قطع باند توقف (φ) فیلتر F_a ، در ساختار FRM طبق روابط زیر محاسبه می گردد.

باتوجه به مقادیر $\omega_{p_2}, \omega_{s_2}, M$ حالت دوم فیلتر FRM اتفاق افتاده است.

$$\omega_{p_2} = \frac{2m\pi - \varphi}{M} \rightarrow 0.7\pi = \frac{2m\pi - \varphi}{8} \rightarrow \varphi = 0.4\pi$$

$$\omega_{s_2} = \frac{2m\pi - \theta}{M} \rightarrow 0.71\pi = \frac{2m\pi - \theta}{8} \rightarrow \theta = 0.32\pi$$

مشخصات فیلترهای ماسک گذار میان گذر طبق مرجع [۲۶] برابر است با جدول (۴-۱) می باشد.

جدول (۴-۱) مشخصات زیرفیلترهای FRM میان‌گذر مطلوب

فرکانس قطع باند توقف	فرکانس قطع باند عبور	زیر فیلتر
$0/4\pi$	$0/32\pi$	F_a
$0/32\pi$	$0/4\pi$	F_c
$0/29\pi$ و $0/71\pi$	$0/45\pi$ و $0/55\pi$	F_{Ma}
$0/21\pi$ و $0/79\pi$	$0/3\pi$ و $0/7\pi$	F_{Mc}

۳.۲.۴ طراحی زیرفیلترها توسط الگوریتم تکاملی

در این پایان‌نامه از سه الگوریتم PSO ساده، PSO با وزن اینرسی غیرخطی و همچنین الگوریتم ترکیبی (ترکیب الگوریتم فرهنگی و PSO) برای طراحی فیلتر استفاده شده و نتایج استفاده از هر کدام با یکدیگر مقایسه شده است.

۱.۳.۲.۴ طراحی فیلتر FIR توسط الگوریتم PSO

برای طراحی یک فیلتر FIR توسط الگوریتم PSO، باید پارامترهای فیلتر همچون ω_s ، ω_p و مرتبه‌ی فیلتر (N) و همچنین مشخصات الگوریتم PSO مثل تعداد ذرات اولیه، C_1 ، C_2 ، کمترین خطا تا لحظه اکنون و ماکزیمم تکرار در ابتدای برنامه مقاردهی شوند که در این پایان‌نامه به ترتیب با مقادیر 50، 1.49، 1.49، $+\infty$ ، 1000 مقاردهی می‌شوند. برای اینکه ذرات در ابتدای برنامه شرایط یکسانی داشته باشند و تحت تأثیر مقدار اولیه سرعت قرار نگیرند، مقدار اولیه سرعت برای تمام ذرات صفر مقاردهی می‌شود. در تکرار اول بردار سرعت توسط رابطه‌ی (۳-۲) مقدار به خود می‌گیرد. تعداد بعد ذرات اولیه برابر با مرتبه‌ی فیلتر است. به‌طور مثال برای طراحی یک فیلتر با مرتبه‌ی ۳۰، نیازمند ذرات اولیه‌ی ۳۰ بعدی می‌باشیم.

از آنجایی که رابطه‌ی خطا در حوزه‌ی فرکانس تعریف می‌گردد ((۳-۵) و ((۳-۶))، پاسخ فرکانسی ذرات اولیه (هرکدام پاسخ ضربه یک فیلتر با مرتبه‌ی N هستند) را بدست می‌آوریم. در این پایان‌نامه بعد از محاسبه‌ی $E(\omega)$ ، تابع خطا (تابع هدف) را از رابطه‌ی ((۳-۶) محاسبه می‌کنیم. هر چند محاسبه‌ی FFT ذرات در هر تکرار الگوریتم حجم محاسبات را افزایش می‌دهد ولی این اتفاق در محاسبه‌ی ضرایب فیلتر دیجیتال در دیگر الگوریتم‌های تکاملی همچون الگوریتم ژنتیک، الگوریتم لانه زنبوری، الگوریتم فرهنگی و سایر الگوریتم‌هایی که رابطه‌ی خطای آنها در حوزه‌ی فرکانس تعریف می‌شود، اتفاق می‌افتد.

در PSO ساده وزن اینرسی ثابت در نظر گرفته شده است و در الگوریتم PSO با وزن اینرسی غیرخطی، این پارامتر طبق رابطه‌ی ((۴-۶) در هر تکرار بروزرسانی می‌گردد.

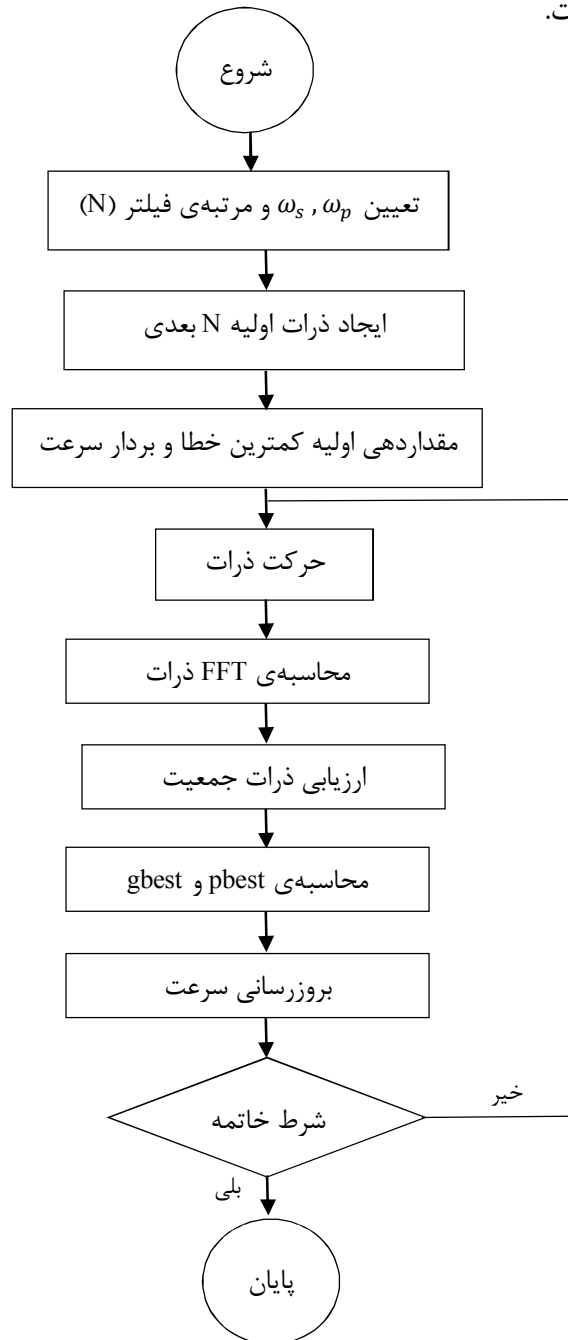
$$\omega^{t+1} = 0.999\omega^t \quad (۴-۶)$$

بهبود الگوریتم PSO در حالت دوم نسبت به حالت اول در نتایج شبیه‌سازی در فصل ۵ کاملاً مشهود است. روند طراحی فیلتر با استفاده از الگوریتم PSO به‌طور خلاصه در شکل ((۴-۴) قابل مشاهده است. زمان اجرای طراحی یک فیلتر پایین‌گذر با مرتبه ۴۰، تعداد ذرات ۵۰ و تعداد تکرار ۵۰۰ با روش PSO در MATLAB برابر ۳۲/۶۶۸ ثانیه است.

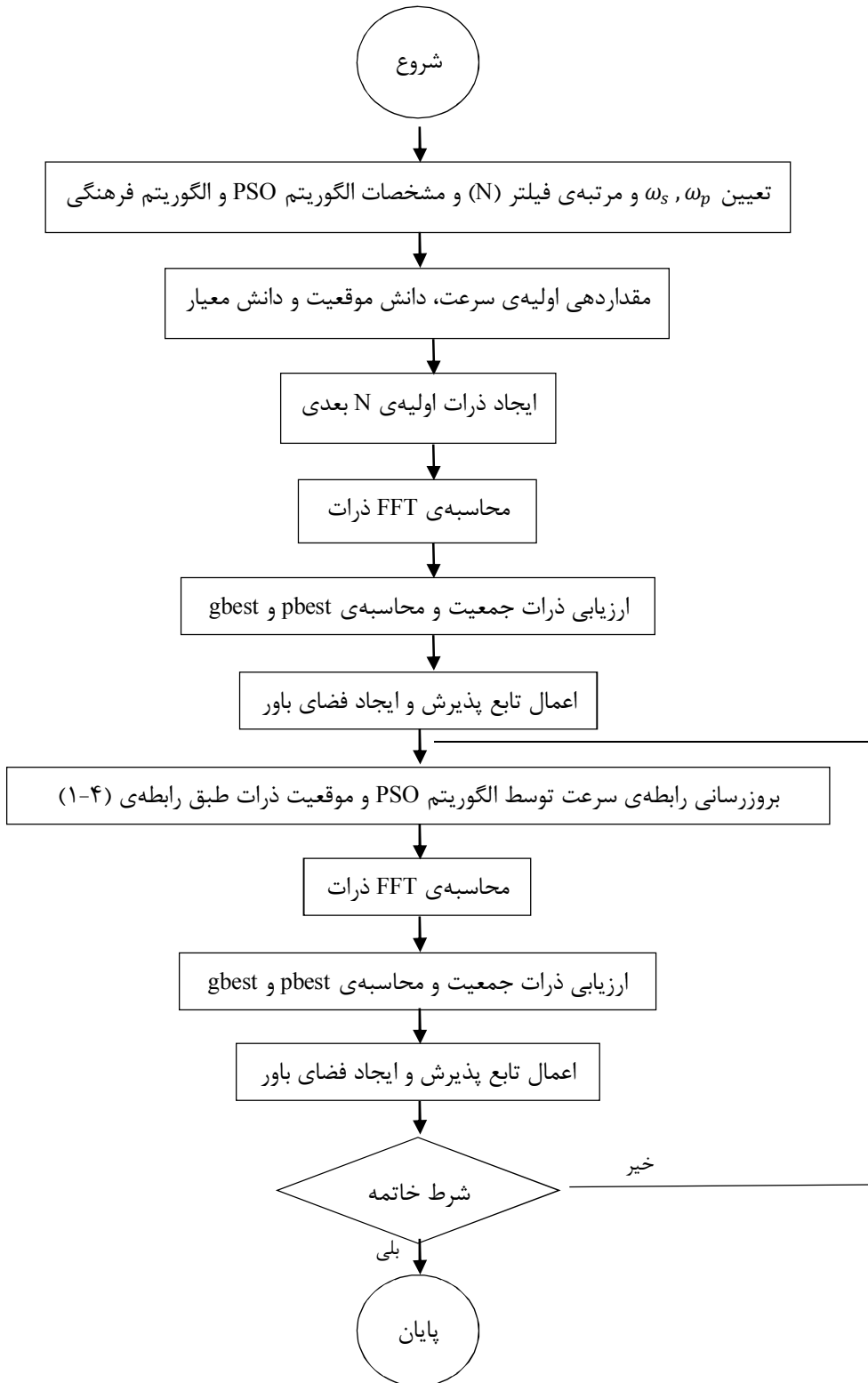
۲.۳.۲.۴ طراحی فیلتر FIR توسط الگوریتم ترکیبی

