

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی برق و رباتیک - گروه مهندسی الکترونیک
پایان نامه کارشناسی ارشد مهندسی برق - الکترونیک

پایان نامه

طراحی مدار اینورتر سه فاز و شبیه سازی آن روی FPGA

طناز طبسی

استاد راهنما

دکتر سید علی سلیمانی ایوری

بهمن ماه ۱۳۹۴

چکیده

همزمان با پیشرفت تکنولوژی در قطعات نیمه‌هادی مانند ترانزیستورها و تریتورها، تقاضای گسترده‌ای برای راه‌اندازهای جدید موتور با توان بالا و ولتاژ متوسط بوجود آمده است. این تجهیزات به طور وسیعی در پتروشیمی، استخراج معدن، صنایع فولاد، ترابری و سایر صنایع مورد استفاده قرار می‌گیرند تا در انرژی الکتریکی صرفه جویی شود، بهره‌وری افزایش یابد و کیفیت محصول بهبود یابد. از جمله راه‌اندازهای پرکاربرد در صنایع، اینورترها می‌باشند که به صورت تکفاز و سه‌فاز جهت کنترل سرعت موتور مورد استفاده قرار می‌گیرند. طراحی صحیح این راه‌اندازها درجهتی که تا حد ممکن از اغتشاش و اتلاف انرژی جلوگیری شود، همواره مورد توجه محققان این حوزه بوده است. با توجه به مزایای زیاد اینورترها فن کلیدزنی انتخاب شده نقش قابل توجهی در کاهش یا حذف هارمونیک‌های ولتاژ خروجی اینورتر و کاهش تلفات کلیدزنی دارد.

در این پایان نامه معماری کارآمدی جهت پیاده‌سازی صحیح بخش کنترلی سرعت موتور متناوب سه‌فاز روی FPGA ارائه شده است. به منظور کاهش اعوجاج و اتلاف انرژی نسبت به طراحی‌هایی که تاکنون در این زمینه صورت گرفته شده از روش مدولاسیون پهنای پالس سینوسی جهت تعیین فرکانس موج حامل استفاده شده است. این طراحی جدید به گونه‌ای است که کلیدزنی صحیح اینورتر سه‌فاز جهت بهینه کردن پارامترهای موثر در عملکرد سیستم فراهم شده است. در مرحله اول مدار در محیط شبیه‌ساز متلب پیاده‌سازی شده است. معماری جدید طوری است که موتور در فاصله زمانی کم سرعت خود را به سرعت نهایی که سرعت مطلوب است می‌رساند. پس از کسب موفقیت در طراحی اولیه، شبیه‌سازی با نرم افزار ISE System Generator روی FPGA صورت می‌پذیرد. علت انتخاب این نرم افزار سهولت شبیه‌سازی در آن است. بخش‌های کنترل کننده و ساخت پالس مدوله شده توسط بلوک‌های FPGA تنظیم شده است. کارآمدی طراحی در حوزه دیجیتال نیز موجب می‌شود پاسخ نهایی همانند شبیه‌سازی اولیه سرعت موتور را با روند مطلوبی به سرعت مرجع برساند. در این مرحله مدار بر روی FPGA از نوع Spartan3 شبیه‌سازی شده است و با توجه به حجم سخت‌افزار انتخاب شده خروجی مدار با سرعت بالایی به پاسخ مطلوب نهایی می‌رسد. به طور مثال اگر سرعت نهایی ۱۴۰۰ در نظر گرفته شود، طبق طراحی انجام شده سرعت در کمتر از ۰.۷ ثانیه با خطای ۲ درصد به مقدار نهایی می‌رسد.

کلید-واژه‌ها: اینورتر سه‌فاز، مدولاسیون عرض پالس سینوسی، کنترل کننده تناسبی انتگرالی، کنترل کننده ضد اشباع، آرایه دريچه‌ای برنامه پذیر منطقی

فهرست مطالب

صفحه	عنوان
۱	فصل اول مقدمه
۲	۱-۱ مقدمه
۵	۲-۱ ساختار پایان نامه
۷	فصل دوم پیشینه و مبانی پایان نامه
۸	۱-۲ موتور القایی و کنترل دور
۱۰	۱-۱-۲ روش کنترل اسکالر
۱۶	۲-۱-۲ روش کنترل برداری
۱۷	۳-۱-۲ روش کنترل مستقیم گشتاور
۱۸	۲-۲ کنترل کننده های دور موتورهای القایی
۲۳	۳-۲ معرفی توپولوژی های مختلف اینورتری
۲۳	۱-۳-۲ اینورترهای با لینک مستقیم
۲۶	۲-۳-۲ اینورتر با مدار باز کننده تای جریان
۳۰	۳-۳-۲ اینورترهای بدون لینک مستقیم
۳۳	۴-۲ مدولاسیون عرض پالس (PWM)
۳۴	۵-۲ مدولاسیون عرض پالس سینوسی
۳۷	۶-۲ مدولاسیون بردار فضایی
۳۹	۷-۲ کنترل کننده تناسبی-انتگرالی و روش تجربی تنظیم ضرائب آن
۴۰	۱-۷-۲ تنظیم ضرائب کنترل کننده تناسبی-انتگرالی به روش پاسخ فرآیند
۴۰	۲-۷-۲ تنظیم ضرائب کنترل کننده تناسبی-انتگرالی به روش زیگلر نیکلز مرتبه یک

۴۱	۳-۷-۲ تنظیم ضرائب کنترل کننده تناسبی- انتگرالی با سایر روش
۴۳	فصل سوم پیاده سازی سخت افزاری مدار اینورتر سه فاز
۴۴	۱-۳ روش دیجیتال
۴۶	۲-۳ استفاده از تراشه های مخصوص
۴۶	۱-۲-۳ استفاده از میکروکنترلرها
۴۷	۳-۳ پردازش سیگنال دیجیتال (DSP)
۴۸	۱-۳-۳ فرمت داده ها در پردازش سیگنال دیجیتال
۴۹	۴-۳ آرایه دریچه ای برنامه پذیر منطقی (FPGA)
۵۲	۱-۴-۳ ساختار کلی FPGA
۵۳	۲-۴-۳ مقایسه FPGA با MPGA
۵۴	۳-۴-۳ مراحل پیاده سازی یک طرح بر روی FPGA
۵۵	۴-۴-۳ انواع متفاوت معماری های FPGA
۵۶	۵-۴-۳ معیارهای اساسی انتخاب و استفاده از FPGA
۵۶	۶-۴-۳ مقایسه مشخصه ها
۵۷	۷-۴-۳ داده های PREP
۵۸	۸-۴-۳ مقایسه FPGA با Microcontroller
۶۱	فصل چهارم شبیه سازی و نتایج
۶۲	۱-۴ شبیه سازی
۶۴	۲-۴ شبیه سازی در حوزه آنالوگ
۶۵	۱-۲-۴ شبیه سازی موتور سه فاز
۶۶	۲-۲-۴ کنترل کننده مدار

۷۲	۳-۲-۴ مدار هدایت موتور
۷۵	۴-۲-۴ سنجش داده
۷۹	۳-۴ شبیه‌سازی در حوزه دیجیتال
۸۳	۱-۳-۴ پیاده‌سازی کنترل کننده تناسبی انتگرالی در حوزه دیجیتال
۸۶	۲-۳-۴ مدولاسیون پهنای پالس سینوسی در حوزه دیجیتال
۹۵	۴-۴ استفاده از کنترل فازی برای بهبود عملکرد مدار
۹۹	فصل پنجم جمع‌بندی، نتیجه‌گیری و پیشنهادات
۱۰۰	۱-۵ نتایج شبیه‌سازی و پیشنهادات
۱۰۲	پیوست ها
۱۱۲	مراجع

فهرست اشکال

صفحه	عنوان
۱۰	شکل ۱-۲: خلاصه ای از انواع روش‌های کنترل موتورهای القایی
۱۱	شکل ۲-۲: افزودن مقاومت در روتور برای کنترل سرعت با تغییر لغزش
۱۲	شکل ۳-۲: منحنی گشتاور سرعت موتور متناوب
۱۳	شکل ۴-۲: تغییرات منحنی گشتاور سرعت بر حسب تغییرات فرکانس
۱۴	شکل ۵-۲: مدار معادل موتور القایی
۱۴	شکل ۶-۲: تغییرات ولتاژ بر حسب فرکانس برای کنترل دور
۱۵	شکل ۷-۲: مشخصه گشتاور سرعت بر اساس V/f ثابت یک موتور القایی
۱۶	شکل ۸-۲: بلوک دیاگرام کنترلی حلقه باز برای روش V/f ثابت
۱۶	شکل ۹-۲: بلوک دیاگرام کنترلی حلقه بسته برای روش V/f ثابت
۲۲	شکل ۱۰-۲: مدار ساده اینورتر تکفاز
۲۲	شکل ۱۱-۲: مدار ساده اینورتر سه‌فاز با ورودی مستقیم
۲۳	شکل ۱۲-۲: ساختار کلی میکرواینورترهای با لینک مستقیم
۲۴	شکل ۱۳-۲: اینورتر لینک مستقیم دارای رزونانس گذر ولتاژ صفر
۲۵	شکل ۱۴-۲: اینورتر با لینک مستقیم با استفاده از مبدل بوست زتا
۲۵	شکل ۱۵-۲: اینورتر فلاپیک با ساختار تمام پل
۲۶	شکل ۱۶-۲: اینورتر با مبدل سوئیچینگ نیم پل
۲۷	شکل ۱۷-۲: ساختار کلی اینورتر با مدار باز کننده تای جریان
۲۸	شکل ۱۸-۲: اینورتر فلاپیک با مدار باز کننده تای جریان
۲۸	شکل ۱۹-۲: اینورتر فلاپیک با مدار تزریق کنند جریان هارمونیک دوم

۲۹	شکل ۲-۲۰: مدار Inteleaved flyback به همراه Current unfolding
۳۰	شکل ۲-۲۱: اینورتر دارای بازکننده تای جریان
۳۰	شکل ۲-۲۲: ساختار پایه میکرواینورترهای بدون لینک DC
۳۱	شکل ۲-۲۳: توپولوژی بدون لینک مستقیم با سیکلکانورتر ولتاژ متناوب فرکانس بالا
۳۱	شکل ۲-۲۴: توپولوژی بدون لینک مستقیم با مبدل پوش پول
۳۲	شکل ۲-۲۵: توپولوژی بدون لینک مستقیم سه طبقه
۳۳	شکل ۲-۲۶: اینورتر بدون لینک مستقیم دو طبقه
۳۴	شکل ۲-۲۷: مقایسه دو سیگنال آنالوگ
۳۴	شکل ۲-۲۸: سیگنال مدولاسیون عرض پالس سینوسی و مدار مقایسه کننده مدولاسیون
۳۵	شکل ۲-۲۹: نمونه ای از یک مدولاسیون اشباع شده
۳۶	شکل ۲-۳۰: مدولاسیون با نمونه برداری منظم
۳۶	شکل ۲-۳۱: مدولاسیون تک قطبی
۳۷	شکل ۲-۳۲: نحوه تولید و سیگنال مدوله شده سینوسی حاصل از مدولاسیون دو قطبی
۳۷	شکل ۲-۳۳: نمای ساده اینورتر سه فاز در مدولاسیون بردار فضایی
۳۹	شکل ۲-۳۴: مدولاسیون بردار فضایی
۴۴	شکل ۳-۱: مقایسه دو سیگنال آنالوگ با استفاده از ثبات
۴۵	شکل ۳-۲: استفاده از مبدل ADC
۵۰	شکل ۳-۳: نمای درونی یک FPGA
۵۱	شکل ۳-۴: نمونه‌ای از بکارگیری FPGA در آزمایشگاه
۶۲	شکل ۴-۱: بلوک دیاگرام سیستم کنترل سرعت موتور القایی
۶۳	شکل ۴-۲: مراحل کنترل سرعت موتور القایی

۶۵	شکل ۴-۳: شمای کلی سیستم مدار کنترل سرعت موتور
۶۷	شکل ۴-۴: بلوک دیاگرام کنترل کننده تناسبی انتگرالی
۶۷	شکل ۴-۵: مدل ساده کنترل کننده تناسبی انتگرالی
۶۹	شکل ۴-۶: سیگنال خطا و تحریک در کنترل کننده تناسبی انتگرالی ساده
۷۱	شکل ۴-۷: شبیه‌سازی کنترل کننده تناسبی انتگرالی ضد اشباع
۷۲	شکل ۴-۸: سیگنال خطا و تحریک در کنترل کننده ضد اشباع
۷۳	شکل ۴-۹: شبیه‌سازی اینورتر سه‌فاز در متلب
۷۴	شکل ۴-۱۰: شبیه‌سازی بلوک مدولاسیون عرض پالس سینوسی
۷۵	شکل ۴-۱۱: موج مدوله شده حاصل از شبیه‌سازی مدار در متلب
۷۵	شکل ۴-۱۲: سرعت مرجع و سرعت واقعی موتور
۷۶	شکل ۴-۱۳: مشخصه سرعت خروجی به ازای ضرایب مختلف کنترل کننده
۷۷	شکل ۴-۱۴: مشخصه گشتاور-سرعت موتور
۷۸	شکل ۴-۱۵: مشخصه سرعت موتور با وجود اعمال گشتاور
۷۸	شکل ۴-۱۶: سرعت موتور بر اساس look up table ورودی مشخص
۸۴	شکل ۴-۱۷: ساختار بکار رفته برای کنترل کننده PI در محیط System generator
۸۵	شکل ۴-۱۸: سیگنال خطا و تحریک در کنترل کننده ضد اشباع در System generator
۸۷	شکل ۴-۱۹: ساختار بکار رفته جهت ایجاد موج سینوسی در محیط System generator
۸۷	شکل ۴-۲۰: موج سینوسی ایجاد شده توسط ساختار در محیط System generator
۸۸	شکل ۴-۲۱: ساختار تولید شکل موج حامل مثلثی در محیط System generator
۸۸	شکل ۴-۲۲: موج مثلثی تولید شده توسط ساختار محیط System generator
۸۹	شکل ۴-۲۳: سیگنال‌های ورودی به بخش مقایسه کننده در محیط System generator

۹۰	شکل ۴-۲۴: پاسخ خروجی سرعت به ازای ورودی مرجع مشخص با شبیه‌سازی در
۹۰	شکل ۴-۲۵: سرعت موتور بر اساس lookup table ورودی مرجع مشخص در پیاده‌سازی
۹۳	شکل ۴-۲۶: خروجی سرعت در مقادیر متفاوت پریود نمونه برداری
۹۴	شکل ۴-۲۷: پاسخ خروجی سرعت به ازای مقدار پایین بیت درگاه ورودی
۹۵	شکل ۴-۲۸: استفاده از داده‌های غنی جهت تعیین توابع فازی سیستم
۹۶	شکل ۴-۲۹: ساختار کنترل فازی و کنترل کننده تناسبی انتگرالی جهت تعیین و اعمال
۹۷	شکل ۴-۳۰: پاسخ بهتر خروجی با ضرایب تعیین شده با کنترل کننده فازی
۹۷	شکل ۴-۳۱: اعمال ضرایب تعیین شده با روش فازی به کنترل کننده PI در SysGen
۹۸	شکل ۴-۳۲: پاسخ بهتر خروجی با ضرایب تعیین شده با کنترل کننده فازی در حوزه
۱۰۳	شکل پیوست (الف)-۱: نمونه ای از شبکه های عصبی چندلایه
۱۰۶	شکل پیوست (ب)-۱: قسمت‌های مختلف یک سیستم فازی
۱۰۹	شکل پیوست (ج)-۱: ساختار شبکه ANFIS
۱۱۱	شکل پیوست (د)-۱: ساختار System Generator
۱۱۱	شکل پیوست (د)-۲: مدار شبیه‌سازی شده در System Generator

فهرست جداول

صفحه	عنوان
۸	جدول ۱-۲: مشخصات موتور القایی
۳۸	جدول ۲-۲: حالت‌های مختلف کلیدزنی در مدولاسیون بردار فضایی
۶۶	جدول ۱-۴: مشخصات موتور القایی سه‌فاز مورد استفاده در پایان نامه
۸۰	جدول ۲-۴: بلوک‌های معادل متلب در پیاده‌سازی دیجیتالی کنترل کننده
۱۰۴	جدول پیوست (ب)-۱: مقایسه مدل ممدانی و TS

فصل اول

مقدمه

۱-۱ مقدمه

موتورهای الکتریکی که از ادوات بسیار مهم در صنایع کوچک و بزرگ محسوب می‌شوند، به دو دسته متناوب و مستقیم تقسیم می‌شوند. برای مدت‌ها ماشین‌های مستقیم به عنوان موتورهای با سرعت قابل تنظیم مورد استفاده قرار می‌گرفتند و کنترل گشتاور متغیر بر روی این ماشین‌ها اجرا می‌شد. مهم‌ترین مشکل این ماشین‌ها بالا بودن هزینه تعمیر و نگهداری آن‌ها و همچنین بازده پایین‌تر نسبت به ماشین‌های متناوب می‌باشد. بعد از دهه ۱۹۸۰ با پیشرفت الکترونیک قدرت و تکنولوژی کنترل ماشین‌های متناوب، ماشین‌های مستقیم از رده خارج شدند اما هنوز صنایع قدیمی و صناعی که به کنترل خیلی دقیق نیاز دارند همچون جرثقیل‌های غول پیکر از ماشین مستقیم استفاده می‌کنند [۱]. برای کنترل دور موتورهای متناوب از درایور که یک سیستم الکترونیک قدرت است، استفاده می‌شود. درایورهای متناوب با عناوینی چون درایورهای تنظیم سرعت^۱ (ASD)، درایورهای تنظیم فرکانس (AFD)^۲، درایورهای فرکانس متغیر (VFD)^۳، درایور سرعت متغیر (VSD)^۴ و یا مبدل‌های فرکانس (FC)^۵ نیز شناخته می‌شوند. اولین موتور متناوب در سال ۱۸۹۹ طراحی شد. موتورهای متناوب با استفاده از القای الکترومغناطیسی انرژی الکتریکی را به انرژی مکانیکی تبدیل می‌کنند. از ویژگی موتورهای متناوب می‌توان به موارد زیر اشاره نمود.

- سرعت ثابت که با فرکانس تغذیه تعیین می‌شود.
- گشتاور ثابت

سرعت ثابت در تمامی پروسه‌ها و کاربردها مناسب نیست و لازم است که بتوان سرعت را متناسب با نیاز تغییر داد.

ماشین آلات صنعتی اکثراً با موتورهای توان بالا که دارای تجهیزات تنظیم سرعت می‌باشند، کار می‌کنند. این موتورها معمولاً در یک سرعت ثابت به کار می‌روند. در کنترل سرعت از درایور متناوب

¹ Adjust Speed Drive

² Adjust Frequency Drive

³ Variable Speed Drive

⁴ Variable Speed Drive

⁵ Frequency Converter

سرعت متغیر استفاده می‌شود. به علاوه درایورهای متناوب دور متغیر معمولاً در کاربردهای پیچیده، محیط‌های نامناسب و سخت مورد استفاده قرار می‌گیرند. کنترل فرآیند و کاهش مصرف انرژی دو دلیل اصلی برای استفاده از درایورهای متناوب می‌باشد که منجر به عملکرد هموارتر، کنترل شتاب و سرعت موتور، جبران‌سازی تغییرات در فرآیندهای متغیر و کنترل گشتاور می‌شود.

در این پایان‌نامه، شبیه‌سازی و کنترل دور موتور القایی به کمک روش‌های کنترل کلاسیک و مدرن و با پیاده‌سازی کنترل‌کننده از طریق¹ FPGA انجام می‌شود. مبنای انتخاب FPGA کاهش حجم سخت‌افزار، هزینه پایین، قابلیت اطمینان بالا، سرعت بالا در کلید زنی در آن است.

پردازش سیگنال، مبنای پردازش اطلاعات است و روش‌های متعددی برای استخراج اطلاعات بدست آمده از طبیعت یا از ماشین‌های ساخت بشر را شامل می‌شود. عموماً هدف از پردازش سیگنال، ساده‌تر شدن بررسی محتوای آن است. امروزه حجم عمده پردازش سیگنال بصورت دیجیتالی انجام می‌شود. پردازش دیجیتالی سیگنال‌ها کاربردهای فراوانی در زمینه‌های مختلف علمی و صنعتی دارد و استفاده از این تکنیک‌ها به صورت چشم‌گیری در حال توسعه است. از جمله کاربردهای این روش می‌توان به پردازش گفتار، پردازش تصویر و سیستم‌های کنترل رباتیک اشاره نمود. به دلیل وجود مسائلی نظیر رانش‌های حرارتی، تolerانس قطعات و... در بسیاری از کاربردها، از جمله در پردازش گفتار دستیابی به تolerانس‌های دقیق مورد نیاز برای فیلترها و قسمت‌های مختلف سیستم به روش آنالوگ، غیر ممکن یا بسیار دشوار است. در حالی که در روش پردازش دیجیتالی، پیاده‌سازی اجزاء مختلف سیستمی، به سهولت و با هر درجه‌ای از دقت که مورد نیاز باشد، امکان پذیر است. علاوه بر این چون کل سیستم به صورت نرم‌افزاری (الگوریتم‌های پردازش) ساخته می‌شود، اصلاح و توسعه آن بسیار ساده خواهد بود.

در این پایان‌نامه از روش‌های پردازش دیجیتال به منظور پیاده‌سازی الگوریتم کنترلی در درایور موتور القایی بهره‌برداری می‌شود. به منظور انجام این پردازش‌ها از FPGA یا آرایه دریچه‌ای برنامه‌پذیر منطقی استفاده می‌شود. این قطعات دارای بلاک‌های منطقی برنامه‌پذیر و اتصالات بین بلاکی قابل پیکربندی هستند. برخی از آنها تنها یک بار قابلیت برنامه‌پذیری دارند و برخی دیگر چندین بار قابلیت برنامه‌پذیری را دارا هستند.

¹ Field Programmable Gate Array

مبدل‌های جریان مستقیم به متناوب را اینورتر می‌نامند. وظیفه‌ی یک اینورتر تبدیل یک ولتاژ ورودی مستقیم به یک ولتاژ خروجی متناوب و متقارن با دامنه و فرکانس مورد نظر است. اینورترها کاربرد گسترده‌ای در صنایع برای کنترل سرعت، منابع تغذیه وقفه ناپذیر، گرمکن القایی، درایورهای فرکانس متغیرو... دارند [۱]. اگر ولتاژ ورودی اینورتر ثابت و غیر قابل کنترل باشد می‌توان با تغییر بهره‌ی اینورتر یک ولتاژ متغیر را در خروجی بدست آورد. در داخل اینورتر این کار معمولاً به وسیله یک روش کنترل مدولاسیون صورت می‌گیرد. شکل موج‌های ولتاژ خروجی در اینورترهای ایده‌آل باید سینوسی باشد با این حال در عمل این شکل موج‌ها غیرسینوسی بوده و دارای یک سری هارمونیک‌های مشخص می‌باشد. با توجه به ماهیت اکثر مصرف‌کننده‌ها اغلب از شکل موج‌های سینوسی با اعوجاج بسیار کم استفاده می‌شود. با در اختیار داشتن قطعات نیمه‌هادی قدرت سریع مانند ترانزیستور نیمه رسانا-اکسید-فلز اثر میدان^۱، ترانزیستور دو قطبی با درگاه عایق شده^۲ و... می‌توان با استفاده از روش‌های کلیدزنی، هارمونیک‌های ولتاژ خروجی را به نحو چشم‌گیری کاهش داد [۲].

تاکنون روش‌های متعددی برای طراحی و بهبود عملکرد اینورتر سه‌فاز بکار رفته است که در زیر به برخی از آن‌ها اشاره می‌شود:

- استفاده از روش مدولاسیون پهنای پالس^۳ سینوسی جهت کنترل سرعت موتور.
- بکارگیری تکنیک‌هایی جهت بهبود مدولاسیون پهنای پالس سینوسی (استفاده از فیلتر پایین گذر، استفاده از موج دندان اره‌ای، تزریق هارمونیک سوم و...).
- استفاده از روش مدولاسیون فضای برداری^۴ [۳].
- استفاده از اینورترهای چند سطحی^۵ جهت نزدیک‌تر شدن خروجی اینورتر به شکل موج سینوسی.
- بکارگیری توپولوژی‌های مختلف جهت بهبود عملکرد اینورتر.

¹ MOSFET

² IGBT

¹ SPWM

² SVM

³ Multi-Level

۲-۱ ساختار پایان نامه

در این پایان نامه ابتدا به معرفی موتورهای القایی و روش‌های مختلف کنترل دور موتور از جمله روش‌های کنترل گشتاور، کنترل برداری و روش تثبیت ولتاژ نسبت به فرکانس پرداخته می‌شود. انواع اینورترهای مورد استفاده به منظور درایور این دسته از موتورها نیز مورد بررسی قرار می‌گیرد. سپس درباره مدولاسیون پهنای پالس، اصول پردازش دیجیتال سیگنال و ساختار تراشه‌های FPGA بحث می‌شود. فصل چهارم نیز به شبیه‌سازی و ارائه نتایج اختصاص یافته است. در این فصل اینورتر سه‌فاز با استفاده از مدولاسیون پهنای پالس سینوسی طراحی می‌گردد. سپس مدار مربوطه با استفاده از نرم‌افزار شبیه‌ساز System Generator بر روی آی‌سی‌های قابل برنامه‌ریزی FPGA پیاده‌سازی می‌گردد. به عبارت دیگر به منظور پیاده‌سازی کنترل کننده از FPGA استفاده می‌شود. همانطور که پیش‌تر گفته شد یکی از دلایل انتخاب FPGA قابلیت سخت‌افزاری آن است. طراحی اینورتر در جهتی که بیشترین بازده ممکن و کمترین اعوجاج در خروجی را داشته باشد و زمان اجرای برنامه در محیط نرم‌افزاری سیستم حتی الامکان کوتاه باشد، نوآوری این پایان نامه خواهد بود. بکارگیری تکنیک‌هایی جهت بهبود عملکرد مدار به طراح در این مسیر کمک می‌کند. در خصوص روش‌های بکاربرده شده جهت بهتر شدن مدار در فصل‌های آینده توضیح داده می‌شود.

فصل دوم

پیشینه و مبانی پایان نامه

۲-۱ موتور القایی و کنترل دور

موتورهای الکتریکی به دو دسته متناوب و مستقیم تقسیم می‌شوند. برای مدت‌ها ماشین‌های مستقیم به عنوان موتورهای با سرعت قابل تنظیم مورد استفاده قرار می‌گرفتند و کنترل گشتاور متغیر بر روی این ماشین‌ها اجرا می‌شد. مهم‌ترین مشکل این ماشین‌ها بالا بودن هزینه تعمیر و نگهداری آن‌ها و همچنین بازده پایین‌تر نسبت به ماشین‌های متناوب می‌باشد. بعد از دهه ۱۹۸۰ با پیشرفت الکترونیک قدرت و تکنولوژی کنترل ماشین‌های متناوب، ماشین‌های مستقیم از رده خارج شدند اما هنوز صنایع قدیمی و صناعی که به کنترل خیلی دقیق نیاز دارند همچون جرثقیل‌های غول پیکر از ماشین مستقیم استفاده می‌کنند [۱]. در این پایان‌نامه از موتور متناوب القایی آسنکرون با مشخصات ارائه شده در جدول (۱-۲) استفاده شده است. در ادامه به معرفی موتورهای القایی و روش‌های کنترل دور آن‌ها پرداخته می‌شود. مشخصات ارائه شده در این جدول صرفاً اعداد اولیه انتخابی توسط خود طراح می‌باشد و مدار با سایر مقادیر نیز پاسخ مطلوبی دارد. تمرکز پایان‌نامه بیشتر روی قسمت طراحی مدار اینورتر می‌باشد.

جدول ۱-۲: مشخصات موتور القایی

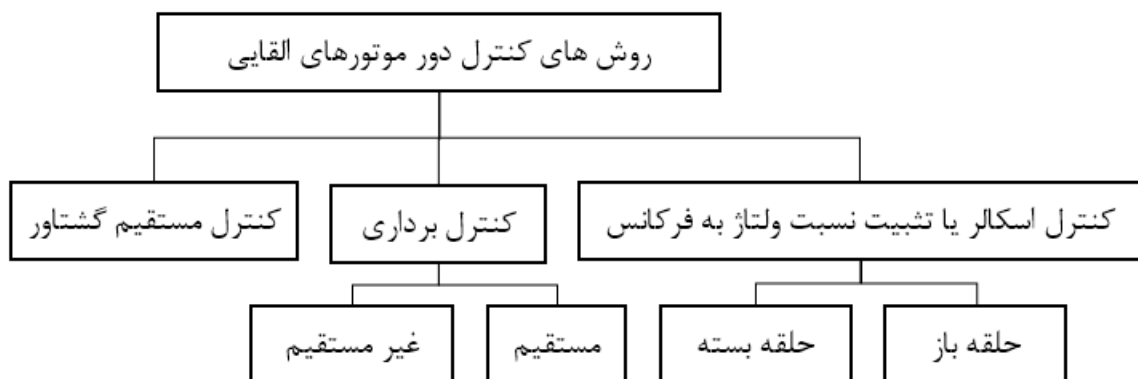
مشخصه	مقدار	یکا
توان نامی	۳۷۳۰	ولت آمپر
ولتاژ نامی (خط به خط)	۲۲۰	ولت (RMS)
فرکانس نامی	۵۰	هرتز
مقاومت اهمی استاتور	۱/۰۸	اهم
اندوکتانس استاتور	۵/۹	میلی هانری
ممان اینرسی	۰/۰۲	کیلوگرم

اگرچه مدارهای کنترل کننده دور موتور ادوات پیچیده‌ای هستند اما چون از ساختار مدارات الکترونیک قدرت استاتیک در آن‌ها استفاده می‌شود و فاقد قطعه متحرک می‌باشند، عمر مفید بالایی دارند. با استفاده از کنترل کننده‌های دور موتور مدرن می‌توان از اتلاف انرژی مصرفی در ترمزهای مکانیکی

و یا مقاومت‌های الکتریکی جلوگیری نمود. کنترل دور موتور مزایایی دارد که از جمله آن می‌توان به موارد زیر اشاره نمود:

- عمر مفید بالا (به دلیل استفاده از مدارات الکترونیک قدرت)
- توانایی درایو در بازگرداندن انرژی مصرفی در ترمزهای مکانیکی و یا مقاومت‌های الکتریکی به شبکه
- (بعضاً هزینه انرژی بازیافت شده از این طریق، در کمتر از یکسال معادل هزینه سرمایه گذاری سیستم بازیافت انرژی می‌شود).
- کاهش جریان راه‌انداز کشیده شده از شبکه (جریان راه‌اندازی کمتر از ۱۰ درصد جریان نامی می‌شود).
- کاهش مصرف انرژی در سیستم‌های دارای موتور (در گذشته با وجود موتورهای دور ثابت، کنترل جریان سیال با دمپرها صورت می‌گرفت).
- کاهش تنش‌های الکتریکی (به دلیل راه‌اندازی و توقف نرم) و در پی آن کاهش تنش‌های مکانیکی و این خود باعث کاهش هزینه‌های تعمیر و نگهداری می‌شود.
- افزایش دامنه تغییرات ممکن برای سرعت موتور نسبت به روش‌های مکانیکی
- اضافه شدن امکانات نرم‌افزاری برای مدیریت عملکرد کنترل دور [۱].

کنترل‌کننده‌های دور موتور انواع مختلفی دارند. آن‌ها قادر هستند انواع موتورهای متناوب و مستقیم را کنترل کنند. البته در نوعی دیگری از طبقه بندی می‌توان عملکرد کنترل‌کننده‌های دور را به سه دسته افراز نمود [۲]. شکل ۱-۲ خلاصه ای از انواع روش‌های کنترل موتورهای القایی نمایش داده شده است [۳ و ۴].



شکل ۱-۲: خلاصه ای از انواع روش های کنترل موتورهای القایی [۳]

۱-۱-۲ روش کنترل اسکالر

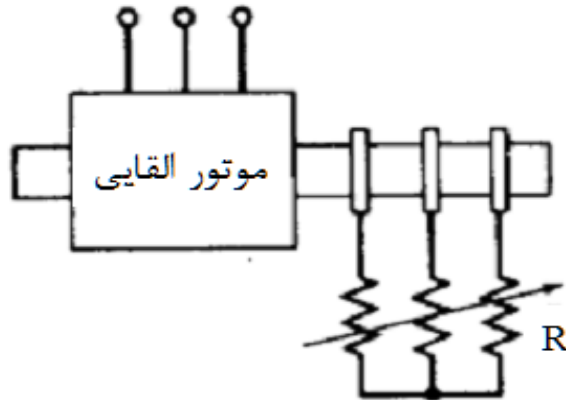
اگرچه روش های کنترل برداری نسبت به روش اسکالر پیچیده تر و گران تر می باشند ولی می توانند سرعت موتور را از صفر تا بالاتر از سرعت نامی کنترل کنند. روش کنترل اسکالر دارای محدودیت بازه کنترل دور می باشند [۱]. در یک موتور الکتریکی القایی، سرعت روتور یا n_r به صورت رابطه (۱-۲) قابل محاسبه می باشد.

$$n_r = \frac{(1 - s) \times 120 \times f_s}{p} \quad (1-2)$$

در این رابطه، f_s فرکانس تغذیه موتور، p تعداد قطبها و s لغزش موتور می باشند. برای تغییر سرعت موتور به روش اسکالر می توان این پارامترها را تغییر داد که در ادامه به طور مختصر شرح داده شده است.

الف) کنترل سرعت با تغییر لغزش:

جهت تغییر لغزش موتور می بایست یک پتانسیومتر در روتور موتورهای القایی سیم پیچی شده اضافه شود که با تغییر این پتانسیومتر منحنی گشتاور- سرعت تغییر کرده و دور موتور نیز تغییر می کند. از این روش کمتر استفاده می شود، چون افزودن مقاومت باعث ایجاد تلفات شده و بازده موتور کاهش پیدا می کند. این روش در شکل ۲-۲ نشان داده شده است [۵و۴].



شکل ۲-۲: افزودن مقاومت در روتور برای کنترل سرعت با تغییر لغزش [۵]

ب) کنترل سرعت با تغییر قطب:

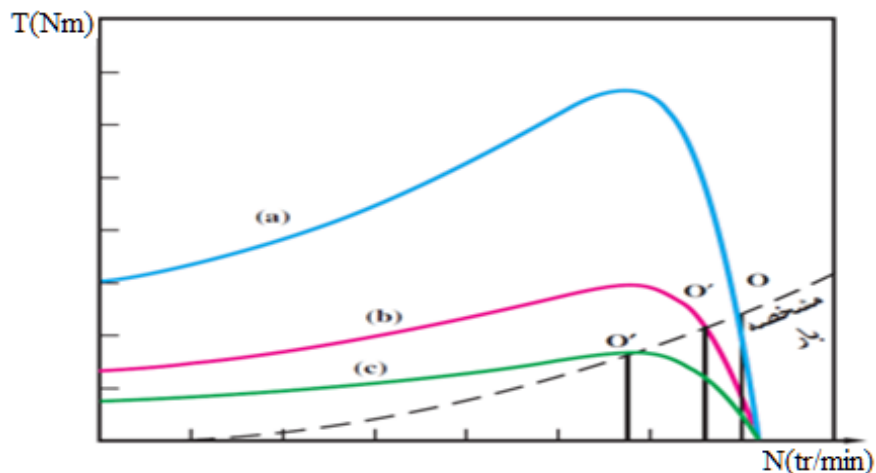
سرعت رابطه‌ی معکوسی با تعداد قطب‌های موتور دارد. بر این اساس می‌توان به جای یک سیم پیچ با استفاده از چند سیم پیچ با تعداد قطب‌های مختلف در استاتور، به سرعت‌های مختلف دست یافت. برای مثال در دو سیم پیچی ۶ و ۸ قطبی و در فرکانس ۵۰ هرتز، می‌توان به دو سرعت ۱۰۰۰ و ۷۵۰ دور بر دقیقه دست یافت. در این روش تغییرات سرعت پیوسته نمی‌باشد و فقط می‌توان چند سرعت گسسته داشت. ضمن اینکه این روش باعث کاهش بازده ماشین شده و هزینه آن را افزایش می‌دهد [۶].

پ) کنترل سرعت با تغییر ولتاژ:

با توجه به رابطه (۲-۲) گشتاور مکانیکی موتور یا T_{mech} با مجذور ولتاژ تغذیه یا V_{th} مرتبط است. در این رابطه ω_{syn} سرعت سنکرون، R'_2 و X'_2 مقادیر مقاومت اهمی و رلوکتانس راه‌اندازی موتور و R_{th} و X_{th} مقدار مقاومت اهمی و رلوکتانس موتور است [۱].

$$T_{mech} = \frac{1}{\omega_{syn}} \frac{V_{th}^2}{\left[\left(R_{th} + \frac{R'_2}{s} \right)^2 + (X_{th} + X'_2)^2 \right]} \frac{R'_2}{s} \quad (2-2)$$

تغییرات ولتاژ، منحنی گشتاور سرعت را بصورت منحنی شکل ۲-۳ تغییر می‌دهد. مشخصه موتور با ولتاژ بالاتر از نامی با (a)، مشخصه موتور در ولتاژ نامی با (b)، مشخصه موتور با ولتاژ پایین‌تر از نامی با (c) و مشخصه بار با خط‌چین نشان داده شده است.

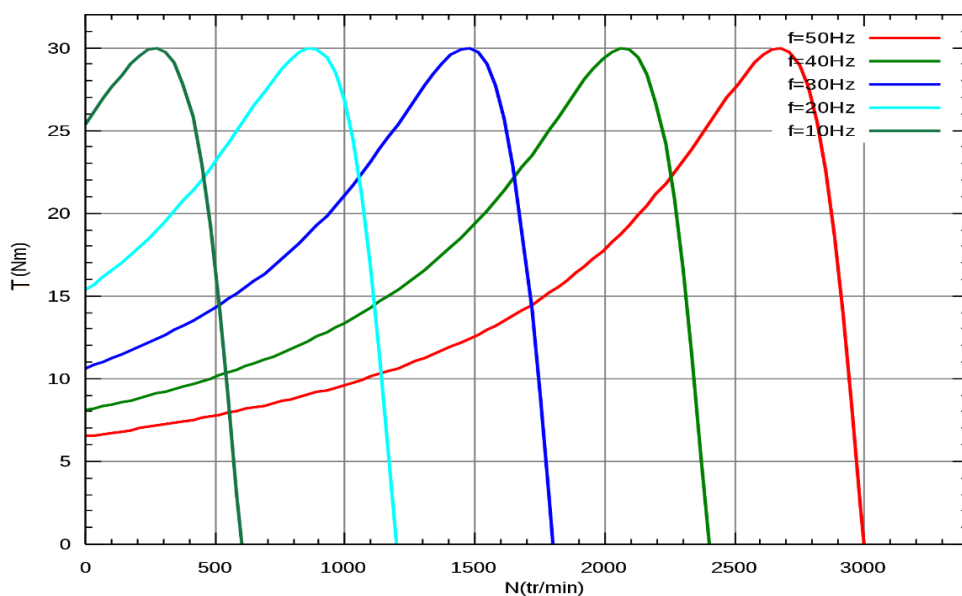


شکل ۲-۳: منحنی گشتاور سرعت موتور متناوب [۱]

پس می‌توان با تغییر ولتاژ سرعت را کنترل کرد. البته باید توجه داشت که بخاطر مسایل عایقی ولتاژ از مقدار نامی خود نمی‌تواند بیشتر شود. ضمن اینکه کاهش زیاد ولتاژ باعث تغییر شار شده که اثرات نامطلوبی بر روی موتور دارد.

ت) کنترل سرعت با تغییر فرکانس منبع:

یکی از روش‌های تغییر دور موتور، تغییر فرکانس منبع تغذیه است. بنابراین به یک تغییر دهنده فرکانس یا مبدل یا همان اینورتر نیاز است. یک یا چند موتور القایی می‌توانند بوسیله‌ی این مبدل‌ها تغذیه شوند. شکل ۲-۴ تغییرات منحنی گشتاور - سرعت بر حسب تغییرات فرکانس را نشان می‌دهد.