برای طراحی فیلتر FIR با این روش باید پارامترهای فیلتر و مشخصات الگوریتم PSO همانند قسمت قبل در ابتدا برنامه تعریف گردد. علاوه بر آنها مشخصات الگوریتم فرهنگی همچون تابع پذیرش، مقداردهی اولیه‌ی دانش موقعیت و دانش معیار انجام می‌پذیرد. در این پایان‌نامه تابع پذیرش برابر $0/2$ ، دانش معیار برابر $+\infty$ و حد بالا و پایین دانش معیار و تابع شایستگی آنها به ترتیب $+\infty$ ، $-\infty$ ، $+\infty$ می‌باشند. مراحل انجام کار را می‌توان در شکل ((۴-۵) مشاهده کرد. نتایج

شبیه‌سازی در فصل ۵ نشان خواهد داد که الگوریتم ترکیبی نسبت به الگوریتم PSO ساده و الگوریتم PSO با وزن اینرسی غیرخطی از دقت بالاتری برخوردار است ولی حجم محاسبات در الگوریتم ترکیبی به علت ترکیب دو الگوریتم تکاملی افزایش می‌یابد به طوریکه زمان اجرای طراحی یک فیلتر پایین‌گذر با مرتبه ۴۰، تعداد ذرات ۵۰ و تعداد تکرار ۵۰۰ با روش الگوریتم ترکیبی در MATLAB برابر ۳۵/۶۴۸ ثانیه است.



شکل (۴-۴) طراحی فیلتر با استفاده از الگوریتم PSO.



شکل (۴-۵) طراحی فیلتر با استفاده از الگوریتم ترکیبی.

۴.۲.۴ طراحی فیلتر FRM

بعد از طراحی زیرفیلترهای $F_{M_c}(z), F_{M_a}(z), F_c(z^M), F_a(z^M)$ توسط الگوریتم‌های تکاملی، فیلتر FRM طبق رابطه‌ی (۴-۶) بدست می‌آید.

$$F_{b_p} = F_a(z^M)F_{M_a} + F_c(z^M)F_{M_c} \quad (۴-۶)$$

۵.۲.۴ پیاده‌سازی روی FPGA

FPGAها از عناصر منطقی که برای کار خاصی محدود نشده‌اند و نیز از اتصالات قابل برنامه‌ریزی تشکیل شده‌اند. در نتیجه می‌توان آنها را به هر مدار منطقی یا دیجیتال دلخواه تبدیل نمود. پیاده‌سازی آسان مدارهای دیجیتالی بر روی آنها، امکان آزمون سریع، برنامه‌ریزی مجدد آنها، قابلیت پردازش موازی، از مزایای این تراشه محسوب می‌گردد.

در این پایان‌نامه طراحی FPGA با نرم‌افزار system generator انجام می‌شود. این نرم‌افزار توسط شرکت Xilinx معرفی شده‌است و یک ابزار طراحی الگوریتم‌های پردازش سیگنال می‌باشد. کمترین قابلیت آن ایجاد یک محیط شبیه‌سازی همانند سیمولینک MATLAB برای طراحی FPGA است.

در این نرم‌افزار طراحی‌های الگوریتم‌های پردازش سیگنال به صورت تصویری و با کمک مجموعه بلوک‌هایی که نرم‌افزار در اختیار کاربر قرار می‌دهد، انجام می‌شود. تمام مراحل شبیه‌سازی از قبیل سنتز، آماده‌سازی فایل شبیه‌سازی تا تولید یک فایل برنامه‌ریزی برای FPGA به صورت خودکار انجام می‌شود.

مهمترین کاربرد system generator در پردازش سیگنال‌های دیجیتال است بطوریکه طراحی فیلتر به وسیله‌ی این نرم‌افزار به راحتی امکان‌پذیر است ولی در طراحی فیلتر با کدنویسی با سختی‌های زیادی از جمله کار با انواع داده‌ها (داده‌های ممیز ثابت و ممیز شناور)، مشکلات تبدیل آنها به یکدیگر، سختی شبیه‌سازی و طولانی بودن کار کدنویسی مواجه می‌شویم.

برای پیاده‌سازی یک فیلتر میان‌گذر FRM بر روی FPGA توسط نرم‌افزار system generator از بلوک‌های منبع تصادفی^۱، پورت ورودی، پورت خروجی، جمع‌کننده، ضرب‌کننده، بلوک تأخیر و تحلیل‌گر طیفی^۲ استفاده می‌شود.

^۱ Random Source

^۲ Spectrum Analyzer

۵ فصل پنجم شبیه‌سازی

شبیه‌سازی

مقدمه

همان طور که در فصل قبل اشاره کردیم برای محاسبه ی ضرایب فیلتر از سه الگوریتم تکاملی استفاده شده است. محیط اجرای این الگوریتم ها نرم افزار MATLAB است و نتایج حاصل از شبیه سازی برتری الگوریتم ترکیبی نسبت به دو روش دیگر را نشان می دهد. با نرم افزار system generator امکان پیاده سازی فیلتر میان گذر FRM بر روی FPGA فراهم می شود.

۱.۵ طراحی ضرایب فیلتر توسط الگوریتم های بهینه سازی

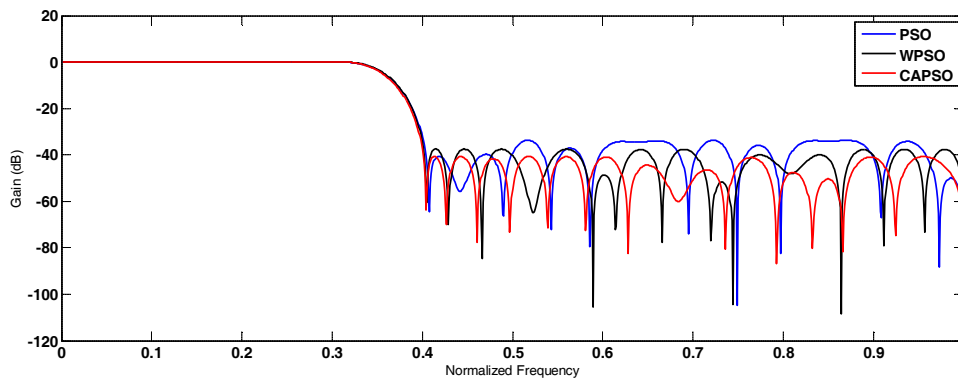
در الگوریتم های بهینه سازی، مقدار فیلتر در فرکانس قطع به مرتبه ی فیلتر و همچنین فرکانس قطع باند عبور (ω_p) و فرکانس قطع باند توقف (ω_s) بستگی دارد. در ادامه به تأثیر پهنای باند گذر متفاوت و همچنین مرتبه ی فیلتر بر روی پارامترهای اصلی فیلتر می پردازیم. همان طور که در بخش قبل اشاره کردیم برای محاسبه ی ضرایب فیلتر از سه الگوریتم تکاملی استفاده شده است. محیط اجرای این الگوریتم ها نرم افزار MATLAB است و مقایسه بین این روش ها در شکل ها و جدول ها زیر بیان گردیده است.

۱.۱.۵ تأثیر پهنای باند گذر در طراحی فیلتر

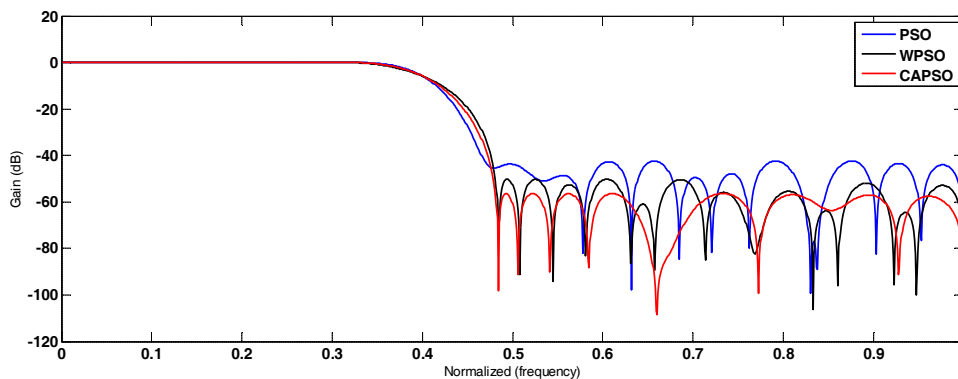
در این بخش تأثیر پهنای باند گذر متفاوت بر روی پارامترهای فیلتر از جمله حداکثر خطا باند توقف (δ_s) و بیشترین مقدار پاسخ فرکانسی فیلتر در باند توقف بررسی می شود. تأثیر پهنای باند گذر

متفاوت برای دو فیلتر با مشخصات $f_1 = \begin{cases} \omega_{p1} = 0.32\pi \\ \omega_{s1} = 0.4\pi \end{cases}$ و $f_2 = \begin{cases} \omega_{p2} = 0.32\pi \\ \omega_{s2} = 0.48\pi \end{cases}$ و مرتبه ۵۰

مورد بررسی قرار می دهیم و نتایج آن به شرح زیر می باشد.



شکل (۱-۵) پاسخ فرکانسی فیلتر f_1 بر حسب dB (پهنای باند گذر 0.08π).



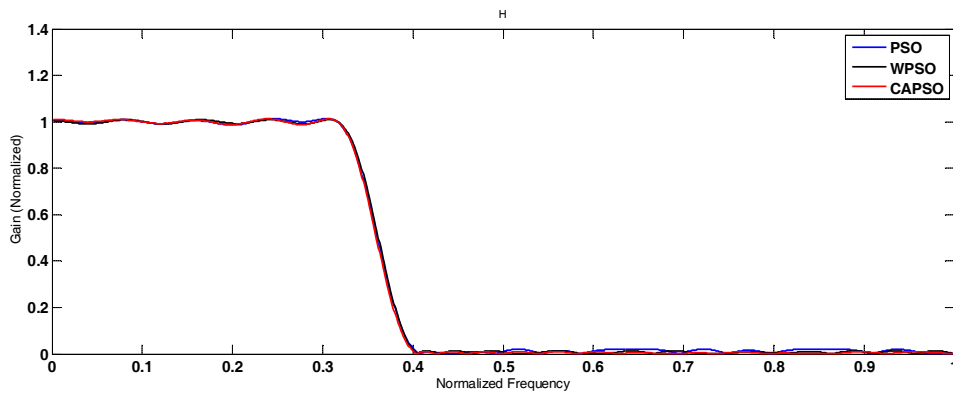
شکل (۲-۵) پاسخ فرکانسی فیلتر f_2 بر حسب dB (پهنای باند گذر 0.16π).

جدول (۱-۵) ماکزیمم مقدار پاسخ فرکانسی فیلتر در باند توقف.

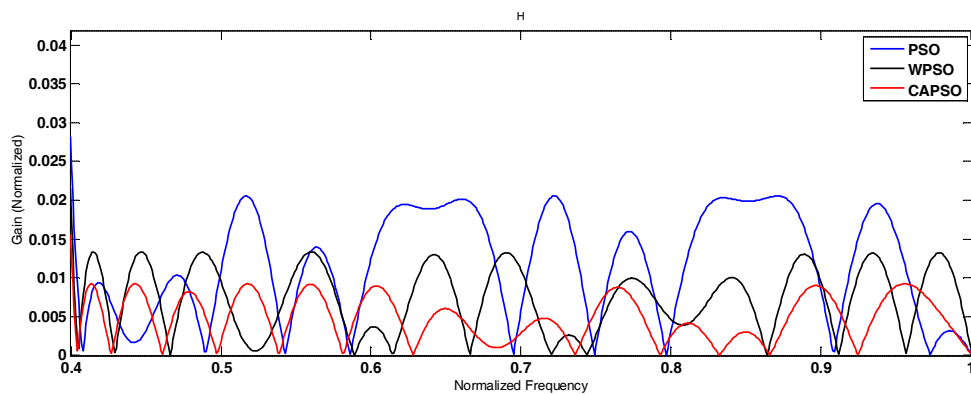
	PSO(dB)	WPSO(dB)	CAPSO(dB)
f_1	-۳۴/۷۷	-۳۷/۵۲	-۴۰/۷۵
f_2	-۴۳/۸۶	-۵۰/۳۴	-۵۵/۳۲

شکل (۱-۵) و (۲-۵) پاسخ فرکانسی فیلتر f_1 و f_2 را بر حسب dB نشان می‌دهد. طبق جدول (۱-۵) که نتایج حاصل از شبیه‌سازی را ارائه می‌کند با افزایش پهنای باندگذر، تضعیف فیلتر افزایش یافته است. همچنین طبق جدول (۱-۵) الگوریتم ترکیبی در پهنای باندگذر 0.08π تضعیف باند توقف را ۱۷٪ و ۸٪ نسبت به الگوریتم PSO و WPSO افزایش داده است. در پهنای باندگذر 0.16π

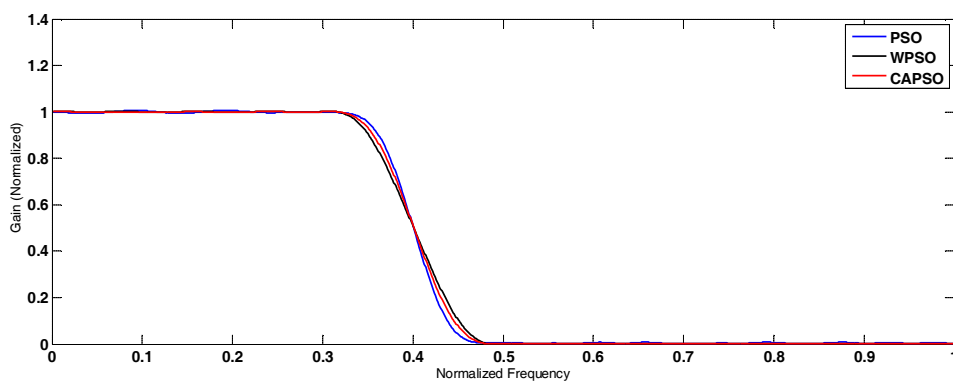
نیز الگوریتم ترکیبی بهترین عملکرد را داشته است بطوریکه تضعیف باند توقف ۲۶٪ و ۱۰٪ نسبت به الگوریتم PSO و WPSO افزایش یافته است. حداکثر خطا ناحیه‌ی توقف (δ_s) از پارامترهای مهم فیلتر به شمار می‌رود. در نتیجه تأثیر پهنای باندگذر متفاوت و مرتبه‌ی فیلتر بر روی آن مورد بررسی قرار گرفته است.



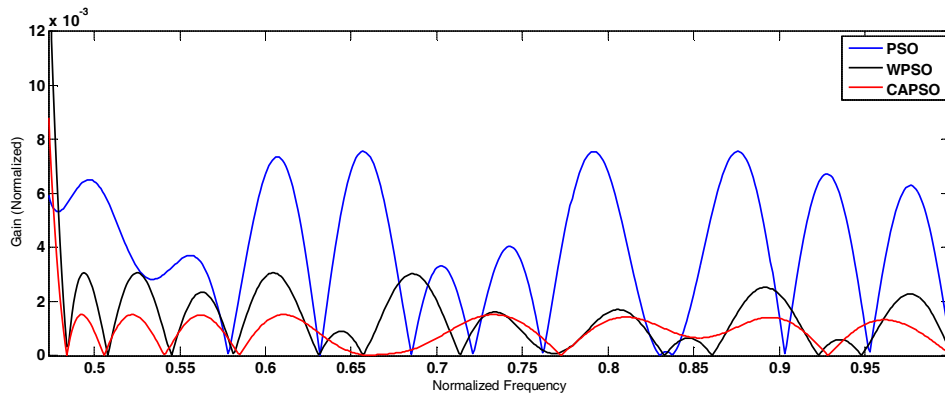
شکل (۳-۵) پاسخ فرکانسی f_1 .



شکل (۴-۵) پاسخ فرکانسی ناحیه‌ی توقف f_1 .



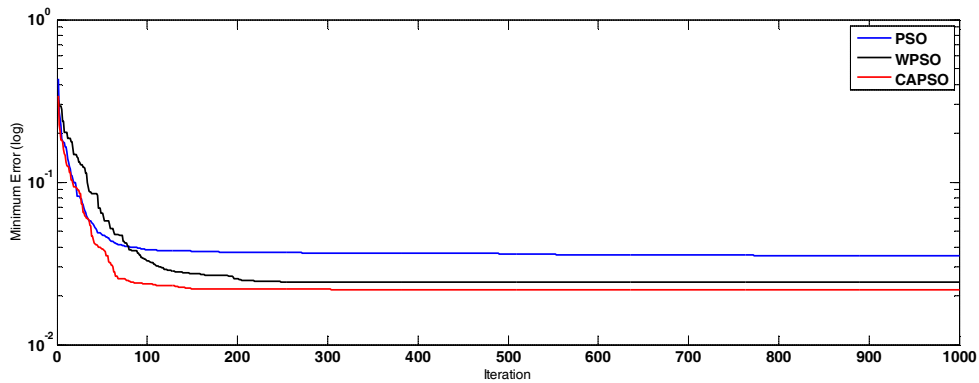
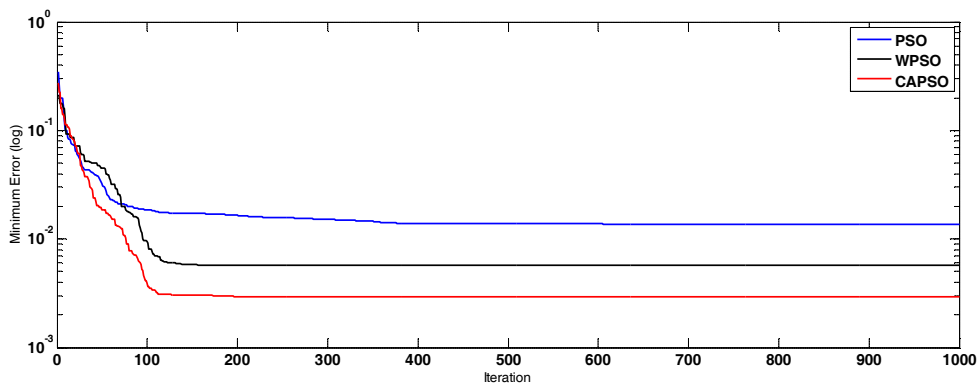
شکل (۵-۵) پاسخ فرکانسی f_2 .