شکل ۲-۴: تغییرات منحنی گشتاور سرعت بر حسب تغییرات فرکانس [۱]

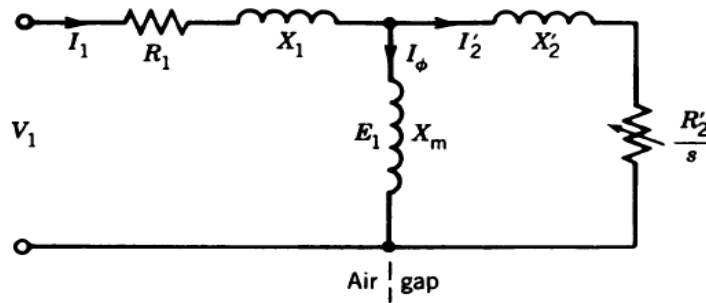
عمده ترین عیبی که کنترل سرعت موتورهای القایی با تغییر فرکانس برای موتور بوجود می آورد، گرم شدن هسته استاتور به سبب افزایش فرکانس از حد نامی می باشد. چون تلفات آهنی هسته استاتور با مجذور فرکانس رابطه مستقیم دارد. به عنوان مثال از کاربرد این روش می توان از موتور ماشین های کارخانه فولاد و موتورهای محرک ماشین نساجی نام برد [۷۶].

ث) روش تثبیت نسبت ولتاژ به فرکانس (یا کنترل V/F ثابت)

ساده ترین روش در کنترل موتورهای متناوب روش تثبیت نسبت ولتاژ به فرکانس می باشد. این روش، به طور گسترده در کاربردهای صنعتی مورد استفاده قرار می گیرد. این نوع کنترلرها از نوع اسکالر بوده و به صورت حلقه باز با پایداری خوب عمل می کنند. مزیت این روش سادگی سیستم های کنترلی آن است. اما این نوع کنترل کننده ها برای کاربردهای با پاسخ سریع مناسب نمی باشند [۳]. درایوهای زیمنس خانواده Sinamics، Micromaster و اشنایدرالکترونیک خانواده ATV از این روش به علاوه روش های دیگر کنترل سرعت استفاده می کنند.

لازم است روابط روش تثبیت نسبت ولتاژ به فرکانس به صورت دقیق تر مورد بررسی قرار گیرد. چنانچه مدار معادل موتور القایی مطابق شکل ۲-۵ در نظر گرفته شود، می توان نوشت.

$$V_1 = (R_1 + jX_1)i_1 + E_1 \quad (3-2)$$



شکل ۲-۵: مدار معادل موتور القایی [۳]

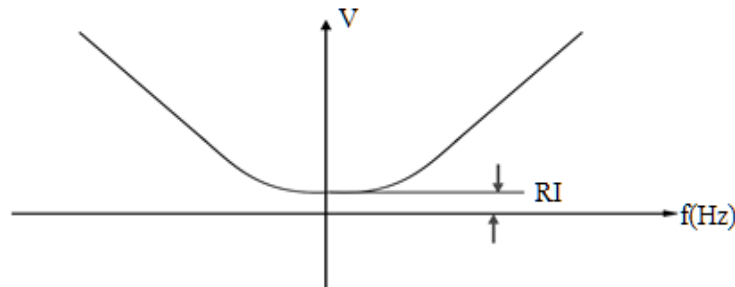
در رابطه (۳-۲)، E_1 که ولتاژ القایی است براساس رابطه (۲-۴) مقدار RMS آن محاسبه می‌شود.

$$E_{1,rms} = 4.44fN\phi_m \quad (4-2)$$

در این رابطه، N تعداد حلقه‌های سری در هر فاز و ϕ_m شار هر قطب در شکاف هوایی می‌باشد. اگر از افت ولتاژ مقاومت استاتور صرف نظر شود، رابطه (۲-۵) به دست می‌آید.

$$V_1 \approx E_1 \rightarrow \phi_m \propto \frac{V}{f} \quad (5-2)$$

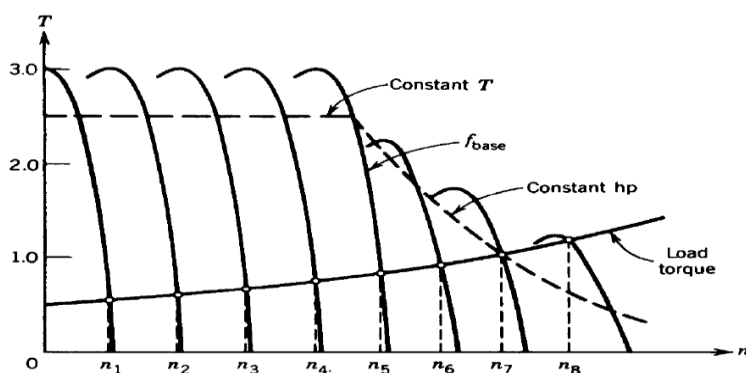
طبق رابطه (۲-۵) برای جلوگیری از به اشباع رفتن موتور باید ولتاژ پایانه موتور متناسب با فرکانس تغییر نماید. به این روش، تثبیت نسبت ولتاژ به فرکانس یا کنترل V/f ثابت گفته می‌شود. لازم به یادآوری است که در فرکانس‌های پایین (حدود ۲۰ درصد فرکانس نامی) افت ولتاژ در دو سر R_1 و X_1 در مقابل V قابل صرف نظر نمی‌باشد و لذا دیگر نمی‌توان شار را ثابت در نظر گرفت. بنابراین برای کنترل دور باید تغییرات ولتاژ بر حسب فرکانس بصورت شکل ۲-۶ تغییر یابد [۳ و ۱].



شکل ۲-۶: تغییرات ولتاژ بر حسب فرکانس برای کنترل دور [۱]

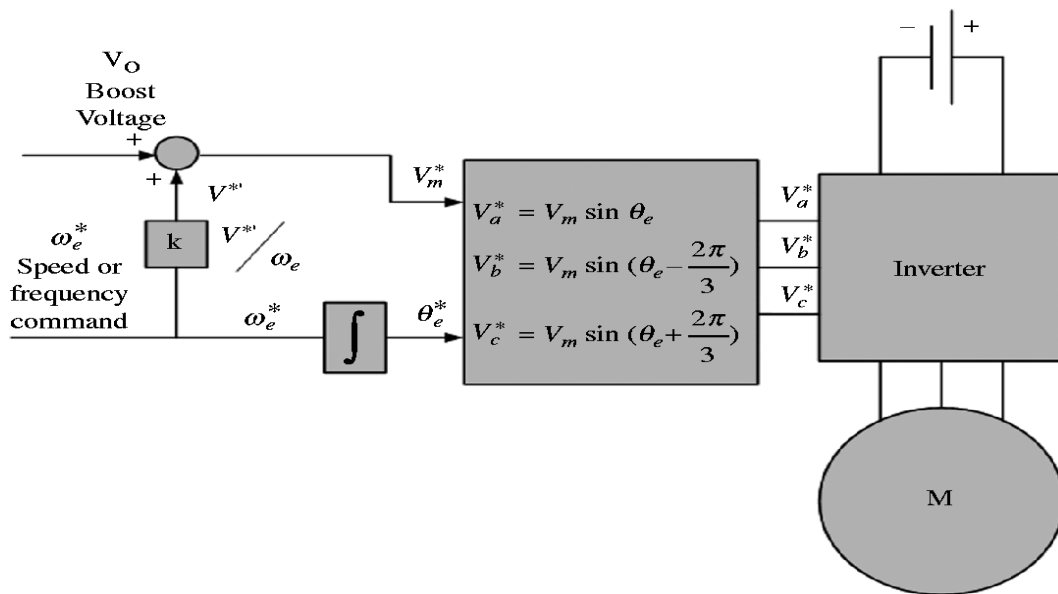
باید توجه داشت که حداکثر مقدار ولتاژی که در پایانه موتور توسط اینورتر حاصل می‌شود در فرکانس مبنا (f_{base}) است. در فرکانس‌های زیر فرکانس مبنا، شار شکاف هوایی با تغییر نسبت ولتاژ به فرکانس ثابت نگه داشته می‌شود پس در این ناحیه گشتاور بیشینه ثابت باقی خواهد ماند، اما در ناحیه فرکانس‌های بالاتر از فرکانس مبنا دیگر ولتاژ را نمی‌توان متناسب با فرکانس افزایش داد و در این حالت شار در شکاف هوایی کاهش یافته و بنابراین گشتاور بیشینه نیز کاهش می‌یابد [۷].

منحنی مشخصه گشتاور سرعت بر اساس V/f ثابت یک موتور القایی در شکل ۷-۲ نشان داده شده است.

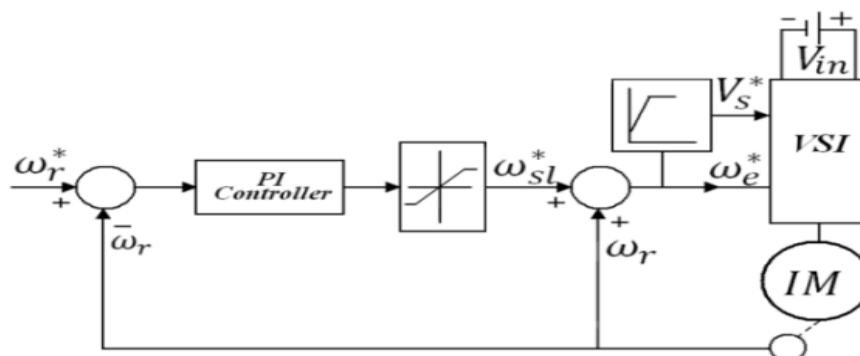


شکل ۷-۲: مشخصه گشتاور سرعت بر اساس V/f ثابت یک موتور القایی [۱]

این روش به دو صورت حلقه باز و حلقه بسته پیاده‌سازی می‌شود. بلوک دیاگرام کنترلی حلقه باز به صورت شکل ۸-۲ و بلوک دیاگرام حلقه بسته به صورت شکل ۹-۲ می‌باشد.



شکل ۲-۸: بلوک دیاگرام کنترلی حلقه باز برای روش V/f ثابت [۷]



شکل ۲-۹: بلوک دیاگرام کنترلی حلقه بسته برای روش V/f ثابت [۷]

۲-۱-۲ روش کنترل برداری

این روش برای حالت‌هایی که سیستم دارای دینامیک پیچیده است، استفاده می‌شود. روبات‌ها و ماشین ابزارها نمونه‌هایی از کاربردهای با دینامیک بالا هستند. در این کاربردها روش‌های کنترلی برداری استفاده می‌شود. در روش کنترل برداری با تفکیک مولفه‌های جریان استاتور به دو مولفه تولیدکننده گشتاور و تولیدکننده شار الکترومغناطیسی و کنترل آن‌ها با استفاده از رگولاتورهای بخصوص ترتیبی داده می‌شود که موتور متناوب نظیر موتور مستقیم کنترل شود و به این ترتیب تمام مزایای موتور مستقیم از

جمله پاسخ گشتاور سریع آن در موتورهای متناوب نیز در دسترس خواهد بود. این روش کنترل سرعت در درایوهای میکرومستر^۱ شرکت زیمنس و درایوهای آلتیور^۲ شرکت اشنایدر استفاده می‌شود [۳].

۲-۱-۳ روش کنترل مستقیم گشتاور

در روش کنترل مستقیم گشتاور، دورموتور از طریق تنظیم جریان و گشتاور آن تنظیم می‌شود که یک روش حلقه بسته است. انتخاب دقیق و درست کنترل دور در هر کاربرد، متضمن کارکرد درست و عمر بالای دستگاه خواهد بود [۸].

بنا بر توضیحات داده شده، به طور کلی روش‌های کنترل دور در موتورهای القایی، به چهار دسته اصلی زیر تقسیم می‌شوند که رایج‌تر از سایر روش‌ها هستند.

الف) کنترل دور به وسیله کنترل ولتاژ

ب) کنترل دور توسط کنترل فرکانس

ج) کنترل دور به وسیله کنترل ولتاژ و کنترل فرکانس به صورت همزمان

د) کنترل سرعت از طریق کنترل جفت قطب

در این میان، بهترین روش کنترل دور به وسیله کنترل همزمان ولتاژ و فرکانس است. زیرا میزان فوران میدان مغناطیسی دوار موتورهای القایی متناسب با نسبت ولتاژ به فرکانس می‌باشد. بنابراین اگر به همان نسبت که ولتاژ را تغییر می‌دهیم فرکانس را نیز تغییر دهیم، مقدار دامنه‌ی میدان مغناطیسی دوار موتور تغییری نمی‌کند و در کار ماشین اختلالی به وجود نخواهد آمد. از سوی دیگر با تغییر مشخصات منبع تغذیه منحنی گشتاور بر حسب سرعت نیز تغییر خواهد نمود. از معایب این روش می‌توان به قیمت بسیار بالای مبدل‌هایی که به صورت همزمان فرکانس و ولتاژ را کنترل می‌نمایند اشاره نمود. این مبدل‌ها ساختمان پیچیده‌ای دارند و تعمیر و نگهداری آن‌ها مشکل می‌باشد. بنابراین در بسیاری از مواقع از نظر اقتصادی استفاده از این روش برای کنترل سرعت مقرون به صرفه نیست.

¹ Micromaster

² Altivar

با توجه به این که روش V/f ثابت روش برگزیده طراح برای کنترل سرعت در این پایان نامه است، بنابراین انتظار می‌رود در صورتی که شبیه‌سازی مدار حلقه بسته مانند شکل ۲-۹ صورت پذیرد، مشخصه گشتاور سرعت مشابه مشخصه شکل ۲-۷ حاصل شود. این مساله در فصل شبیه‌سازی بیشتر مورد بررسی قرار می‌گیرد.

۲-۲ کنترل‌کننده‌های دور موتورهای القایی

کنترل سرعت موتورهای الکتریکی از گذشته تاکنون از جمله مسائل مهم، پیچیده و پرهزینه در صنعت بوده است. در زمان‌های گذشته برای سیستم‌هایی که در آن‌ها نیاز به موتورهای الکتریکی با سرعت متغیر بوده به ناچار بایستی از موتورهای مستقیم استفاده می‌شد. استفاده از موتورهای مستقیم مشکلاتی را به همراه داشت از جمله این مشکلات می‌توان به نیاز به یکسوسازهای پر قدرت، هزینه تعمیر و نگهداری بالا، وزن و حجم بیشتر در مقایسه با موتورهای متناوب هم رده، قیمت بالا و... اشاره نمود. این مشکلات و برخی دلایل دیگر باعث شده امروزه کمتر از موتورهای مستقیم استفاده شود و استفاده از موتورهای متناوب با سرعت متغیر به سرعت در حال رشد باشد. برای کنترل دور موتورهای متناوب امروزه راه‌اندازهای زیادی توسط شرکت‌های مختلف ارائه می‌گردد [۷۵].

راه‌اندازها قادرند دور موتور را از صفر تا چندین برابر دور نامی موتور و به طور پیوسته تغییر دهند. تنظیم دور موتورهای الکتریکی در کاربردهای زیادی باعث صرفه جویی در انرژی می‌شود. ضمن اینکه راه‌اندازها جریان کشیده شده از شبکه را به میزان زیادی کاهش می‌دهند، به طوری که این جریان خیلی کمتر از جریان اسمی موتور است. این مبدل‌ها می‌توانند موتور را به طور نرم و کاملاً کنترل شده راه‌اندازی و متوقف نمایند. زمان راه‌اندازی و توقف را می‌توان به دقت تنظیم نمود. این زمان‌ها می‌توانند کسری از ثانیه و یا صدها دقیقه باشد. توانایی مبدل در راه‌اندازی و توقف نرم موجب کاهش قابل ملاحظه تنش‌های مکانیکی در کوپلینگ‌ها و سایر ادوات دوار می‌گردد [۶].

راه‌اندازها یا کنترل‌کننده‌های دور موتور می‌توانند مشخصه‌های بار را با مشخصه‌های موتور تطبیق دهند. این راه‌اندازها توان راکتیو ناچیزی از شبکه می‌کشند و لذا نیازی به اصلاح ضریب بار ندارند. همچنین این کنترل‌کننده‌ها موتور را نرم راه‌اندازی می‌کنند، لذا علاوه بر کاهش تنش‌های الکتریکی، از

شوکه‌های مکانیکی به بار نیز جلوگیری می‌شود. این شوک‌های مکانیکی می‌توانند باعث استهلاک سریع قسمت‌های مکانیکی و قسمت‌هایی از بار شوند. راه‌اندازی نرم هزینه‌های نگهداری را کاهش داده و به افزایش عمر مفید محرکه‌ها و قسمت‌های دوار منجر خواهد شد. جریان کشیده شده از شبکه در هنگام راه‌اندازی موتور با استفاده از درایور کمتر از ۱۰٪ جریان نامی موتور است [۹].

در صورتی که نیاز بار ایجاب کند با استفاده از این کنترل‌کننده‌ها، موتور می‌تواند در سرعت‌های پائین کار کند. کار در سرعت‌های کم منجر به کاهش هزینه‌های تعمیر و نگهداری ادوات مکانیکی خواهد شد.

راه‌انداز موتورهای متناوب یا اینورترها، جریان متناوب را با استفاده از یک پل دیود و خازن، به جریان مستقیم تبدیل می‌کنند. سپس با سویچ کردن جریان با فرکانس مورد نظر، جریانی با فرکانس دلخواه تولید می‌کنند. هر قدر در یک موتور متناوب فرکانس را بالاتر رود، سرعت گردش موتور بالاتر می‌رود. ضمن اینکه گشتاور در فرکانس‌های پایین‌تر و یا بالاتر از فرکانس برق شهر تقریباً حفظ می‌شود [۹].

به طور کل از لحاظ نوع تغذیه اینورتر و باری که اینورتر آن را تغذیه می‌کند، می‌توان اینورترها را به دو گروه زیر تقسیم کرد.

- اینورترهای منبع ولتاژ VSI^۱.
- اینورترهای منبع جریان CSI^۲.

در اینورترهای منبع جریان ورودی اینورتر جریان مستقیم و خروجی آن ولتاژ متناوب سینوسی است. این مبدل‌ها بیشتر در کاربردهای درایوهای ماشین‌های بزرگ صنعتی یا در موارد توان بالا کاربرد دارند. در اینورترهای منبع ولتاژ، ورودی ولتاژ مستقیم و خروجی جریان متناوب سینوسی می‌باشد. در هر دو اینورترها توان قابلیت انتقال در هر دو سمت را دارد [۹]. به عبارت دیگر در صورتی که ولتاژ و جریان هم علامت باشند سیستم به صورت اینورتر و در صورتی که مختلف‌العلامه باشند سیستم بصورت یکسوساز عمل می‌کند.

از لحاظ دسته بندی اتصال اینورترها، می‌توان آن‌ها را به دو دسته تقسیم بندی نمود.

¹ Voltage Source Inverters

² Current Source Invertors

- اینورترهای حقیقی
- اینورترهای مجازی

اگر شبکه متصل به اینورتر، اکتیو باشد، برای مثال در کاربردهای تولید انرژی‌های نوین، اینورتر مجازی می‌باشد. به عبارت دیگر اینورتر یک مبدل پل تریستوری با زاویه آتش بزرگتر از ۹۰ درجه خواهد بود. اما در صورتی که این شبکه پسیو باشد و اینورتر به این شبکه پسیو متصل باشد، اینورتر حقیقی بوده و عمل تبدیل مستقیم به متناوب را انجام می‌دهد.

از لحاظ نوع کموتاسیون نیز می‌توان اینورترها را به دو دسته زیر تقسیم نمود.

- اینورترهای با کموتاسیون طبیعی، کموتاسیون خط
- اینورترهای با کموتاسیون اجباری

در سیستم‌های متصل به شبکه معمولاً از کموتاسیون‌های طبیعی استفاده می‌کنند، لیکن در کموتاسیون اجباری از طریق مدار جانبی کموتاسیون صورت می‌گیرد. از لحاظ نوع شبکه می‌توان اینورترها را به دو دسته تکفاز و سه‌فاز تقسیم کرد. اینورترهای تکفاز نیز دارای انواع مختلفی می‌باشند مانند اینورترهای نیم موج، تمام موج و پوش پول^۱ که هر کدام در کاربردهای مخصوصی مورد استفاده دارند [۷۶].

از نظر نوع مدار تحریک عناصر سوئیچ می‌توان اینورترها را به صورت زیر تقسیم‌بندی نمود.

- اینورترهای موج مربعی
- اینورترهای با مدولاسیون پالسی

در اینورترهای موج مربعی عمل کنترل ولتاژ از طریق یکسوساز انجام می‌شود، تا دامنه موج متناوب خروجی کنترل شود. این نوع اینورترها فقط عمل کنترل فرکانس را انجام می‌دهند. در این حالت شکل موج خروجی مربعی می‌باشد. در اینورترهای با مدولاسیون پالسی، یکسوسازها معمولاً بصورت دیودی بوده و عمل کنترل ولتاژ و فرکانس فقط از طریق اینورتر صورت می‌گیرد. این کار از طریق اعمال الگوهای

^۱Push Pull

مختلف پالس به کلیدهای اینورتر صورت می‌گیرد. الگوهای مختلفی برای نزدیک‌تر کردن سیگنال شکل موج خروجی به موج سینوسی وجود دارند از جمله مدولاسیون پهنای پالس، مدولاسیون بردار فضایی، مدولاسیون عرض پالس سینوسی و... که هر کدام کاربردهای ویژه ای دارند [۹۵].

از دیدگاه تعداد سطوح سیگنال خروجی، اینورترها به سه دسته زیر تقسیم می‌شوند.

- اینورترهای دو سطحی که در آنها شکل موج خروجی دارای دو سطح خروجی مثبت و منفی است.
- اینورترهای سه سطحی که در آنها علاوه بر دو سطح مثبت و منفی، شکل موج سطح صفر نیز اضافه می‌گردد. عمل حذف ولتاژی سطح صفر را اضافه می‌کند.
- اینورترهای چند سطحی که در آنها از چند عنصر کلیدزنی در هر بازوی پل استفاده می‌شود. با ترکیب مناسب این عناصر باهم می‌توان به چندین سطح در سیگنال خروجی رسید. با اتصال موازی اینورترها نیز این عمل را می‌توان انجام داد. این کار در کاهش ابعاد سیستم فیلترینگ مفید است.

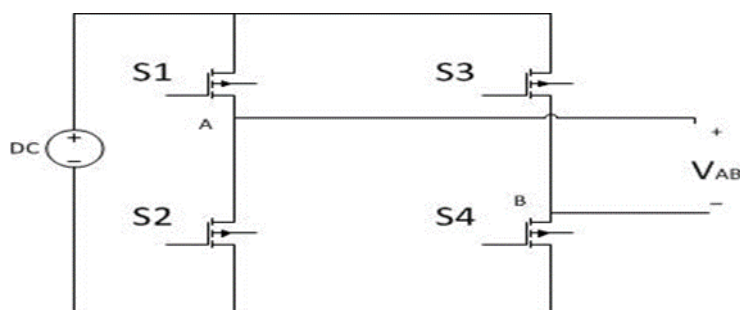
اینورترهای صنعتی که بیشتر برای کاربردهای فرکانس بالا استفاده می‌گردند، اینورترهای تشدید نامیده می‌شوند. در اینورترهای تشدید کلیدزنی عناصر در لحظه صفر شدن ولتاژ یا جریان صورت می‌گیرد. بنابراین در مقدار تلفات سوئیچینگ، کاهش قابل ملاحظه‌ای حاصل می‌شود. این اینورترها به دو دسته شرح داده شده در ذیل، تقسیم می‌شوند.

- اینورترهای با تشدید بار که در آنها از یک بار سلفی خازنی برای ایجاد رزونانس استفاده می‌شود. بسته به مقادیر مختلف ضریب میرایی و فرکانس اینورتر، این سیستم‌ها عملکردهای متفاوتی دارند که در هر حالت برای کاربرد خاصی مورد استفاده قرار می‌گیرند. اینورترهای با تشدید بار خود دارای دو نوع می‌باشند که شامل اینورترهای تشدید با مدار تشدید سری و اینورترهای تشدید با مدار تشدید موازی است. در اینورترهای تشدید با مدار تشدید سری، از یک سیستم رزونانسی سری در خروجی به همراه بار استفاده می‌گردد، که سلف سری باعث پیوستگی جریان خروجی خواهد شد. بنابراین این اینورتر بایستی به یک منبع ولتاژ متصل گردد، یعنی یک اینورتر منبع ولتاژ می‌باشد. در اینورترهای تشدید با مدار تشدید موازی از

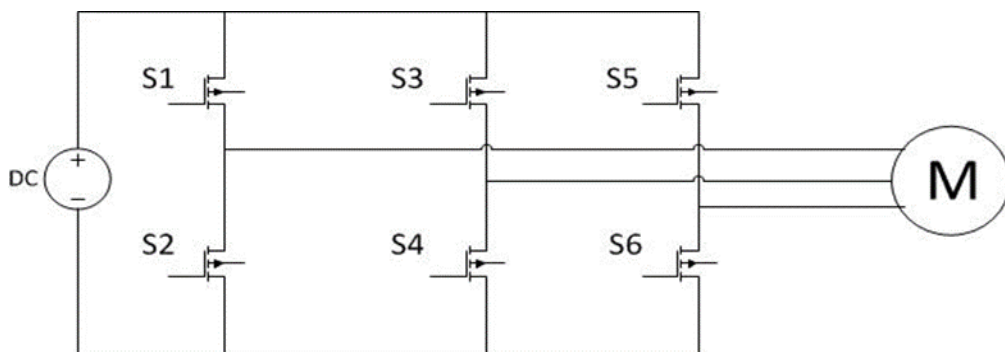
یک سیستم رزونانسی موازی در خروجی اینورتر به همراه بار استفاده می‌شود که وجود خازن موازی باعث پیوستگی در ولتاژ خروجی خواهد شد. بنابراین این اینورتر بایستی به یک منبع جریان متصل گردد، یعنی یک اینورتر منبع جریان می‌باشد.

- اینورترهای با لینک مستقیم تشدید می‌شوند که در آن‌ها به ولتاژ مستقیم ورودی به اینورتر اجازه داده می‌شود تا حول یک مقدار ثابت نوساناتی را داشته باشد. در این حالت ولتاژ ورودی طی زمان محدودی صفر می‌ماند و اجازه سوئیچینگ در این لحظات به کلیدهای اینورتر داده می‌شود [۹].

گفته شد که از لحاظ فاز شکل موج خروجی اینورترها به دو دسته‌ی تکفاز و سه‌فاز تقسیم می‌شوند. انتخاب هر یک از این موتورها بسته به بار یا موتور خروجی دارد. ساختار مدار ساده اینورتر تکفاز و سه‌فاز به ترتیب در شکل‌های ۲-۱۰ و ۲-۱۱ نمایش داده شده‌اند. سوئیچ‌های S1 تا S6 می‌تواند سوئیچ ایده‌آل، ماسفت، تریستور و... باشد. برای کاربردهای سرعت بالا و فرکانس کاری زیاد می‌توان از ماسفت استفاده نمود [۱۰].



شکل ۲-۱۰: مدار ساده اینورتر تکفاز [۱۰]



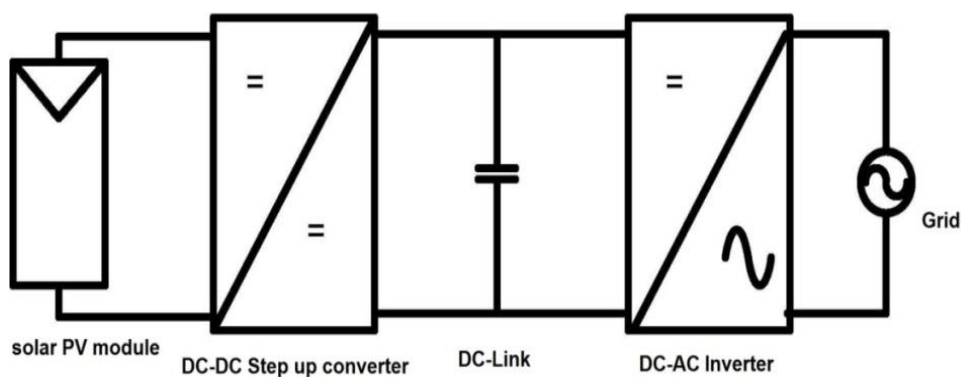
شکل ۲-۱۱: مدار ساده اینورتر سه‌فاز با ورودی مستقیم [۱۰]

۳-۲ معرفی توپولوژی‌های مختلف اینورتری

در این بخش توپولوژی‌های متعدد اینورترها معرفی می‌شوند و ویژگی‌های آن‌ها مورد ارزیابی قرار می‌گیرند. توپولوژی‌ها به سه دسته تقسیم می‌شوند: اینورترهای با لینک مستقیم، اینورترهای با مدار بازکننده تای جریان و اینورترهای بدون لینک مستقیم. در ادامه هر کدام از این سه دسته مورد بررسی قرار می‌گیرند.

۱-۳-۲ اینورترهای با لینک مستقیم

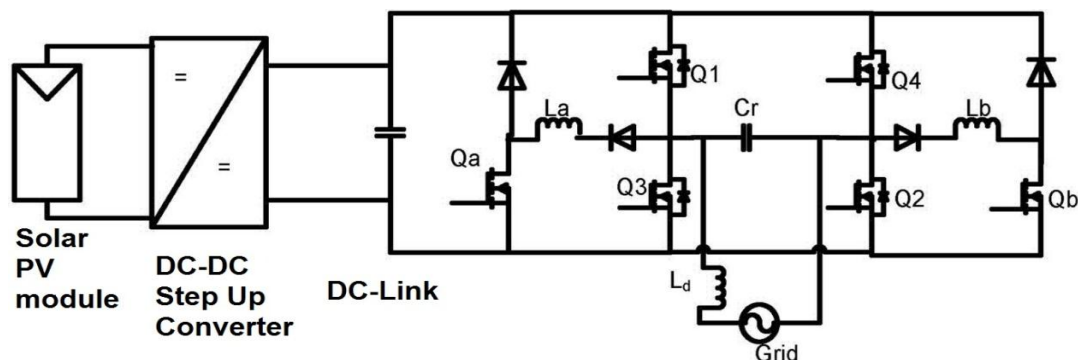
توپولوژی کلی اینورترهای با لینک مستقیم در شکل ۲-۱۲ نمایش داده شده است [۱۱]. مبدل افزایشی DC-DC ولتاژ مازول فتو ولتایی را افزایش می‌دهد. ولتاژ این مازول‌ها عموماً در حدود ۲۲ تا ۴۵ ولت می‌باشد که توسط این طبقه به ولتاژی که بزرگتر از ولتاژ پیک شبکه باشد تبدیل می‌گردد. سپس در طبقه بعدی، یک اینورتر، این ولتاژ مستقیم را به وسیله تکنیک‌های PWM به ولتاژ متناوب شبکه تبدیل می‌کند. خازن لینک مستقیم انرژی را ذخیره می‌کند و وظیفه آن جداسازی توان است. به عبارت دقیق‌تر این خازن واسطی بین توان مستقیم و توان متناوب است. وجود این خازن الزامی است، چون توان مستقیم ثابت است ولی توان لحظه‌ای در یک ولتاژ متناوب با زمان تغییر می‌کند.



شکل ۲-۱۲: ساختار کلی اینورترهای با لینک مستقیم [۱۱].

اینورترهای با لینک مستقیم از تعداد بالایی برخوردارند. در ادامه ۴ حالت مختلف را مورد بررسی قرار می‌دهیم. توپولوژی که در شکل ۲-۱۳ نمایش داده شده است از مرجع [۱۲] انتخاب شده است. طبقه

اینورتری، از تکنیک گذر ولتاژ صفر رزونانسی^۱ استفاده می‌کند. با استفاده از این تکنیک، کلیدزنی در فرکانس ۲۵ کیلو هرتز انجام می‌شود. استراتژی کنترلی به گونه‌ای بهبود یافته است تا تلفات طبقه اینورتر کاهش یابد. نمونه آزمایشی در توان ۲۰۰ وات ساخته شده است و بازده ماکزیممی در حدود ۹۶ درصد برای آن بیان شده است. با این حال این توپولوژی دارای ایزولاسیون الکتریکی نمی‌باشد.

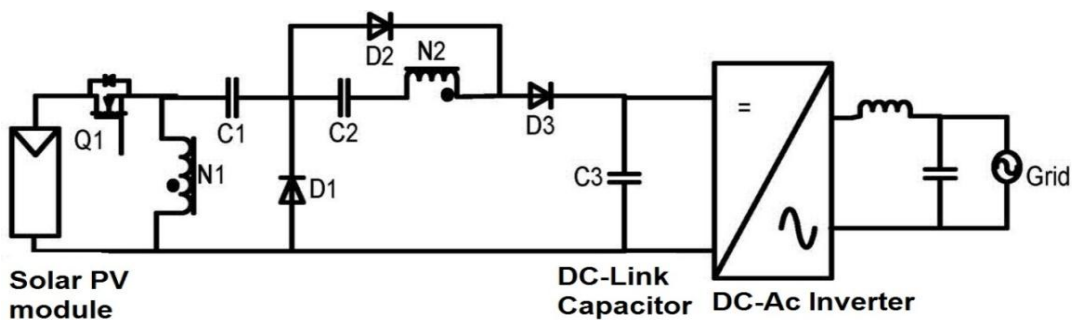


شکل ۲-۱۳: اینورتر لینک مستقیم دارای رزونانس گذر ولتاژ صفر [۱۲].

در منبع [۱۳] همان طور که در شکل ۲-۱۴ نیز نمایش داده شده است برای طبقه افزایش ولتاژ در میکرواینورتر، از یک مبدل بوست بدون ترانسفورماتور استفاده شده است. توپولوژی بر پایه مبدل بوست زتا می‌باشد که تفاوت آن با این مبدل در این است که سلف ورودی آن با یک زوج سلف تزویج شده جایگزین شده است. مزیت عمده ای که این توپولوژی دارد این است که انرژی که در اندوکتانس نشستی وجود دارد، به طور موثری می‌تواند بازیابی شود. این امر علاوه بر این که موجب افزایش راندمان می‌شود، تاثیر مستقیمی بر کاهش استرس ولتاژ بر روی کلید Q1 دارد. در نمونه ۲۵۰ واتی آن که در فرکانس ۵۰ کیلو هرتز کلیدزنی می‌شود، در بار ۵۰ وات دارای ماکزیمم بازده ۹۷.۳ درصدی می‌باشد و در بار نامی آن، بازدهی ۹۴.۸ درصدی دارد.

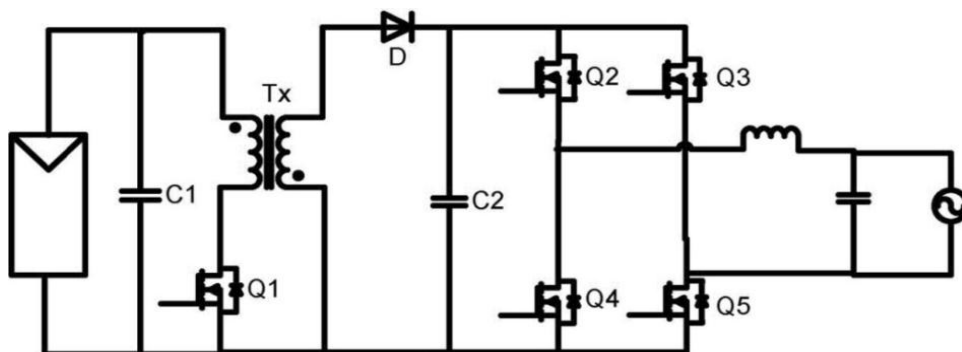
^۱ Zero voltage resonant transition

^۱ Zeta boost converter



شکل ۲-۱۴: اینورتر با لینک مستقیم با استفاده از مبدل بوست زتا [۱۳]

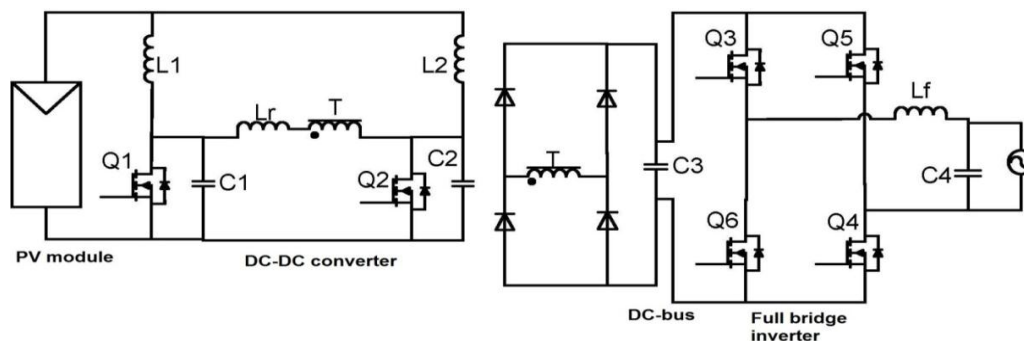
توپولوژی که در مرجع [۱۴] آورده شده است شامل یک مبدل DC-DC فلیپک است که به وسیله آن ولتاژ ورودی به ولتاژی بزرگتر از ولتاژ پیک تبدیل می‌شود. در طبقه بعد یک اینورتر تمام پل، ولتاژ مستقیم را توسط PWM به ولتاژ متناوب تبدیل می‌کند. این ساختار در شکل ۲-۱۵ نمایش داده شده است. فرکانس کلیدزنی در هر دو مبدل ۲۵ کیلو هرتز است. مزیت عمده این توپولوژی این است که به واسطه ایزوله سازی ترانسفورماتوری نیازی به زمین کردن طرف مستقیم و متناوب مدار نمی‌باشد. همچنین تعداد کم اجزای تشکیل دهنده آن باعث افزایش قابلیت اطمینان و نیز کاهش قیمت سیستم می‌گردد. برای این ساختار مقداری برای بازده عنوان نشده است. با این حال از آن جایی که در این توپولوژی از کلیدزنی سخت^۱ استفاده شده است طبیعتاً انتظار می‌رود بازده پایینی داشته باشد.



شکل ۲-۱۵: اینورتر فلیپک با ساختار تمام پل [۱۴]

¹ Hard switching

در مرجع [۱۵] توپولوژی مشابه ساختار قبلی معرفی شده است. با این تفاوت که از یک فیلتر LC در ورودی طبقه مبدل تمام پل استفاده شده است. در مراجع [۱۶] و [۱۷] همان طور که در شکل ۲-۱۶ آورده شده، از یک مبدل سویچینگ نیم پل استفاده شده است.^۱



شکل ۲-۱۶: اینورتر با مبدل سوئیچینگ نیم پل [۱۶] و [۱۷]

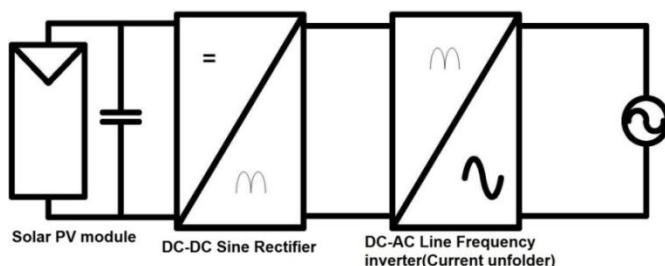
مهمترین نکته‌ای که در این تبدیل ولتاژ مستقیم به مستقیم وجود دارد این است که این مبدل به خوبی می‌تواند در هر دو حالت عملکرد پیوسته و گسسته کار کند. برای طبقه مبدل DC-DC در یک نمونه ۸۵ وات که در حالت عملکرد CCM^۲ و در فرکانس ۱ مگاهرتز کار می‌کند بازده ۹۰ درصدی از سیستم به دست آمده است. این بازده پایین نشانگر این است که اگرچه کلیدزنی نرم نقش موثری در به دست آوردن یک میکرواینورتر با راندمان بالا دارد، ولی دیگر تمهیدات نیز باید در نظر گرفته شوند. دیگر تمهیدات شامل انتخاب اجزای کلیدزنی، فرکانس کلیدزنی و... می‌شود.