شکل (۵-۶) پاسخ فرکانسی باند توقف f_2 .جدول (۵-۲) حداکثر خطا باند توقف f_1 و f_2

$\delta_s(\text{normalized})$	PSO	WPSO	CAPSO
f_1	۰/۰۲۰۰۱	۰/۰۱۳۳۱	۰/۰۰۹۲۰۸
f_2	۰/۰۰۷۳۵۳	۰/۰۰۳۰۴۶	۰/۰۰۱۵۱۶

شکل های (۵-۳) و (۵-۵) پاسخ فرکانسی فیلتر f_1 و f_2 در حالت عادی (اندازه بر حسب dB نیست) و شکل های (۵-۴) و (۵-۶) پاسخ فرکانسی فیلتر f_1 و f_2 را در ناحیه ی توقف را نشان می-دهند. جدول (۵-۲) نشان می دهد که با افزایش باندگذر حداکثر خطای باند توقف در هر سه الگوریتم کاهش یافته است و همچنین الگوریتم ترکیبی عملکرد بهتری نسبت به سایر الگوریتم ها دارد به گونه ای که در پهنای باندگذر 0.08π حداکثر خطای باند توقف $2/17$ و $1/44$ برابر نسبت به الگوریتم PSO و WPSO کاهش یافته است. در پهنای باندگذر 0.16π نیز الگوریتم پیشنهادی بهترین عملکرد را داشته است بطوریکه حداکثر خطا باند توقف $6/5$ و $2/7$ برابر نسبت به الگوریتم PSO و WPSO کاهش یافته است.

در ادامه مقدار کمترین خطا در تکرارهای الگوریتم ها را بررسی می کنیم.

شکل (۷-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 .شکل (۸-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_2 .

جدول (۳-۵) کمترین خطا.

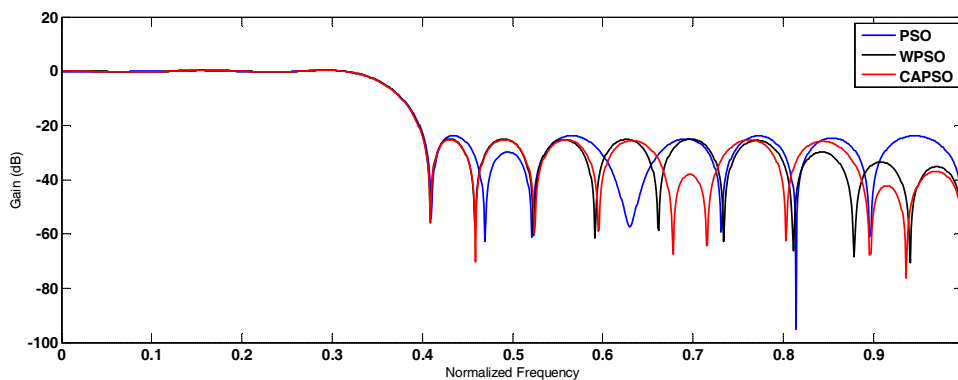
	PSO	WPSO	CAPSO
f_1	۰/۰۳۵۱۹	۰/۰۲۴۲۲	۰/۰۲۱۷۸
f_2	۰/۰۱۳۶۸	۰/۰۰۵۶۷۸	۰/۰۰۲۹۴۷

کمترین خطا نسبت به فیلتر ایده آل طبق جدول (۳-۵) در الگوریتم ترکیبی برای حالت اول ۱/۶۱ و ۱/۱۱ برابر و برای حالت دوم ۴/۶۴ و ۱/۹۲ برابر نسبت به الگوریتم PSO و WPSO بهبود یافته است. یک مزیت مهم الگوریتم ترکیبی نسبت به الگوریتم WPSO، سرعت همگرایی بالاتر آن است. این مزیت در مواردی که کاهش عملیات محاسباتی مد نظر است و ویژگی مهمی محسوب می‌گردد (شکل (۷-۵) و (۸-۵)).

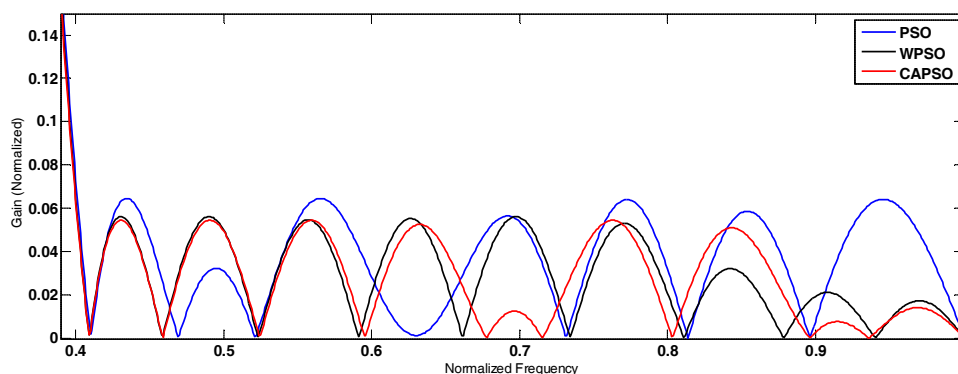
۲.۱.۵ تأثیر مرتبه‌ی فیلتر در طراحی فیلتر

از عوامل تأثیرگذار بر روی پارامترهای فیلتر، مرتبه‌ی فیلتر است. از نظر تئوری به میزانی که مرتبه‌ی فیلتر بیشتر باشد، فیلتر تضعیف بیشتری در باند توقف دارد. این ویژگی مطلوب در قبال پیچیدگی ساختاری فیلتر بدست می‌آید. در این بخش فیلتر پایین‌گذر f_1 در مرتبه‌های ۳۰، ۴۶ و ۶۰ را با هر سه الگوریتم بهینه‌سازی شبیه‌سازی می‌کنیم و تغییر پارامترهای فیلتر را با افزایش مرتبه‌ی آن بررسی می‌کنیم.

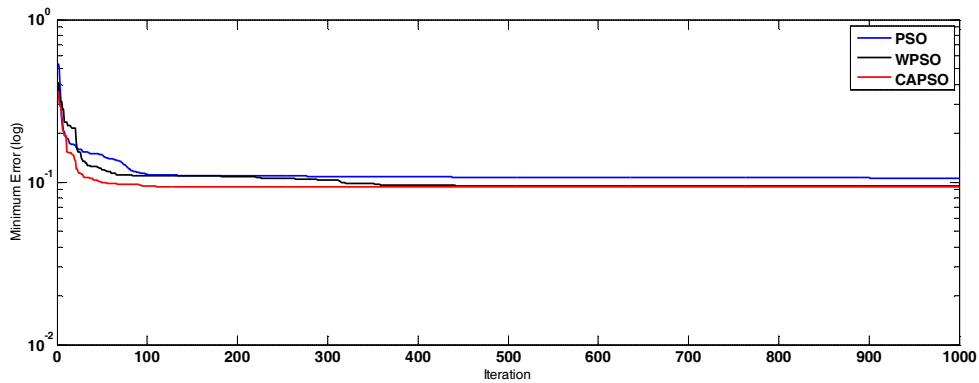
۱.۲.۱.۵ مرتبه $N=30$



شکل (۹-۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۳۰.



شکل (۱۰-۵) پاسخ فرکانسی باند توقف فیلتر f_1 مرتبه ۳۰.



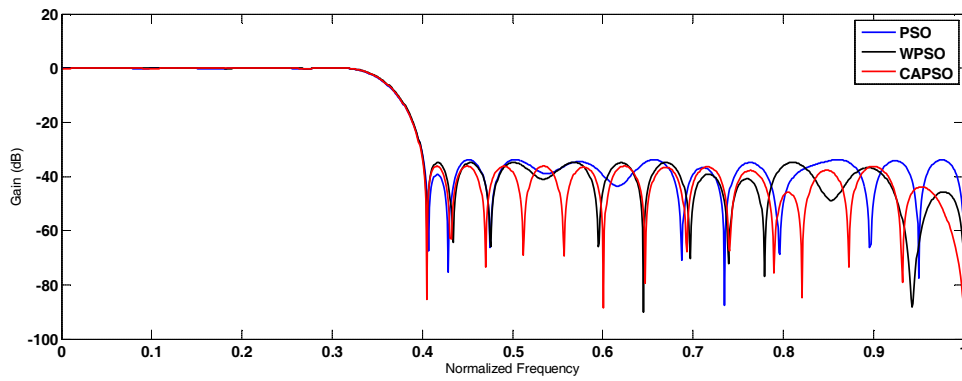
شکل (۱۱-۵) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۳۰.

جدول (۴-۵) فیلتر f_1 مرتبه ۳۰.

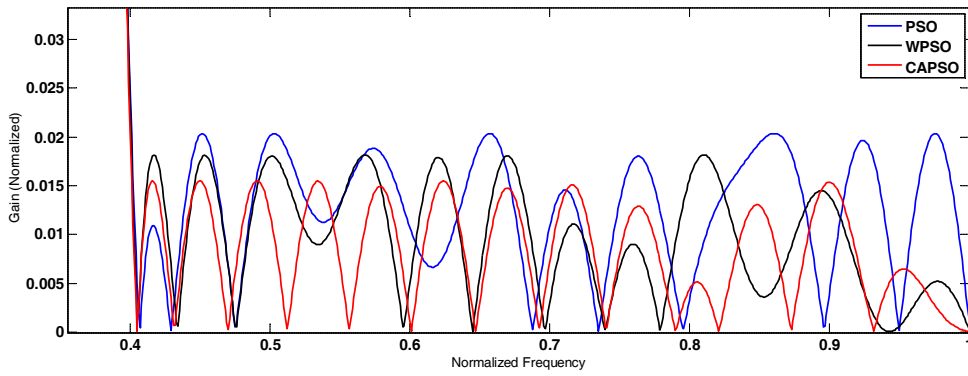
$N=30$	تضعیف باند توقف (dB)	حداکثر خطای باند توقف (normalized)	کمترین خطا
PSO	-۲۳/۸	۰/۰۶۴۵۶	۰/۱۰۵۴
WPSO	-۲۵	۰/۰۵۶۲۲	۰/۰۹۴۸۳
CAPSO	-۲۵/۲۷	۰/۰۵۴۴۹	۰/۰۹۱۲۸

شکل (۹-۵) پاسخ فرکانسی فیلتر f_1 با مرتبه ۳۰، شکل (۱۰-۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۳۰ در ناحیه‌ی توقف را نشان می‌دهند و شکل (۱۱-۵) کمترین خطا در هر تکرار الگوریتم را مشخص می‌کند. جدول (۴-۵) نتایج حاصل از این شبیه‌سازی را بیان می‌کند. مرتبه ۳۰ نسبت به سایر مرتبه‌ها از پیچیدگی کمتری برخوردار است. در این حالت الگوریتم ترکیبی تضعیف باند توقف را ۶٪ و ۱٪ نسبت به الگوریتم PSO و WPSO افزایش داده است.