۲-۳-۲ اینورتر با مدار بازکننده تای جریان

اینورترهای با مدار بازکننده تای جریان عموماً دارای دو طبقه هستند. خروجی طبقه اول یک جریان سینوسی یک سو شده است که دارای فرکانسی دو برابر فرکانس شبکه است و در فرکانس کلیدزنی بالا تولید می‌شود. در طبقه دوم تای این جریان باز می‌شود و در فرکانس شبکه به شبکه تزریق می‌گردد. توپولوژی کلی در شکل ۲-۱۷ نمایش داده شده است.

^۲ half wave zero voltage switching resonant converter

^۱ Continuous Conduction Mode



شکل ۲-۱۷: ساختار کلی اینورتر با مدار باز کننده تای جریان [۱۱]

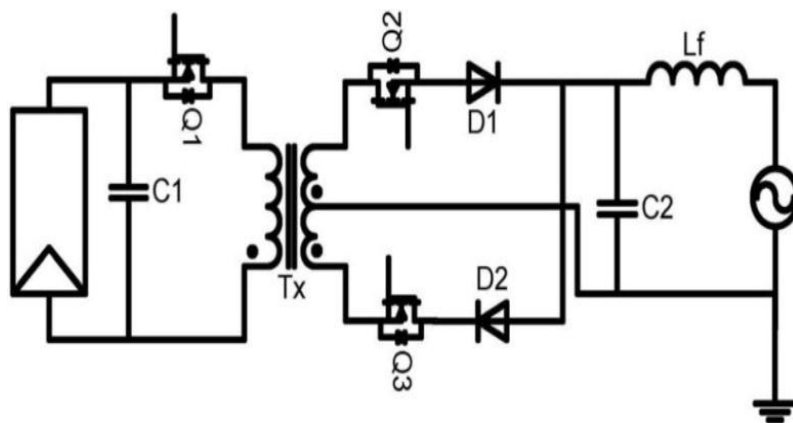
در مرجع [۱۷] تعدادی از این توپولوژی‌ها معرفی شده‌اند. اغلب این توپولوژی‌ها، بر پایه ترانسفورماتور می‌باشند و لذا این مدل از میکرواینورترها به طور ذاتی ایزولاسیون الکتریکی را تامین می‌کنند. در این قسمت ۴ مدل از توپولوژی‌های دارای مدار باز کننده تای جریان معرفی می‌شوند.

توپولوژی که در شکل ۲-۱۸ نشان داده شده است شامل یک طبقه مبدل فلاپیک است که ولتاژ ماژول را در هر لحظه به ولتاژ شبکه تبدیل می‌کند. بدین ترتیب یک جریان مستقیم به فرم یک جریان سینوسی یکسو شده به وجود می‌آید که به وسیله طبقه باز کننده تای جریان، تبدیل به جریان متناوب می‌گردد و به شبکه تزریق می‌شود. این توپولوژی می‌تواند در حالت عملکرد^۱ DCM و یا^۲ BCM کار کند. مراجع متعددی به بررسی این ساختار پرداخته‌اند از جمله [۱۸]، [۱۹]. مزیت اصلی این ساختار در کم بودن تعداد اجزای سیستم است که منجر به افزایش قابلیت اعتماد سیستم و نیز کاهش قیمت می‌شود. علاوه بر این مزایا، ایزولاسیون الکتریکی نیز تامین می‌گردد. ایراد عمده این توپولوژی در این است که به واسطه کلیدزنی سخت، تلفات افزایش می‌یابد و نیز هارمونیک مرتبه دو در جریان منجر به این می‌شود که فرایند^۳ MPPT دشوارتر باشد [۲۰]. خازن های بزرگی که در دکوپله سازی توان به کار می‌رود می‌تواند منجر به کاهش قابلیت اعتماد سیستم منجر شود. بیشینه راندمانی که از این توپولوژی در نمونه ۲۰۰ واتی آن به دست آمده حدود ۹۶ درصد بوده است.

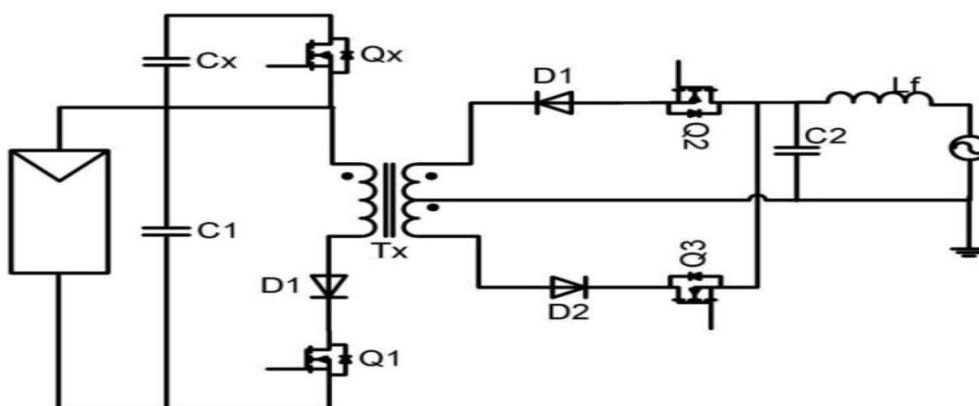
¹ Discontinuous Conduction mode

² Boundary Conduction mode

³ Maximum Power Point Tracking

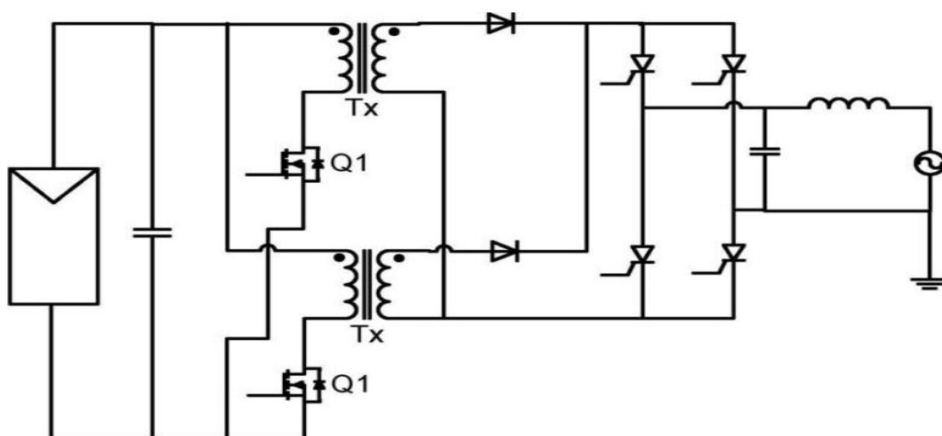


شکل ۲-۱۸: اینورتر فلیبک با مدار بازکننده تای جریان [۱۷]، [۱۸]



شکل ۲-۱۹: اینورتر فلیبک با مدار تزریق کننده جریان هارمونیک دوم [۲۱]

توپولوژی نشان داده شده در شکل ۲-۱۹ نمونه بهبود یافته مدار شکل ۲-۱۸ می باشد. این توپولوژی، مداری را به کار گرفته است که به وسیله آن هارمونیک دوم جریان به ورودی تزریق می شود که منجر به کاهش ریپل جریان ورودی می شود [۲۱]، [۲۲]. کلید اصلی $Q1$ به جهت تامین جریان متوسطی برابر با جریان متوسط توپولوژی قبلی، استفاده می شود. این کار به وسیله زمان وظیفه ثابت انجام می شود. این در حالی است که کلید Qx برای تامین مولفه هارمونیک مرتبه دوم به کار گرفته می شود. این تغییر ساختار منجر به این می شود که ریپل جریان منبع نسبت به توپولوژی قبلی کمتر شود. با این حال، این تغییر ساختار، تاثیر منفی بر قابلیت اعتماد سیستم و بازده آن می گذارد. برای نمونه ۱۰۰ واتنی آن بازده ۷۰ درصد گزارش شده است.

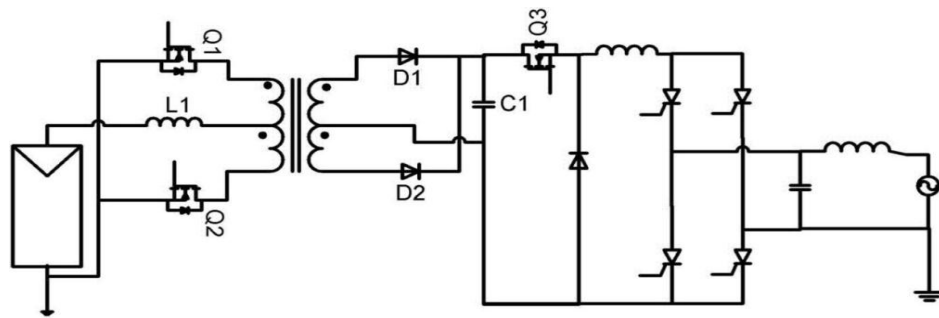


شکل ۲-۲۰: مدار Interleaved flyback به همراه Current unfolding [۲۳]

توپولوژی نشان داده شده در شکل ۲-۲۰ یک فلاپیک اینترلیوید^۱ است که دارای بخش بازکننده تای جریان نیز می باشد. ریپل جریان منبع به واسطه عملیات اینترلیوینگ کاهش یافته است. بنابراین ریپل جریان مربوط به خازن دکوپله سازی کاهش یافته است. این تمهید می تواند منجر به افزایش قابلیت اعتماد سیستم گردد. برای نمونه ۱۹۵ واتی آن که در فرکانس ۱۷۲ کیلوهرتز کلیدزنی می شود بازده ماکزیمم ۹۴ درصد گزارش شده است.

در مرجع [۲۴] یک اینورتر سه طبقه (که دارای طبقه بازکننده تای جریان نیز می باشد) عنوان شده که در شکل ۲-۲۱ نمایش داده شده است. طبقه اول این توپولوژی یک منبع جریان مبدل پوش پول است که ولتاژ را افزایش می دهد. در طبقه بعد به وسیله یک مبدل باک، جریان سینوسی که به طور کامل یکسو شده است به طبقه بعد تزریق می شود. مدار بازکننده تای جریان نیز جریان را در فرکانس شبکه به شبکه تزریق می کند. نمونه آزمایشی که برای این ساختار ساخته شده است به همراه مدار کنترلی آن دارای بازده ۸۰ درصدی می باشد.

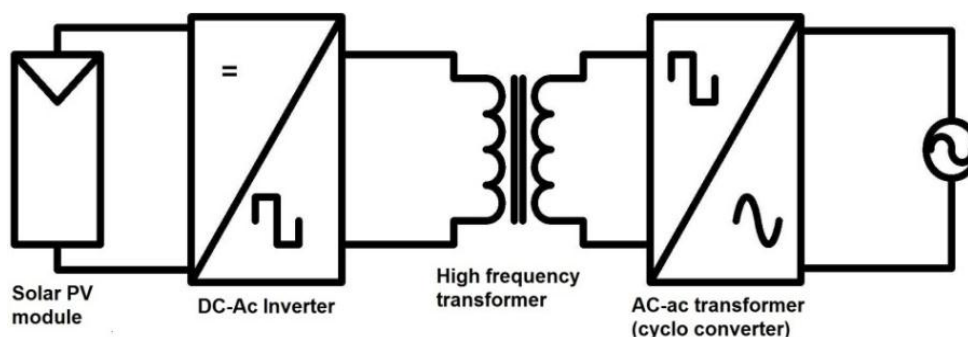
^۱ Interleaved flyback



شکل ۲-۲۱: اینورتر دارای بازکننده تای جریان [۲۴]

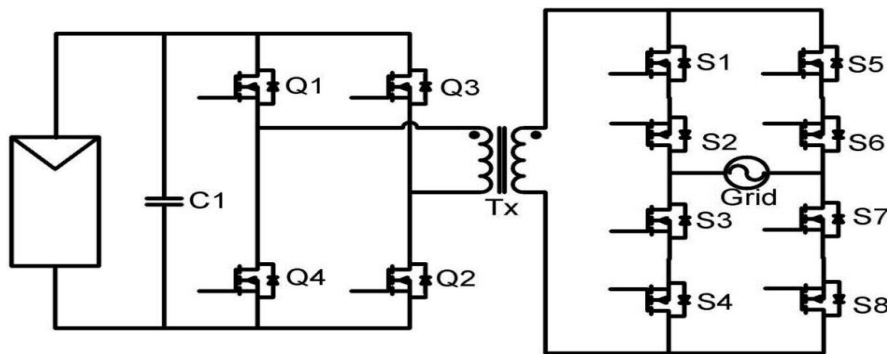
۳-۳-۲ اینورترهای بدون لینک مستقیم

این توپولوژی ها حداقل شامل دو طبقه توان هستند. طبقه اول به وسیله یک ترانسفورماتور فرکانس بالا ولتاژ ورودی را به یک ولتاژ متناوب در فرکانس بالا تبدیل می کند. در طبقه دوم یک سیکلوکانورتر، ولتاژ متناوب فرکانس بالا را به جریان سینوسی در فرکانس شبکه تبدیل می کند و آن را به شبکه تزریق می کند. ساختار پایه اینورترهای بدون لینک مستقیم در شکل ۲-۲۲ نمایش داده شده است. در این قسمت ۴ توپولوژی از این خانواده معرفی می شود.



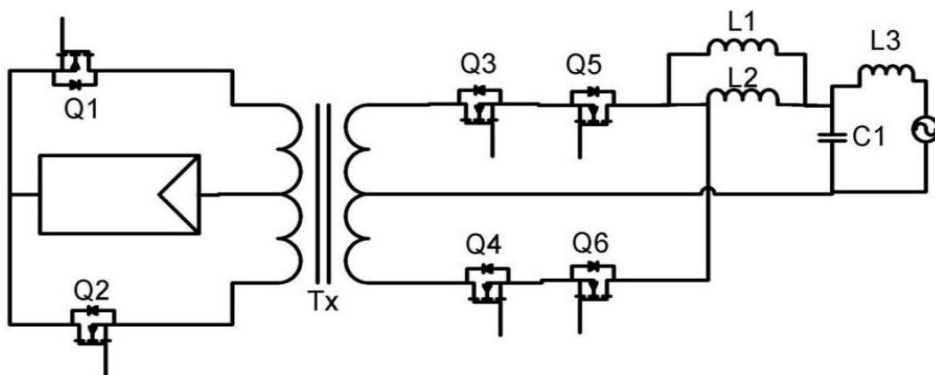
شکل ۲-۲۲: ساختار پایه اینورترهای بدون لینک مستقیم [۱۱]

در مرجع [۲۵] توپولوژی که بیان شده است شامل یک مبدل مستقیم به متناوب است که در فرکانس ۲۰ کیلوهرتز به صورت کلیدزنی سخت، کلیدزنی می شود و ولتاژ ورودی را به ولتاژی که متناسب با ولتاژ شبکه باشد تبدیل می کند. در طبقه بعد یک سیکلوکانورتر، ولتاژ متناوب فرکانس بالا را به صورت جریان سینوسی در فرکانس شبکه تبدیل می کند و آن را به شبکه تزریق می کند. این توپولوژی در شکل ۲-۲۳ نشان داده شده است. نمونه آزمایشی برای توان یک کیلووات ساخته شده است ولی بازدهی برای آن ذکر نشده است.



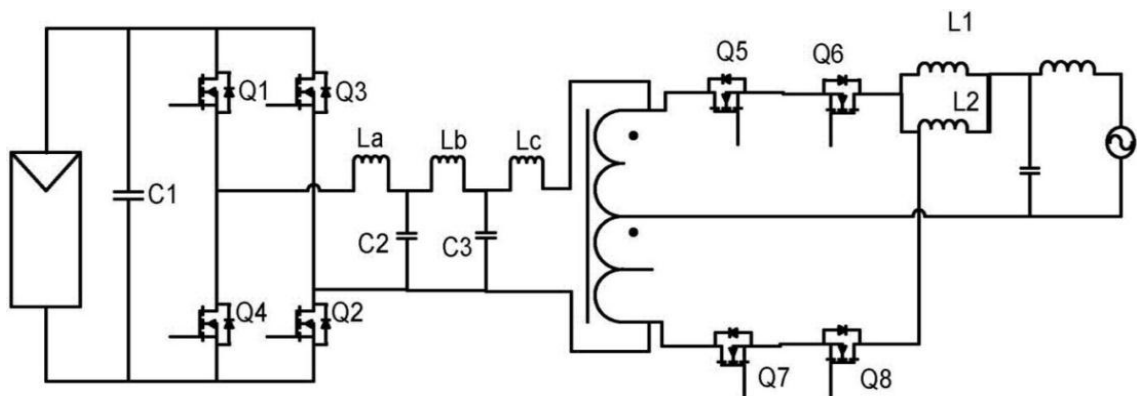
شکل ۲-۲۳: توپولوژی بدون لینک مستقیم با سیکلوکانورتر ولتاژ متناوب فرکانس بالا [۲۵]

توپولوژی که در منبع [۲۶] به آن اشاره شده در شکل ۲-۲۴ نمایش داده شده است. این مدار یک بخش مبدل پوش پول دارد که در فرکانس ۴۰ کیلوهرتز کلیدزنی می‌شود و خروجی آن به کمک یک ترانسفورماتور ولتاژ، به ولتاژ مورد نیاز برای شبکه تبدیل می‌شود. طبقه خروجی، یک سیکلوکانورتر هدایت اجباری است. برای این ساختار نمونه ۳۰۰ واتی آن ساخته شده است.



شکل ۲-۲۴: توپولوژی بدون لینک مستقیم با مبدل پوش پول [۲۶]

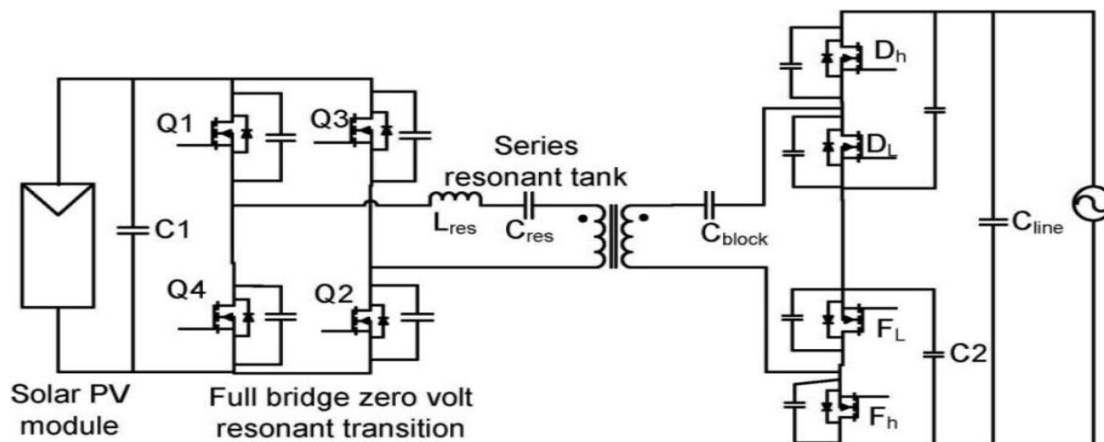
سومین توپولوژی که از این خانواده بررسی می‌شود در منبع [۲۷] معرفی شده است. این ساختار شامل سه طبقه اصلی است: اینورتر تمام پل منبع ولتاژی، یک مدار مبدل امپدانس به ادمیتانس که شامل عناصر LC است و یک سیکلوکانورتر که شامل کلیدهای دوجهته و یک ترانسفورماتور نقطه وسط است. مدار مبدل امپدانس به ادمیتانس منبع ولتاژ را به منبع جریان تبدیل می‌کند. در نمونه آزمایشی که در توان ۳۰ وات تهیه شده است فرکانس کلیدزنی ۲۰ کیلوهرتز انتخاب شده است. این توپولوژی در شکل ۲-۲۵ نمایش داده شده است.



شکل ۲-۲۵: توپولوژی بدون لینک مستقیم سه طبقه [۲۷]

یک توپولوژی نوین برای اینورترهای بدون لینک مستقیم در مرجع [۲۸] آورده شده است. مبدل از دو طبقه اصلی ساخته شده که در شکل ۲-۲۶ نمایش داده شده است. طبقه اول شامل یک مبدل تمام پل رزونانسی ولتاژ صفر است. این طبقه یک جریان شبه سینوسی تولید می‌کند. یک ترانسفورماتور فرکانس بالا ولتاژ را به ولتاژ متناسب با شبکه تبدیل می‌کند. طبقه دوم یک سیکلوکانورتر نیم موج است که جریان را به شبکه تزریق می‌کند. نقش خازن C_{block} در ثانویه ترانسفورماتور، فیلتر کردن جریان مستقیم است. به عبارت دقیق‌تر از جریان یافتن جریان مستقیم در سیم پیچ‌های ترانسفورماتور که می‌تواند منجر به اتصال کوتاه گردد، جلوگیری می‌کند. از مزایای این سیکلوکانورتر آن است که در هر لحظه از زمان کلیدزنی، تنها ۱.۵ قسمت از تلفات، مربوط به تلفات حالت روشن است که نسبت به سیکلوکانورترهای متداول پشت به پشت که این مقدار ۲ است بهبود یافته است. همچنین در هر لحظه از زمان تنها یکی از دو نیم پل کار می‌کند که منجر به کاهش تلفات کلیدزنی شده است. از آنجایی که توپولوژی، دارای تعداد عناصر تشکیل دهنده کمی می‌باشد ساختار به صورت فشرده و کوچک خواهد بود.

نمونه آزمایشی که از این ساختار تهیه شده است دارای توان ۱۷۵ وات می‌باشد که بازده آن ۹۵.۹ درصد گزارش شده است.



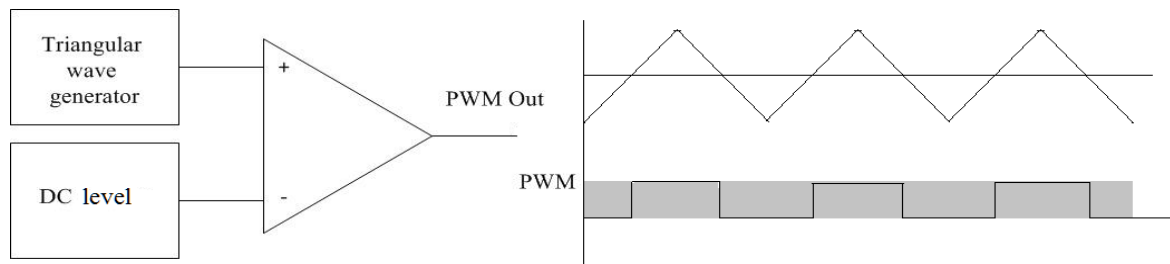
شکل ۲-۲۶: اینورتر بدون لینک مستقیم دو طبقه [۲۸]

۲-۴ مدولاسیون عرض پالس^۱

در مدولاسیون عرض پالس یک ولتاژ مستقیم ورودی به اینورتر داده شده و ولتاژ متناوب خروجی با تنظیم بازه‌های روشن و خاموش عناصر کلید زنی بدست می‌آید. بازه‌های روشن و خاموش اینورتر با سیگنال‌های متفاوت مدولاسیون پهنای پالس کنترل می‌شوند. سیگنال‌های PWM پالس‌هایی با فرکانس ثابت و دامنه و عرض پالس متغیر می‌باشند. نمونه ساده‌ای از این مدولاسیون کلید زنی بین لامپ و یک باتری است. جهت کنترل میزان توان انتقالی به لامپ می‌توان میزان سوئیچ زنی کلید را تغییر داد [۹].

در این روش از مقایسه دو سیگنال آنالوگ، سیگنال مدوله شده مطابق شکل ۲-۲۷ به دست می‌آید. در این حالت یک موج مثلثی با یک سطح ولتاژ مستقیم مقایسه شده و در خروجی مقایسه کننده، سیگنال PWM به دست آمده است. در این روش نباید سطح سیگنال مستقیم از دامنه موج مثلثی (موج حامل) بیشتر شود، در غیر این صورت یک PWM اشباع شده بوجود می‌آید. جهت تولید سیگنال مثلثی می‌توان از تراشه‌های آماده‌ای استفاده کرد که با اضافه کردن چند المان پسیو سیگنال‌های مربعی و سینوسی را در خروجی تحویل می‌دهند [۲۹]. در خصوص تراشه‌های آماده جهت پیاده‌سازی سخت افزاری در فصل بعد بیشتر توضیح داده خواهد شد.

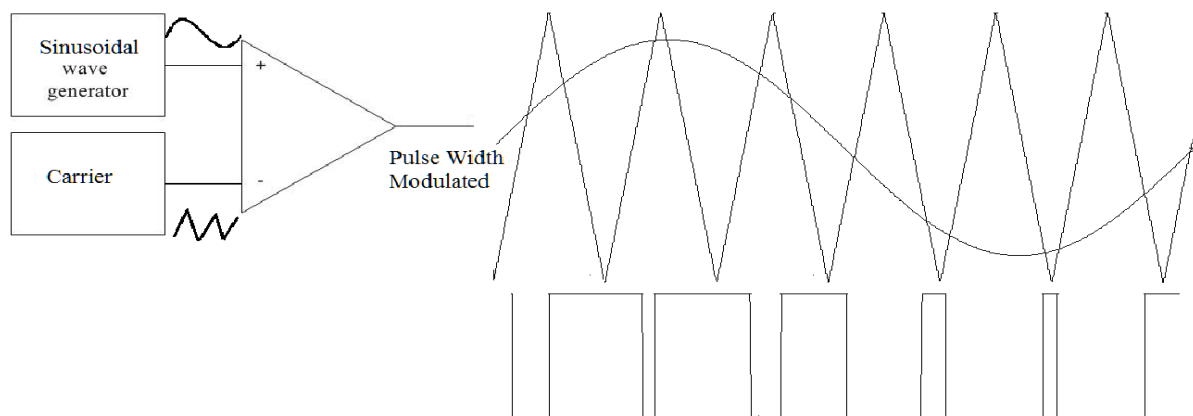
^۱ Pulse width modulation (PWM)



شکل ۲-۲۷: مقایسه دو سیگنال آنالوگ [۲۹]

۲-۵ مدولاسیون عرض پالس سینوسی

این مدولاسیون بر اساس مقایسه یک سیگنال سینوسی مرجع و یک سیگنال مثلثی استوار است. شکل ۲-۲۸ نمونه‌ای از این مقایسه و سیگنال سینوسی مدوله شده را نشان می‌دهد [۲۹].

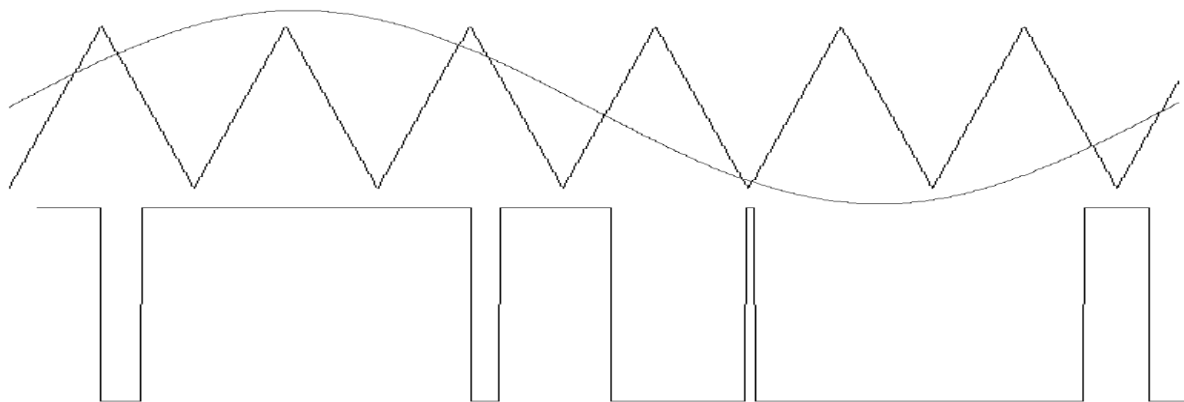


شکل ۲-۲۸: سیگنال مدولاسیون عرض پالس سینوسی و مدار مقایسه کننده مدولاسیون [۲۹]

هر نوع از مدولاسیون‌ها دارای پارامترهای مهمی است که می‌بایست مد نظر قرار گیرند. اولین پارامتر میزان مدولاسیون دامنه است که با m_a نشان داده می‌شود که از رابطه (۳-۱) بدست می‌آید.

$$m_a = \text{دامنه موج مثلثی} / \text{دامنه موج سینوسی} \quad (۳-۱)$$

دامنه تغییرات m_a به صورت $0 < m_a < 1$ می‌باشد. در صورتی که m_a بیشتر از یک شده یک مدولاسیون اشباع شده خواهیم داشت که در شکل ۲-۲۹ یک نمونه آن نشان داده شده است [۳۰].



شکل ۲-۲۹: نمونه ای از یک مدولاسیون اشباع شده [۳۰]

پارامتر دیگری که در این مدولاسیون مطرح است، میزان مدولاسیون فرکانس است که از معادله (۲-۶) بدست می‌آید [۳۰]. در این رابطه m_f معمولاً از یک بزرگتر است.

$$m_f = \text{فرکانسی موج سینوسی / فرکانس موج مثلثی} \quad (۲-۶)$$

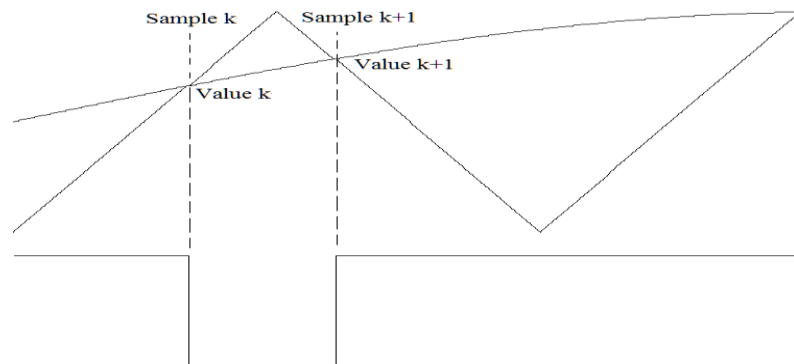
این مدولاسیون از نظر نحوه نمونه برداری برای مقایسه دو سیگنال سینوسی و مثلثی به دو دسته تقسیم می‌شوند که عبارتند از:

الف: مدولاسیون پهنای پالس سینوسی با نمونه برداری طبیعی

این همان رویه عادی ایجاد سیگنال‌های مدولاسیون عرض پالس سینوسی است که سیگنال‌های تغییر وضعیت از مقایسه لحظه به لحظه سیگنال سینوسی و مثلثی بدست می‌آید.

ب: مدولاسیون عرض پالس با نمونه برداری منظم

اگرچه روش نمونه برداری قبل برای بیشتر مدارهای آنالوگ مناسب است، اما این روش تولید سیگنال مدوله شده سینوسی نسبت به نویز حساس بوده و با کوچکترین عامل خارجی کیفیت را از دست می‌دهد. به همین جهت از روش دیگری به نام منظم استفاده می‌کنیم که در آن از موج سینوسی در فواصل پررودیک از سیگنال مثلثی نمونه برداری شده و این نمونه‌ها جهت ایجاد مدولاسیون عرض پالس سینوسی با سیگنال مثلثی مقایسه می‌شوند. شکل ۲-۳۰ نمونه‌ای از این نوع مدولاسیون را نشان می‌دهد [۳۰].

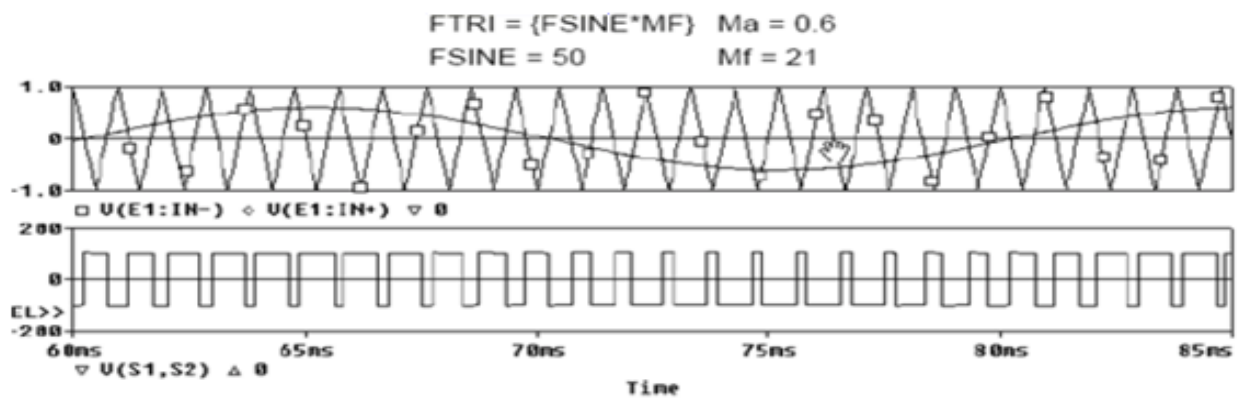


شکل ۲-۳۰: مدولاسیون با نمونه برداری منظم [۳۰]

مدولاسیون پهنای پالس سینوسی از نظر تعداد قطب‌های قابل تولید به دو دسته تقسیم می‌شوند.

الف: مدولاسیون سینوسی تک قطبی

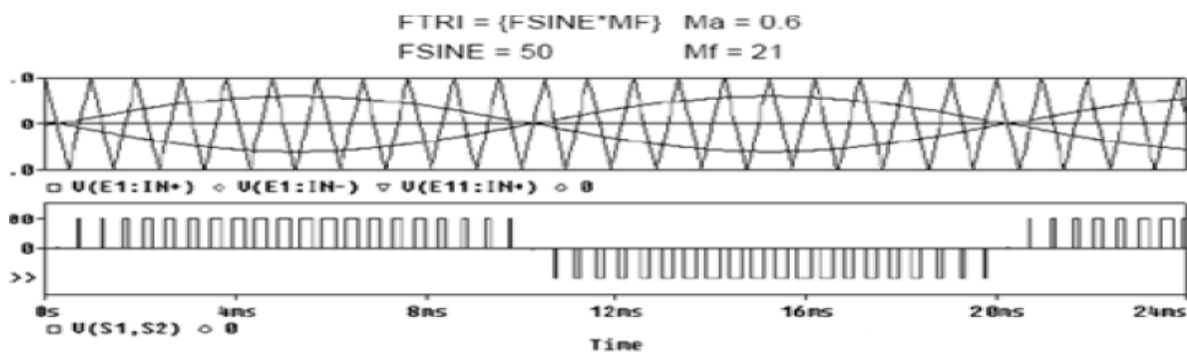
این روش از مقایسه یک شکل موج سینوسی و مثلثی حاصل می‌شود. شکل ۲-۳۱ نمونه‌ای از آن را نشان می‌دهد.



شکل ۲-۳۱: مدولاسیون تک قطبی [۳۰]

ب: مدولاسیون سینوسی دوقطبی

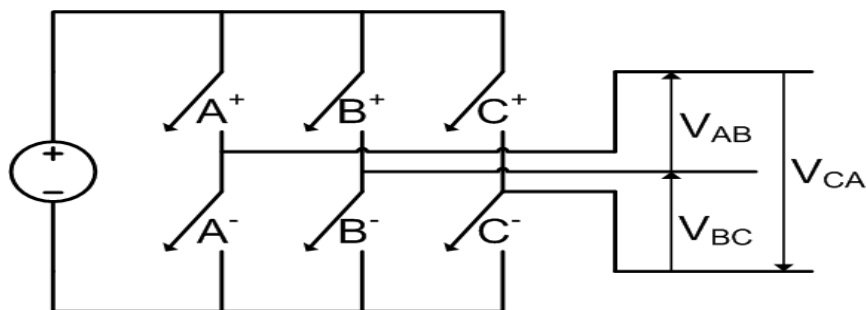
در این روش دو سیگنال سینوسی با اختلاف فاز 180° درجه با یک موج مثلثی مقایسه می‌شود که از نظر قطبیت مخالف یکدیگر هستند. شکل ۲-۳۲ نحوه تولید و سیگنال مدوله شده سینوسی حاصل از مدولاسیون دوقطبی را نشان می‌دهد [۳۰].



شکل ۲-۳۲: نحوه تولید و سیگنال مدوله شده سینوسی حاصل از مدولاسیون دو قطبی [۳۰]

۲-۶ مدولاسیون بردار فضایی^۱

مدولاسیون بردار فضایی الگوریتمی است برای کنترل مدولاسیون عرض پالس که برای تولید پالس متناوب استفاده می‌شود. یک حوزه توسعه در این روش در بکار بردن آن جهت کم کردن اعوجاج هارمونیک کل می‌باشد [۳]. همانطور که در شکل ۲-۳۳ قابل مشاهده است، در یک اینورتر سه فاز که تغذیه مستقیم را به یک موتور سه فاز وصل می‌کند، هرگز نباید سوئیچ‌های مربوط به یک بازو همزمان با هم روشن شوند چرا که موجب اتصال کوتاه شدن تغذیه می‌شود. بنابراین در هر وضعیت سوئیچ‌های مربوط به یک بازو بایستی به طور مکمل باهم روشن و خاموش باشند. در نتیجه هشت حالت مختلف برای کلیدزنی^۲ بوجود می‌آید که در جدول ۲-۲ نشان داده شده است.



شکل ۲-۳۳: نمای ساده اینورتر سه فاز در مدولاسیون بردار فضایی [۱۰]

^۱Space Vector Modulation

^۲Switching

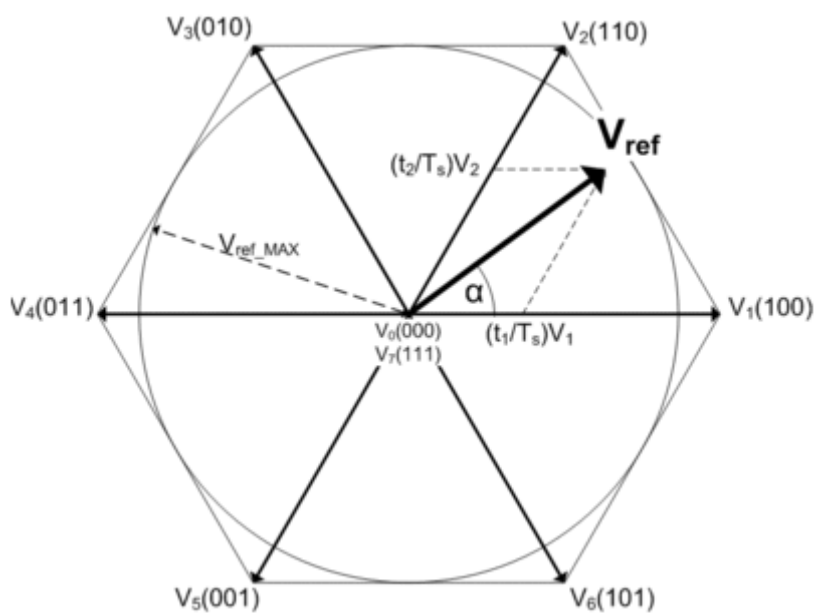
جدول ۲-۲ حالت‌های مختلف کلیدزنی در مدولاسیون بردار فضایی [۱۰]

Vector	A ⁺	B ⁺	C ⁺	A ⁻	B ⁻	C ⁻	V _{AB}	V _{BC}	V _{CA}	
V ₀ = {000}	OFF	OFF	OFF	ON	ON	ON	0	0	0	zero vector
V ₁ = {100}	ON	OFF	OFF	OFF	ON	ON	+V _{dc}	0	-V _{dc}	active vector
V ₂ = {110}	ON	ON	OFF	OFF	OFF	ON	0	+V _{dc}	-V _{dc}	active vector
V ₃ = {010}	OFF	ON	OFF	ON	OFF	ON	-V _{dc}	+V _{dc}	0	active vector
V ₄ = {011}	OFF	ON	ON	ON	OFF	OFF	-V _{dc}	0	+V _{dc}	active vector
V ₅ = {001}	OFF	OFF	ON	ON	ON	OFF	0	-V _{dc}	+V _{dc}	active vector
V ₆ = {101}	ON	OFF	ON	OFF	ON	OFF	+V _{dc}	-V _{dc}	0	active vector
V ₇ = {111}	ON	ON	ON	OFF	OFF	OFF	0	0	0	zero vector

بنابراین ولتاژ خروجی به صورت موج سینوسی بوجود می‌آید. جهت پیاده‌سازی مدولاسیون بردار فضا به هر کدام از حالات کلیدزنی یک بردار در صفحه مختصات اختصاص داده می‌شود. یک سیگنال ولتاژ مرجع با فرکانس مرجع f_s به عنوان نمونه در نظر گرفته می‌شود. سیگنال مرجع از ترکیب دو بردار سوئیچ مجاور و یک یا دو بردار صفر ساخته می‌شود. بردارهای سوئیچ ثابت هستند اما بردار مرجع با دامنه و فرکانسی که دامنه و فرکانس مطلوب مورد نظر در خروجی است، حول مرکز دایره همانند شکل ۲-۳۳ می‌چرخد. تدابیر متعدد برای انتخاب بردارها وجود دارد و استفاده می‌شود. این تعدد روش‌ها موجب بدست آوردن اعوجاج هارمونیک^۱ و تلفات کلیدزنی مختلف می‌گردد. می‌توان به بازده بالاتر و پاسخ هارمونیک^۲ نسبت به سایر روش‌ها رسید [۳۱].