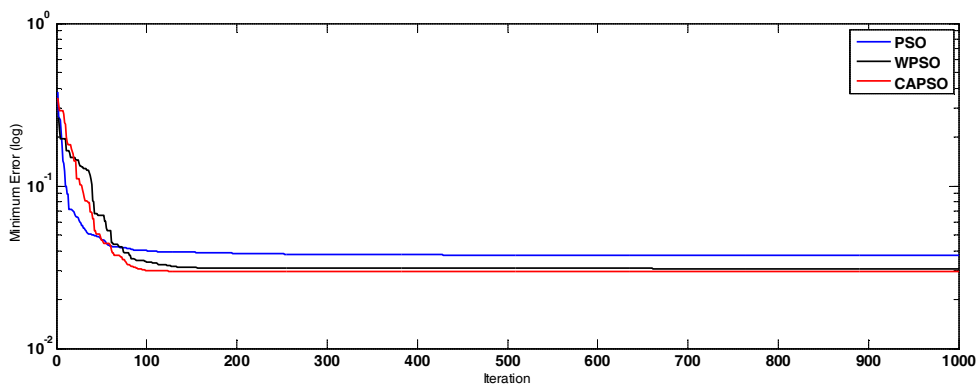
۲.۲.۱.۵ مرتبه $N=46$



شکل (۵-۱۲) پاسخ فرکانسی فیلتر f_1 مرتبه ۴۶.



شکل (۵-۱۳) پاسخ فرکانسی باند توقف فیلتر f_1 مرتبه ۴۶.



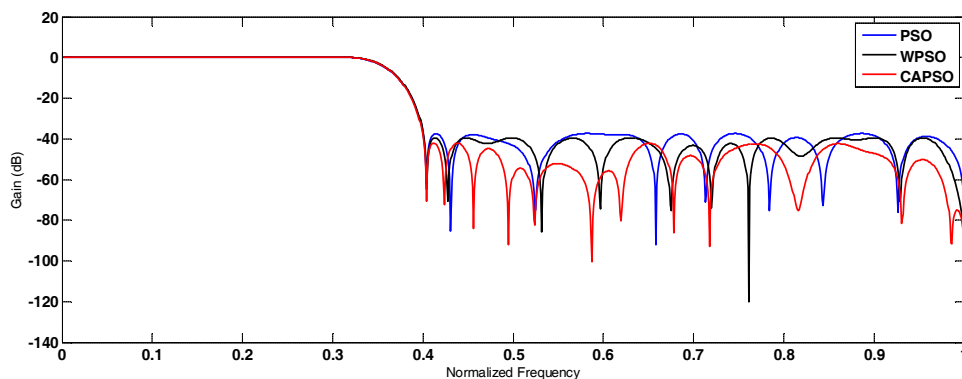
شکل (۵-۱۴) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۴۶.

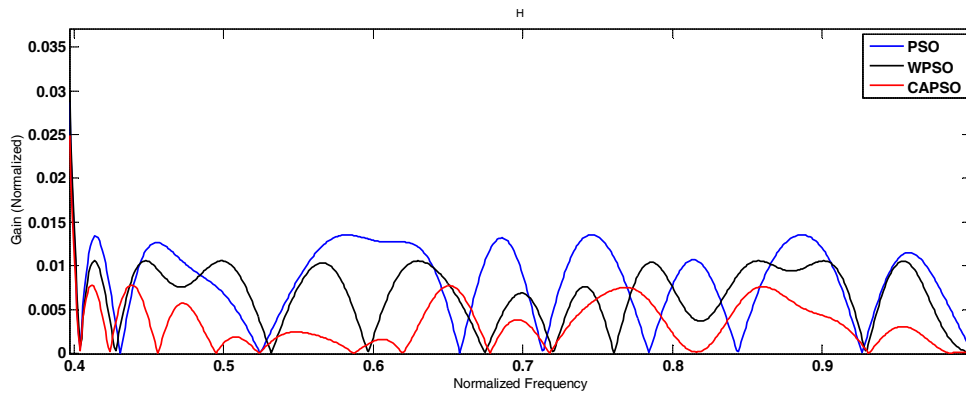
جدول (۵-۵) فیلتر f_1 مرتبه ۴۶.

N=۴۶	تضعیف باند توقف (dB)	حداکثر خطای باند توقف (normalized)	کمترین خطا
PSO	-۳۳/۸۳	۰/۰۲۰۳۴	۰/۰۳۷۶۳
WPSO	-۳۴/۸۵	۰/۰۱۸۱۶	۰/۰۳۱۰۶
CAPSO	-۳۶/۲۱	۰/۰۱۵۴۸	۰/۰۲۹۷۴

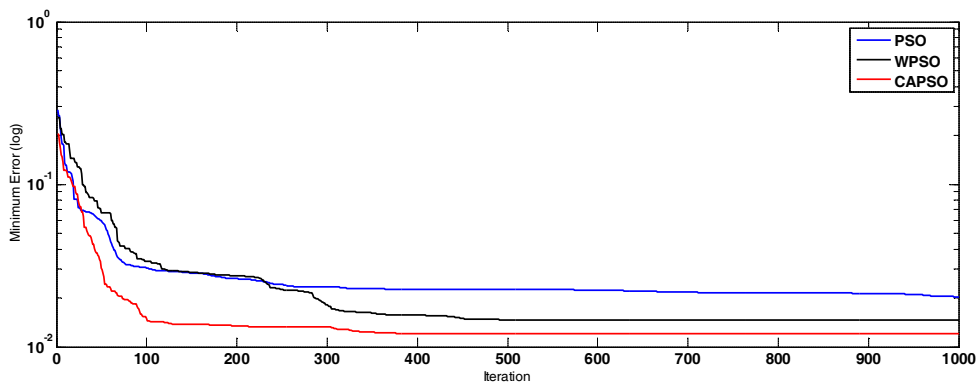
شکل (۵-۱۲) پاسخ فرکانسی فیلتر f_1 با مرتبه ۴۶، شکل (۵-۱۳) پاسخ فرکانسی فیلتر f_1 مرتبه ۴۶ در ناحیه‌ی توقف را نشان می‌دهند و شکل (۵-۱۴) کمترین خطا در هر تکرار الگوریتم را مشخص می‌کند. جدول (۵-۵) نتایج حاصل از این شبیه‌سازی را بیان می‌کند. مرتبه ۴۶ نسبت به مرتبه ۳۰ به ترتیب افزایش تضعیف باند توقف ۰/۴۲، ۰/۳۹ و ۰/۴۳ برای الگوریتم‌های PSO، WPSO و CAPSO را دارد. در این حالت نیز الگوریتم ترکیبی تضعیف باند توقف را ۰/۷ و ۰/۳ نسبت به الگوریتم PSO و WPSO افزایش داده است.

۳.۲.۱.۵ مرتبه ۶۰ $N=۶۰$

شکل (۵-۱۵) پاسخ فرکانسی فیلتر f_1 مرتبه ۶۰.



شکل (۵-۱۶) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۶۰.



شکل (۵-۱۷) کمترین خطا در هر تکرار الگوریتم فیلتر f_1 مرتبه ۶۰.

جدول (۵-۶) فیلتر f_1 مرتبه ۶۰.

$N=60$	تضعیف باند توقف (dB)	حداکثر خطای باند توقف (normalized)	کمترین خطا
PSO	-۳۷/۴۶	۰/۰۱۳۵۵	۰/۰۲۰۳۶
WPSO	-۳۹/۵۱	۰/۰۱۰۵۸	۰/۰۱۴۶۴
CAPSO	-۴۲/۲۶	۰/۰۰۷۷۱۲	۰/۰۱۲۱

شکل (۵-۱۵) پاسخ فرکانسی فیلتر f_1 با مرتبه ۶۰، شکل (۵-۱۶) پاسخ فرکانسی فیلتر f_1

مرتبه ۶۰ در ناحیه‌ی توقف را نشان می‌دهند و شکل (۵-۱۷) کمترین خطا در هر تکرار الگوریتم را

مشخص می‌کند. جدول (۶-۵) نتایج حاصل از این شبیه‌سازی را بیان می‌کند. مرتبه‌ی ۶۰ نسبت به مرتبه‌ی ۳۰ به ترتیب افزایش تضعیف باند توقف ۵۷٪، ۵۸٪ و ۶۷٪ برای الگوریتم‌های PSO، WPSO و CAPSO را دارد. در این حالت نیز الگوریتم ترکیبی تضعیف باند توقف را ۱۲٪ و ۷٪ نسبت به الگوریتم PSO و WPSO افزایش داده است.

۲.۵ طراحی فیلتر بانده‌گذر تیز FRM با الگوریتم‌های تکاملی

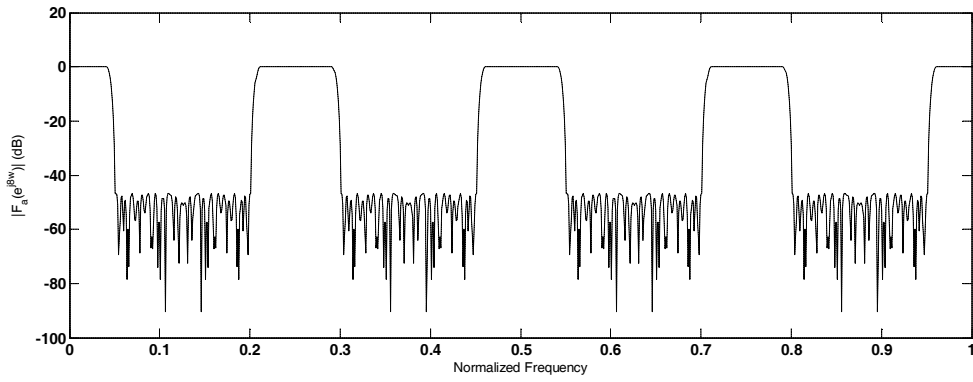
همانطور که در فصل ۳ اشاره شد تکنیک FRM روشی برای ساخت فیلتر با بانده‌گذر تیزتر و تعداد ضرایب فیلتر کمتر می‌باشد. بانده‌گذر تیز با توجه به کاربرد فیلتر تعریف متفاوتی پیدا می‌کند. محققین از تکنیک FRM برای طراحی فیلترهایی با بانده‌گذر بین 0.001π تا 0.05π استفاده می‌کنند. هدف از این بخش طراحی فیلتر میان‌گذر FRM با مشخصات (۱-۵) به وسیله‌ی الگوریتم‌های تکاملی است.

$$M = 8, w_{s1} = 0.29\pi, w_{p1} = 0.3\pi, w_{p2} = 0.7\pi, w_{s2} = 0.71\pi \quad (1-5)$$

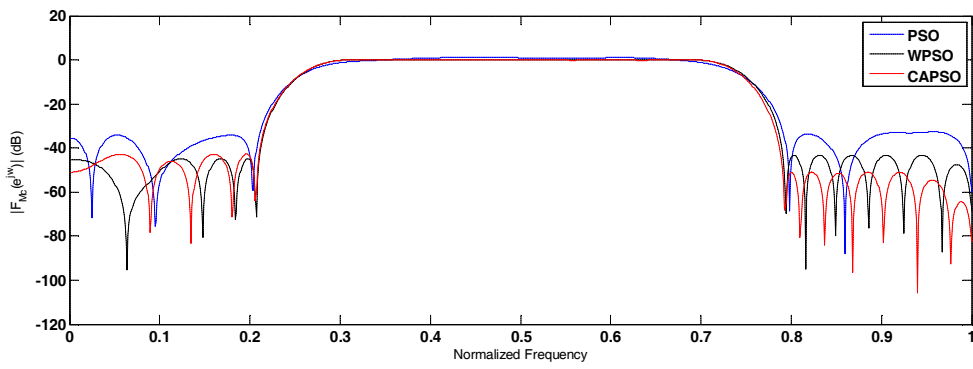
تعداد ذرات جامعه در الگوریتم‌های تکاملی ۵۰ انتخاب شده است و مشخصات هر یک از زیرفیلترها به شرح جدول (۷-۵) می‌باشد.

جدول (۷-۵) زیرفیلترهای فیلتر میان‌گذر.

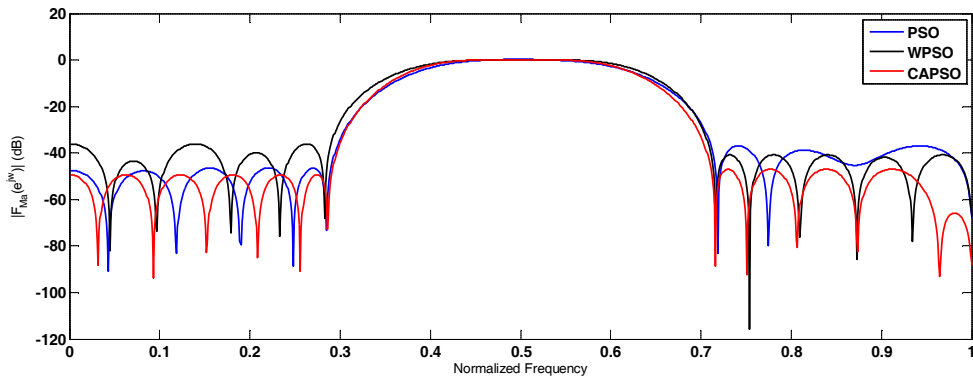
زیر فیلتر	مرتبه فیلتر	فرکانس قطع باند عبور	فرکانس قطع باند توقف
F_a	۶۵	0.32π	0.4π
F_c	—	0.4π	0.32π
F_{Ma}	۳۲	0.45π و 0.55π	0.29π و 0.71π
F_{Mc}	۵۶	0.3π و 0.7π	0.21π و 0.79π



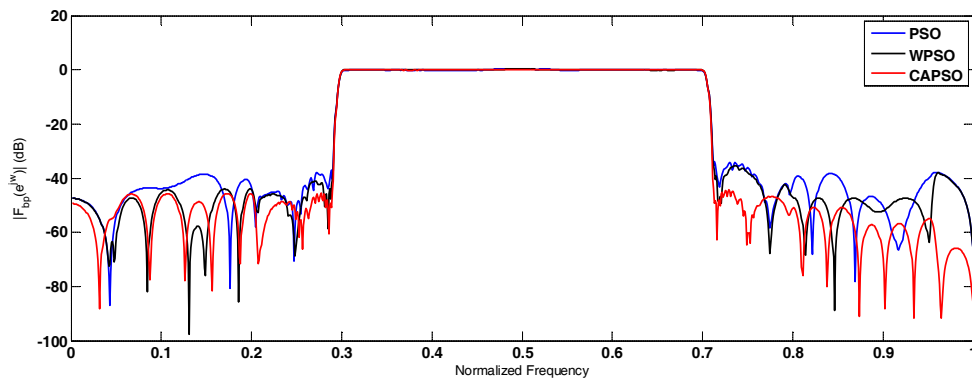
شکل (۵-۱۸) پاسخ فرکانسی $F_a(z^8)$.



شکل (۵-۱۹) پاسخ فرکانسی F_{Mc} .



شکل (۵-۲۰) پاسخ فرکانسی F_{Ma} .



شکل (۵-۲۱) پاسخ فرکانسی فیلتر میان گذر FRM.

شکل (۵-۱۸) پاسخ فرکانسی فیلتر $F_a(z^8)$ با مرتبه‌ی ۶۵، شکل (۵-۱۹) پاسخ فرکانسی فیلتر $F_{Mc}(z)$ و شکل (۵-۲۰) پاسخ فرکانسی فیلتر $F_{Ma}(z)$ با مرتبه‌ی ۳۲ می‌باشند. پاسخ فرکانسی فیلتر میان گذر FRM در شکل (۵-۲۱) قابل مشاهده است. نتایج حاصل از شبیه‌سازی برای الگوریتم‌های PSO، WPSO و CAPSO به ترتیب در سه جدول (۵-۸)، (۵-۹)، (۵-۱۰) آورده شده‌است. جداول مذکور برتری الگوریتم ترکیبی را نسبت به دو روش دیگر مشخص می‌کند.

جدول (۵-۸) طراحی فیلتر میان گذر FRM توسط الگوریتم PSO.

زیر فیلتر	مرتبه‌ی فیلتر	تضعیف باند توقف اول (dB)	تضعیف باند توقف دوم (dB)
F_a	۶۵	-۳۵/۴۵	-----
F_{Ma}	۳۲	-۴۶/۵۷	-۳۶/۹۶
F_{Mc}	۵۶	-۳۴/۸۷	-۳۴/۰۷
FRM	۱۵۳	-۳۸/۲	-۳۴/۳۲

جدول (۹-۵) طراحی فیلتر میان گذر FRM توسط الگوریتم WPSO.

زیر فیلتر	مرتبہ	تضعیف باند توقف اول (dB)	تضعیف باند توقف دوم (dB)
F_a	۶۵	-۴۰/۹	-----
F_{Ma}	۳۲	-۳۶/۱۹	-۴۰/۸۱
F_{Mc}	۵۶	-۴۴/۸۷	-۴۳/۳۲
FRM	۱۵۳	-۴۱	-۳۶/۷

جدول (۱۰-۵) طراحی فیلتر میان گذر FRM توسط الگوریتم CAPSO.

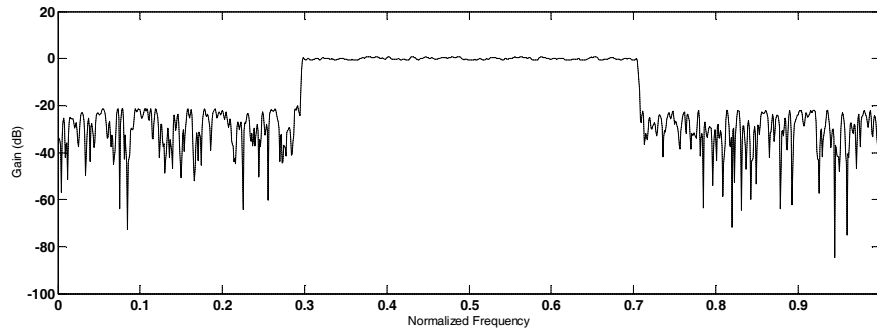
زیر فیلتر	مرتبہ	تضعیف باند توقف اول (dB)	تضعیف باند توقف دوم (dB)
F_a	۶۵	-۴۶/۹۹	-----
F_{Ma}	۳۲	-۴۹/۴۷	-۴۶/۹۶
F_{Mc}	۵۶	-۴۲/۶۹	-۵۱/۰۱
FRM	۱۵۳	-۴۵/۶۱	-۴۴/۳۳

نتایج جدول‌های بالا نشان می‌دهد که الگوریتم ترکیبی عملکرد بهتری در طراحی فیلتر میان گذر باند باریک دارد بطوریکه تضعیف باند توقف اول در این الگوریتم ۱۹٪ و ۱۱٪ و تضعیف باند توقف دوم ۲۹٪ و

۲۱٪ نسبت به الگوریتم PSO و WPSO افزایش یافته است. نتایج بدست آمده نشان می‌دهد که الگوریتم ترکیبی علاوه بر آنکه دقت بهتری نسبت به الگوریتم PSO و WPSO دارد، نسبت به [۲۶] که دقیقاً همین فیلتر را با الگوریتم ژنتیک طراحی کرده است دقت بهتری دارد.

اگر بخواهیم فیلتر میان گذر با مشخصات (۵-۱) با استفاده از الگوریتم ترکیبی بطور مستقیم (بدون استفاده از تکنیک FRM) طراحی کنیم با مشکل مواجه می‌شویم زیرا باند گذر این فیلتر آنقدر کوچک است که الگوریتم ترکیبی قادر به طراحی آن با تضعیف باند توقف جدول (۵-۱۰) نمی‌باشد.