¹ THD

² Harmonic Response



شکل ۲-۳۴: مدولاسیون بردار فضایی [۳۱].

۷-۲ کنترل کننده تناسبی - انتگرالی و روش تجربی تنظیم ضرائب

متداول ترین کنترل کننده مورد استفاده در صنعت کنترل کننده تناسبی - انتگرالی است که در بسیاری از فرآیندهای کنترلی نظیر کنترل سرعت موتور، کنترل دما، کنترل فشار و ... کاربرد دارد. این کنترل کننده مقدار خطا بین خروجی فرآیند و مقدار مطلوب خروجی محاسبه می کند. هدف کنترل کننده، به حداقل رساندن خطا با تنظیم ورودی‌های کنترل فرآیند است. کنترل کننده تناسبی - انتگرالی از یک بخش تناسبی و یک بخش انتگرال گیر تشکیل شده است که تابع تبدیل آن به صورت رابطه (۷-۲) تعریف می شود.

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} \right) \quad (7-2)$$

در این رابطه K_p بهره بخش تناسبی و T_i ثابت زمانی انتگرال گیر است. روش‌های مختلفی به منظور تنظیم ضرائب تناسبی و انتگرالی این کنترلر وجود دارد که در اینجا به رایج‌ترین آن‌ها در کنترل صنعتی اشاره می شود.

۲-۷-۱ تنظیم ضرائب کنترل کننده تناسبی - انتگرالی به روش پاسخ فرآیند

روش تنظیم تجربی ضرائب PI با استفاده از پاسخ فرآیند به شرح زیر است.

- صفر کردن ضریب انتگرالی و افزایش ضریب تناسبی تا نوسانی شدن خروجی
- ضریب تناسبی مناسب برابر نصف ضریب تناسبی است که به ازاء آن خروجی نوسانی می شود (K_p) .
- ضریب تناسبی برابر K_p تنظیم شده و ضریب انتگرالی به تدریج افزایش می یابد تا خروجی نوسانی شود.
- ضریب انتگرالی مناسب برابر $\frac{1}{3}$ ضریبی است که به ازاء آن خروجی نوسانی می شود $(= K_i)$
 $(\frac{K_p}{T_i})$.
- ضریب انتگرالی برابر K_i تنظیم می شود [۳۲-۳۳].

۲-۷-۲ تنظیم ضرائب کنترل کننده تناسبی - انتگرالی به روش زیگلر نیکلز

مرتبه یک حلقه باز

در این روش پس از تثبیت خروجی، سیگنال کنترلی به صورت پله واحد تغییر می کند و پاسخ خروجی به این پله مورد بررسی قرار می گیرد. ثابت زمانی، تاخیر و بهره پاسخ پله حلقه باز سیستم محاسبه شده و پاسخ سیستم به صورت تابع تبدیل مرتبه یک رابطه (۸-۲) مدل می شود.

$$G = \frac{K}{\tau s + 1} e^{-\tau_d s} \quad (8-2)$$

در این رابطه K بهره پاسخ پله، τ ثابت زمانی و τ_d تاخیر است. با معلوم بودن تابع انتقال رابطه (۸-۲) ضرائب کنترل کننده به صورت رابطه (۹-۲) محاسبه می شوند.

$$\alpha = \frac{\tau_d}{\tau}, \quad K_p = \frac{0.9}{K\alpha}, \quad T_i = 3.3\tau_d \quad (9-2)$$

علاوه بر روش‌های ذکر شده از روش‌های دیگری مانند طراحی حوزه زمان و فرکانس، P&O^۱ و... نیز برای طراحی کنترل‌کننده PI استفاده می‌شود که با توجه به مشخص نبودن یا پرهزینه بودن شناسایی سیستم‌های صنعتی غالباً روش‌های تحلیلی مانند طراحی حوزه زمان و حوزه فرکانس برای آن‌ها مورد استفاده قرار نمی‌گیرد. ضرائب محاسبه شده به این روش‌ها غالباً بهینه نیست و با تغییرات جزئی در آن‌ها می‌توان به بهره‌های بهینه دست یافت. یکی از روش‌های دستیابی به بهره بهینه در کنترل PI استفاده از الگوریتم ژنتیک است [۳۳].

۲-۷-۳ تنظیم ضرائب کنترل‌کننده تناسبی - انتگرالی با سایر روش‌ها

تنظیم ضرایب کنترل‌کننده می‌تواند توسط شبکه‌های عصبی، شبکه‌های فازی یا شبکه‌های فازی-عصبی صورت پذیرد. کاربرد و تاریخچه این شبکه‌ها در پیوست‌ها شرح داده شده است. استفاده از این ابزارها برای تنظیم ضرایب کنترل‌کننده‌ای که وظیفه‌ی کنترل سرعت موتور در این پایان‌نامه را بر عهده دارد، به بهبود پاسخ خروجی کمک می‌کند. در این خصوص در فصل چهارم بیشتر توضیح داده می‌شود.

¹ Perturb and Observation

فصل سوم

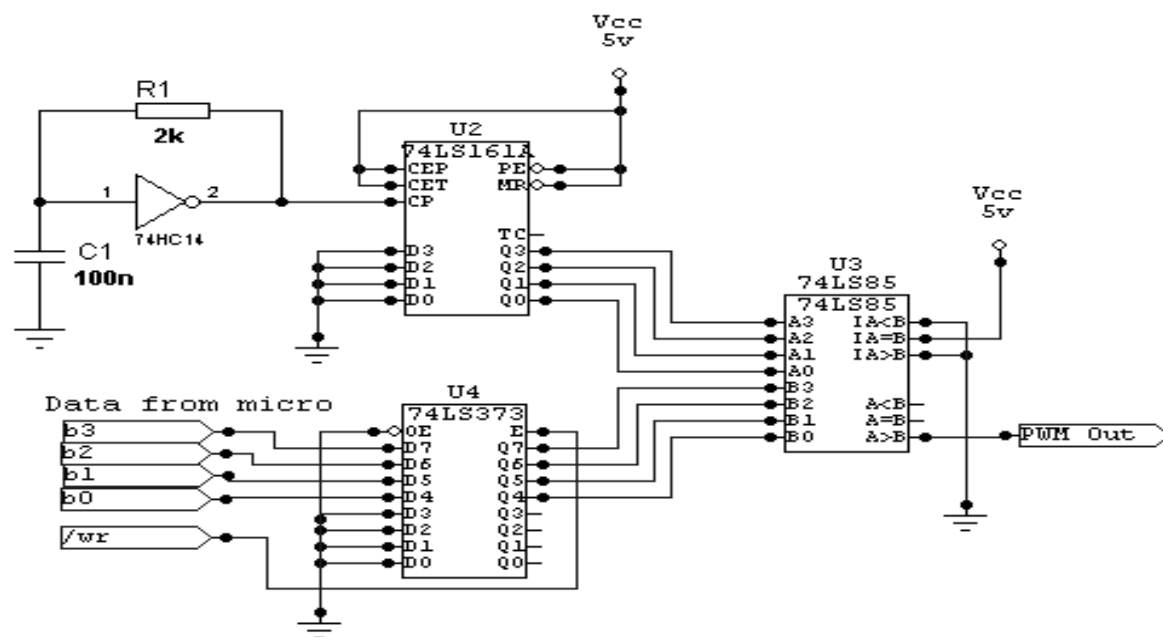
پیاده‌سازی سخت افزاری مدار

اینورتر سه‌فاز

در این فصل در خصوص روش‌های پیاده‌سازی سخت افزاری مدار طراحی ساخت پالس ورودی اینورتر صحبت خواهد شد. در پیاده‌سازی سخت افزاری سیگنال مدوله شده پهنای پالس را می‌توان علاوه بر روش آنالوگ، به صورت دیجیتال، با استفاده از مدارهای مجتمع و یا با استفاده از میکروکنترلرها می‌توان تولید نمود.

۳-۱ روش دیجیتال

در این روش یک شمارنده مدام در حالت شمارش افزایشی (از مقدار صفر تا FF هگزا دسیمال و دوباره بازگشت به مقدار صفر) بوده که مقدار آن با مقدار موجود در یک ثبات که قبلاً ذخیره شده و یا با حاصل تبدیل یک مبدل آنالوگ به دیجیتال مقایسه می‌شود. حاصل آن پالس‌هایی است که عرض آن‌ها متناسب با داده موجود در ثبات یا داده حاصل از این مبدل است. این روش در شکل ۳-۱ نشان داده شده است. عمل شمارش توسط IC74LS161 انجام می‌شود که کلاک آن توسط نوسان‌ساز ساخته شده با 74HC14 تولید می‌شود [۳۱].

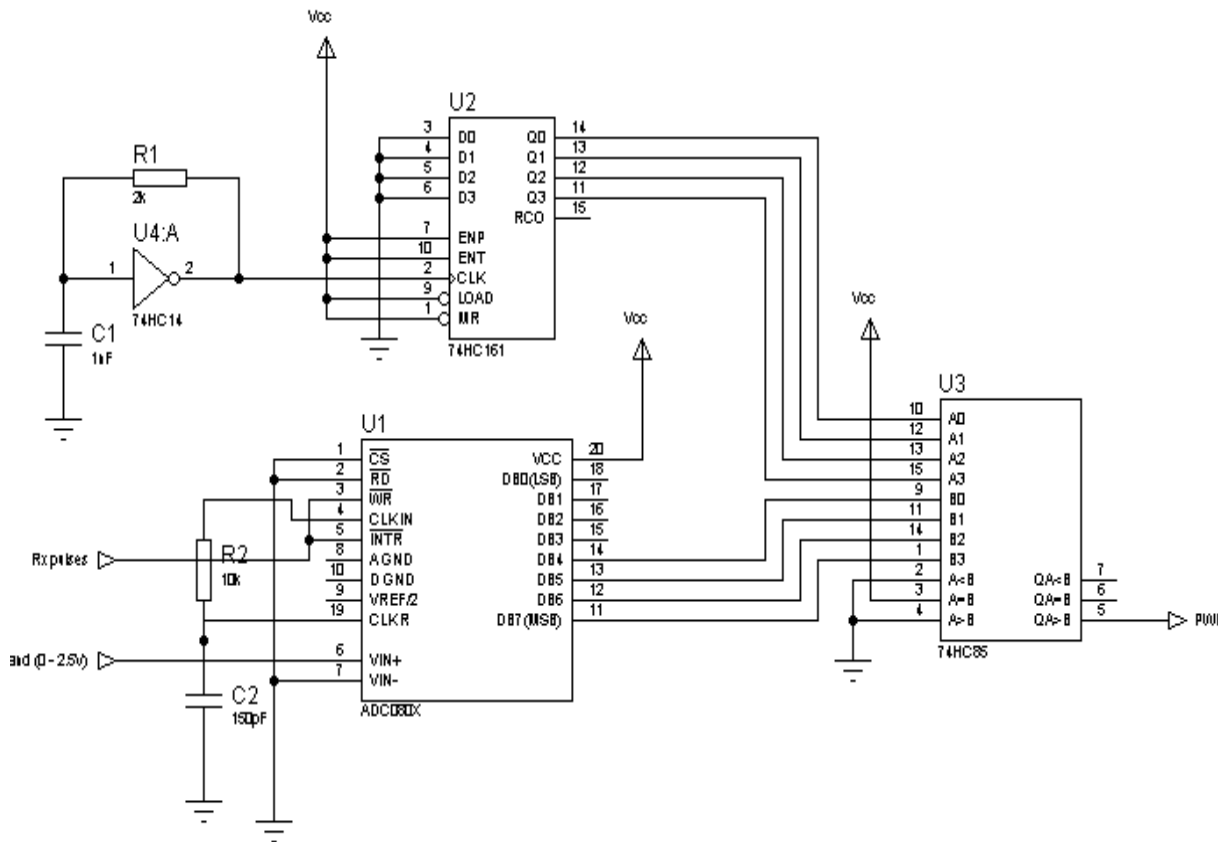


شکل ۳-۱: مقایسه دو سیگنال آنالوگ با استفاده از ثبات [۳۱]

فرکانس نوسان ساز این اسیلاتور از رابطه (۱-۳) دست می آید.

$$F = \frac{1}{6.3 \times R1 \times C1} \quad (1-3)$$

داده چهار بیتی برای ثابت 74LS373 از طریق میکروکنترلر تامین می شود. مقدار فرکانس سیگنال PWM حاصل از این روش ۱۶ برابر کمتر از فرکانس کلاک شمارنده است. از این رو برای کامل شدن یک پریود کامل از سیگنال PWM به ۱۶ پالس شمارنده نیاز است. با توجه به شکل ۲-۳، فرکانس شمارنده ۸۰ کیلو هرتز و در نتیجه فرکانس PWM، ۵ کیلو هرتز خواهد شد. استفاده از مبدل آنالوگ به دیجیتال در شکل ۲-۳ نشان داده شده است.



شکل ۲-۳: استفاده از مبدل آنالوگ به دیجیتال [۳۱]

۳-۲ استفاده از تراشه‌های مخصوص

در شکل ۳-۲ از مبدل آنالوگ به دیجیتال هشت بیتی که فقط چهار بیت آن مورد استفاده قرار گرفته، بهره‌برداری شده است. اگرچه مبدل به صورت خودکار پالس دهی شده اما نیاز به یک ورود جهت شروع کردن عمل تبدیل دارد. این عمل ورودی می‌تواند توسط مداری که درخواست تبدیل سیگنال آنالوگ به دیجیتال را داشته باشد، برای فواصل زمانی خاص و مرتب پالس دهی شود.

تراشه‌هایی تولید شده‌اند که می‌توانند یک سطح ولتاژ مستقیم را به یک خروجی PWM تبدیل کنند و عمدتاً برای منابع تغذیه سوئیچینگ به کار می‌روند. در این تراشه‌ها معمولاً چرخه کار^۱ سیگنال PWM خروجی از صفر تا صد تغییر نمی‌کند. این مساله باعث افت بازده و محدودیت توان در کنترل موتورها به روش PWM می‌شود [۳۲].

۳-۲-۱ استفاده از میکروکنترلرها

در میکروکنترلر AT9058535 از تایمر ۱۶ بیتی برای حالت تولید سیگنال PWM می‌توان استفاده کرد. یک تایمر در حالت PWM به عنوان یک شمارنده صعودی و نزولی عمل کرده و از مقدار صفر تا بالاترین مقدار ممکن که بستگی به دقت‌های PWM انتخابی دارد، عمل شمارش را انجام می‌دهد و پس از رسیدن به این مقدار دوباره صفر شده و این حلقه تکرار می‌گردد [۳۲].

میکروکنترلر AVR دارای یک مبدل آنالوگ به دیجیتال بوده که می‌تواند مقادیر آنالوگ را به صورت لحظه‌ای دریافت کرده آن‌ها را تبدیل به داده دیجیتال کرده و با مقایسه این داده‌ها با مقدار موجود در شمارنده عمل تولید پالس‌های PWM را انجام دهد. مبدل این مدل از نوع میکروکنترلر هشت کاناله و ۱۰ بیتی می‌باشد و زمان تبدیل آن در حد ۱۲۰ تا ۵۰۰ میکروثانیه است.

¹Duty Cycle

۳-۳ پردازش سیگنال دیجیتال^۱

پردازش سیگنال دیجیتال یکی از قدرتمندترین تکنولوژی های امروزی است که از کامپیوترها برای پردازش سیگنال مانند تشخیص صدا، فیلتر کردن نویز، فشرده سازی داده، تقویت صدا و ویدئو استفاده می کند. این کاربردها بدون پیشرفت در سخت افزارهای تخصصی پردازنده های سیگنال دیجیتال امکان پذیر نیست.

دنیای پردازنده های سیگنال دیجیتال در طول ۳۰ سال گذشته پیشرفت قابل توجهی داشته است. ایجاد و کاربرد روش های DSP از دهه ۱۹۶۰ میلادی با استفاده از کامپیوترهای پردازنده مرکزی دیجیتالی به منظور محاسبات عددی آغاز شد. این تکنیک در آن زمان چندان مرسوم نبود چرا که کامپیوترهای مناسب فقط در دانشگاه ها و مراکز علمی تحقیقاتی وجود داشتند. ارائه ریزپردازنده ها در اواخر دهه ۱۹۷۰ و اوایل دهه ۱۹۸۰ میلادی این امکان را ایجاد کرد که روش های DSP در گستره وسیعی از کاربردها استفاده شوند. اما پردازنده های همه منظوره مانند خانواده Intel X86 برای اهداف پر قدرت ریاضی در DSP مناسب نیستند. لذا در خلال سال های دهه ۱۹۸۰ اهمیت توسعه این روش، تولیدکنندگان بزرگ قطعات نیمه هادی مانند Texas Instrument، Analog Devices، و Motorola را به سمت طراحی و ساخت تراشه های DSP هدایت کرد. پردازنده های DSP دارای معماری خاصی هستند که برای انواع عملیات لازم در DSP طراحی می شوند. این پردازنده ها قادرند میلیون ها عملیات ممیز شناور را در ثانیه انجام دهند و هر روز در حال توسعه و افزایش قدرت محاسباتی هستند. ابزارآلات پردازش سیگنال دیجیتال در بسیاری از محصولات مصرفی محبوب یافت می شوند. شکی نیست که پردازش سیگنال دیجیتال نقش مهمی در زندگی مدرن ایفا می کند. هدف از پردازش سیگنال به طور معمول آن است که با حفظ محتوای اطلاعاتی، سیگنال به فرمی تبدیل شود که برای انتقال و یا ذخیره سازی مناسب باشد. پردازش سیگنال دیجیتال شامل اعمال الگوریتم بر روی سیگنال هایی است که بصورت دیجیتال نمایش داده می شوند. امروزه حجم عمده پردازش سیگنال بصورت دیجیتالی انجام می شود. به دلیل مزایای فراوان

¹ Digital Signal Processing

روش‌های پردازش دیجیتال که در بالا به پاره‌ای از آن‌ها به اختصار اشاره شد، دامنه کاربرد پردازنده‌های DSP بسیار توسعه یافته است [۳۳].

۳-۳-۱ فرمت داده‌ها در پردازش سیگنال دیجیتال

اغلب موارد برای نمایش داده‌ها در این نوع پردازنده‌ها از روش ممیز ثابت استفاده می‌کنند. اگرچه در کاربردهای علمی استفاده از روش ممیز شناور معمول‌تر است. در روش ممیز ثابت در اعداد باینری نیز مانند اعداد مبنای ۱۰ نقطه ممیز در جای ثابتی می‌ماند. این روش با ممیز شناور تفاوت دارد. در روش ممیز شناور اعداد از دو جزء نما و مانع تشکیل می‌شوند و نقطه ممیز بر اساس ارزش عددی نما تغییر مکان می‌دهد. در روش ممیز شناور دامنه وسیعی از اعداد قابل نمایش هستند و دیگر مشکل سرریز اعداد وجود نخواهد داشت. در کاربردهای DSP نیز توجه به اعتبار اعداد نکته مهمی است. بطور مثال نباید اجازه داد اعتبار حاصل یک عملیات بعثت سرریز از بین برود. ممکن است عجیب بنظر برسد که چرا در پردازنده‌های سیگنال دیجیتال با اینکه اعتبار اعداد از اهمیت خاصی برخوردار است از روش ممیز ثابت برای نمایش اعداد استفاده می‌شود. در اکثر کاربردها پردازنده‌های DSP با محدودیت‌هایی مواجه‌اند که از جمله مهم‌ترین آن‌ها باید به ارزان بودن قیمت این پردازنده‌ها و مصرف توان پایین در آن‌ها باید اشاره کرد. روش ممیز ثابت در مقایسه با ممیز شناور به پردازنده‌های ارزان‌تر و با مصرف توان کمتر منتهی می‌شود که سرعت قابل قبولی را نیز نتیجه می‌دهد. پیاده‌سازی روش ممیز شناور به سخت‌افزار پیچیده‌تری نسبت به روش ممیز ثابت نیاز دارد. به این دلایل است که تعداد کمی از پردازنده‌های DSP از روش ممیز شناور برای نمایش اعداد استفاده می‌کنند. محدودیت قیمت و مصرف توان از جمله نکات مهمی است که بر طول کلمه مورد نظر برای نمایش اعداد نیز تاثیر می‌گذارد. در پردازنده‌های DSP از کمترین طول کلمه‌ای که دقت مورد نظر آن کاربرد خاص را تامین می‌کند استفاده می‌شود. در اکثر پردازنده‌های DSP که از روش ممیز ثابت برای نمایش اعداد استفاده می‌کنند، طول کلمه ۱۶ بیت در نظر گرفته می‌شود که برای اکثر کاربردهای پردازش سیگنال دیجیتال مناسب است. در برخی کاربردهای خاص مانند پردازش سیگنال صوتی که امکان داشتن دقت مورد نظر با ۱۶ بیت وجود ندارد برخی پردازنده‌ها از ۲۰ یا ۲۴ یا حتی ۳۲ بیت برای نمایش اعداد استفاده می‌کنند. برای اینکه کیفیت سیگنال طی پردازش‌هایی که بر روی داده‌های ممیز ثابت صورت می‌گیرد در حد خوبی باقی بماند در اکثر

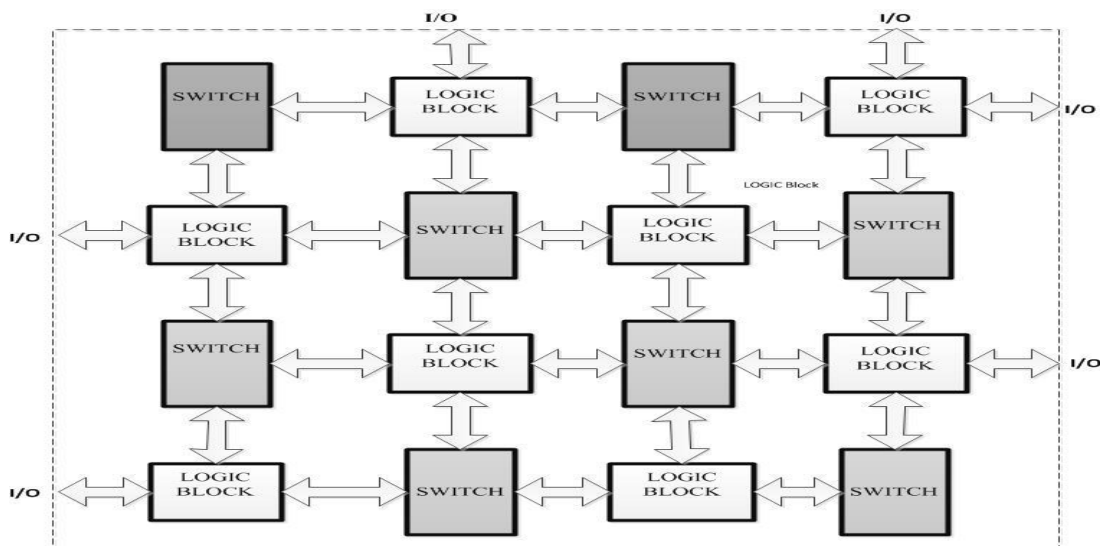
پردازنده‌های سیگنال دیجیتال از امکانات سخت‌افزاری دیگری نیز برای نگهداشتن اعتبار اعداد پس از مجموعه‌ای از محاسبات استفاده می‌شود. بطور مثال در اکثر پردازنده‌ها از یک یا چند ثبات انبارکننده برای نگهداری حاصل جمع چند عملیات ضرب استفاده می‌شود. اکثر ثبات‌های انبارکننده دارای طول بیشتر نسبت به سایر ثبات‌های موجود در پردازنده هستند. آن‌ها اغلب دارای بیت‌های اضافی بنام بیت‌های محافظ هستند که می‌توانند دامنه اعداد قابل نمایش را برای جلوگیری از سرریز گسترش دهند [۳۳ و ۳۴].

۳-۴ آرایه دريچه‌ای برنامه‌پذیر منطقی^۱

FPGA که به عبارتی آرایه دريچه‌ای برنامه‌پذیر منطقی و به تعبیری ساده‌تر آرایه گیت برنامه‌پذیر در محل خوانده می‌شود، تراشه‌هایی هستند با معماری داخلی همچون شکل ۳-۳ از پیش تعیین شده توسط شرکت سازنده که قابلیت پیکربندی برای اهداف مختلف را توسط طراحان فراهم می‌آورد. انواع دیگری از این نوع تراشه نیز موجود است؛ پیش از FPGA ها، مدارهای مجتمع با کاربرد خاص^۲ موجود بود که هنوز هم در برخی پایان نامه‌های بزرگ و پیچیده مورد استفاده قرار می‌گیرد. مهمترین اشکال ASIC فرآیند طراحی و ساخت زمان‌بر و پرهزینه می‌باشد که در نتیجه‌ی ثابت بودن طرح نهایی در سیلیکون ایجاد می‌شود که فقط با تولید نسخه‌ی جدید می‌توان مدار را ویرایش کرد، به همین دلیل قطعه در مدت طولانی‌تری به بازار می‌رسد. در FPGA این مشکلات برطرف شده است [۳۵].

¹ Field Programmable Gate Array (FPGA)

² Application Specific Integrated Circuit (ASIC)



شکل ۳-۳: نمای درونی یک FPGA [۳۶]

این قطعات دارای بلاک‌های منطقی برنامه‌پذیر و اتصالات بین بلاکی قابل پیکربندی هستند. برخی از آن‌ها به تراشه‌ها تنها یک بار قابل برنامه‌ریزی هستند و برخی دیگر چندین بار قابلیت برنامه‌ریزی را دارا هستند [۳۵].

برای برنامه‌نویسی و طراحی FPGA دو روش وجود دارد که شامل زبان‌های توصیف سخت‌افزار^۱ و طراحی مدار است. از نگاه دیگر این تراشه‌ها تحت عنوان آرایه گیت برنامه‌پذیر محلی شناخته می‌شوند که نشان دهنده برنامه‌پذیری درون سیستمی آن‌ها است. همین ویژگی، امکان تغییر طرح پیاده‌سازی شده را به سادگی فراهم می‌آورد؛ بدون آن که نیاز به تولید نسخه جدید باشد. در نتیجه زمان عرضه به بازار بسیار کوتاه‌تر می‌شود [۳۵ و ۳۶].

تراشه‌های FPGA برای پیشرفت گروه‌های مهندسی کوچک مناسب هستند. زیرا به سادگی می‌توان طرح را به سادگی و با هزینه کم بر روی آنها پیاده‌سازی نمود. امروزه از FPGA برای پیاده‌سازی دستگاه‌های مخابراتی، رادیوهای نرم‌افزاری، رادارها، پردازش تصویر و دیگر کاربردهای پردازش سیگنال و قطعات سیستم روی تراشه حاوی عناصر نرم‌افزاری و سخت‌افزاری، مطابق شکل ۳-۴ استفاده می‌شود [۳۵ و ۳۶].

^۱ Verilog



شکل ۳-۴: نمونه‌ای از بکارگیری FPGA در آزمایشگاه

پیشرفت تکنولوژی در ساخت قطعات برنامه پذیر منجر به طراحی در مدارهای مجتمع با مقیاس بزرگ^۱ و استفاده از آن به جای مدارهای مجتمع با مقیاس کوچک^۲ شده است. از ویژگی های VLSI می-توان به تراکم بالا و کاربردهای متنوع اشاره کرد. در یک جمع بندی کلی مزایای طراحی به روش ASIC عبارت است از [۳۵]:

- کاهش هزینه، ابعاد و حجم سیستم
- افزایش قابلیت اطمینان سیستم به دلیل انتقال اکثر مدارات جانبی به داخل تراشه
- کاهش زمان، هزینه مونتاژ راه اندازی و نگهداری
- کاهش مدت زمان طراحی و ساخت و عرضه به بازار
- کاهش توان مصرفی، نویز و اغتشاش.

اولین تراشه قابل برنامه ریزی که به بازار عرضه شد، حافظه‌های فقط خواندنی برنامه پذیر بود. این حافظه‌ها دارای خطوط آدرس به عنوان ورودی و خطوط داده به عنوان خروجی بوده و شامل دسته‌ای از گیت‌های AND غیرقابل برنامه ریزی و آرایه OR قابل برنامه ریزی بودند. این حافظه‌ها فاقد قابلیت‌های لازم برای پیاده‌سازی مدارهای منطقی هستند. بنابراین آرایه‌ای به بازار عرضه شد که دارای یک گیت AND قابل برنامه ریزی و یک گیت OR تثبیت شده است و تحت عنوان PAL شناخته می-شدند [۳۵ و ۳۷].

¹ Very Large Scale Integration (VLSI)

² Small Scale Integration (SSI)

بعد از PAL دسته دیگری از تراشه‌های منطقی برنامه‌پذیر تحت عنوان GAL ارائه شد که دارای گیت-های AND قابل برنامه‌ریزی و متصل به آرایه OR بودند. در GAL به جای فیوز از سلول‌هایی از نوع ترانزیستورهای اثر میدان^۱ که قابل پاک شدن به صورت الکتریکی هستند، استفاده شده است. این تراشه‌ها آرایش‌های متنوعی دارد که هر یک توسط شماره یکتایی مشخص می‌شود [۳۷].

تراشه‌های ماسک شده قابل برنامه‌ریزی که به صورت آرایه‌ای از ترانزیستورهای پیش ساخته هستند و برای پیاده‌سازی مدارهای منطقی، در کارخانه‌های سازنده به یکدیگر متصل می‌شوند به همراه سایر تراشه‌ها نیز شروع به رشد کرد [۳۷].

پیشرفت در ابزارهای طراحی و نیز تراشه‌های قابل برنامه‌ریزی منجر به عرضه FPGA شد که امروزه از نظر تکنولوژی در زمره بزرگترین مدارهای مجتمع موجود در بازار هستند. محصولات با تکنولوژی نیم میکرون، حدود ده میلیون ترانزیستور را در گستره‌ای به ابعاد ۱/۸ سانتی متر در ۱/۵ سانتی متر جای داده‌اند. اگر چه این محصولات ظرفیتی بیش از ۳۰۰۰۰۰ گیت و ۳۰۰ پایه ورودی - خروجی را به کاربر عرضه می‌کند ولی با این همه هنوز از تراشه‌های سابق استفاده فراوانی به عمل می‌آید. در آینده، قطعات FPGA همچنان رشد کرده و با توسعه ابزارهای طراحی قوی‌تر که دستورات پیشرفته‌تری از زبان‌های کنونی^۲ را پشتیبانی کنند، راه برای به کارگیری و گسترش هر چه بیشتر این گونه تراشه‌ها هموار گردد [۳۵].

۳-۴-۱ ساختار کلی FPGA

تراشه FPGA از یک سری عناصر منطقی که برای کار خاصی محدود شده‌اند و دارای اتصالات قابل برنامه‌ریزی هستند، تشکیل شده است. بنابراین هر دو جزء اصلی تشکیل دهنده یک مدار و اتصالات بین آنها، قابل برنامه‌ریزی است. همانطور که در شکل ۳-۳ مشخص است سه جزء اصلی آن عبارتند از بلوک‌های منطقی، عناصری که برای اتصالات به کار می‌روند و بلوک‌های ورودی - خروجی.

^۱ CMOS

^۲ VHSIC Hardware Description Languages (VHDL) & Altera Hardware Description Languages (AHDL)

ساختار و محتویات بلوک‌های منطقی می‌تواند خیلی ساده (در حد گیت NAND) و یا خیلی پیچیده (نظیر چند مالتی پلکسر یا جدول بررسی به همراه یک فلیپ فلاپ) باشند. عناصر متصل کننده معمولاً بین بلوک‌های منطقی قرار می‌گیرند و از قطعات فلزی که می‌توانند به هم یا به بلوک‌های منطقی متصل شوند، تشکیل شده‌اند. اتصال این قطعات با سوئیچ‌های قابل برنامه‌ریزی انجام می‌شود. طراحی بلوک‌های منطقی و عناصر اتصالی باید با هم به گونه‌ای باشد که پیاده‌سازی مدارات منطقی مختلف را روی FPGA ممکن سازد. معمولاً بین پیچیدگی و انعطاف پذیری هر دوی بلوک‌های منطقی و منابع اتصالی یک نسبت معکوس وجود دارد، یعنی با زیاد شدن یکی دیگری کم می‌شود و بر عکس [۳۵].

۳-۴-۲ مقایسه FPGA با MPGA

پیشرفت تکنولوژی CMOS باعث شده است که وضعیت FPGAها در برابر MPGAها بهبود یابد. پس از اینکه ظرفیت FPGA به چند هزار گیت رسید، صحبت از جایگزینی FPGAها جنبه عملی‌تر یافت. به علت تنوع تکنولوژی‌های به کار رفته در ساخت FPGA و نیز وجود ابزارهای مختلف و پیشرفت هر یک از آنها یک تکنولوژی خاص از FPGAها را نمی‌توان به سادگی انتخاب کرد. اگر چه هم اکنون فرکانس کار برای FPGAها نوعاً حدود ۱۲۰ مگاهرتز تا ۸۰۰ مگاهرتز می‌باشد، اما محدوده واقعی کاملاً به نوع طرح پیاده شده بستگی دارد. تراشه‌های FPGA معمولاً به اندازه کافی پین ورودی - خروجی فراهم می‌کنند به طوریکه FPGAهایی با بیش از ۳۰۰ پین نیز وجود دارد.

FPGA در مقایسه با MPGA دارای دو مزیت عمده می‌باشد:

- هزینه پایین در تیراژ کم
- سرعت بالای آماده‌سازی

سه عیب عمده‌ای که FPGAها نسبت به MPGAها دارند عبارتند از:

- سرعت کم به دلیل تاخیر زیاد سوئیچ‌های برنامه‌پذیر در مسیرهای اتصالات
- چگالی پایین‌تر منطقی به دلیل سطح اشغال شده توسط سوئیچ‌ها
- قیمت بالا برای تولید با تیراژ بالا [۳۵].

۳-۴-۳ مراحل پیاده‌سازی یک طرح بر روی FPGA

روش طراحی مدارات منطقی مستقل از تراشه نهایی است که برای پیاده‌سازی انتخاب می‌گردد. در فرآیند طراحی مهمترین بخش، شبیه‌سازی است. از آنجا که طراحی به صورت بخش بخش می‌باشد، کلیه این بخش‌ها را می‌توان جداگانه و با اعمال ورودی و دریافت‌های خروجی‌های مورد نظر تست کرده و در نهایت کل طرح را به طور یک جا تست نمود.

الف) ورودی سیستم

زبان‌های ساده توصیف سخت‌افزار (AHDL و DBEL و ...) و پیرایشگرهای موج (که شکل موج ورودی - خروجی را به عنوان ورودی به سیستم قبول می‌کنند) و زبان‌های توصیف سخت‌افزار در سطوح مختلف (مانند VHDL و Verilog) باید در این بخش تعریف شوند.

ب) ترکیب و ترجمه

در ادامه طراحی می‌توان هر قسمت از طراحی را با قسمت‌های دیگر ترکیب کرد و کل مدار را به صورت معادلات منطقی در آورد که خروجی طبقه کامپایل و ترکیب باشد.

پ) بهینه‌سازی

در این قسمت به وسیله یک سری از الگوریتم‌ها و با توجه به معادلات جبر بولی و الگوریتم‌های بهینه‌سازی و با توجه به ساختار خاص بلوک‌های برنامه‌ریزی در FPGA، معادلات بهینه به دست می‌آید.

ت) نگاشت تکنولوژی

ترجمه یک مدار از گیت‌های پایه به مدارهایی با المان‌های مداری دیگر (مانند سلول‌های منطقی FPGA یا سلول‌های پایه CPLD) را نگاشت تکنولوژی گویند. معمولاً در برنامه نگاشت تکنولوژی یک فاز بهینه‌سازی منطقی با توجه به ساختار بلوک‌های برنامه‌پذیر، برای مینیمم کردن مساحت مصرفی یا بالا بردن سرعت موجود می‌باشد.

ث) جایگزینی و سیم کشی اتصالات داخلی

بعد از انجام عمل نگاشت برنامه جایگزینی برای محل قرار گیری هر بلوک منطقی مدار تصمیم می‌گیرد که هر سلول منطقی در کدام سلول فیزیکی FPGA قرار گیرد. این مساله دارای پیچیدگی زیادی است. نرم‌افزارهای هر تراشه توسط شرکت فروشنده آن تولید می‌شود.

ج) شبیه‌سازی

دو نوع شبیه‌سازی نرم‌افزاری انجام می‌شود:

- شبیه‌سازی منطقی: این شبیه‌ساز از نظر منطقی عملکرد مدار را تست می‌کند. در این عمل، اطلاعاتی از زمان بندی و همچنین عملیات بهینه ساز منطقی در اختیار نیست.
- شبیه‌ساز زمانی: در این مرحله که پیش از مسیریابی و رفع سایر مشکلات صورت می‌گیرد. پارامترهای تاخیر، توان و خازن‌های مدار و... نیز به محیط شبیه‌سازی منتقل شده و تصویر واقعی از عملکرد تراشه را به دست می‌دهد [۳۶].

۳-۴-۴ انواع متفاوت معماری‌های FPGA

چهار نوع مختلف معماری‌های FPGA را که کارخانجات متفاوت استفاده می‌کنند، عبارتند از:

- آرایه دو بعدی متقارن
- سطری
- دریایی از گیت‌ها
- مدارات سلسله مراتبی

نوع اول از یک آرایه دو بعدی از بلوک‌های منطقی که منابع اتصالات به صورت سطری و ستونی بین این بلوک‌ها واقع شده‌اند، تشکیل یافته است. این نوع معماری توسط شرکت Xilinx استفاده می‌شود.

نوع دوم شبیه به نسل‌های اولیه MPGAها می باشد و در حال حاضر، توسط شرکت آکتل^۱ پیاده سازی می‌شود.

نوع سوم متشکل از بلوک‌های کوچک منطقی است که با لایه‌هایی برای برقراری اتصال بین این بلوک‌ها پوشیده شده است که آن هم توسط شرکت آکتل در معماری‌های جدیدش استفاده می‌شود.

نوع چهارم معماری FPGA بر گرفته شده از معماری PLD و CPLDها می‌باشد و توسط شرکت آلترا^۲ در سری ماکس^۳ ساخت این کارخانه استفاده شده است.

۳-۴-۵ معیارهای اساسی انتخاب و استفاده از FPGA

هنگام طراحی، ابتدا بلوک‌های کلی بخش‌های اساسی سیستم را روی کاغذ پیاده می‌کنند. این بلوک‌ها شامل بلوک پردازنده، بلوک‌های حافظه و برخی ورودی - خروجی‌ها و مدارهای واسطه می‌باشند. برخی از این بلوک‌ها را می‌توان با یک تراشه طراحی کرد و برخی قسمت‌ها به تعداد تراشه‌های بیشتری نیاز دارند.