شکل (۵-۲۲) طراحی فیلتر میان گذر با مشخصات (۵-۱) و مرتبه‌ی ۱۵۰۰ را بطور مستقیم نشان می‌دهد. همان طور که مشاهده می‌شود با وجود مرتبه‌ی بالای فیلتر، تضعیف باند توقف آن در حدود ۲۴dB می‌باشد درحالیکه فیلتر طراحی شده با تکنیک FRM در جدول (۵-۱۰) دارای ۱۵۳ ضریب می‌باشد.

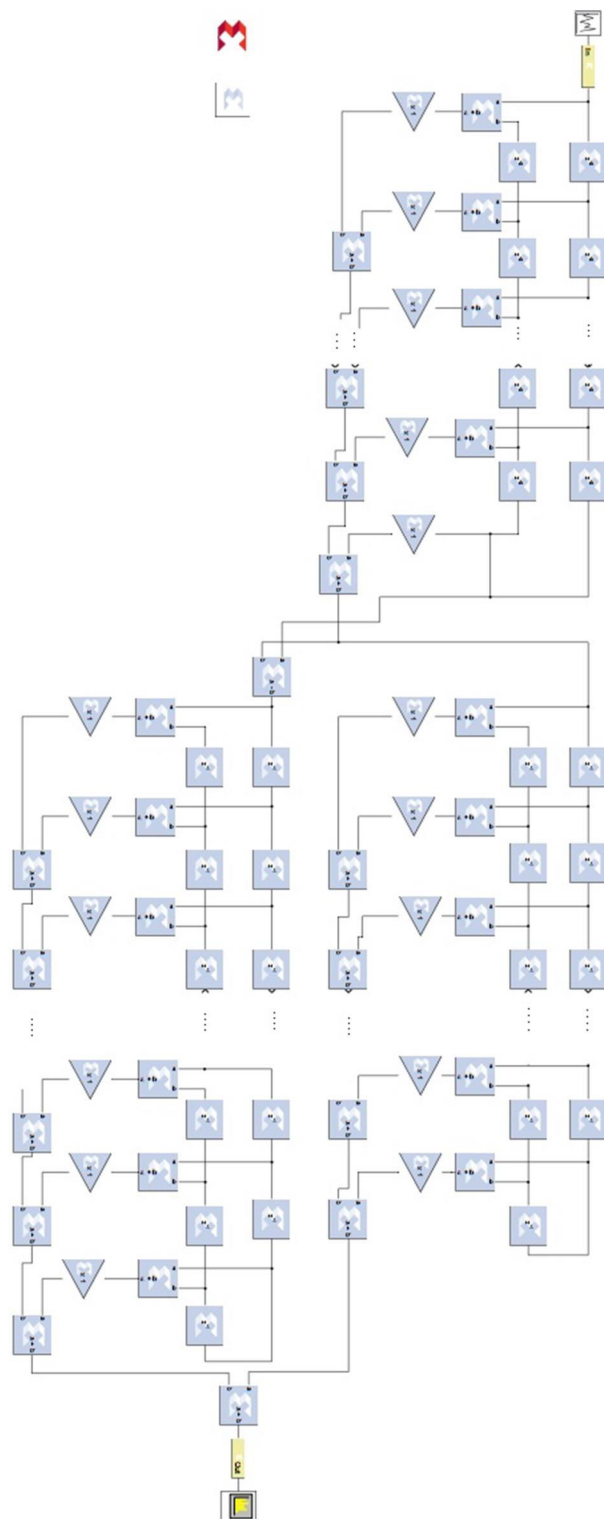


شکل (۵-۲۲) پاسخ فرکانسی فیلتر میان گذر بدون استفاده از تکنیک FRM.

۳.۵ شبیه سازی فیلتر میان گذر تیز FRM با FPGA

بعد از محاسبه ضرایب زیر فیلترها توسط الگوریتم‌های تکاملی در بخش قبل، از این ضرایب برای پیاده‌سازی فیلتر در محیط system generator استفاده می‌کنیم. از آنجایی که فیلترهای طراحی شده دارای فاز خطی هستند ضرایب آنها متقارن می‌باشد. در نتیجه در پیاده‌سازی این فیلترها بر روی FPGA نیاز به $\frac{N}{2}$ (زوج) و $\frac{N+1}{2}$ (فرد) ضرب‌کننده خواهد داشت. شکل (۵-۲۳) نحوه پیاده‌سازی

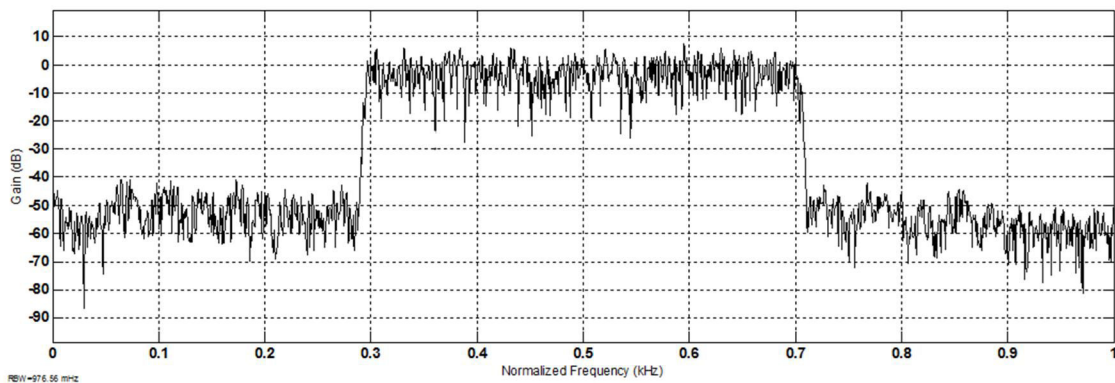
قسمتی از فیلتر $F_a(z^8)$ در محیط system generator را نشان می‌دهد.



شکل (۵-۲۳) نحوه پیاده سازی فیلتر میان گذر FRM با system generator.

شکل (۲۴-۵) پاسخ فرکانسی فیلتر میان گذر FRM به منبع ورودی تصادفی را نشان می دهد. نوسانات زیاد باند عبور نسبت به حالتی که همین فیلتر را با MATLAB شبیه سازی می کنیم، به دلیل منبع ورودی تصادفی است.

جدول (۱۱-۵) نتایج عددی حاصل از شبیه سازی فیلتر مذکور را در محیط system generator نشان می دهد.



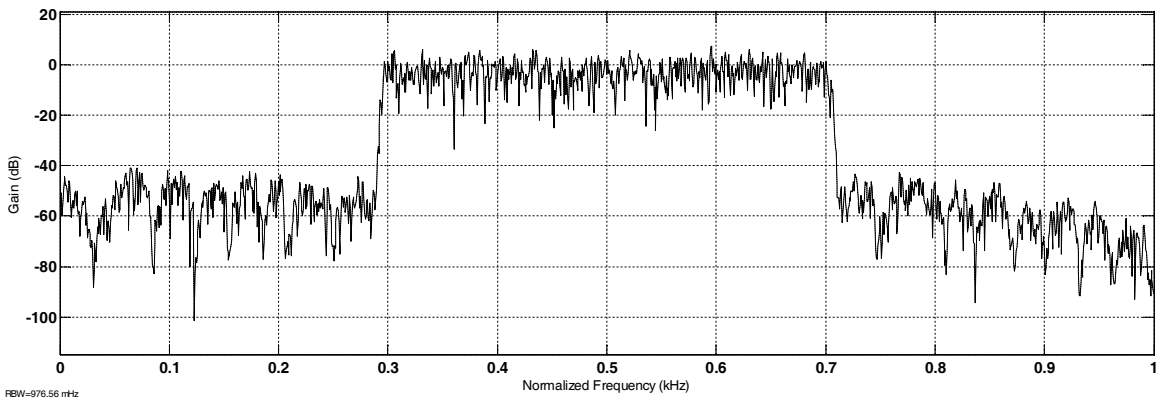
شکل (۲۴-۵) پاسخ فرکانسی فیلتر میان گذر FRM در FPGA.

جدول (۱۱-۵) بیشترین و کمترین مقدار باند عبور و توقف در FPGA.

	تضعیف باند توقف اول (dB)	تضعیف باند توقف اول (dB)	بیشترین مقدار باند عبور (dB)	کمترین مقدار باند عبور (dB)
FPGA	-۴۰/۶۸	-۴۰/۰۵	۵۱۲/۲۷	-۲۵/۱۶

شکل (۲۵-۵) پاسخ فرکانسی فیلتر میان گذر FRM در محیط simulink را نمایش می دهد.

در جدول (۱۲-۵) نتایج حاصل از این شبیه سازی نشان داده شده است.

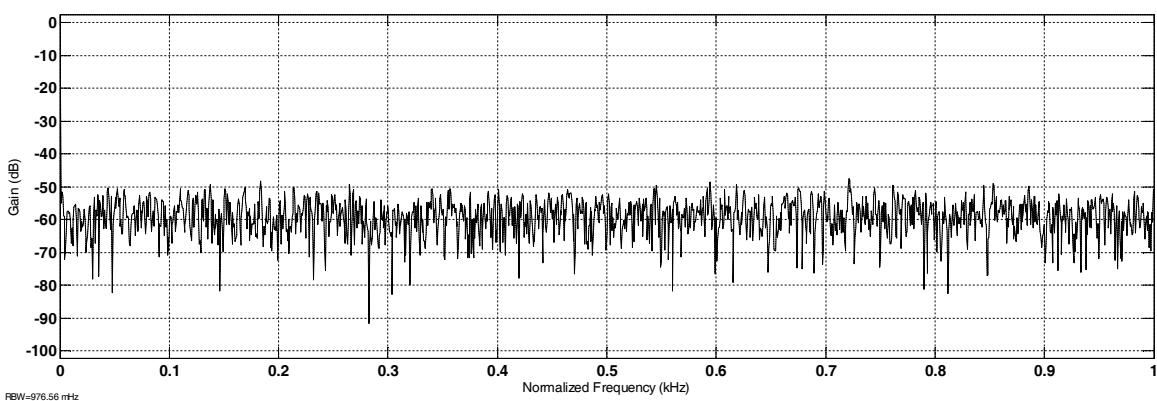


شکل (۵-۲۵) پاسخ فرکانسی فیلتر میان گذر FRM در simulink.