لازم است طراح بتواند میزان گیت مورد نیاز و نیز تعداد پایه‌های ورودی - خروجی و مشخصه مورد نیاز برای طراحی را تخمین بزند. با در نظر گرفتن این سه فاکتور مهم در هر خانواده می‌توان به انتخاب نهایی نزدیک شد. انتخاب نهایی نیز می‌تواند با توجه به فاکتور مهم قیمت صورت گیرد [۳۵].

۳-۴-۶ مقایسه مشخصه‌ها

از نظر مشخصه‌ها هنگامی می‌توان FPGAها را با یکدیگر مقایسه کرد که اندازه گیری مشخصه در خانواده‌های مختلف از راه مشابهی صورت گرفته باشد. از جمله این مشخصه‌ها فرکانس کاری است که به معنی بیشترین فرکانسی است که در آن فلیپ فلاپ‌ها قادر به کار هستند.

¹ Actel

² Altera

³ Max

بنابراین لازم است مشخص شود که برای اندازه‌گیری مشخصه FPGAهای مختلف چه اندازه‌گیری‌هایی باید صورت گیرد. برای انجام یک مقایسه ساده بر اساس داده‌های تولید کنندگان، چهار آزمایش انجام می‌شود. نتایج این آزمایش‌ها نشان می‌دهد که تا چه حد می‌توان بر این تخمین‌ها تکیه کرد. چهار آزمایش مذکور عبارتند از:

- تاخیر ورودی - خروجی یک AND چهار ورودی
- تاخیر ورودی - خروجی یک مقایسه کننده هشت ورودی
- بیشینه فرکانس کلاک مربوط به یک شیفت رجیستر
- بیشینه فرکانس کار یک شمارنده ۱۶ بیتی

پس از آنکه مشخص شد کدام خانواده قادر به تامین مشخصه لازم است، انتخاب تراشه بر اساس تعداد گیت و پین و قیمت صورت می‌گیرد [۳۸].

۳-۴-۷ داده‌های PREP

موسسه PREP، بنیاد مستقلی است که توسط مجموعه تولیدکنندگان تراشه‌های قابل برنامه‌ریزی حمایت می‌شود و با استفاده از یک سری آزمایش‌های مخصوص، مشخصه و ظرفیت تولیدات مختلف را مقایسه می‌کند. پیاده‌سازی مدارهای زیر مبنای این مقایسه‌ها می‌باشد.

- یک مدار مسیر داده (شامل یک مالتی پلکسر ۴ به ۱ و یک رجیستر و - یک شیفت رجیستر)
- یک شمارنده (شامل دو رجیستر و یک مالتی پلکسر ۴ به ۱)
- یک ماشین حالت کوچک (شامل ۸ حالت و ۸ ورودی و خروجی)
- یک ماشین حالت بزرگ (شامل ۱۶ حالت و ۸ ورودی و خروجی)
- یک مدار محاسبه ریاضی (شامل یک ضرب کننده ۴ در ۴ به همراه یک جمع کننده ۸ بیتی و یک رجیستر ۸ بیتی)
- یک اکومولاتور ۱۶ بیتی
- یک شمارنده ۱۶ بیتی با قابلیت بار شدن و فعال کننده هم زمان
- یک شمارنده ۱۶ بیتی با قابلیت بار شدن و امکان آغاز شمارش با مقدار جدید بار شده

- یک دیکدر ادرس ۱۶ بیتی

جزئیات این مدارها را با مراجعه به کتاب داده هر یک از تولید کنندگان تحت حمایت PREP قابل بررسی است [۳۸].

۳-۴-۸ مقایسه FPGA با Microcontroller

ریزکنترلگر^۱ نوعی ریزپردازنده است که دارای حافظه‌ی دسترسی تصادفی^۲ و حافظه‌ی فقط خواندنی^۳، تایمر، پورت‌های ورودی و خروجی و درگاه ترتیبی پورت سریال^۴ در درون خود تراشه می‌باشد و می‌تواند به تنهایی بر روی ابزارهای دیگری کنترل اعمال کند. به عبارت دیگر یک میکرو کنترلر، مدار مجتمع کوچکی است که از یک CPU کوچک و اجزای دیگری نظیر نوسان‌ساز کریستالی، تایمر، درگاه‌های ورودی و خروجی آنالوگ و دیجیتال و حافظه تشکیل شده است.

در واقع یک ریزپردازنده در درون میکروکنترلر قرار داده شده است که ریزکنترلگر با استفاده از آن می‌تواند محاسبات منطقی و حسابی را انجام دهد.

وجود RAM و ROM و پورت‌های ورودی و خروجی در میکروکنترلرها آنها را یک انتخاب ایده‌آل برای کاربردهایی می‌کند که قیمت و فضا در آنها مهم است. در بسیاری از کاربردها مثل کنترل از راه دور تلویزیون نیازی به محاسبات سنگین در حد یک ۴۸۶ یا حتی ۸۰۸۶ وجود ندارد. در این گونه موارد و بسیاری دیگر از کاربردها فضای مورد استفاده توان مصرفی و قیمت هر واحد مهم تر از قدرت محاسبات است. در این گونه موارد اغلب لازم است تعدادی پورت ورودی و خروجی خوانده شده و بیت‌های مشخصی را خاموش یا روشن کند. میکروکنترلرها عموماً برای کاربردهای کوچک طراحی می‌شوند، بنابراین برخلاف ریزپردازنده‌ها در اینجا مهمترین مسائل، سادگی و مصرف کم توان است.

¹ Microcontroller

² Random Access Memory

³ Read Only Memory

⁴ Serial Ports

همانطور که گفته شد یک میکروکنترلر از CPU، تایمر ها ، پورت ها و حافظه های مختلف تشکیل شده است که با پروگرام کردن آن CPU شروع به اجرای خط به خط برنامه نوشته شده می کند. در حالی که در FPGA ، CPU وجود ندارد و با نوشتن کد به یک FPGA در واقع سخت افزار مدار را طراحی می شود.

در یک FPGA بر خلاف میکروکنترلر قابلیت اجرای پردازش های موازی وجود دارد و این امر باعث بالا رفتن سرعت عملیات در یک FPGA می شود. همچنین یک FPGA به دلیل استفاده مستقیم از گیت ها و بلوک های منطقی و عملیاتی دیگر خیلی سریع تر از میکروکنترلر هست. به همین دلیل از FPGA ها برای ساخت سیستم های پرسرعت و حساس که دارای پیچیدگی های خاص هم می باشند استفاده می شود.

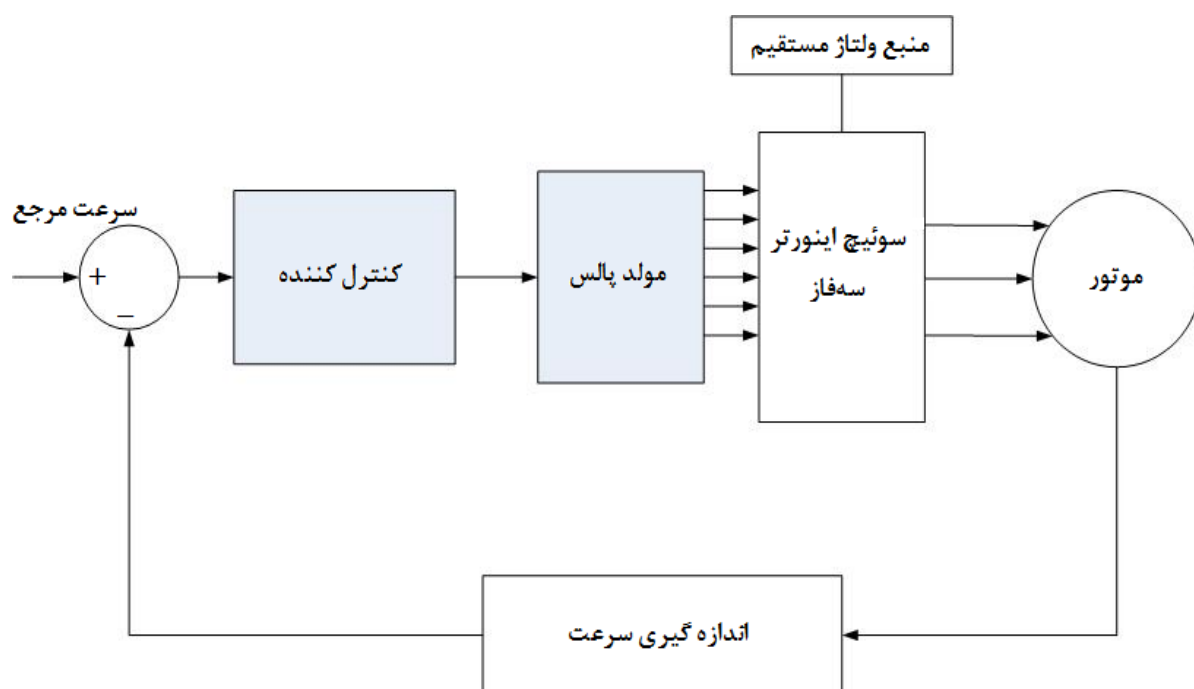
معمولا FPGA و میکروکنترلر هر کدام کاربرد های خاص خود را دارند و به جای یکدیگر استفاده نمی شوند. به دلیل آنکه در این پایان نامه سرعت بالا در رسیدن مدار به پاسخ نهایی مد نظر است، FPGA به میکروکنترلر ارجحیت دارد [۳۹].

فصل چهارم

شبیه‌سازی و نتایج

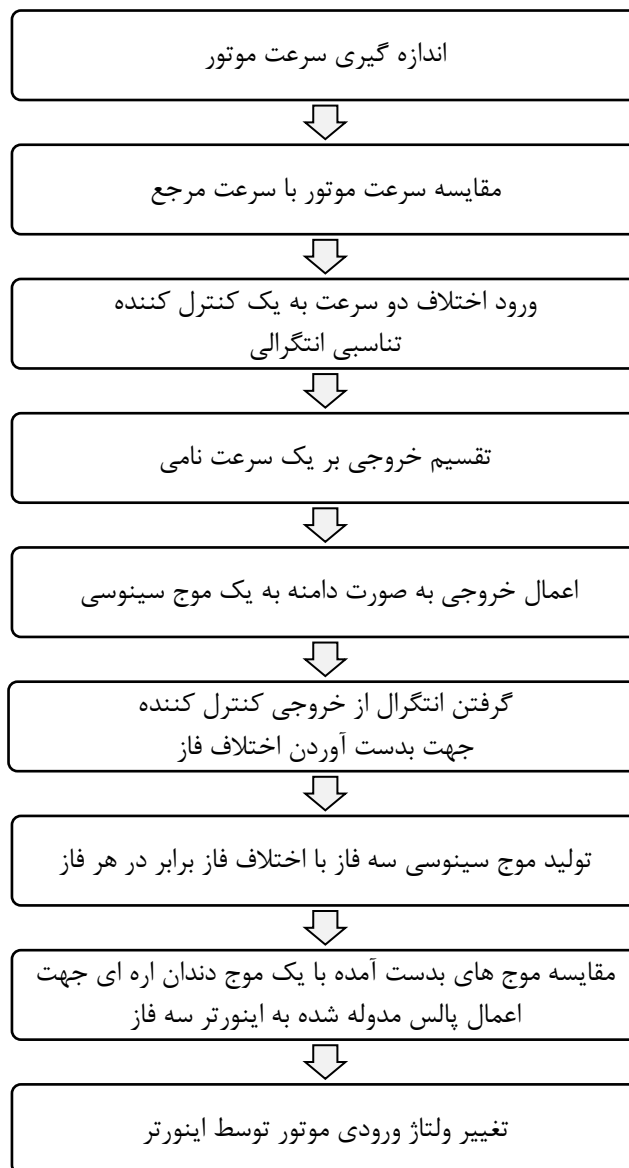
۴-۱ شبیه‌سازی

همانگونه که در فصل دوم اشاره شد مدار کنترل حلقه بسته موتور سه‌فاز از بخش‌های مختلفی تشکیل شده است. شکل ۴-۱ این بخش‌ها و روند کنترل سرعت موتور را نشان می‌دهد. در این فصل نحوه‌ی شبیه‌سازی هر یک از این قسمت‌ها بیان خواهد شد. به عبارت دیگر به ارائه‌ی یک مدل شبیه‌سازی شده از یک موتور القایی سه‌فاز به همراه مدار اینورتر سه‌فاز که کلیدزنی آن را مدولاسیون پهنای پالس سینوسی در واحد ساخت پالس فراهم می‌آورد، پرداخته می‌شود و بخش کنترل کننده نیز شرح داده می‌شود.



شکل ۴-۱: بلوک دیاگرام سیستم کنترل سرعت موتور القایی

برای روشن شدن مسیر انجام فرآیند کنترل سرعت موتور در شکل ۴-۲ مراحل طی شده توسط سیستم جهت کنترل موتور نشان داده می‌شود. مادامی که سرعت به سرعت مطلوب نرسیده است مراحل طی شده در شکل ۴-۲ مجدداً از اول تا آخر تکرار می‌شود تا خروجی نهایی به سرعت مطلوب برسد.



شکل ۴-۲: مراحل کنترل سرعت موتور القایی

ابتدا جهت طراحی و سنجش عملکرد مدار کنترل سرعت موتور از نرم افزار متلب استفاده می شود. کلیه قسمت های مدار که در شکل ۴-۲ شرح داده شد، در متلب شبیه سازی می شود. بلوک کنترل کننده، مولد پالس ورودی اینورتر سه فاز و... هر یک بوسیله بلوک های موجود در متلب پیاده سازی می شود. پس از طراحی مدار در محیط شبیه ساز متلب، قسمت های کنترل کننده و ساخت پالس که با رنگ طوسی در

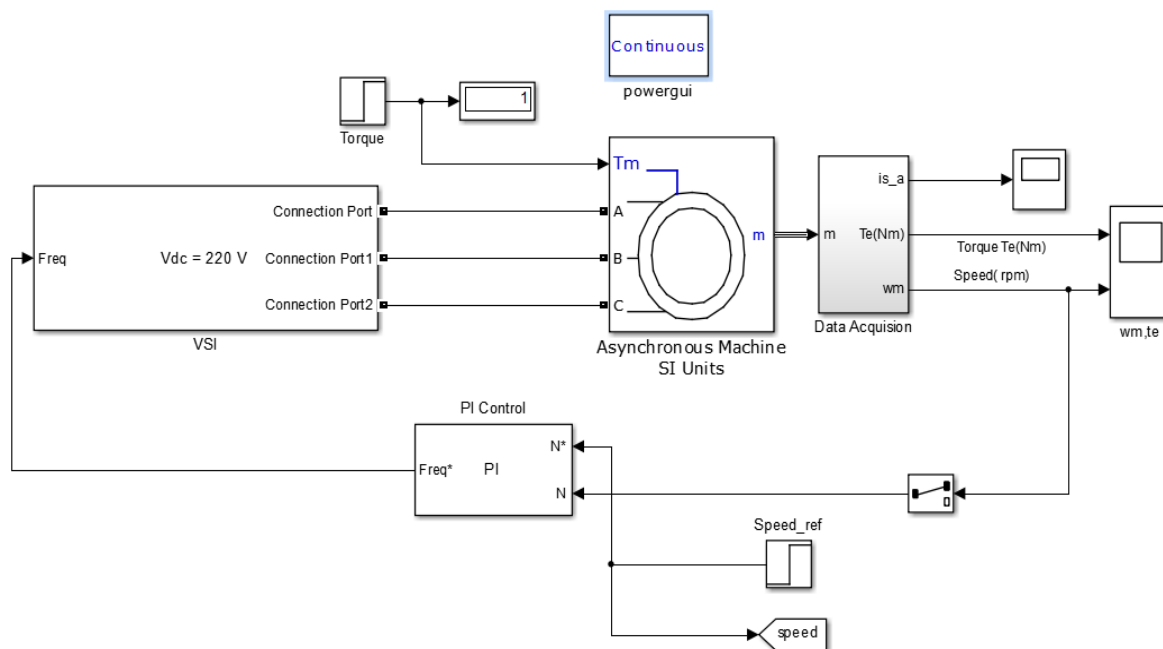
شکل ۱-۴ مشخص شده‌اند، توسط نرم‌افزار^۱ System Generator در FPGA پیاده‌سازی می‌گردد. در انتها برای بهبود عملکرد مدار در هر دو حوزه تمهیداتی در نظر گرفته می‌شود. برای اینکه روند شبیه‌سازی‌های این فصل به خوبی روشن شود، شبیه‌سازی در دو مرحله آنالوگ و دیجیتال انجام می‌شود.

در پیاده‌سازی حوزه آنالوگ موتور سه‌فاز، کنترل کننده تناسبی- انتگرالی و مدار هدایت موتور شبیه‌سازی می‌شود. در پیاده‌سازی دیجیتال بخش‌های کنترل کننده و ساخت پالس مدوله شده با بلوک‌های FPGA طراحی می‌شوند. پس از آن جهت بهبود عملکرد مدار از کنترل کننده فازی - عصبی جهت تنظیم ضرایب کنترل کننده استفاده می‌شود.

۲-۴ شبیه‌سازی در حوزه آنالوگ

همانطور که در بخش قبل گفته شد جهت طراحی مدار کنترل سرعت موتور از محیط شبیه‌ساز نرم‌افزار متلب استفاده می‌شود. این محیط امکانات‌های متعدد و سودمندی را برای طراحی هر مداری در اختیار کاربر قرار می‌دهد. با توجه به قابلیت‌های محیط متلب امکان شبیه‌سازی هر مداری به آسانی فراهم می‌شود. مدار طراحی شده برای کنترل سرعت موتور در شکل ۳-۴ نمایش داده شده است. این مدار از چند واحد اصلی تشکیل شده است که هر واحد وظیفه‌ی خاصی را در مدار به عهده دارد و هر بخش از یک یا چند زیربخش تشکیل شده است.

^۱ پیوست (الف)



شکل ۴-۳: شمای کلی سیستم مدار کنترل سرعت موتور

بخش‌های اصلی مدار عبارت است از:

- موتور سه‌فاز
- کنترل کننده مدار
- مدار هدایت موتور
- سنسور داده

۴-۲-۱ شبیه‌سازی موتور سه‌فاز

موتور مورد استفاده برای شبیه‌سازی رفتار موتور القایی سه‌فاز در برنامه‌ی شبیه‌سازی یکی از بلوک‌های استاندارد پیش فرض نرم‌افزار متلب^۱ در نظر گرفته می‌شود که در شکل ۴-۳ نشان شده است. جدول ۴-۱

^۱Asynchronous Machine

مشخصات این موتور را نمایش می‌شود. (بجز پارامترهای بیان شده در این جدول دیگر پارامترها برابر مقدار پیش فرض و یا محاسبه شده توسط نرم‌افزار متلب بوده و دست نخورده باقی می‌مانند).

جدول ۴-۱: مشخصات موتور القایی سه‌فاز مورد استفاده در پایان

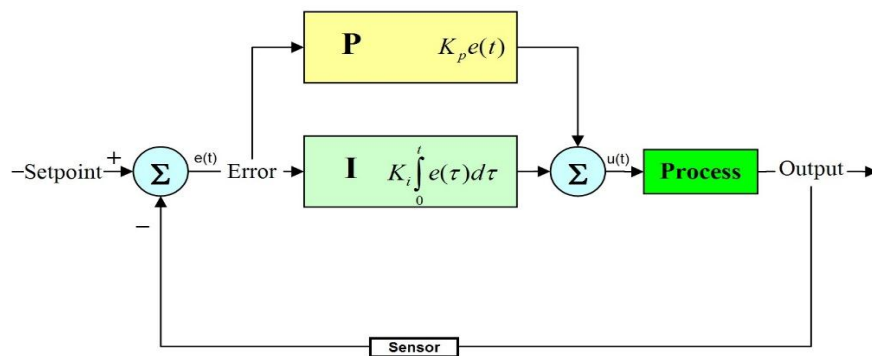
مشخصه موتور	اندازه	واحد
توان نامی	۳۷۳۰	ولت آمپر
ولتاژ نامی (خط)	۲۲۰	ولت (RMS)
فرکانس نامی	۵۰	هرتز
مقاومت اهمی استاتور	۱/۰۸	اهم
اندوکتانس استاتور	۵/۹	میلی هانری
ممان اینرسی	۰/۰۲	کیلوگرم مترمربع

هدف از این پایان نامه کنترل دور موتور سه‌فاز می‌باشد. مدار باید بر اساس سرعت مرجع که به مدار داده می‌شود در حداقل زمان ممکن سرعت خود را به مقدار مطلوب برساند.

۴-۲-۲ کنترل کننده مدار

هدف کنترل کننده، به حداقل رساندن خطا با تنظیم ورودی‌های کنترل فرآیند است. جهت کنترل سرعت موتور از کنترل کننده تناسبی انتگرالی^۱ استفاده می‌شود. کنترلر PI برای جبران خطای ماندگار به کار می‌رود، زیرا تا وقتی که خطایی در خروجی وجود داشته باشد، جمله انتگرال تغییر پیدا می‌کند و در نتیجه خطای خروجی رفته رفته کاهش می‌یابد. پایداری، سرعت و نداشتن خطای حالت ماندگار از ویژگی‌های این کنترل کننده است که در شکل ۴-۴ نمایش داده شده است.

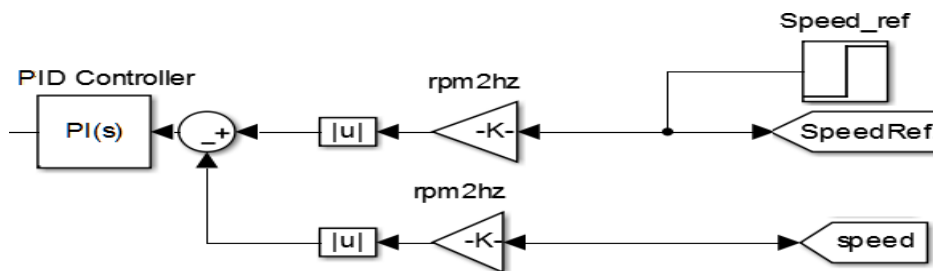
² Proportional Integral (PI)



شکل ۴-۴: بلوک دیاگرام کنترل کننده تناسبی - انتگرالی

ساختار حلقه بسته به طراح کمک می کند که توانایی سیستم را برای بازتولید ورودی به صورت درست و صحیح افزایش دهد، حساسیت نسبت خروجی به ورودی را کاهش دهد، از اثر عوامل غیر خطی و اختلالات یا نویزهای خارجی بکاهد و پهنای باند سیستم را بیفزاید. علت استفاده کنترل کننده PI در این پایان نامه سهولت پیاده سازی آن در حوزه دیجیتال است.

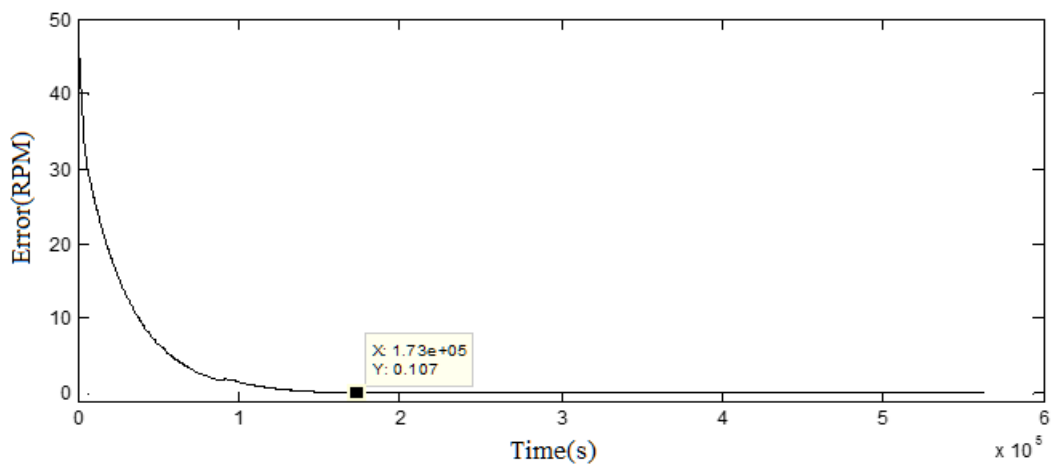
شکل ۴-۵ شبیه سازی کنترل کننده PI ساده را در محیط متلب نشان می دهد. برای پیاده کردن کنترل کننده تناسبی - انتگرالی شکل ۴-۴ در اینجا از بلوک آماده متلب استفاده می شود. کفایت ضرایب کنترل کننده به تنظیمات بلوک اعمال شود. ورودی های کنترل کننده شامل سرعت مرجع و سرعت موتور است که قرار است به سرعت مرجع برسد.



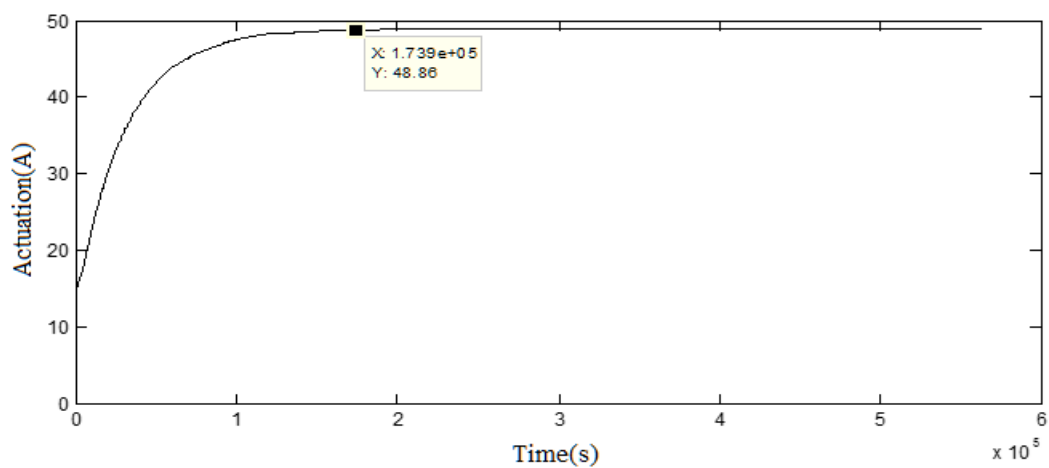
شکل ۴-۵: مدل ساده کنترل کننده تناسبی - انتگرالی

سیگنال خطا که ورودی کنترل کننده است و سیگنال تحریک که خروجی آن و ورودی مولد پالس است برای کنترل کننده تناسبی - انتگرالی در شکل ۴-۶ نمایش داده شده است. همانطور که در شکل

مشخص است سیگنال خطا رفته رفته کم می‌شود تا اختلاف سرعت به حداقل برسد و سرعت واقعی موتور به مقدار مطلوب نزدیک شود. فرآیند کم شدن خطا و میل کردن سیگنال تحریک به مقدار ثابت حدود ۱۷ ثانیه طول می‌کشد.

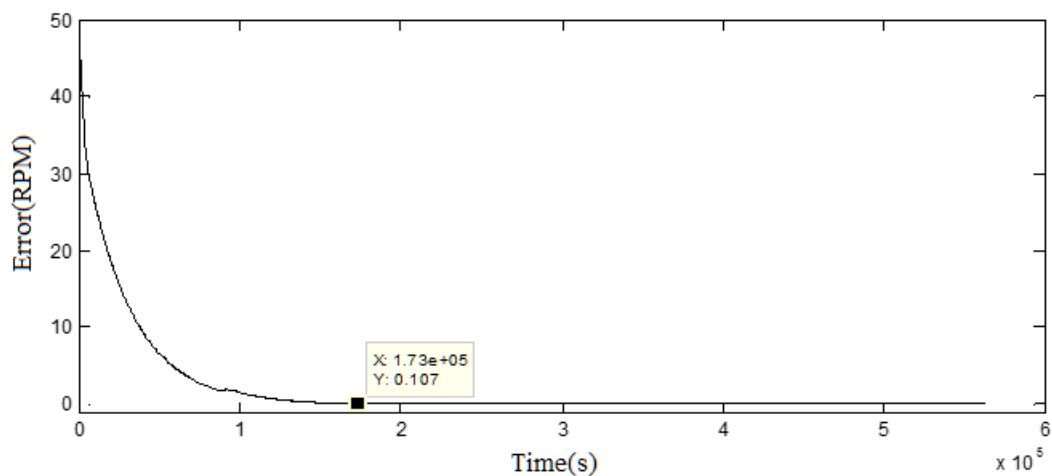


(الف)

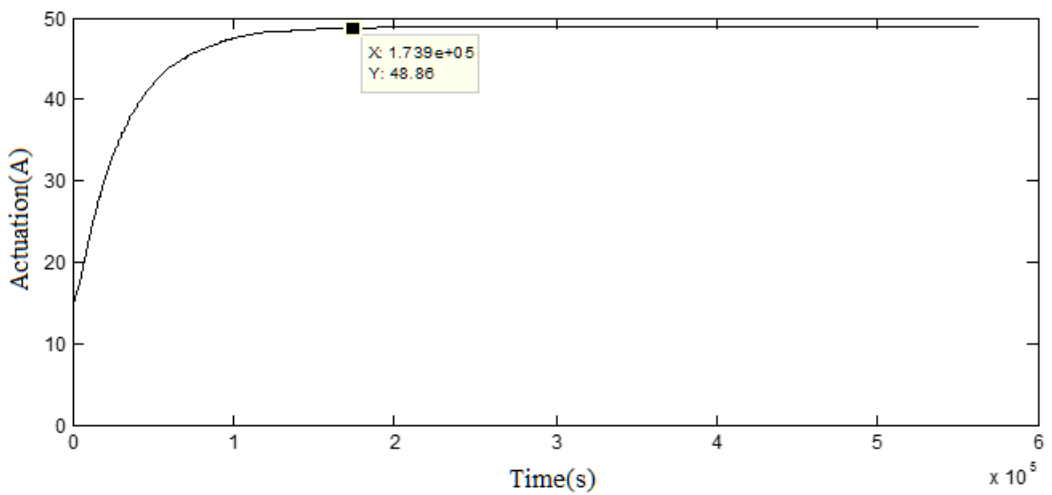


(ب)

شکل ۴-۶: (الف) سیگنال خطا و (ب) سیگنال تحریک در کنترل کننده تناسبی - انتگرالی ساده



(الف)



(ب)

شکل ۴-۶: (الف) سیگنال خطا و (ب) سیگنال تحریک در کنترل کننده تناسبی - انتگرالی ساده

در بسیاری از فرآیندها فرمان کنترل کننده از سقف اشباع بالاتر می‌رود و چون کنترل کننده دارای محدودیت می‌باشد، در عمل حلقه فیدبک کارایی خود را از دست می‌دهد. عاملی که بیشترین تاثیر در به اشباع رفتن کنترل کننده دارد ترم انتگرال گیر کنترل کننده می‌باشد. بخاطر عمل انتگرال گیری، خطا روی هم انباشته شده و ممکن است از حد اشباع عبور کند. به این فرآیند جمع شونده^۱ گفته می‌شود.

^۱ Windup

اگر جلوی جمع شونده‌گی گرفته نشود چون زمان زیادی نیاز است تا خطای منفی جمع گردد و اثر جمع شونده‌گی را از بین ببرد پاسخ سیستم خیلی کند شده و کنترل کننده نمی‌تواند ورودی را دنبال کند. روش‌های زیادی برای جلوگیری از جمع شونده‌گی^۱ ارائه شده است که به دو صورت آنالوگ و دیجیتال قابل پیاده‌سازی می‌باشد.

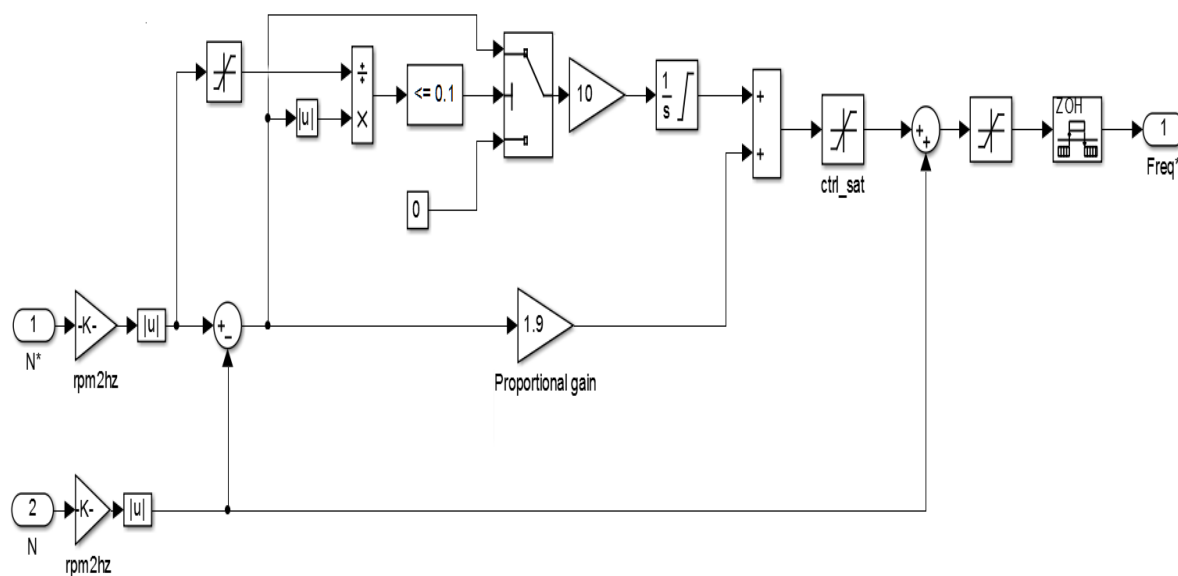
ایده اصلی تمام روش‌های ضد اشباع این است که زمانی که کنترل کننده به اشباع می‌رود، چون انتگرال گیری از خطا تاثیری ندارد پس بهتر است که انتگرال گیر خاموش شود. بصورت دیجیتالی کافی است که اگر مقدار خروجی کنترل کننده از ماکزیمم خود بیشتر شد مقدار انتگرال گیر را صفر کنیم.

در طراحی این مدار نیز اگر اختلاف سرعت واقعی از سرعت مرجع زیاد باشد، ترم انتگرالگیر بزرگ شده و خروجی به اشباع می‌رود. این رخداد موجب ناپایداری سیستم می‌شود. برای بهتر شدن عملکرد مدار می‌توان از کنترل کننده تناسبی - انتگرالی با ساختار متفاوت استفاده نمود. این کنترل کننده به گونه‌ای طراحی شده است که از اشباع جلوگیری می‌کند که اصطلاحاً به آن کنترل کننده ضد اشباع می‌گویند.

شمای کلی کنترل PI ضد اشباع شبیه‌سازی شده در شکل ۴-۷ نشان داده شده است. ساختار این کنترل کننده ضد اشباع بدین صورت است که قدرمطلق سیگنال خطا بر قدرمطلق سیگنال مرجع تقسیم می‌شود. ضمن آنکه خود سیگنال مرجع یا همان فرمان کنترلر از یک بلوک اشباع عبور داده می‌شود که از تقسیم بر صفر جلوگیری می‌کند. حد پایین بلوک اشباع را ۲ و حد بالای آن بی نهایت انتخاب می‌شود و با این عمل سیگنال بین ۲ تا بی نهایت محدود می‌شود که منجر به جلوگیری از تقسیم شدن آن بر صفر شدن می‌گردد. حاصل تقسیم با عدد ۰/۱ مقایسه می‌شود. اگر حاصل تقسیم کمتر از ۰/۱ باشد، خروجی بلوک مقایسه کننده ۱ می‌گردد در غیر این صورت خروجی صفر خواهد شد. اگر خروجی مقایسه کننده ۱ باشد، بلوک سوئیچ بعد از آن سیگنالی که به پورت بالایی‌اش وارد می‌شود را عبور می‌دهد (یعنی خود سیگنال خطا) و اگر خروجی صفر باشد سوئیچ سیگنالی که به پورت پایین آن متصل است (یعنی عدد صفر) را عبور می‌دهد.

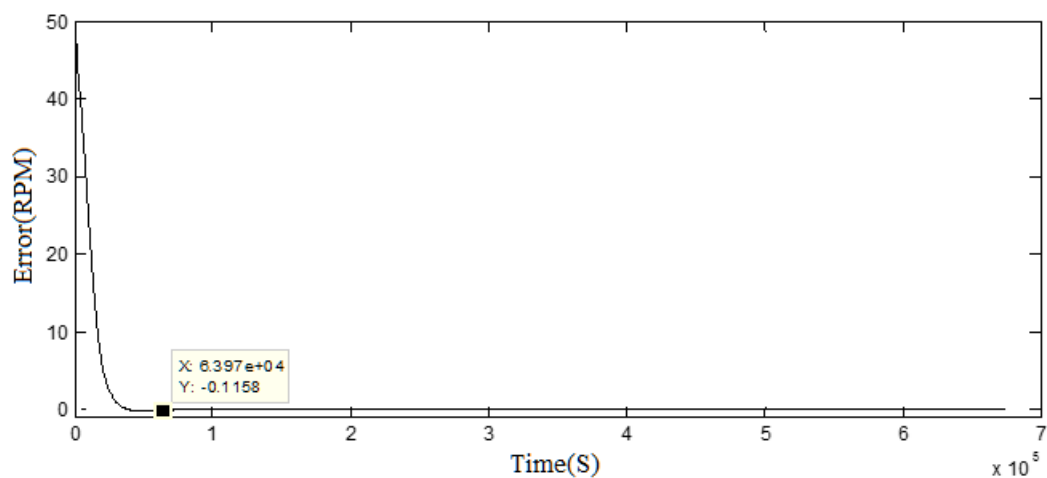
² Anti-windup

با توجه به آن چه گفته شد مادامی که تفاوت سرعت واقعی و سرعت مرجع مقدار بزرگی است به ورودی بلوک انتگرال گیر مقدار صفر وارد می‌شود. زمانی که سیستم از اشباع خارج شده و حاصل تقسیم سیگنال خطا بر میزان مرجع کوچک شده و خود سیگنال خطا به بخش انتگرالگیر کنترل کننده وارد خواهد شد. ضرایب کنترل کننده از روش فرآیند بدست آمده‌اند.

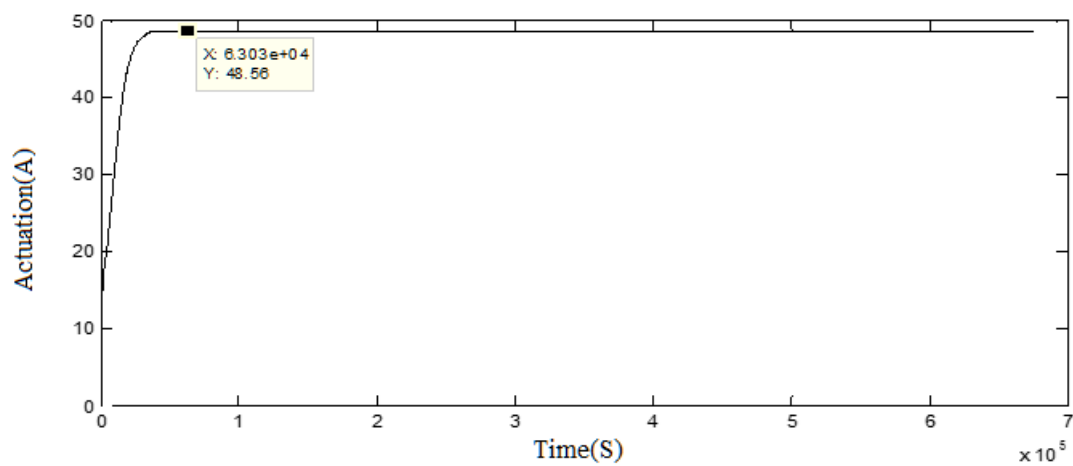


شکل ۴-۷: شبیه‌سازی کنترل کننده تناسبی انتگرالی ضد اشباع

سیگنال خطا و سیگنال تحریک این بار در شکل ۴-۸ برای مدل بهبود یافته کنترل کننده ترسیم شده است. از مقایسه شکل‌های ۴-۶ و ۴-۸ می‌توان دریافت که بهبود عملکرد مدار به سادگی صورت گرفته است. در کمتر از ۷ ثانیه سیگنال خطای مدل ضد اشباع به صفر میل می‌کند و سیگنال تحریک نیز به مقدار ثابتی می‌رسد.