جدول (۵-۱۲) بیشترین و کمترین مقدار باند عبور و توقف در simulink.

	تضعیف باند توقف اول (dB)	تضعیف باند توقف اول (dB)	بیشترین مقدار باند عبور (dB)	کمترین مقدار باند عبور (dB)
simulink	۴۰/۹۱	۴۲/۵۳	۵/۷۲۵	-۲۵/۰۹

شکل (۵-۲۶) تفاضل پاسخهای فرکانسی بدست آمده در محیط simulink و FPGA را نمایش می-دهد و در جدول (۵-۱۳) کمترین و بیشترین مقدار تفاضل قابل مشاهده است.

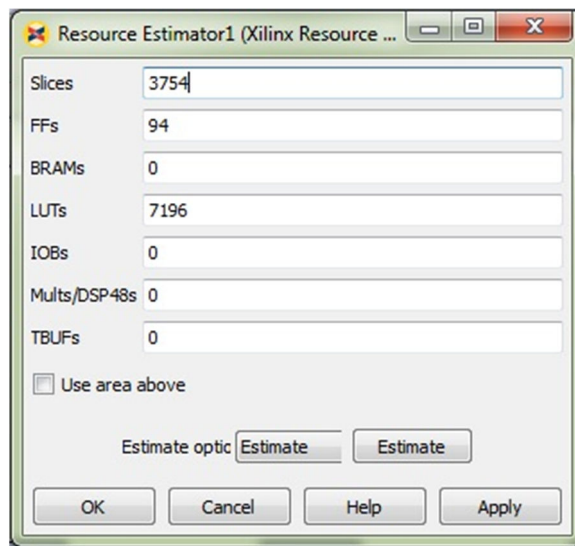


شکل (۵-۲۶) تفاضل پاسخ فرکانسی فیلتر میان گذر FRM در simulink و FPGA.

جدول (۵-۱۳) کم ترین و بیش ترین مقدار تفاضل پاسخ فرکانسی FPGA و simulink.

	کم ترین تفاضل (dB)	بیش ترین تفاضل (dB)
Different	-۹۱/۷۴	-۴۷/۴۸

در طراحی فیلتر FRM میان گذر با FPGA، بخشی از حافظه FPGA اشغال می شود که می توان با استفاده از گزینه Resource Estimator در نرم افزار system generator این حجم اشغال شده را مشاهده کرد. شکل (۵-۲۶) نتایج حاصل از Resource Estimator را نشان می دهد



شکل (۵-۲۷) حجم حافظه اشغال شده فیلتر میان گذر FRM در FPGA.

فصل ششم

جمع بندی و نتیجه گیری

جمع‌بندی و نتیجه‌گیری

الگوریتم‌های تکاملی روش‌های مناسبی برای طراحی ضرایب فیلترها می‌باشند. الگوریتم بهینه‌سازی دسته ذرات روشی قوی برای این منظور است. در این پایان‌نامه برای بهتر کردن الگوریتم بهینه‌سازی دسته ذرات اقدام به ترکیب این الگوریتم با الگوریتم فرهنگی شده‌است و حاصل این کار دست یافتن به روشی با دقت بیشتر، سرعت همگرایی بالاتر و توانایی دنبال کردن پیچیدگی فیلتر در مرتبه‌های بالاتر می‌باشد.

از دیگر اهداف این پایان‌نامه طراحی فیلتر میان‌گذر باند باریک و شبیه‌سازی آن بر روی FPGA می‌باشد. روش‌های ارائه شده در این پایان‌نامه در جهت سعی بر کاهش حجم محاسبات، کاهش ضرایب فیلتر نهایی، دقت و سرعت بیشتر در رسیدن به فیلتر نهایی خواهد بود.

با استفاده از تکنیک ماسک‌گذاری بر روی پاسخ فرکانسی ارائه شده در این پایان‌نامه توانستیم فیلتری که طراحی مستقیم آن با استفاده از الگوریتم‌های تکاملی تقریباً غیرعملی می‌باشد را با تعداد ضرایب مناسب شبیه‌سازی کنیم. کاهش ضرایب در پیاده‌سازی این فیلتر بر روی FPGA بسیار سودمند خواهد بود. زیرا برای شبیه‌سازی این ضرایب نیاز به بلوک ضرب‌کننده است و این ضرب‌کننده‌ها حجم زیادی از حافظه را به خود اختصاص می‌دهند.

از چند منظر برای بهبود فیلتر میان‌گذر باند باریک می‌توان اقدام نمود. به وسیله روش‌های بهتر تکنیک ماسک‌گذاری بر روی پاسخ فرکانسی می‌توان پیچیدگی نهایی فیلتر مطلوب را کاهش داد. تعریف تابع خطای مناسب از موضوعاتی است که هنوز جای کار دارد. همچنین در شبیه‌سازی این فیلتر بر روی FPGA رسیدن به راهکارهایی که حجم نهایی اشغال شده حافظه کل را کاهش دهد بسیار سودمند است.

منابع و مراجع

- [1] O. Gustafsson, H. Johansson, and L. Wanhammar, "Design and efficient implementation of narrow-band single filter frequency masking FIR filters," in *Signal Processing Conference, 2000 10th European*, 2000, pp. 1-4.
- [2] C. Z. Yang and Y. Lian, "New structures for single filter based frequency-response masking approach," in *APCCAS 2006-2006 IEEE Asia Pacific Conference on Circuits and Systems*, 2006, pp. 69-72.
- [3] Y. Lian, "A new frequency-response masking structure with reduced complexity for FIR filter design," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, 2001, pp. 609-612.
- [4] L. Cen, "Frequency response masking filter design using an oscillation search genetic algorithm," *Signal Processing*, vol. 87, pp. 3086-3095, 2007.
- [5] J. I. Ababneh and M. H. Bataineh, "Linear phase FIR filter design using particle swarm optimization and genetic algorithms," *Digital Signal Processing*, vol. 18, pp. 657-668, 2008.
- [6] W. Lee, V. Rehbock, K. Teo, and L. Caccetta, "A weighted least-square-based approach to FIR filter design using the frequency-response masking technique," *IEEE Signal Processing Letters*, vol. 11, pp. 593-596, 2004.
- [7] A. R. Baig and M. Rashid, "Honey bee foraging algorithm for multimodal & dynamic optimization problems," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 169-169.
- [8] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department 2005.
- [9] E. K. Singh and G. Kaur, "Design of Low Pass FIR Filter using Artificial Bee Colony Optimization Technique and its Comparison with Particle Swarm Optimization," *International Journal of Advanced Research in Computer Science and Software Engineering*, p. 9, 2014.
- [10] Q. Yue, C. Ma, and X. Wang, "Canonical signed digit encoding based optimal design for FIR filters," in *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, 2011, pp. 729-732.
- [11] V. Saini, B. Singh, and R. Devi, "Area Optimization of FIR Filter and its Implementation on FPGA," *International Journal of Recent Trends in Engineering*, vol. 1, pp. 55-58, 2009.

- [12] Alan V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*: Prentice Hall, 1999.
- [13] L. R. Rabiner and B. Gold, "Theory and application of digital signal processing," *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, vol. 1, 1975.
- [14] Y. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE transactions on circuits and systems*, vol. 33, pp. 357-364, 1986.
- [15] Y. Lim and Y. Lian, "The optimum design of one-and two-dimensional FIR filters using the frequency response masking technique," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, pp. 88-95, 1993.
- [16] J. Yli-Kaakinen and T. Saramäki, "An efficient algorithm for the optimization of FIR filters synthesized using the multistage frequency-response masking approach," *Circuits, Systems, and Signal Processing*, vol. 30, pp. 157-183, 2011.
- [17] K. James and E. Russell, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
- [18] S. Das, A. Abraham, and A. Konar, "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives," in *Advances of Computational Intelligence in Industrial Systems*, ed: Springer, 2008, pp. 1-38.
- [19] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 84-88.
- [20] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," in *Proceedings of the genetic and evolutionary computation conference*, 2001, pp. 469-476.
- [21] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 101-106.
- [22] S. Mondal, S. Ghoshal, R. Kar, and D. Mandal, "Novel particle swarm optimization for low pass fir filter design," *WSEAS transactions on signal processing*, vol. 8, pp. 111-120, 2012.
- [23] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*, 1994.
- [24] A. P. Engelbrecht, *Computational intelligence: an introduction*: John Wiley & Sons, 2007.
- [25] Q. Niu, X. Wang, and Z. Zhoua, "An efficient cultural particle swarm optimization for economic load dispatch with valve-point effect," *Procedia Engineering*, vol. 23, pp. 828-834, 2011.

- [26] P. Mercier, S. M. Kilambi, and B. Nowrouzian, "Optimization of FRM FIR digital filters over CSD and CDBNS multiplier coefficient spaces employing a novel genetic algorithm," *Journal of Computers*, vol. 2, pp. 20-31, 2007.

Abstract:

FIR filter has many applications in industry. Stability and linear phase are two main of its characteristics. For better efficiency we need to design filter with more coefficients. FIR filter design involves multi-modal, multi parameter optimization, so different optimization techniques can be utilized to determine the impulse response coefficient of it. By frequency response masking FIR filter structure we can design sharp band pass filter with low computational complexity. And better accuracy for designing FIR filter achieved by combining cultural algorithm and particle swarm optimization algorithm. System generator software make it possible to implant FRM bandpass filter on error FPGA.

Keywords:

FIR Filter, FRM Filter, PSO, Cultural Algorithm, FPGA.



Faculty of Electrical and Robotics Engineering
M.Sc. Thesis in Digital Electronic Systems Engineering

FRM-based Band Pass Filter Design and Implementation on
FPGA platform

By: mahsa moghimi

Supervisor:
Dr. Ali Solyemani Aiouri

september2016