(الف)



(ب)

شکل ۴-۸: (الف) سیگنال خطا و (ب) سیگنال تحریک در کنترل کننده ضد اشباع

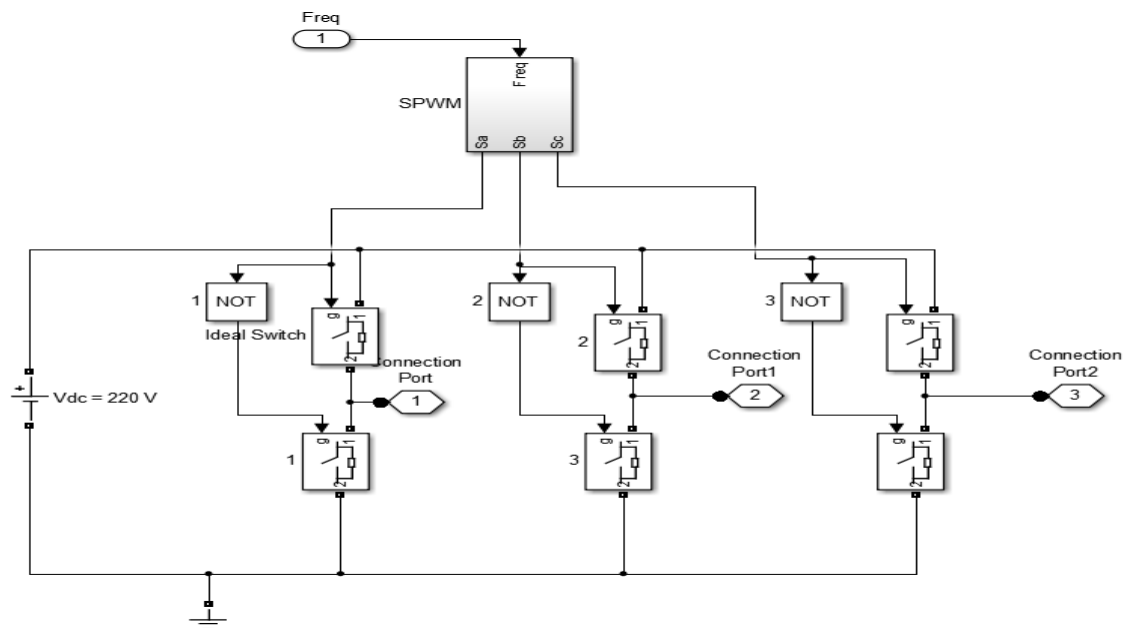
۴-۲-۳ مدار هدایت موتور

واحد ساخت پالس ورودی موتور خود از دو زیربخش تشکیل شده است:

- اینورتر سه فاز
- شبیه ساز مدولاسیون عرض پالس سینوسی

الف) اینورتر سه فاز

بلوک قبل از موتور شامل اینورتر سه فاز با ۶ سوئیچ می باشد که در شکل ۴-۹ نشان داده شده است. همان طور که قبلاً اشاره شد اینورتر سه فاز متداولترین و معروفترین اینورتر مورد استفاده در صنعت و نیز محیط های علمی می باشد. سوئیچ های اینورتر می توانند کلید ایده آل، ماسفت یا هر سوئیچ دیگری باشند. کلیدهای مربوط به یک بازو همزمان نباید روشن باشند در غیر این صورت منجر به اتصال کوتاه شدن و آسیب رساندن به موتور می شود. بنابراین در هر بازو موج ورودی به کلید بالایی دقیقاً معکوس موج ورودی به کلید پایینی است. سه خروجی اینورتر به سه فاز موتور وارد شده و هدایت آن را به عهده دارند.

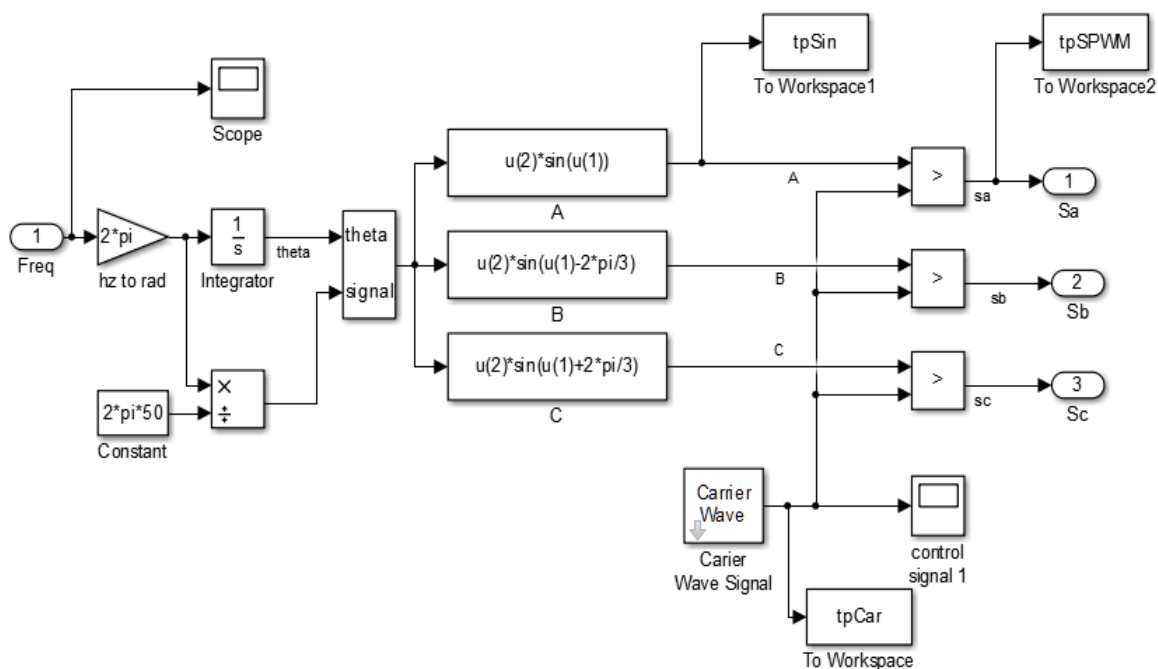


شکل ۴-۹: شبیه سازی اینورتر سه فاز در متلب

ب) شبیه سازی مدولاسیون عرض پالس سینوسی

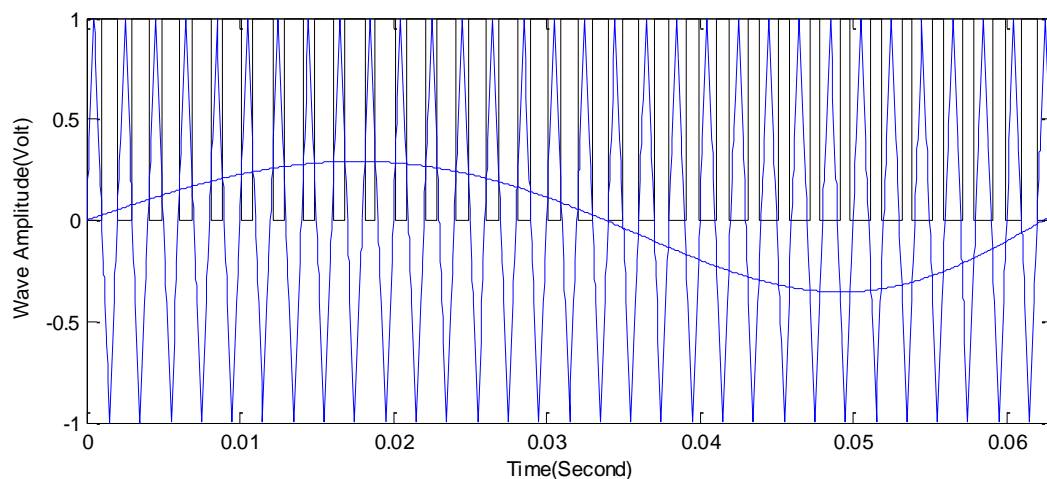
کلیدزنی صحیح اینورتر سه فاز جهت هدایت مناسب موتور سه فاز را واحد مدولاسیون عرض پالس برعهده دارد. شش سیگنال مدوله شده مناسب جهت کلیدزنی صحیح شش سوئیچ اینورتر سه فاز مورد نیاز است. همانطور که گفته شد در هر بازوی اینورتر شکل موج پایینی معکوس بالایی است. پس در واقع مدار سه شکل موج اصلی نیاز دارد که با معکوس کردن هر کدام از آنها شش شکل موج برای شش سوئیچ اینورتر ایجاد می شود. این سه شکل موج مدوله شده از مقایسه سه موج سینوسی با اختلاف فاز ۱۲۰

درجه، با حامل دندان اره‌ای بوجود می‌آیند. برای تولید هر موج سینوسی به دامنه و فاز نیاز است. این دامنه و فاز با توجه به فرکانسی که از خروجی بلوک کنترل کننده می‌آید، ساخته می‌شود. دامنه این شکل موج از تقسیم خروجی کنترل کننده تناسبی انتگرالی بر سرعت نامی بدست می‌آید و برای تولید فاز موج سینوسی ابتدا فرکانس از هر تتر به رادیان تبدیل شده سپس از آن انتگرال گرفته می‌شود و سپس این دو مقدار بدست آمده را به یک موج سینوسی واحد اعمال می‌گردد. موج سینوسی حاصل با شکل موج دندان اره‌ای مقایسه شده و موج خروجی را می‌سازد. بلوک مقایسه کننده موج سینوسی را با مثلی مقایسه می‌کند و اگر موج سینوسی بزرگتر باشد خروجی ۱ و اگر کوچکتر باشد خروجی صفر می‌شود. بدین ترتیب شکل موج شبه مربعی مدولاسیون عرض پالس سینوسی که در این طراحی مد نظر است، در خروجی تولید می‌شود. شکل ۴-۱۰ شبیه‌سازی این بخش را نشان می‌دهد.



شکل ۴-۱۰: شبیه‌سازی بلوک مدولاسیون عرض پالس سینوسی

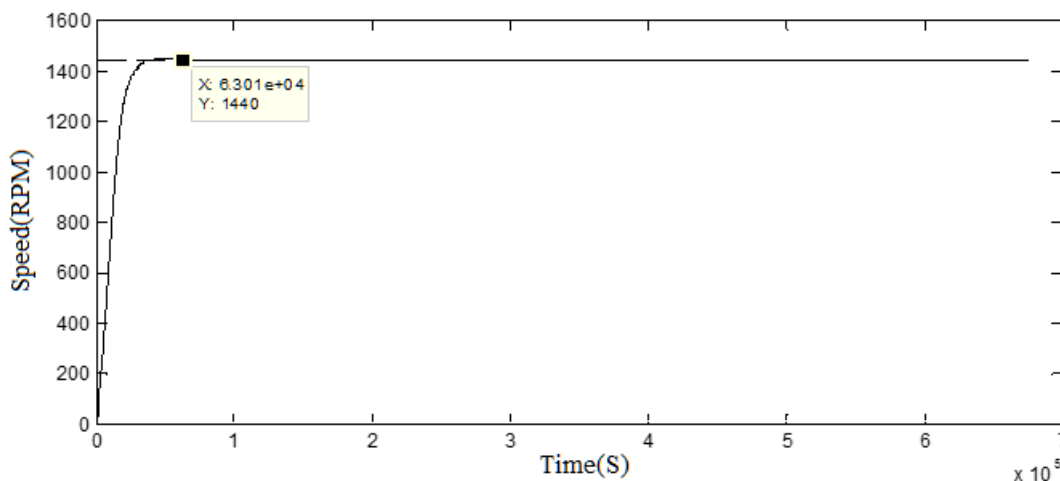
موج سینوسی بوجود آمده از سیگنال کنترلی ورودی، موج مثلی حامل و پالس مدوله شده حاصل از مقایسه آن دو موج با هم در شکل ۴-۱۱ نشان داده می‌شود. این شکل گویای آن است که مدولاسیون عرض پالس سینوسی به خوبی شبیه‌سازی شده است.



شکل ۴-۱۱: موج مدوله شده SPWM حاصل از شبیه‌سازی مدار در متلب

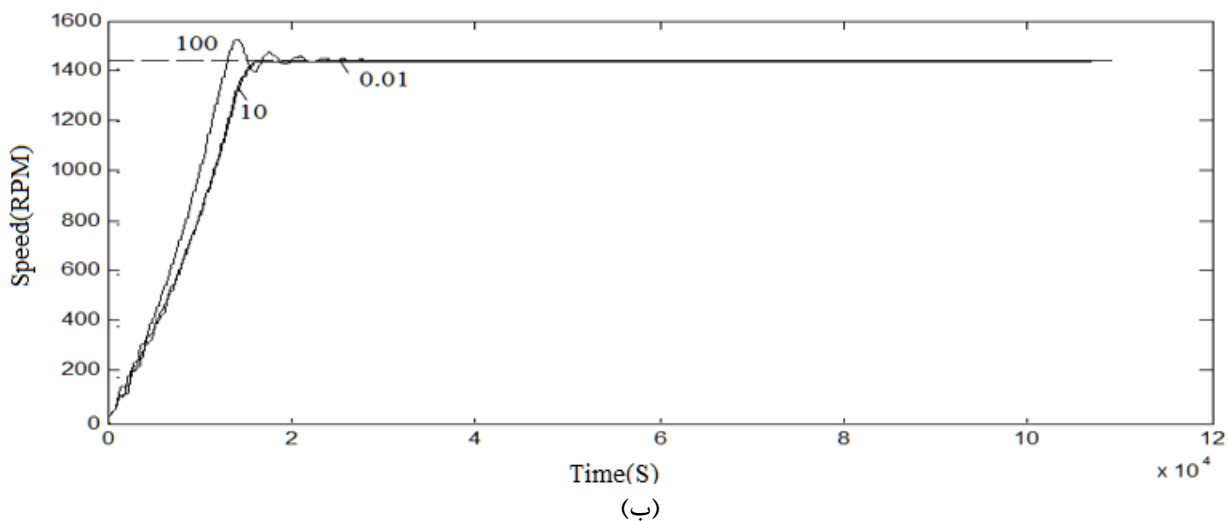
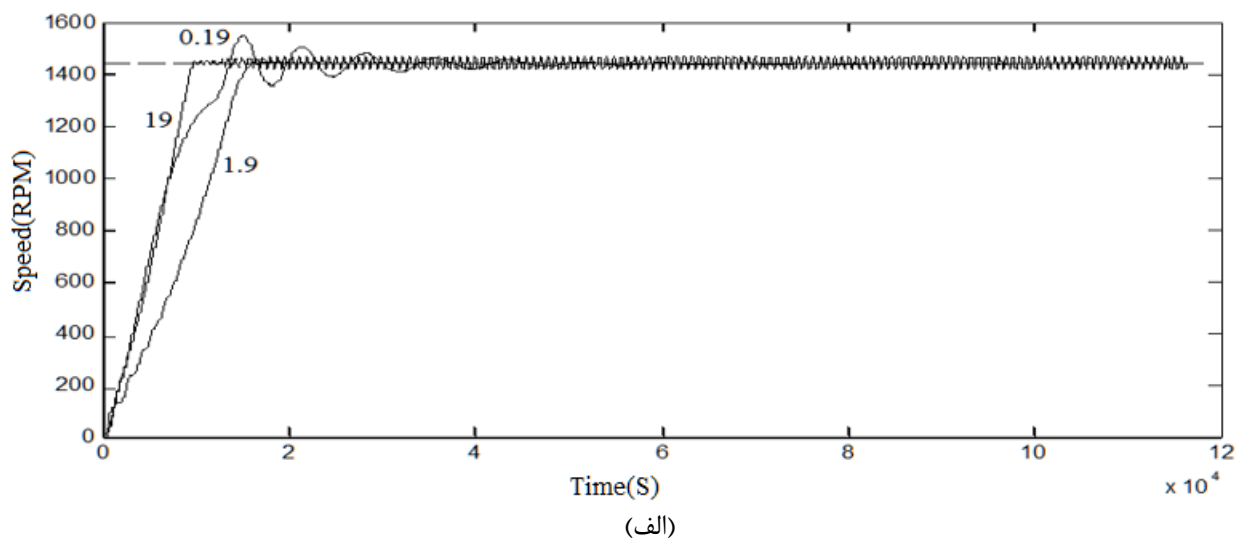
۴-۲-۴ سنجش داده

بلوک سنجش داده وظیفه دارد اطلاعات را از خروجی موتور استخراج کند. می‌توان مشخصه گشتاور، جریان استاتور و سرعت موتور را در این بلوک مشاهده و رسم نمود. برای مثال اگر سرعت مرجع سیستم برابر ۱۴۴۰ RPM باشد. نمودار سرعت مرجع و سرعت واقعی ژنراتور در شکل ۴-۱۲ نشان داده شده است.



شکل ۴-۱۲: سرعت مرجع و سرعت واقعی موتور

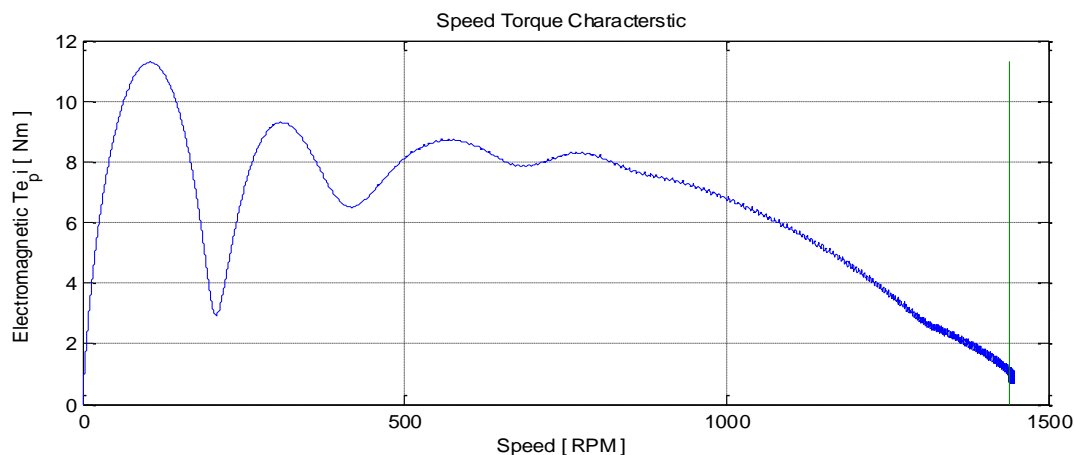
همانطور که در شکل ۴-۱۲ مشاهده می‌شود سرعت مرجع با خط چین نشان داده شده است. طبق طراحی صورت گرفته سرعت واقعی موتور بخوبی سرعت مرجع را دنبال می‌کند و مقدار اورشوت پاسخ بسیار کم است. سرعت موتور در کمتر از ۷ ثانیه به مقدار نهایی خود یعنی مقدار مرجع رسیده است.



شکل ۴-۱۳: مشخصه سرعت خروجی به ازای ضرایب مختلف کنترل کننده (الف) $K_p=0.19, 1.9, 19$ و (ب) $K_i=0.01, 10, 100$

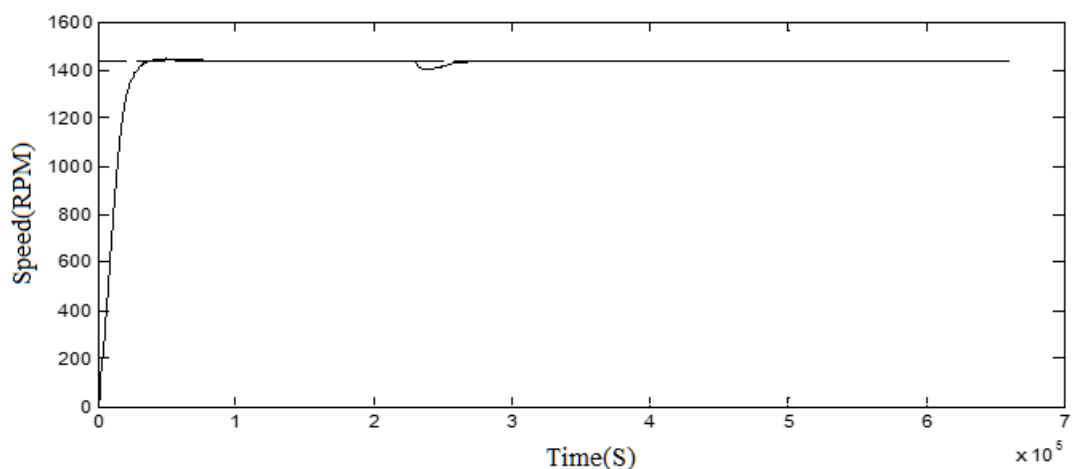
ضرایب کنترل کننده همانطور که در بخش قبل گفته شد از روش کنترل فرآیند بدست آمده است. می توان مشخصه سرعت خروجی را به ازای مقادیر مختلف ضریب انتگرالی و ضریب تناسبی می توان بدست آورد. شکل ۴-۱۳ نحوه انتخاب ضرایب مناسب با روش کنترل فرآیند را نشان می دهد. از میان منحنی های بدست آمده مشخصه خروجی با ضریب تناسبی ۱/۹ و ضریب انتگرالی ۱۰ پاسخ خروجی بهتری ارائه می دهد.

مشخصه گشتاور-سرعت موتور را نیز می توان ترسیم نمود. این مشخصه به صورت شکل ۴-۱۴ قابل نمایش است. طبق آنچه در خصوص منحنی گشتاور-سرعت در فصل دوم گفته شد، مشخصه منطقی حاصل شده است. در ابتدای منحنی قدری نوسان دیده می شود. علت آن است که هنوز کنترل کننده به حالت پایدار نرسیده و خطا زیاد است که در نهایت پایدار می شود.



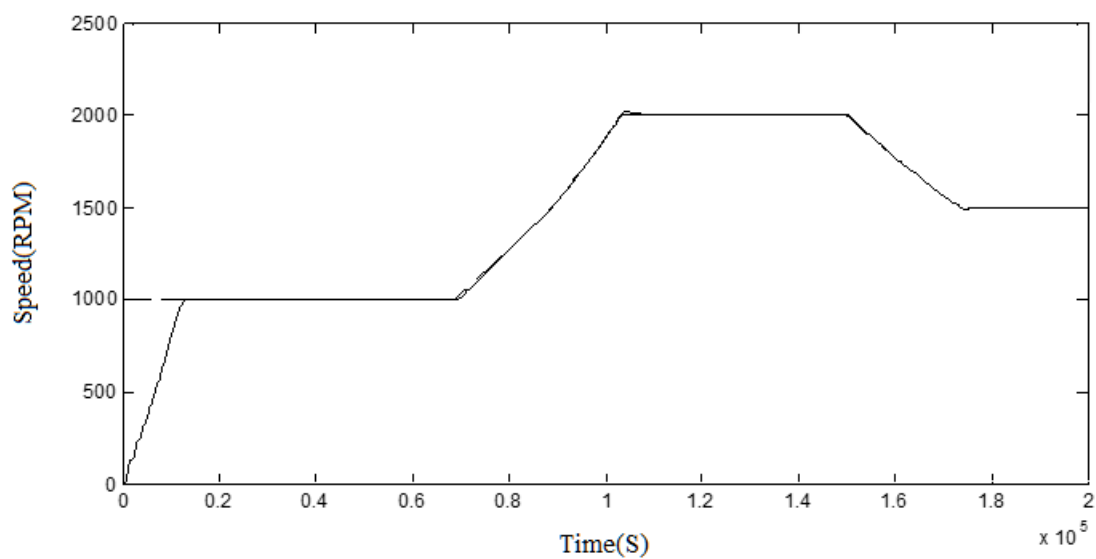
شکل ۴-۱۴: مشخصه گشتاور-سرعت موتور

می توان گشتاور بار موتور را به صورت پله ای به آن اعمال نمود. برای مثال گشتاور بار موتور ابتدا صفر باشد آنگاه در ثانیه ۴ام ناگهان افزایش یابد. مشخصه سرعت خروجی مانند شکل ۴-۱۵ خواهد شد. تنها کمی پایین زدگی در مشخصه سرعت مشاهده می شود که آن نیز نهایتاً برطرف شده است. این نشان می دهد که مدار با وجود گشتاور بار باز هم خود را پایدار می کند و سرعت موتور را به سرعت مطلوب می رساند.



شکل ۴-۱۵: مشخصه سرعت موتور با وجود اعمال گشتاور

در این پایان نامه تنها این مهم است که سرعت موتور از یک سرعت مرجع پیروی کند و در حداقل زمان ممکن به آن برسد. می‌توان آرایه به خصوصی از سرعت را به مدار داد و نحوه‌ی دنبال کردن خروجی را مشاهده نمود.



شکل ۴-۱۶: سرعت موتور بر اساس look up table ورودی مشخص

طبق شکل ۴-۱۶ و با توجه به اینکه نرخ نمونه برداری مدار ۰.۰۰۰۱ ثانیه است، سرعت مرجع ۲۰ ثانیه در ۱۰۰۰ است و بعد به ۲۰۰۰ میرسد و ۱۵ ثانیه در ۲۰۰۰ می ماند و نهایتاً سرعت ۱۵۰۰ می شود. منحنی خط چین سرعت مرجع و منحنی متصل سرعت اصلی را نشان می دهد که به خوبی مرجع توسط مدار دنبال می شود.

۳-۴ شبیه سازی در حوزه دیجیتال

در این بخش سیگنال ورودی اینورتر در حوزه دیجیتال شبیه سازی می شود. همانطور که پیشتر اشاره شد از تراشه FPGA در پیاده سازی سخت افزاری بخش تولید سیگنال ورودی اینورتر می توان استفاده نمود. در این بخش نحوه شبیه سازی بلوک های تولید کننده شکل موج ورودی اینورتر سه فاز، به عبارت دیگر بلوک های کنترل کننده و مدولاسیون عرض پالس در حوزه دیجیتال شرح داده خواهد شد.

در ابتدا قبل از هر توضیحی لازم می باشد به نکات مهم و ساختاری که در برنامه ریزی FPGA به کار برده شده است و نقش مهمی در صحت عملکرد برنامه ایفا می کنند اشاره شود. سعی شده است کلیه کدها به شکل سنکرون و با یک کلاک نوشته شود. به عبارت دیگر، عبارت پردازش^۱ در کد تنها حساس به کلاک سیستم می باشد و به سیگنال دیگری حساسیت ندارد. همچنین تمام کدها با شرط لبه بالارونده نوشته شده است. اگر کاربر در قسمتی از کد مجبور به استفاده از کلاک با فرکانس دیگری شود، تلاش بر آن بوده تا در صورت امکان این کلاک ضریب صحیحی از کلاک اصلی سیستم باشد.

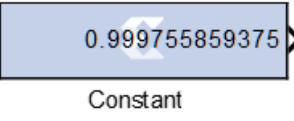

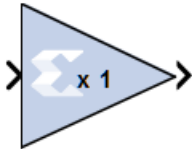
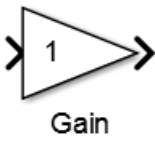
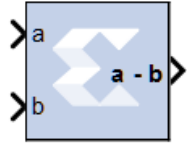

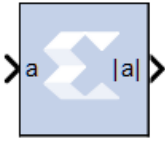

تمام سیگنال های ورودی (داده و ارسال در ماژول فرستنده) به جز کلاک یک بار در عبارت پردازش در ثباتی داخلی بافر شده و پس از آن در تمام کد، به جای استفاده از سیگنال ورودی از سیگنال بافر شده استفاده شده است. همچنین برای تمام سیگنال های خروجی (اشغال و سریال در ماژول فرستنده) یک سیگنال داخلی تعریف شده و تمام ارجاعات به آن سیگنال خروجی را به سیگنال داخلی اعمال شده است. سپس در خارج از عبارت پردازش سیگنال داخلی را به سیگنال پورت خروجی اعمال می گردد. سعی شده است تا جایی که ممکن است از ساختارهای شرطی تودرتو اجتناب کرده و بیش از دو لایه عبارت شرطی را در ماژول نباشد. همچنین تا حد امکان از شرط های ساده استفاده شده است.

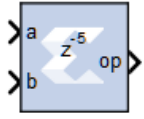

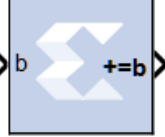
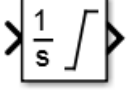
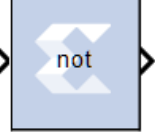

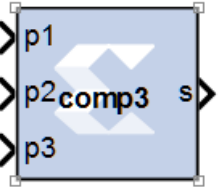
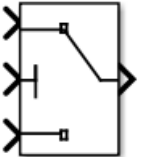
¹ Process

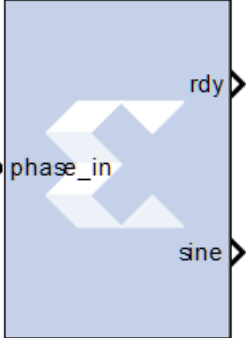
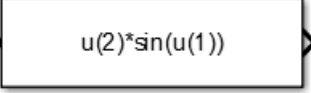
برای سنتز مدار روی FPGA از نرم‌افزار System Generator استفاده شده است. این نرم‌افزار تنها به محیط شبیه‌ساز متلب چند بلوک دیجیتالی اضافه می‌کند. در خصوص این نرم‌افزار در پیوست (د) توضیح داده شده است.

جهت استفاده از بلوک‌های این نرم‌افزار لازم است طراح ابتدا با بلوک‌های اصلی مورد استفاده که در طراحی حوزه دیجیتال کمک می‌کند آشنا شود. جدول ۲-۴ این بلوک‌ها را معرفی و بلوک مشابه آن در متلب را نیز نشان می‌دهد.

جدول ۲-۴ بلوک‌های معادل متلب در پیاده‌سازی دیجیتالی کنترل کننده

وظیفه	بلوک معادل آن در SysGen	وظیفه	بلوک متلب
عدد ثابت باینری در حوزه دیجیتال به مدار وارد می‌کند.	 Constant	عدد ثابت به مدار وارد می‌کند.	 Constant
عمل ضرب را در FPGA انجام می‌دهد.	 CMult4	بلوک ضرب کننده است.	 Gain
جمع یا تفریق باینری انجام می‌دهد.	 AddSub	عمل جمع یا تفریق انجام می‌دهد.	 Sum
مثبت عدد باینری را محاسبه می‌کند.	 Absolute	قدر مطلق ورودی‌اش را حساب می‌کند.	 Abs

<p>عملیات تقسیم در FPGA را انجام می دهد.</p>	 <p>Divide</p>	<p>عمل تقسیم (همچنین ضرب) را انجام می دهد.</p>	 <p>product</p>
<p>انباشتگر ورودی ها را مرتباً باهم جمع می کند.</p>	 <p>Accumulator</p>	<p>بلوک انتگرال گیر</p>	 <p>Integrator</p>
<p>عمل منطقی معکوس کردن علامت را روی عدد باینری انجام می دهد.</p>	 <p>Inverter</p>	<p>بلوک منطقی که مقدار ورودی را از نظر علامت معکوس می کند.</p>	 <p>logical operator</p>
<p>در کلیه بلوک های شرطی می توان با استفاده از بلوک MCode یک کد mfile ساده در نوشت و به آدرس کد در مشخصات بلوک ارجاع داد.</p>	 <p>MCode</p>	<p>یک سوئیچ شرطی است که ورودی اول و آخر را با وسط مقایسه می کند و هرکدام بیشتر باشد عبور می دهد.</p>	 <p>Switch</p>

<p>دامنه و فاز سینوس ورودی را به صورت لحظه ای حساب می کند.</p>	 <p>DDS Compiler 4.0 4</p>	<p>تابع خاص در مطلب را می توان از طریق این بلوک به مدار داد. مثل توابع نمایی، سینوسی و ...</p>	 <p>Function block</p>
----------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

لازم به ذکر می باشد جهت ارتباط و اتصال سایر بخش ها به این بلوک ها در ورودی و خروجی به ترتیب درگاه ورودی^۱ و درگاه خروجی^۲ استفاده شده است. از این بلوک ها جهت ورود و خروج داده استفاده می شود. بلوک درگاه خروجی تنظیمات خاصی نداشته و تعداد بیت های خروجی نیز با توجه به سیگنالی که به آن وصل شده به صورت خودکار تعیین می گردد. بلوک درگاه ورودی قابلیت تعیین ورودی به شکل های بولین^۳ و ممیز ثابت و ممیز شناور می باشد. نکته مهم در رابطه با این بلوک این است که زمان نمونه برداری از خروجی بلوک را می توان تعیین کرد با توجه به این مطلب که این زمان نمی تواند از مقدار پریود شبیه ساز سیستم^۴ کمتر باشد.

¹ GATEWAY IN

² GATEWAY OUT

³ Boolean

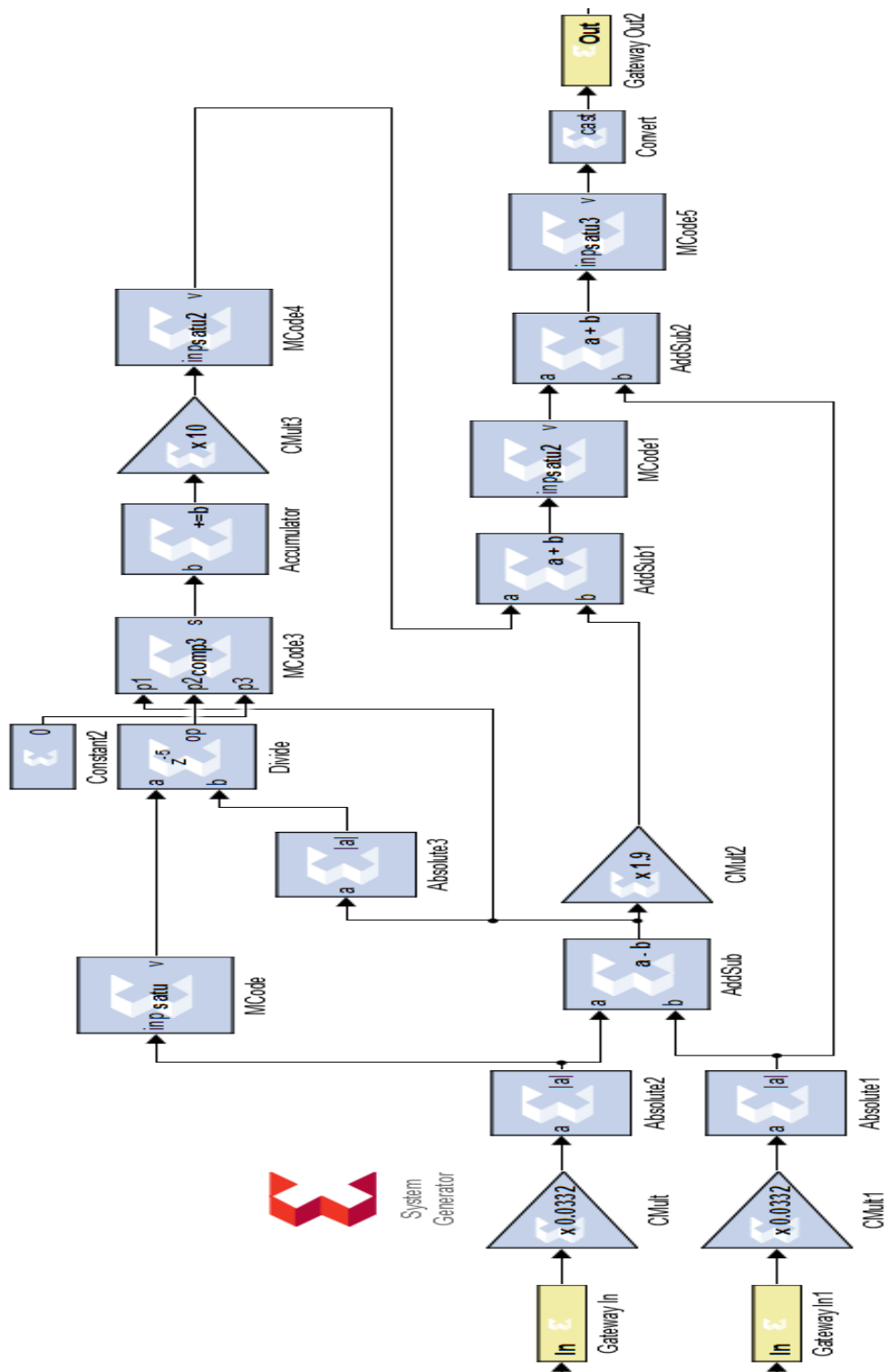
⁴ Simulink system Period

۴-۳-۱ پیاده‌سازی کنترل کننده تناسبی انتگرالی در حوزه دیجیتال

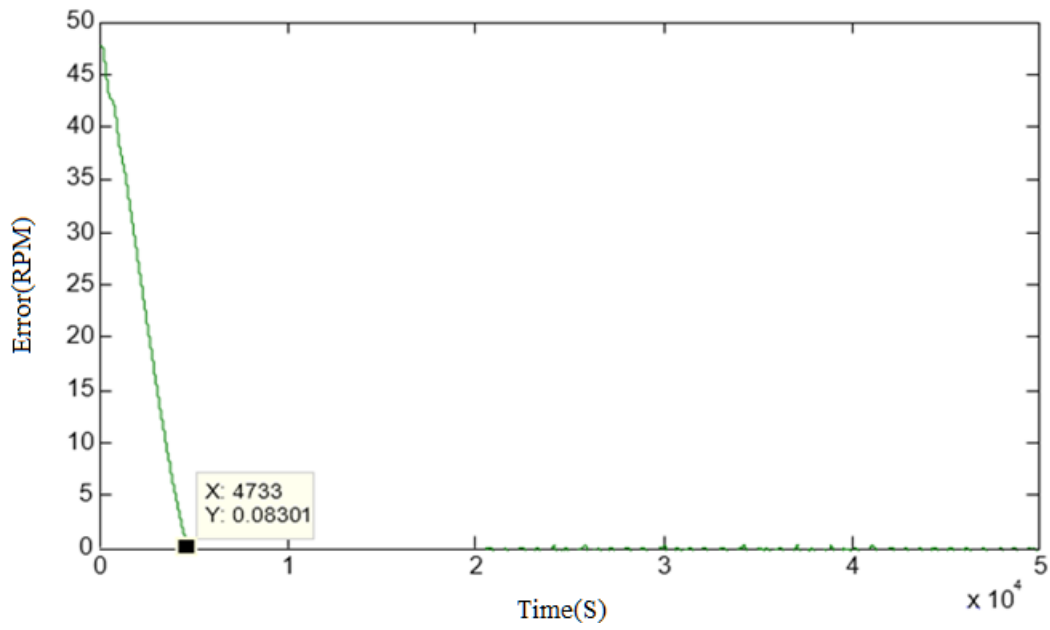
این بخش از شبیه‌سازی در حوزه‌ی دیجیتال کاملاً مشابه شبیه‌سازی آنالوگ آن انجام می‌شود به عبارت دیگر به جای بلوک‌های ضرب کننده از بلوک ضرب کننده ثابت^۱ در FPGA و به جای تفریق کننده سرعت از سرعت مرجع، از بلوک جمع و تفریق کننده^۲ استفاده می‌شود. همانطور که قبلاً توضیح داده شد علت استفاده از این کنترل کننده آن است که پیاده‌سازی سهل الوصول تری در FPGA دارد. با اعمال بلوک‌های موجود گفته شده در نرم‌افزار System Genetator ساختاری همانند شکل ۴-۱۷ در محیط نرم‌افزار بوجود خواهد آمد که در واقع طراحی کنترل کننده تناسبی انتگرالی با بلوک‌های FPGA است. طبق آنچه در بخش قبل گفته شد بلوک‌های GATEWAY IN و GATWAY OUT همانند یک درگاه جهت ارتباط ورودی‌های مدار و خروجی آن با سایر بخش‌های مدار که خارج از FPGA هستند به کار می‌روند تا فرمت داده را در این دو حوزه عوض کنند. ورودی اول سرعت مرجع و ورودی دوم سرعت اصلی است. از اختلاف این دو مقدار با هم قدر مطلق گرفته می‌شود این انتگرال توسط بلوک تقسیم کننده بر مقدار مرجع تقسیم می‌شود. بلوک MCode شرط سوئیچ را بررسی می‌کند و خروجی را طبق شرط به انباشتگر می‌فرستد. سپس ترم تناسبی کنترل کننده با آن جمع می‌شود و از یک Mcode دیگر که مقدار را از بالا و پایین محدود می‌کند عبور می‌کند بدین ترتیب عمل کنترل تناسبی انتگرالی مشابه بخش قبل در FPGA پیاده می‌شود. ضمناً FPGA در اینجا spartan3 انتخاب شده است که با توجه به حجم سخت افزار کفایت می‌کند. منابع مورد استفاده از FPGA انتخابی بکارگیری بلوک Resource Estimator از همین نرم‌افزار است.

⁵ Constant multiplier

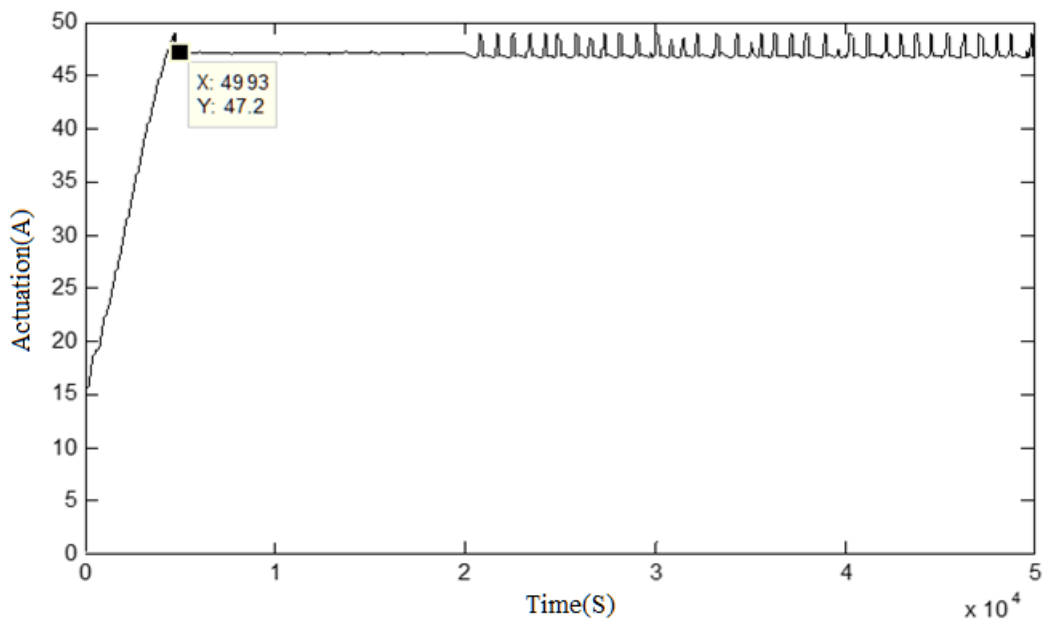
⁶ ADDSUB



شکل ۴-۱۷: ساختار بکار رفته برای کنترل کننده تناسبی انتگرالی در محیط System Generator



(الف)



(ب)

شکل ۴-۱۸: (الف) سیگنال خطا و (ب) سیگنال تحریک در کنترل کننده ضد اشباع حوزه دیجیتال

سیگنال تحریک و سیگنال خطا طبق این شبیه‌سازی به صورت شکل ۴-۱۸ خواهد شد. در کمتر از ۱ ثانیه سیگنال خطا به صفر میل نموده و سیگنال تحریک به مقدار نهایی خود می‌رسد. این مساله صحت پیاده‌سازی را نشان می‌دهد.

۴-۳-۲ مدولاسیون پهناي پالس در حوزه دیجیتال

واحد مدولاسیون پهناي پالس در حوزه دیجیتال خود از سه بخش اصلی تشکیل شده است.

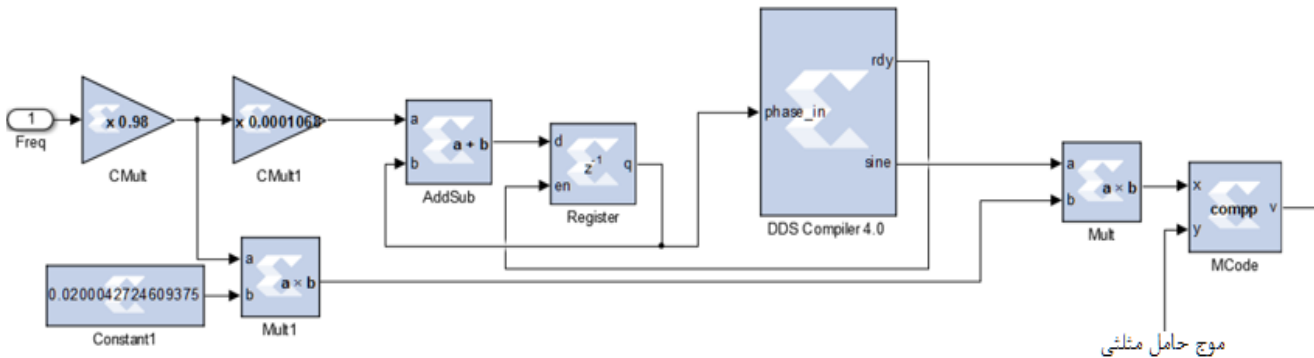
- شبیه‌سازی موج سینوسی
- تولید موج مثلثی
- مقایسه کننده

اگر سیگنال‌های حاصل از دو بخش اول در حوزه دیجیتال به خوبی ساخته شوند پالس مدوله شده حاصل از مقایسه آن‌ها در بخش سوم نیز سرعت موتور را به خوبی کنترل می‌کند.

الف) تولید موج سینوسی در حوزه دیجیتال

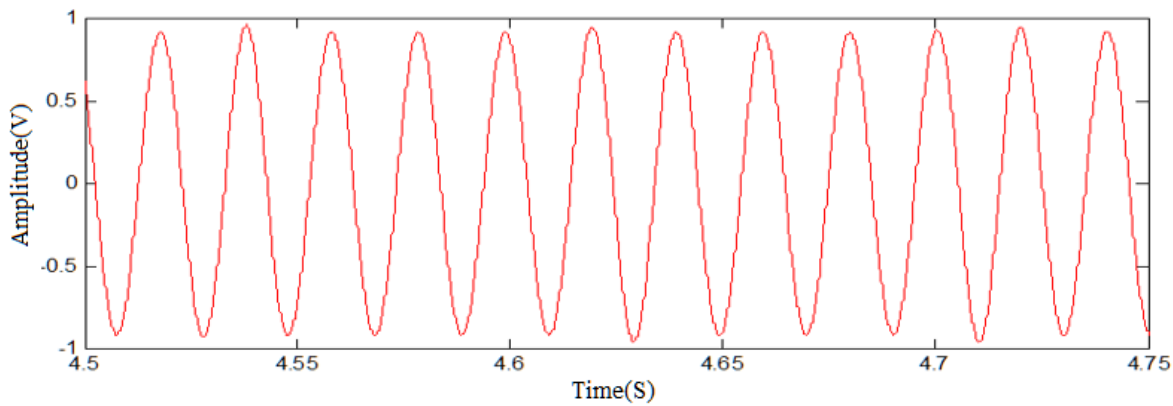
فرکانس از خروجی کنترل کننده وارد بلوک SPWM می‌شود. برای جبران تبدیل فرکانس از حوزه‌ی آنالوگ به دیجیتال ورودی ابتدا در یک ضریب ضرب می‌شود. عمل انتگرال فرکانس ورودی که فاز شکل موج‌های سینوسی مورد نیاز را می‌سازد، توسط دو بلوک کنار هم ثبات و جمع کننده صورت می‌پذیرد. بلوک جمع کننده داده را با داده‌ی قبلی که در ثبات ذخیره شده است جمع می‌کند. خروجی انتگرالگیر به ورودی بلوک کامپایلر می‌رود. وظیفه‌ی بلوک کامپایلر آن است که سینوس فازی که به ورودی آن داده می‌شود را به صورت لحظه‌ای حساب کند. در خروجی کامپایلر یک شمارنده‌ای نیز وجود دارد که مادامی که ۱ باشد سینوس را محاسبه می‌کند و هرگاه بازه‌ی انتگرالگیر به اتمام رسید شمارنده را صفر می‌کند و مجدداً عمل محاسبه‌ی سینوس انجام می‌گردد و در واقع این مقادیر سینوس لحظه‌ای کنار هم قرار می‌گیرند و شکل موج سینوسی کامل در حوزه دیجیتال تولید می‌شود. جهت تولید موج سینوسی در دو فاز دیگر مقادیر $0/3332$ و $0/6667$ را به ترتیب برای فاز ۱۲۰ و ۲۴۰ درجه به مقدار فاز ورودی کامپایلر اضافه می‌کنیم. برای تولید دامنه‌ی موج سینوسی نیز نرم فرکانس (فرکانس $\times 0/02$) در انتها بعد از تولید موج سینوسی اعمال خواهد شد. با اعمال فرآیند گفته شده در نرم‌افزار System Generator ساختاری

همانند شکل ۴-۱۹ در محیط نرم‌افزاری بوجود می‌آید که وظیفه‌ی تولید موج سینوسی در SPWM را خواهد داشت.



شکل ۴-۱۹: ساختار بکار رفته جهت ایجاد موج سینوسی در محیط System generator

شکل موج سینوسی تولید شده توسط این ساختار به صورت شکل ۴-۲۰ بدست می‌آید که مشابه سینوسی در حوزه آنالوگ است.

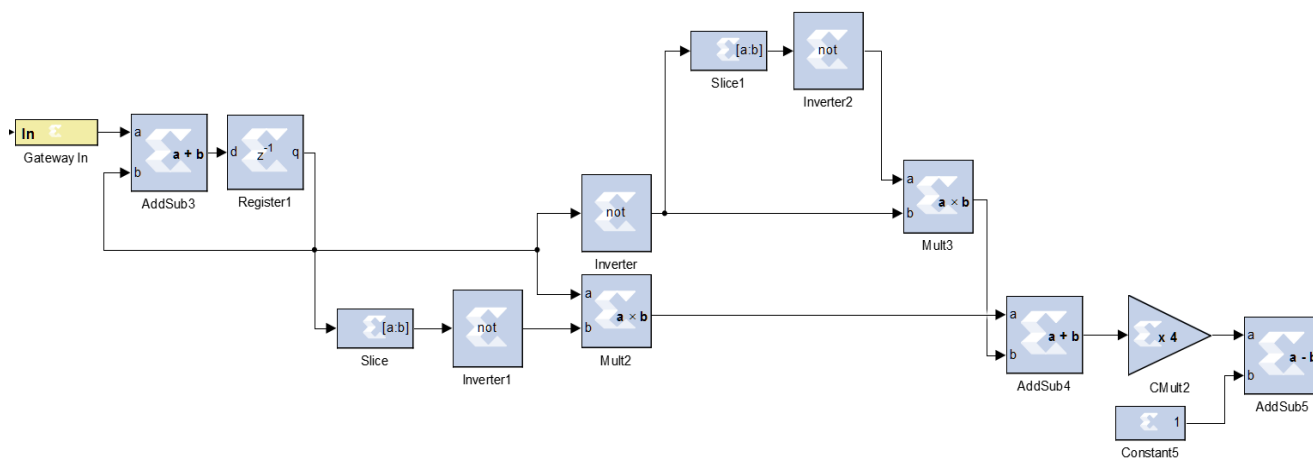


شکل ۴-۲۰: موج سینوسی ایجاد شده توسط ساختار محیط System generator

ب) تولید موج حامل مثلثی در حوزه دیجیتال

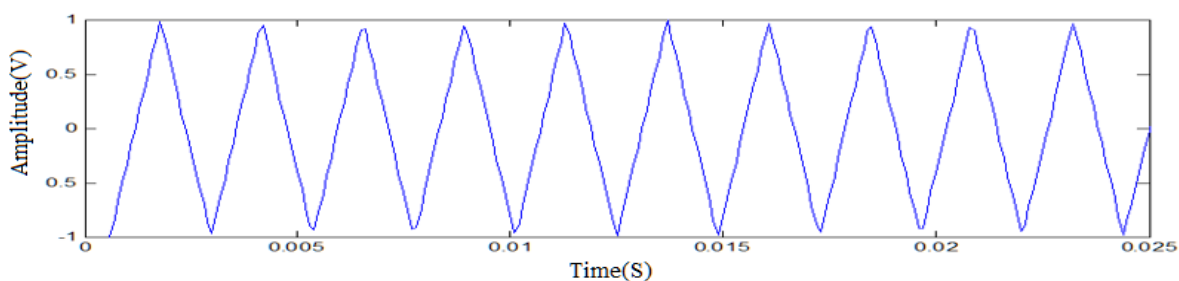
طراحی مدار به گونه ای است که یک مقدار اولیه به صورت افزایشی با خودش جمع می‌شود. بیت‌ها به صورت واحد یکی یکی خارج شده و معکوس می‌شوند. زمانی که موج مثلثی به مقدار نهایی خود یعنی ۱ دیجیتال رسید بیت معکوس شده صفر می‌شوند و ضرب کننده آن غیر فعال می‌شود. سپس ضرب

کننده مسیر دیگر فعال می‌شود. بیت‌های افزایشده از مسیر دوم که معکوس مسیر اول است عبور کرده و به صورت آینه‌ای شروع به کاهش می‌کند. به این ترتیب موج مثلثی ساخته می‌شود. چون دامنه از ۰ تا ۰/۵ است در یک ضریب ۴ ضرب می‌شود و سپس از ۱ تفریق می‌گردد. در این صورت دامنه موج مثلثی به بازه‌ی ۱- تا ۱ منتقل می‌شود. با اعمال فرآیند گفته شده در نرم‌افزار System Generator ساختاری همانند شکل ۴-۲۱ در محیط نرم‌افزاری بوجود خواهد آمد که وظیفه‌ی تولید موج حامل مثلثی در SPWM را خواهد داشت.



شکل ۴-۲۱: ساختار تولید شکل موج حامل مثلثی در محیط System generator

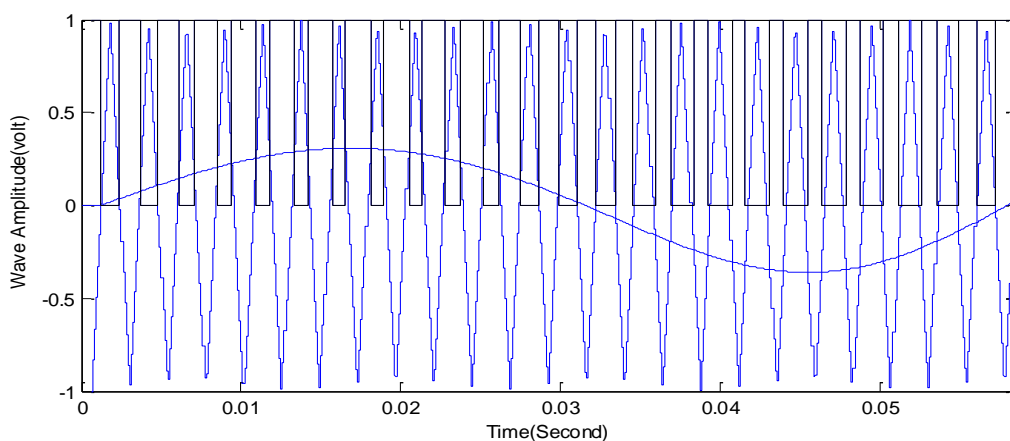
شکل موج مثلثی تولید شده توسط این ساختار به صورت شکل ۴-۲۲ خواهد بود. همانطور که مشاهده می‌شود شکل موج مناسب و قابل قبولی می‌باشد.



شکل ۴-۲۲: موج حامل مثلثی تولید شده توسط ساختار در محیط System generator

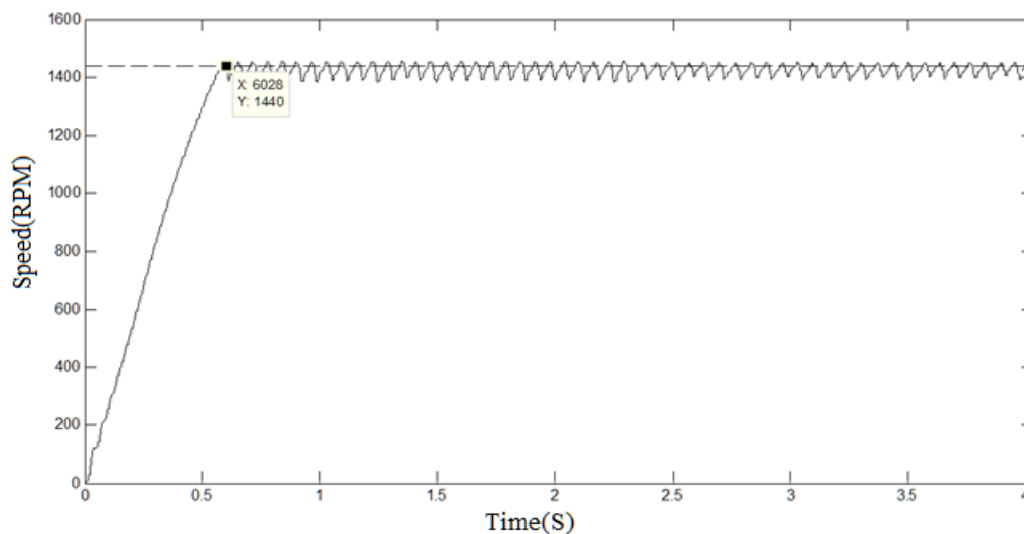
ج) مقایسه کننده

همانطور که در بخش قبل به آن اشاره شد در System Generator بلوکی به نام MCODE قرار دارد که به کاربر اجازه می‌دهد از متلب سیمولینک برای مدلسازی و پیاده‌سازی عملیات کنترلی ساده استفاده کند. در اینجا برای ساخت مقایسه کننده از MCODE استفاده می‌شود که می‌توان یک دستور ساده‌ی مقایسه برای کد آن نوشت و دستور را به تابع آن ارجاع داد. سیگنال‌های ورودی به این بخش سیگنال سینوسی و موج مثلثی می‌باشند.



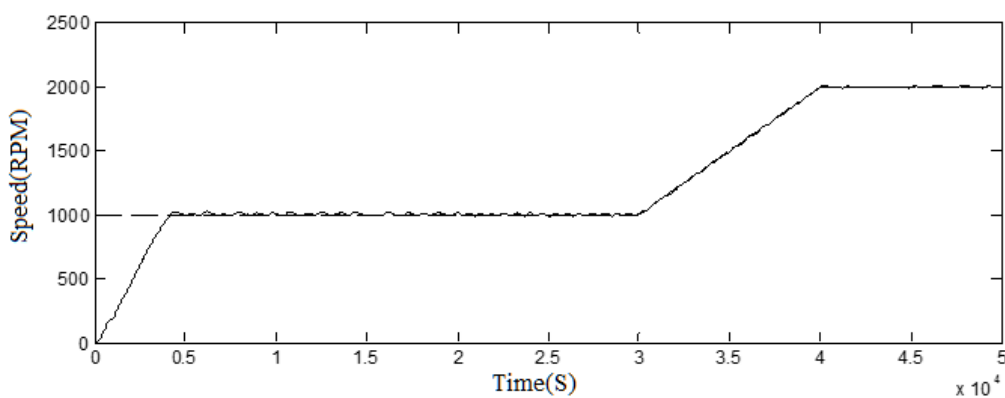
شکل ۴-۲۳: سیگنال‌های ورودی به بخش مقایسه کننده در محیط System generator

سیگنال‌های تولیدی در بخش‌های قبلی با ورود به این مقایسه کننده تشکیل ساختار مقایسه‌ای را داده و خروجی به شکل پالس مربعی همانند شکل ۴-۲۳ تولید خواهد شد. این پالس مربعی ورودی اینورتر سه‌فاز سیستم خواهد بود که موجب تغییر در اندازه‌ی ولتاژ پایانه‌های ورودی موتور خواهد شد. سرعت را به راحتی توسط این پالس مدوله شده کنترل و به مقدار مرجع نزدیک خواهد شد. طبق آنچه که هدف پایان نامه از ابتدا بوده است خروجی سرعت ورودی را دنبال می‌کند. شکل ۴-۲۴ گویای این مساله است. همانطور که مشاهده می‌شود سرعت در کمتر از 0.7 ثانیه به سرعت نهایی که با خط‌چین مشخص شده رسیده است.



شکل ۴-۲۴: پاسخ خروجی سرعت به ازای ورودی مرجع مشخص با شبیه‌سازی در FPGA

در حالت پایدار خروجی سرعت در شکل ۴-۲۴ بالا و پایین‌هایی دیده می‌شود. علت آن است که در حوزه دیجیتال اعداد اصلاحی برای کنترل سرعت در هر لحظه را بلوک‌های دیجیتال رند نموده و رفتار پیوسته‌ای ندارد. این عدم پیوستگی مشکلی ایجاد نمی‌کند و با توجه به خروجی موتور که دنبال کننده سرعت مرجع است، موتور آن را حس نخواهد کرد. در اینجا نیز می‌توان آرایه به خصوصی از سرعت را به مدار داد و نحوه‌ی دنبال کردن خروجی را بررسی کرد. طبق شکل ۴-۲۵ دنبال کردن سرعت واقعی نسبت به سرعت مرجع رضایت‌بخش است. می‌توان نتیجه گرفت که طراحی صحیحی صورت گرفته است.



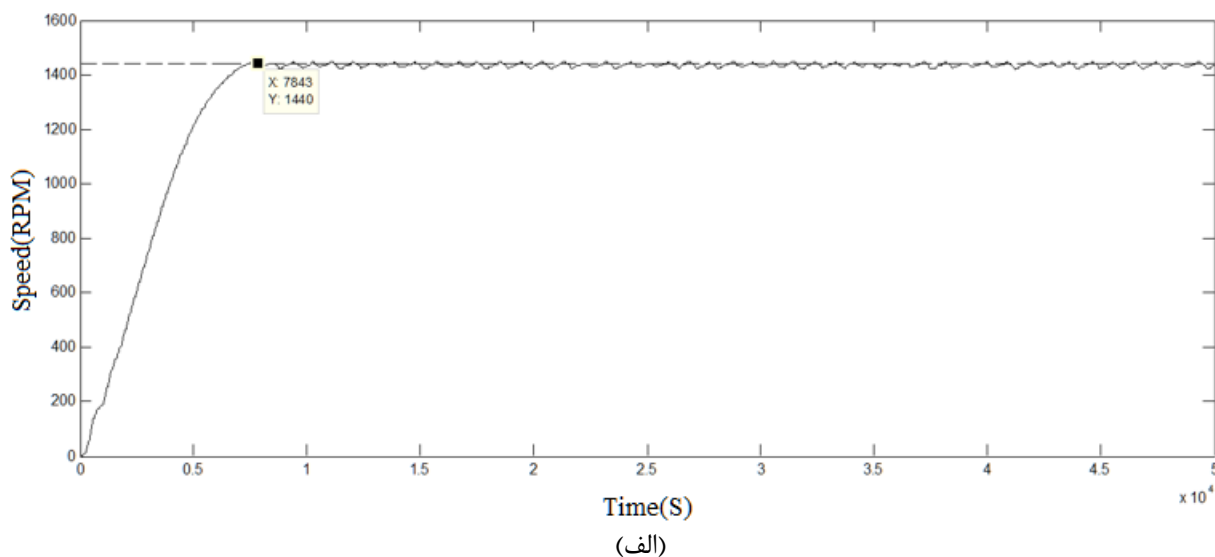
شکل ۴-۲۵: سرعت موتور بر اساس look up table ورودی مشخص در پیاده‌سازی دیجیتال

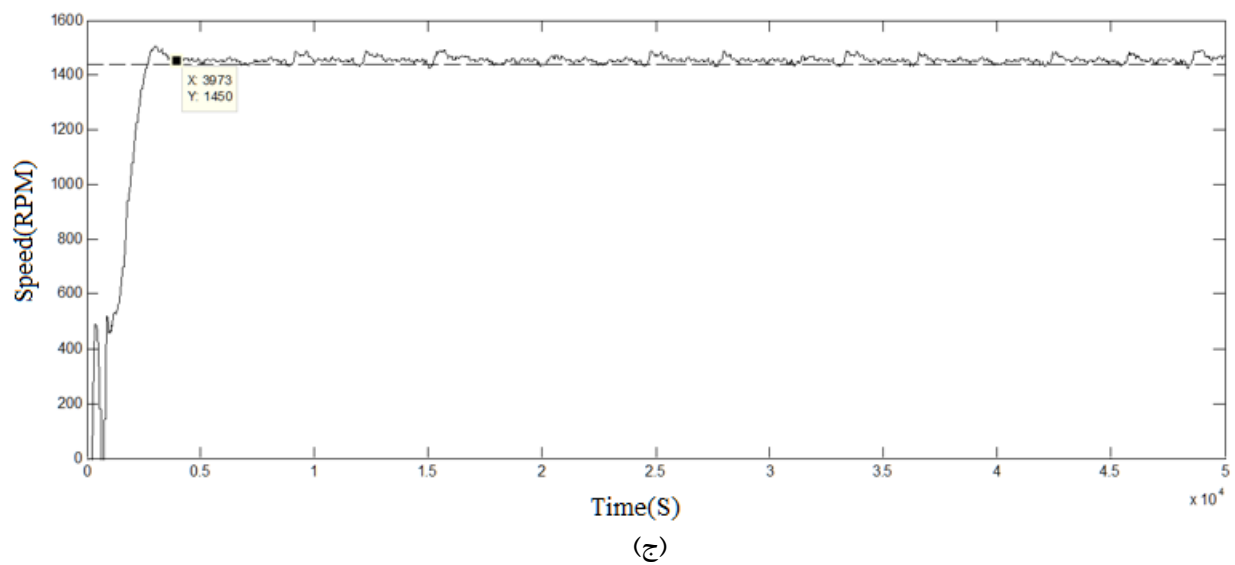
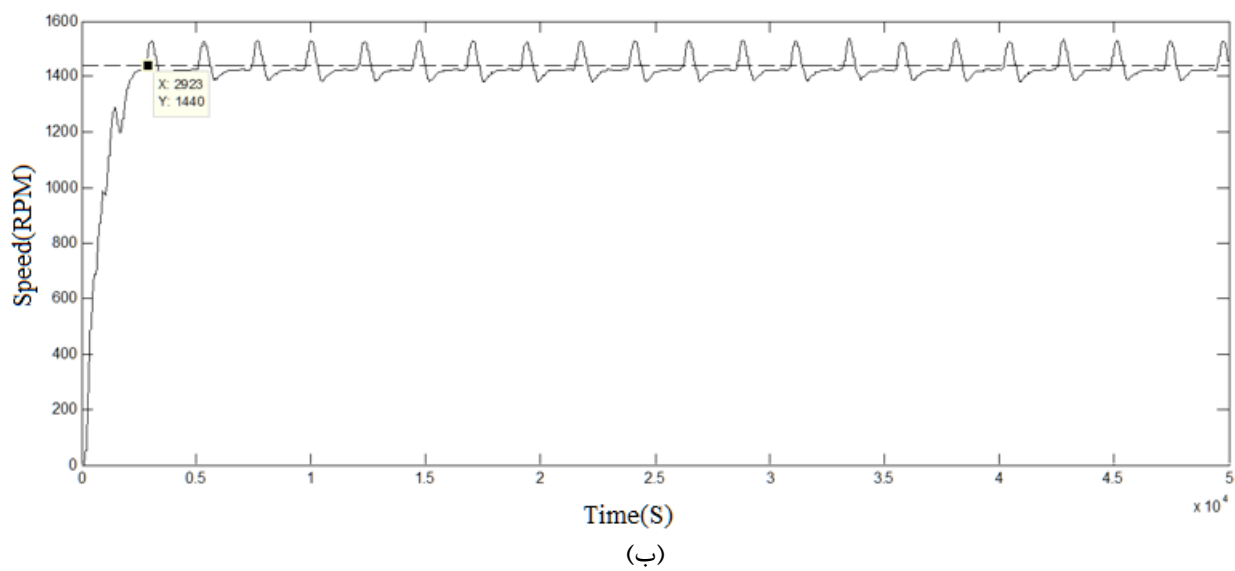
در طول شبیه‌سازی در حوزه دیجیتال دو پرسش مطرح می‌شود.

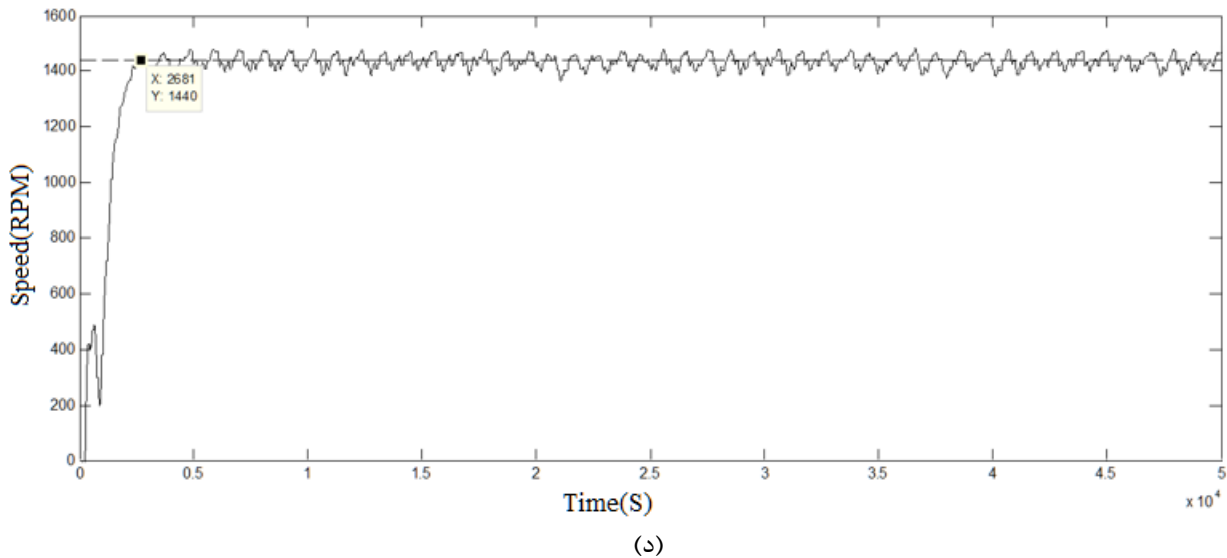
- پریود نمونه برداری بلوک‌های ورودی FPGA که فرمت داده‌ها را از آنالوگ به دیجیتال تبدیل می‌کنند چگونه باید باشد؟
- هر بلوک از شبیه‌ساز System Generator ورودی‌اش را با چه سطح بیتی دریافت نماید؟

در خصوص پرسش اول بایستی خاطر نشان کرد نه تنها پریود نمونه برداری درگاه ورودی FPGA نمی‌تواند از پریود نمونه برداری کل مدار کوچکتر باشد بلکه باید مضرب صحیحی از آن نیز باشد. علاوه بر این باید به این مساله توجه داشت که هرچه پریود نمونه برداری کوچکتر باشد یا به عبارت دیگر فرکانس نمونه برداری بیشتر باشد، FPGA در فواصل زمانی کمتر و با دقت بیشتر از سیگنال آنالوگ ورودی‌اش نمونه می‌گیرد. در این حالت گسستگی سیگنال دیجیتال نرم‌تر اتفاق می‌افتد.

شکل ۴-۲۶ پاسخ سرعت خروجی را به ازای مقادیر متفاوت پریود نمونه برداری نشان می‌دهد. گسستگی‌ها در مقادیر بالا پریود نمونه برداری قابل مشاهده است. به همین دلیل هم در ابتدای منحنی نوسانات خروجی بیشتر است. با توجه به آنچه که گفته شد در این طراحی پریود نمونه برداری در حوزه دیجیتال مساوی نرخ نمونه برداری کل مدار یعنی $1/0000$ انتخاب شده است.



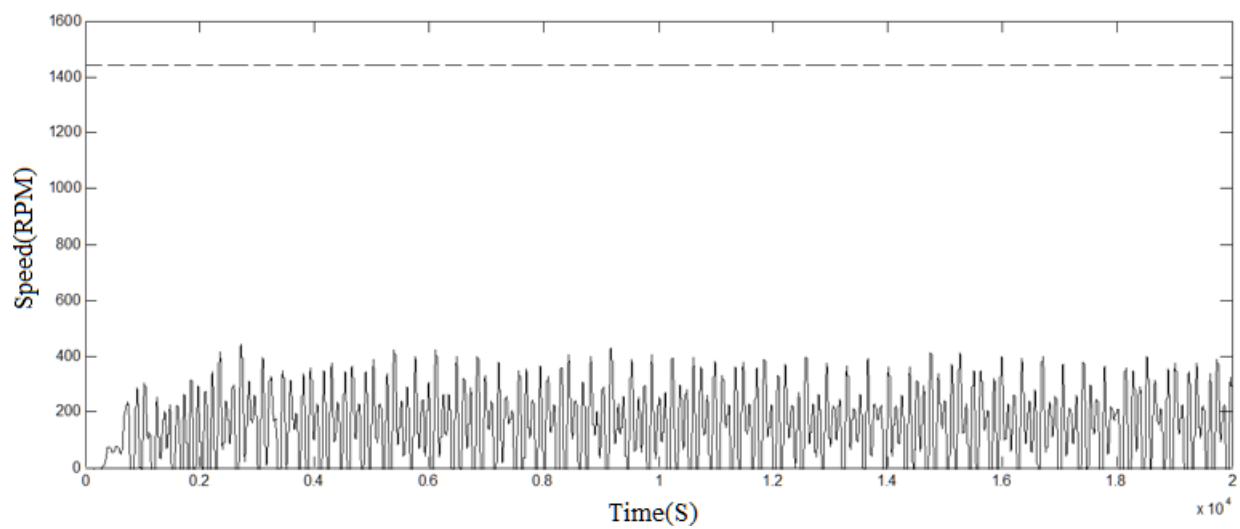




شکل ۴-۲۶: خروجی سرعت در مقادیر متفاوت پریود نمونه برداری الف) $T=0.0001$ و ب) $T=0.0003$ و ج) $T=0.0007$ و د) $T=0.001$

در مورد سوال دوم طراح باید توجه داشته باشد که تعداد بیت‌های هر بلوک دیجیتال دقت مدار را تعیین می‌کند. اگر تعداد بیت‌ها کم باشد دقت داده‌های ورودی هر بلوک کم شده و پاسخ خروجی به مقدار نهایی نخواهد رسید. در این صورت برای کنترل صحیح سرعت موتور تعداد بیت را باید افزایش داد. حداقل تعداد بیت درگاه ورودی که توسط آن مدار به خوبی پاسخ می‌دهد در این طراحی ۱۴ بیت می‌باشد.

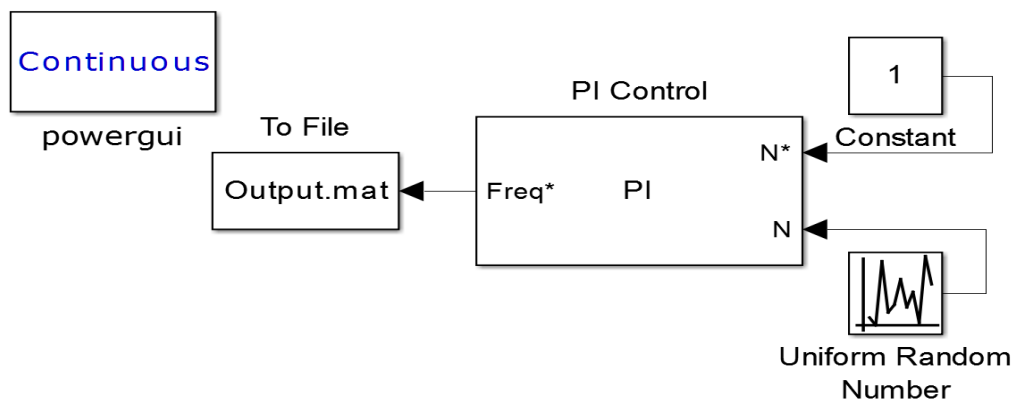
شکل ۴-۲۷ خروجی را به ازای تعداد بیت کمتر از ۱۴ درگاه ورودی نشان می‌دهد. همانطور که مشخص است به دلیل تعداد بیت کم مقدار سرعت خروجی در سمت کنترل کننده درست دریافت نمی‌شود.



شکل ۴-۲۷: پاسخ خروجی سرعت به ازای مقدار پایین بیت درگاه ورودی

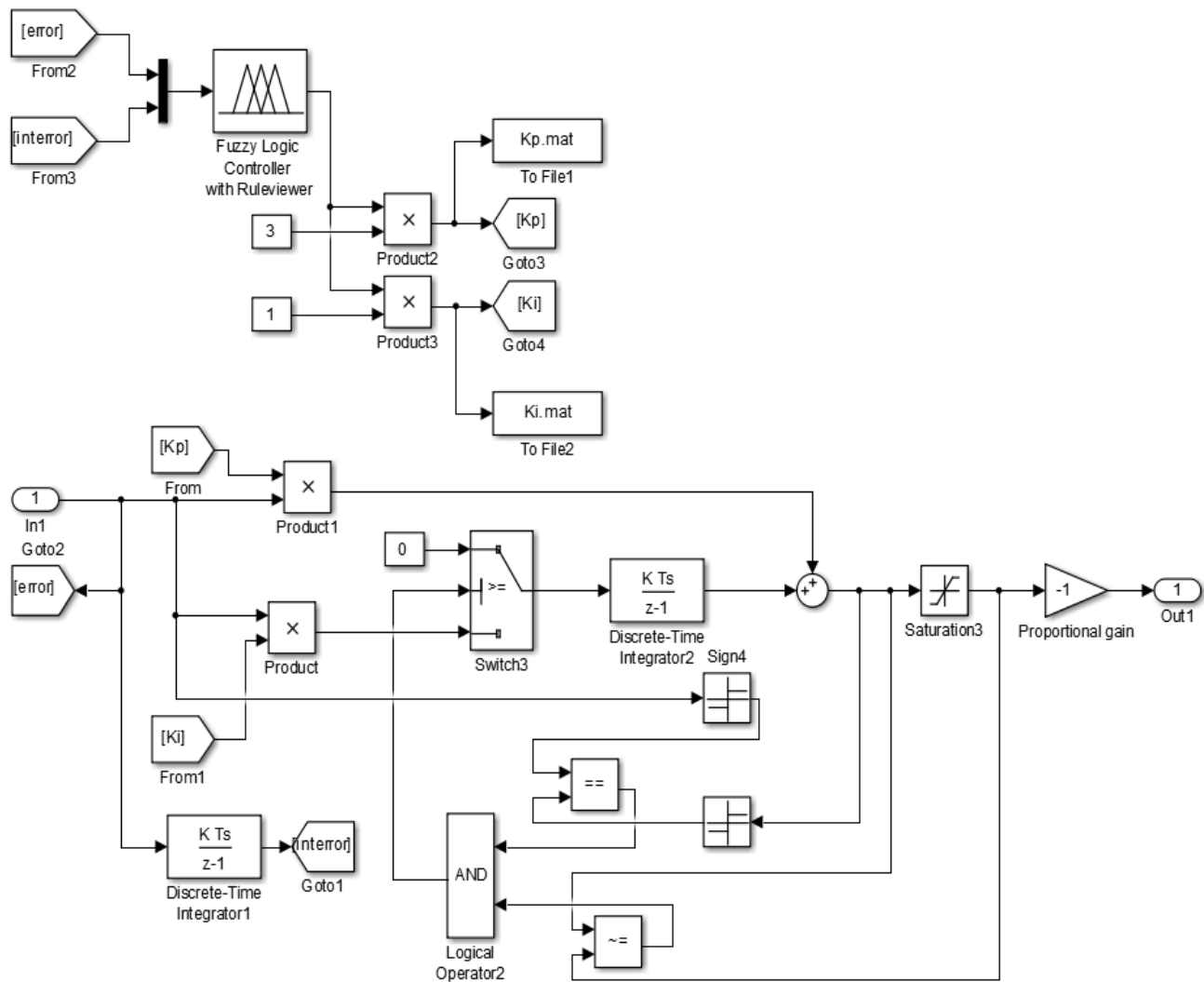
۴-۴ استفاده از کنترل فازی برای بهبود عملکرد مدار

در خصوص شبیه‌سازی در دو بخش قبل باید گفت که به دلیل آنکه ضرایب با روش پاسخ فرآیند انتخاب شده اند و این روش مشابه روش سعی و خطا است، تضمینی بر دقت بودن پاسخ نهایی و عملکرد صحیح مدار کنترلی نیست. علاوه بر این در هر نقطه کاری سیستم یک ضریب مناسب است. بنابراین نیاز به یک روش صریح برای تعیین ضرایب احساس می‌شود. جهت تعیین دقیق‌تر ضرایب می‌توان از کنترل کننده فازی بهره گرفت تا در هر نقطه کاری بهره مناسب را به سیستم اعمال نماید. برای این کار لازم است ابتدا رفتار کنترل کننده PI شناسایی شود. بدین صورت که ابتدا یک ورودی تصادفی بین ۱ و -۱ و یک ورودی ثابت ۱ به کنترل کننده داده می‌شود. خطا و انتگرال خطا و خروجی آن ذخیره می‌شود. علت استفاده از این ورودی‌ها آن است که جهت طراحی توابع فازی داده‌ها غنی باشند، تا گوشه‌های سیستم را به خوبی شناسایی نمایند. شکل ۴-۲۸ نحوه استخراج این داده‌ها را نشان می‌دهد.



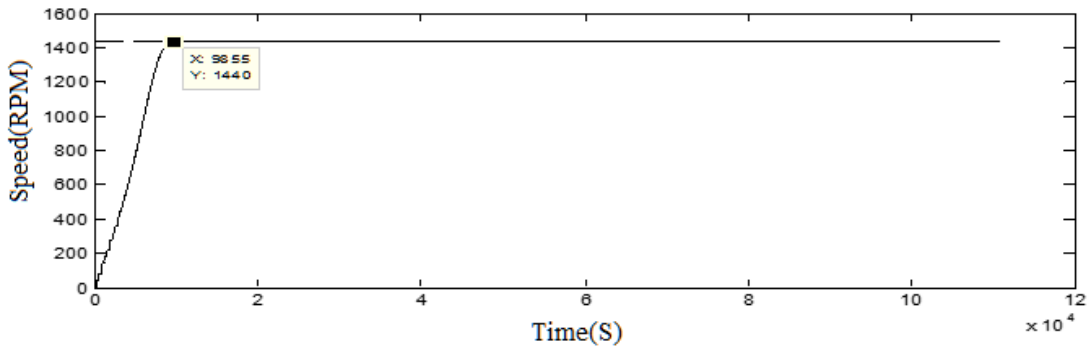
شکل ۴-۲۸: استفاده از داده‌های غنی جهت بدست آوردن خروجی مناسب برای تعیین توابع فازی سیستم

حال با استفاده از ورودی‌های اصلی کنترل کننده یعنی خطا و انتگرال خطا و خروجی که از بالا بدست آمده کنترل کننده فازی طراحی می‌شود. جهت تعیین توابع فازی با استفاده از این ورودی‌ها یک شبکه عصبی آموزش داده می‌شود. شبکه عصبی آموزش داده شده از روش کاهش گرادیان و تخمین حداقل مربعات خطا توابع فازی را تعیین می‌کند. توابع فازی تعیین شده استخراج شده و در بلوک آماده کنترل کننده فازی متلب قرار داده می‌شود. بدین ترتیب ضرایب کنترل کننده با استفاده از ساختار شکل ۴-۲۹ به صورت آفلاین تعیین می‌شوند و به کنترل کننده داده می‌شوند.



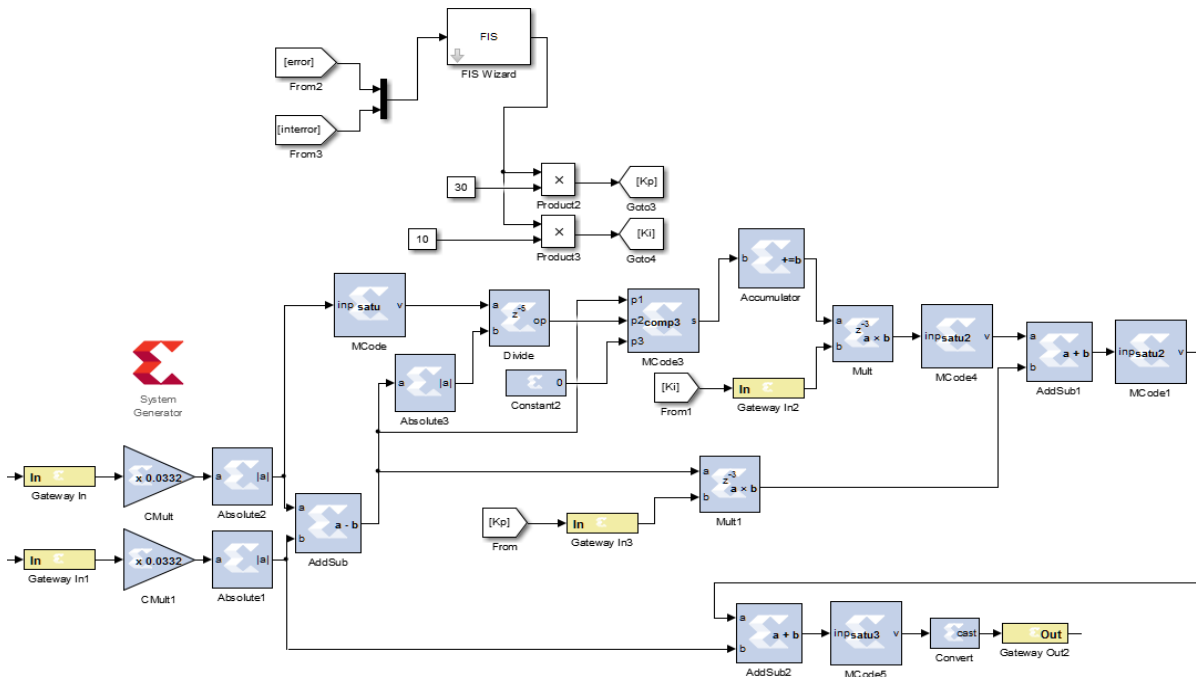
شکل ۴-۲۹: ساختار کنترل فازی و کنترل کننده PI جهت تعیین و اعمال ضرایب

این روش تعیین ضرایب مطمئن تر از روش قبلی است و پاسخ خروجی سریع تری دارد. علاوه بر این در هر نقطه کاری ضرایب مناسب همان نقطه را به کنترل کننده اعمال می کند. همانطور که در شکل ۴-۳۰ مشاهده می شود خروجی در کمتر از ۰/۹ ثانیه به پاسخ نهایی می رسد.



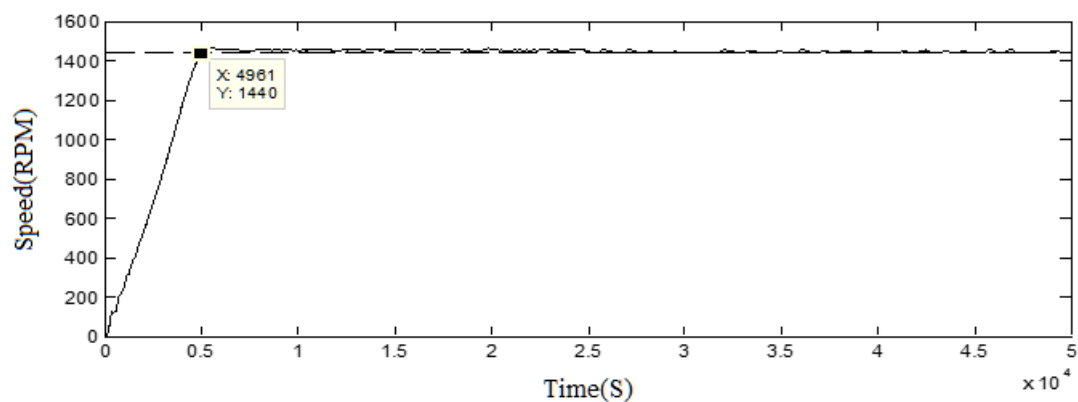
شکل ۴-۳۰: پاسخ بهتر خروجی با ضرایب تعیین شده با کنترل کننده فازی

می‌توان به صورت آفلاین ضرایب را در شبیه‌ساز متلب تعیین نمود و سپس ضرایب تعیین شده را به کنترل کننده تناسبی انتگرالی با یک بلوک ضرب کننده در حوزه دیجیتال وارد کرد. در این حالت نیاز به آموزش مجدد شبکه با بلوک‌های دیجیتال نیست. چون کنترل کننده قادر است خود را با شرایط جدید تطبیق دهد. شکل ۴-۳۱ نحوه‌ی اعمال این ضرایب به کنترل کننده دیجیتالی را نشان داد.



شکل ۴-۳۱: اعمال ضرایب تعیین شده با روش فازی به کنترل کننده PI در محیط System Generator

پاسخ خروجی را می‌توان در شکل ۳۲-۴ مشاهده نمود. خروجی در کمتر از ۰/۴ ثانیه به پاسخ نهایی می‌رسد. علت اینکه نسبت به حالت آنالوگ خروجی تاخیر بیشتری دارد آن است که بلوک‌های FPGA خود کمی تاخیر به سیستم وارد می‌نمایند.



شکل ۳۲-۴: پاسخ بهتر خروجی با ضرایب تعیین شده با کنترل کننده فازی در حوزه دیجیتال

فصل پنجم

جمع‌بندی، نتیجه‌گیری و پیشنهادات

۵-۱ نتایج شبیه‌سازی و پیشنهادات

در این پایان نامه با طراحی مدار کنترل کننده‌ی PI در محیط دیجیتال و با استفاده از مدارات FPGA سعی در بهبود عملکرد کنترلی موتور القایی سه‌فاز بوده و کاهش قابل ملاحظه‌ای در اندازه‌ی مدار رخ داده است که در عمل موجب یکپارچگی سیگنال‌ها در مدار و افزایش سرعت عملکرد مدار می‌باشد. همچنین قیمت تمام شده برای اینگونه مدارات پایین‌تر بوده و به دلیل عدم توانایی در کپی کردن مدار، امنیت آن افزایش می‌یابد. البته کلیه مزایای فوق در قبال افزایش توان مصرفی سیستم با استفاده از این مدارات می‌باشد.

برای شبیه‌سازی از محیط System Generator استفاده شده که هر چند می‌توان در هر زمینه‌ای در طراحی بر اساس FPGA از آن استفاده کرد ولی مهم‌ترین استفاده System Generator در کارهای پردازشی می‌باشد. System Generator با ساده کردن استفاده از بلوک‌های پردازشی و انواع داده ممیز ثابت و ممیز شناور و ساده شدن شبیه‌سازی با کمک گرفتن از امکانات سیمولینک، گزینه‌ای ایده‌آل برای انجام کارهای پردازشی می‌باشد، طوری که کاری که با کدنویسی با سختی‌های فراوان از جمله کار با انواع داده و مشکلات تبدیل نوع، سختی شبیه‌سازی و طولانی بودن کار کدنویسی همراه بوده است، در محیط System Generator به راحتی امکان پذیر می‌باشد. با توجه به شکل‌های بدست آمده در فصل قبل می‌توان دید این سیستم موفق به پایدارسازی سرعت موتور بر اساس میزان مرجع شده است.

به عنوان پیشنهاد برای تحقیقات آتی می‌توان تمهیدات زیر را در نظر گرفت تا نتایج بهتری حاصل شود.

- به جای استفاده از ضرایب ثابت جهت کنترلر PI می‌توان از روش‌های اتوماتیک یادگیری تقویتی- ترکیبی پارامترهای سیستم کنترل سرعت موتور القایی استفاده و این پارامترها را به شکلی بهینه تعیین کرد.
- انواع دیگری از کنترل کننده‌ها مانند کنترل کننده جهت‌یابی میدان^۱ می‌توانند طراحی شوند تا عملکرد مدار بهبود یابد.

^۱ Field Oriented Control (FOC)

- روش‌های دیگر مدولاسیون مانند مدولاسیون فضای برداری در حوزه آنالوگ و دیجیتال پیاده شوند.
- با استفاده از روش‌های پردازش سیگنال دیجیتال اثرات هارمونیک موج سینوسی ایجاد شده و روش انتخابی جهت کاهش اعوجاج بهبود یابد.

پیوست ها

پیوست (الف): معرفی شبکه‌های عصبی

شبکه‌های عصبی که در دهه ۱۹۸۰ میلادی مطرح شدند، حاصل بهم پیوستن گره‌های اجراکننده توابع گره‌ای از طریق اتصالات^۱ هستند. این ساختارها از طریق الگوریتم‌ها و روش‌های مختلف آموزشی قادر به شبیه‌سازی و شناسایی الگوها و سیستم‌ها هستند. دو ویژگی مهم شبکه‌های عصبی قابلیت یادگیری و تنوع ساختار است. یادگیری شبکه‌های عصبی می‌تواند به صورت بدون ناظر، دارای ناظر و یا ترکیبی از این دو صورت گیرد. در یادگیری بدون ناظر، صرفاً داده‌های موجود به کار گرفته می‌شوند درحالیکه در یادگیری دارای ناظر دانش و تجربه انسان در موضوع مورد نظر در یادگیری دخالت می‌کند. غالباً یادگیری بدون ناظر برای دسته‌بندی داده‌ها و یادگیری دارای ناظر برای تنظیم قوانین و توابع تعلق به کار گرفته می‌شود. روش پس انتشار خطا^۲ که در آموزش بسیاری از شبکه‌های عصبی به کار گرفته می‌شود، یک روش یادگیری دارای ناظر محسوب می‌شود [۴۰].

شبکه‌های عصبی مصنوعی شامل تعدادی لایه متشکل از نرون‌ها و وزن‌های آن‌ها هستند که اطلاعات مربوط به نگاشت ورودی-خروجی در وزن‌ها ذخیره می‌شود. ویژگی این شبکه‌ها تعمیم‌دهی نتایج حاصل از تعداد معدودی از داده‌هاست، به نحوی که بتوانند سیستم‌های پیچیده را مدل کنند [۴۱].

تفاوت اصلی سیستم‌های فازی و شبکه‌های عصبی در این است که تشکیل و تنظیم یک شبکه عصبی برای مدل‌سازی سیستم‌ها به میزان بسیار کمتری به دانش انسان و شناخت و تجربه او پیرامون سیستم اولیه نیاز دارد. در حالیکه در مدل‌سازی فازی، هرچه این شناخت کامل‌تر شود مدل دقیق‌تر خواهد شد [۴۰]. شبکه‌های عصبی انواع مختلفی دارند که از جمله آن‌ها می‌توان به شبکه‌های عصبی چندلایه و شبکه RBF^۳ اشاره نمود.

¹ Links

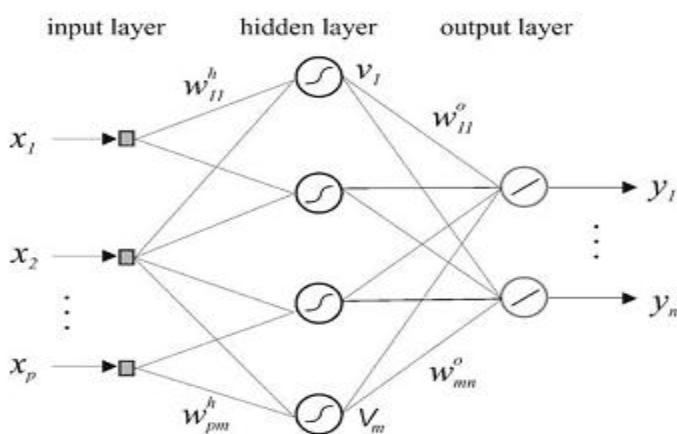
² Error Back Propagation

¹ Radial Basis Function Network

شبکه عصبی چندلایه دارای یک لایه ورودی، یک لایه خروجی و چند لایه پنهان است. لایه اول محاسبات خاصی را انجام نداده و فقط گاهی داده‌ها را وزن‌دهی می‌کند. سپس تابعی غیرخطی روی داده‌های وزن‌دار اثر می‌گذارد. در شکل پیوست (الف)-۱ نمونه‌ای از شبکه‌های عصبی چندلایه نشان داده شده است. شبکه RBF شبکه‌ای دولایه است که به صورت رابطه (الف)-۱ نمایش داده می‌شود. تابع گره-ای معمولاً به صورت گاوسی در نظر گرفته می‌شود و وزن‌ها با روش حداقل مربعات به دست می‌آیند [۴۰].

$$y = f(x) = \sum_{i=1}^m \omega_i \phi_i(x) \quad (\text{الف}_1)$$

در این رابطه ω_i نشان‌دهنده وزن اتصالات شبکه و $\phi_i(x)$ نشان‌دهنده توابع گره‌ای شبکه است. در شبکه‌های عصبی به طور کلی افزایش تعداد لایه‌های پنهان گرچه نتایج را دقیق‌تر می‌کند اما روند یادگیری را سخت‌تر و طولانی‌تر می‌کند. در مورد تعداد نرون‌ها نیز این مطلب صادق است که اگر تعداد نرون‌ها کم باشد آموزش با مشکل مواجه می‌شود در حالیکه افزایش بیش از حد نرون‌ها عملاً سیستم را به سمت حفظ کردن^۱ داده‌ها سوق می‌دهد [۴۰].



شکل پیوست (الف)-۱: نمونه‌ای از شبکه‌های عصبی چندلایه [۴۰]

² Over Learning

پیوست (ب): منطق و کنترل فازی

منطق فازی شیوه‌ای جدید در پیاده‌سازی و هوشمند نمودن ماشین‌ها، روبات‌های صنعتی و کنترل-کننده‌ها است که بر اساس مدلسازی تصمیم‌گیری در مغز انسان عمل می‌کند. با ظهور این شیوه، انقلاب جدیدی در کنترل کننده‌های هوشمند بوجود آمد. طی دهه ۱۹۸۰ منطق فازی در موارد متنوعی از جمله کنترل فرآیندهای صنعتی، تصمیم‌گیری فازی و کنترل سیستم‌های غیرخطی و متغیر با زمان به کار گرفته شد. در سیستم‌های مبتنی بر منطق فازی تصمیم‌گیری براساس ورودی‌ها، برچسب‌های زبانی و درجه تعلق ورودی‌ها به مجموعه‌های فازی انجام می‌شود. در این سیستم‌ها هرچه توابع تعلق و قوانین فازی دقیق‌تر تنظیم شوند، عملکرد سیستم مطلوب‌تر خواهد بود [۴۰].

تاکاگی و سوگینو در سال ۱۹۸۵ مدل‌سازی و شناسایی فازی را مورد بررسی قرار دادند. پس از آن مدل‌سازی فازی کاربردهای عملی زیادی در زمینه کنترل، پیش‌بینی، استدلال و استنباط به دست آورد. قوانین فازی عباراتی به صورت "اگر A آنگاه B" هستند که به دلیل فرم فشرده و مختصری که دارند، غالباً برای بیان حالات غیردقیق استدلالی به کار می‌روند. به عنوان مثال: اگر دما بالا باشد آنگاه حجم کم است. در این مثال فشار و حجم متغیرهای زبانی هستند که کمیت آن‌ها به جای اعداد دقیق با برچسب-های زبانی مثل بالا و کم توصیف شده است [۴۲].

قوانین فازی به دو فرم کلی ممدانی و تاکاگی-سوگینو طبقه‌بندی می‌شوند. در قوانین ممدانی بخش موخره قانون، به صورت یک عبارت زبانی^۱ و در فرم تاکاگی-سوگینو به صورت یک عبارت عددی قاطع^۲ یا تابع عددی خواهد بود. هر دو نوع این عبارت‌ها برای مدل‌سازی و کنترل به کار می‌روند. در جدول (ب)-۱ مقایسه‌ای بین دو فرم کلی قوانین فازی ارائه شده است [۴۰].

جدول (ب)-۱: مقایسه مدل ممدانی و TS

مزایای مدل ممدانی	مزایای مدل TS
پایه‌سازی راحت‌تر	شفافیت بیشتر
فهم راحت‌تر	دقت بالاتر

¹ Linguistic

² Crisp

سیستم‌های فازی که پایه و اساس آن‌ها مجموعه قواعد فازی است، از پنج بخش تشکیل شده‌اند. اجزاء سازنده سیستم‌های فازی عبارتند از:

- مجموعه قواعد فازی^۱: این بخش دربرگیرنده مجموعه قوانین فازی به کار رفته در سیستم است.
- پایگاه داده^۲: این بخش دربرگیرنده توابع تعلق به کار رفته در قوانین فازی است.
- واحد تصمیم‌ساز^۳: این قسمت وظیفه انجام عملیات استنباطی و استنتاجی بر روی قوانین و متغیرهای فازی سیستم را بر عهده دارد.
- بخش فازی‌ساز^۴: این بخش وظیفه تبدیل ورودی قاطع و دقیق به متغیر فازی قابل تطبیق با قوانین را دارد.
- واسط خروجی: این قسمت که با عنوان غیرفازی‌ساز^۵ مطرح می‌شود، وظیفه تبدیل نتایج و خروجی‌های فازی سیستم را به خروجی قاطع بر عهده دارد.

قسمت‌های مختلف یک سیستم‌فازی در شکل پیوست (ب)-۱ نشان داده شده‌اند. استدلال فازی در این سیستم‌ها با مقایسه متغیرهای ورودی با توابع تعلق آغاز می‌شود. ضمن این مقایسه میزان سازگاری ورودی با مقدمه هریک از قوانین تعیین شده و بر این اساس وزن اختصاصی هر قانون تعیین می‌شود. سپس براساس موخره قوانین و وزن اختصاص یافته به آن‌ها خروجی قاطع تولید خواهد شد [۴۲].

در اینجا ذکر چند نکته ضروری است. اولاً هیچ روش استانداردی برای انتقال دانش یا تجربه انسان به صورت مجموعه قوانین فازی وجود ندارد، ثانیاً نیاز به روش‌های موثرتر به منظور تنظیم توابع تعلق برای کاهش خطای خروجی و افزایش شاخص عملکرد همواره احساس می‌شود [۴۰].

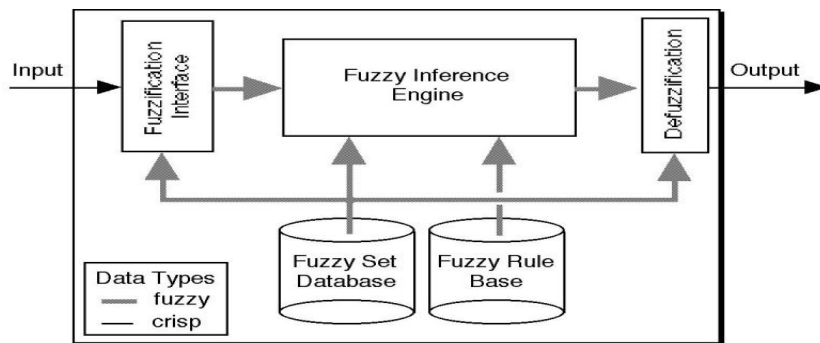
³ Rule_Base

⁴ Data_Base

⁵ Decision Making Unit (Fuzzy Inference Engine)

¹ Fuzzification Inference

² Defuzzification



شکل پیوست (ب)-۱: قسمت‌های مختلف یک سیستم فازی [۴۲]

منطق فازی می‌تواند مستقیماً به عنوان کنترل‌کننده و هم به عنوان تنظیم‌کننده بهره کنترل‌کننده تناسبی-انتگرالی^۱ به کار گرفته شود که در این صورت قواعد فازی برای تغییر ضرائب جملات تناسبی و انتگرالی در نواحی کار مختلف تنظیم می‌شوند. به کنترل‌کننده حاصل فازی-PI گفته می‌شود. ضرائب کنترل‌کننده PI نیز به صورت جدول‌بندی بهره برای نواحی کاری مختلف قابل تنظیم است. تشخیص نواحی کاری در این نوع کنترلر می‌تواند با استفاده از منطق فازی نیز صورت پذیرد که حاصل کنترل-کننده فازی-PI با جدول‌بندی بهره خواهد بود [۴۰].

در صورت تغییر پارامترهای سیستم با گذشت زمان، کنترل‌کننده‌های فازی-PI به صورت تطبیقی نیز قابل استفاده هستند که روش و چگونگی طراحی و پیاده‌سازی آن‌ها در مرجع [۴۳] مطرح شده است. همانطور که گفته شد، بهینه‌سازی ضرائب کنترلر PI می‌تواند با استفاده از الگوریتم ژنتیک انجام گیرد. به علاوه سایر الگوریتم‌های تکاملی و الگوریتم مورچگان نیز برای این موضوع به کار گرفته شده‌اند. الگوریتم مورچگان و الگوریتم‌های ژنتیک برای بهینه‌سازی کنترل‌کننده‌های فازی نیز به کار گرفته می‌شود [۴۲].

^۱ PI

پیوست (ج): شبکه‌های فازی-عصبی

شبکه‌های فازی-عصبی که در اوایل دهه ۱۹۹۰ مطرح شدند، حاصل تلفیق قوانین و ساختارهای فازی با شبکه‌های عصبی مصنوعی هستند که برای شناسایی سیستم‌ها به ویژه اجزاء غیرخطی آن‌ها، پیش‌بینی سری‌های زمانی، کنترل اتوماتیک، پردازش سیگنال و موارد متنوع دیگر به کار می‌روند [۴۴]. پایه و اساس مدل‌های فازی-عصبی در سال ۱۹۹۳ با ارائه ساختار ANFIS نهاده شد.

به طور کلی شبکه‌های فازی-عصبی ساختارهایی با قابلیت یادگیری بالا هستند که امکان استفاده از دانش و تجربه انسان در ساختار آن‌ها به دلیل استفاده از قواعد فازی فراهم آمده است. از مشکلات اصلی موجود برای طراحی و آموزش شبکه‌های فازی-عصبی می‌توان به ابهامات موجود در تعیین توابع تعلق و برچسب‌های زبانی و تنظیم قوانین فازی اشاره نمود. در واقع الگوریتم مشخص و از پیش تعیین شده برای تولید قوانین فازی و برچسب‌های زبانی وجود ندارد و تنظیم قواعد براساس تجربه و دانش انسان صورت می‌گیرد [۴۵].

به طور کلی دو روش برای طراحی و تنظیم اولیه پارامترهای شبکه‌های فازی-عصبی وجود دارد. روش اول مبتنی بر دانش و تجربه انسان از سیستم است که غالباً از آن با عنوان طراحی خلاقانه یاد می‌شود. روش دوم براساس داده‌ها و فرآیند آموزش و تنظیم خودکار قوانین فازی و پارامترها در شبکه است. غالباً روش دوم به دلیل عدم نیاز به شناخت قبلی از سیستم پرکاربردتر است. اما در این روش احتمال ایجاد تناقض در مجموعه قوانین فازی و یا تولید قوانین فازی با مقدمه یکسان اما موخره متفاوت وجود دارد. بنابراین لازم است روش‌ها و الگوریتم‌هایی برای بررسی و رفع تناقض در نظر گرفته شود [۴۶].

در آموزش شبکه‌های فازی-عصبی، دسته‌بندی داده‌های ورودی-خروجی و تولید قوانین پایه‌ای بسیار حائز اهمیت است. تقسیم‌بندی و خوشه‌بندی فازی مشخص کننده تعداد، موقعیت و گستره برچسب‌های زبانی برای دسته‌بندی فضای ورودی-خروجی است. برچسب‌های زبانی مشخص کننده زیر مجموعه‌های فازی به کار رفته در دسته‌بندی داده‌ها هستند. براین اساس می‌توان شبکه‌های فازی-عصبی را به دو دسته زیر تقسیم نمود:

الف) شبکه‌هایی که خوشه‌بندی داده‌های ورودی-خروجی و تعیین قوانین پایه‌ای آن‌ها براساس دانش و تجربه کارشناس انجام می‌شود و تنها پارامترهای شبکه از طریق فرآیند یادگیری تصحیح و تنظیم می‌شوند. شبکه فازی-عصبی ANFIS از این دسته است.

ب) شبکه‌هایی که خوشه‌بندی داده‌های ورودی-خروجی و تولید قوانین پایه‌ای آن‌ها به صورت خودکار و طی الگوریتم یادگیری و از طریق داده‌ها و ویژگی‌های عددی و آماری آن‌ها انجام می‌شود. بنابراین الگوریتم آموزش این ساختارها شامل دو مرحله است. مرحله اول خوشه‌بندی داده‌های ورودی-خروجی و تولید قواعد پایه‌ای و مرحله دوم تصحیح و تنظیم پارامترها. در این شبکه‌ها خوشه‌ها، برچسب‌های زبانی و قواعد پایه‌ای براساس داده‌های آموزش تعیین می‌شوند و هیچ قاعده اولیه‌ای لازم نیست. به عبارت دیگر مجموعه قواعد پایه‌ای قبل از شروع مرحله اول فرآیند آموزش تهی است. به دلیل عدم نیاز به دانش کارشناس، این ساختارها به مراتب محبوب‌تر از دسته قبل هستند و همواره تلاش‌های زیادی برای توسعه و پیشرفت آن‌ها انجام می‌شود. غالباً این شبکه‌ها با عنوان خودسازمان‌ده^۱ شناخته می‌شوند [۴۶].

در شبکه‌های خودسازمان‌ده چنانچه قواعد پایه‌ای به خوبی تنظیم شوند، مدل‌سازی با دقت قابل قبول انجام می‌شود. اما در صورتی که قواعد پایه‌ای به خوبی تنظیم نشده باشند و حداقل دو قانون با مقدمه یکسان و موخره متفاوت وجود داشته باشد، خطای بزرگی در مدل‌سازی ایجاد خواهد شد [۴۶]. بنابراین یادگیری سریع و قابلیت تعمیم دهی بالا از ویژگی‌های مدل‌های فازی-عصبی است. به علاوه نتایج حاصل از مدل‌های فازی-عصبی قابل تفسیر هستند، بر خلاف شبکه‌های عصبی که به صورت جعبه سیاه عمل می‌کنند [۴۰].

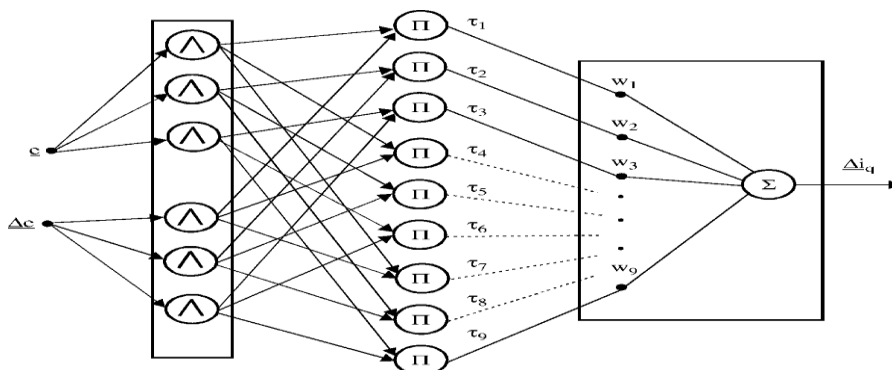
به عنوان مثالی از شبکه‌های فازی-عصبی ساختار ANFIS^۲ که حاصل تلفیق شبکه‌های عصبی تطبیقی و منطق فازی است مختصراً معرفی می‌شود. ساختار شبکه ANFIS که دارای پنج لایه است، در شکل پیوست (ج)-۱ نشان داده شده است. لایه اول دارای گره‌های قابل تنظیم به صورت $O_i^1 = \mu_{A_i}(x)$ است. معمولاً $\mu_{A_i}(x)$ به صورت گاوسی یا زنگوله شکل با مقدار بیشینه‌ای برابر یک و کمینه‌ای برابر صفر

¹ Self-Organizing

² Adaptive Network Based Fuzzy Inference System

می‌باشد. پارامترهای تابع که همان پارامترهای مقدمه قوانین فازی هستند، براساس توصیف زبانی مربوط به متغیرها و زیرفضاهای فازی و براساس روش‌هایبرید تنظیم می‌شوند [۴۴].

گره‌های لایه دوم ثابت اند. این گره‌ها دو سیگنال ورودی را در هم ضرب کرده و حاصل را به عنوان خروجی تحویل شبکه می‌دهند. سیگنال‌های ورودی این گره‌ها در واقع میزان سازگاری ورودی با هر یک از توابع تعلق و خروجی آن‌ها، وزن مربوط به هر یک از قوانین است. گره‌های سوم نیز ثابت بوده و وظیفه آن‌ها محاسبه وزن نرمالیزه هر یک از قوانین، مطابق رابطه (ج-۱) است.



شکل پیوست (ج)-۱: ساختار شبکه ANFIS [۴۵]

$$\bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, 2 \quad (\text{ج-۱})$$

گره‌های لایه چهارم، قابل تنظیم‌اند و پارامترهای آن‌ها همان پارامترهای موخره قوانین است. تابع گره‌ای آن‌ها مطابق رابطه (ج-۲) است که در آن عبارت فازی موخره قانون i ام می‌باشد. در این رابطه O_i^4 خروجی گره i ام از لایه چهارم است.

$$O_i^4 = \bar{w}_i \cdot z_i \quad (\text{ج-۲})$$

لایه پنجم یک گره دارد و خروجی آن حاصل جمع کلیه سیگنال‌های ورودی به این گره یا به عبارت دیگر میانگین وزن دار خروجی قوانین فازی است. توابع گره‌ای این شبکه باید به طور قطعه‌ای مشتق‌پذیر باشند. قوانین به کار رفته نیز از نوع تاکاگی_سوگینو بوده که عبارت موجود در موخره آن‌ها برحسب متغیرهای مقدمه خطی است. الگوریتم هایبرید که ترکیب روش کاهش گرادیان و تخمین کمترین

مربعات (LSE)^۱ می‌باشد، به صورت مستقیم برای شناسایی و تخمین پارامترهای شبکه قابل اعمال است. پارامترهای موخره قوانین همان پارامترهای خطی شبکه هستند که به روش LSE تخمین زده می‌شوند. پارامترهای مقدمه نیز به روش گرادیانی به‌روز خواهند شد [۴۴].

پیوست (د): توضیح مختصر در خصوص نرم‌افزار System Generator

System Generator یک ابزار طراحی پردازش سیگنال^۲ است. این نرم‌افزار توسط شرکت زایلینکس^۳ معرفی شد و مهمترین قابلیت آن ایجاد یک محیط شبیه‌سازی همانند شبیه‌ساز متلب برای طراحی FPGA است. برای استفاده از System Generator هیچ نیازی به اطلاعات قبلی در مورد انواع FPGAهای شرکت زایلینکس یا متدهای طراحی نیست. در این نرم‌افزار طراحی‌های DSP به صورت تصویری، با کمک مجموعه بلوک‌هایی که نرم‌افزار در اختیار کاربر قرار می‌دهد. از ویژگی‌های منحصر بفرد این نرم‌افزار این است که تمامی مراحل پیاده‌سازی از قبیل سنتز و یا تولید یک فایل برنامه‌ریز برای FPGA را به صورت خودکار انجام می‌دهد. با استفاده از قابلیت پردازش سیگنال^۴ در نرم‌افزار می‌توان مدل‌های الگوریتمی متلب را به کار برد. AccelDSP شامل سنتز الگوریتمی قدرتمندی است که ممیز شناور^۵ نرم‌افزار را به عنوان ورودی گرفته و یک مدل ممیز ثابت کامل^۶ را برای استفاده در محیط شبیه‌ساز ارائه می‌دهد. همچنین با استفاده از ویژگی شبیه‌سازی همزمان^۷ سخت‌افزاری در نرم‌افزار می‌توان به شبیه‌سازی سریع دست یافت. این نرم‌افزار به طور اتوماتیک یک شبیه‌سازی سخت‌افزاری برای یک طرح ساخته شده با استفاده از بلوک‌های پردازش سیگنال تولید می‌کند که یکی از ۲۰ سطح سخت‌افزاری پشتیبانی شده را اجرا می‌کند. در واقع System Generator یک محیط مجتمع برای انواع طراحی‌های پردازش سیگنال ارائه می‌دهد که می‌تواند طراحی‌های مختلفی که ممکن است با استفاده از شبیه‌سازی، متلب یا زبان‌های برنامه‌نویسی^۸ ایجاد شده باشد را در یک محیط شبیه‌سازی و اجرا گرد هم آورد.

¹ Least Square Estimation

¹ Digital Signal Processor(DSP)

² Xilinx

³ Accel DSP

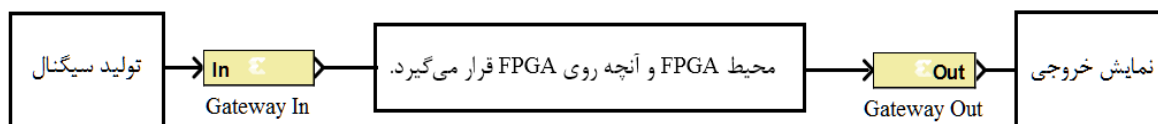
⁴ Floating Point

⁵ Fixed Point

⁶ Cosimulation

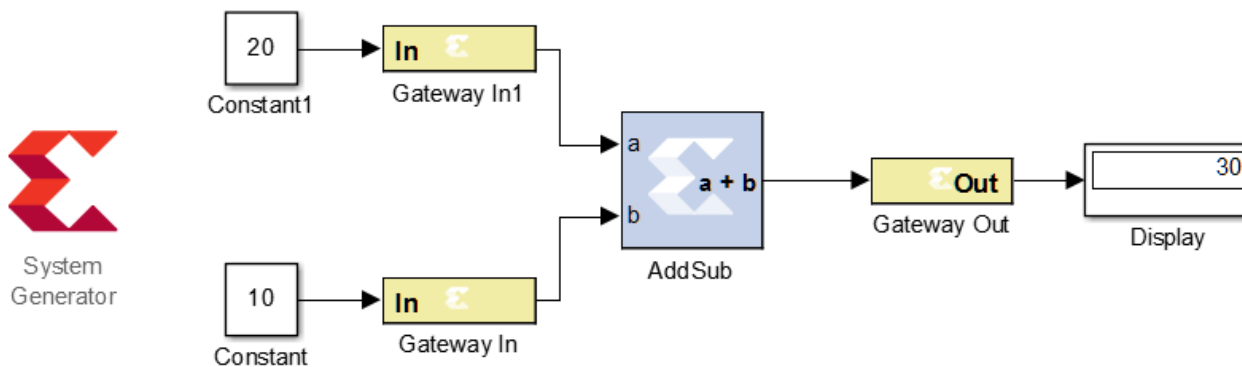
⁸ C ,C++

همچنین System Generator دارای یک جعبه بلوک است که اجازه می‌دهد زبان‌های انتقال ثابت^۱ به شبیه‌ساز تبدیل شده و بطور همزمان در شبیه‌سازهای بخصوص نرم افزار^۲ استفاده شود. جهت پیاده‌سازی پیاده‌سازی دیجیتالی این بلوک‌ها کافی است نرم‌افزار را نصب کنیم. به طور کل محیط کاربر پسند، عدم نیاز به تبدیل فرمت داده‌ها در FPGA، عدم نیاز به داشتن دانش کدنویسی در FPGA و امکان نتیجه گیری ساده یک مدار طراحی شده سه ویژگی برجسته این نرم افزار است. برای آشنایی بیشتر با محیط این شبیه‌ساز به شکل زیر توجه کنید.



شکل پیوست (د)-۱: ساختار System Generator

به عنوان یک مثال ساده مدار زیر شبیه‌سازی جمع ساده دو عدد را نشان می‌دهد.



شکل پیوست (د)-۲: مدار ساده شبیه‌سازی شده در System Generator

¹ Register Transfer Language

² Modelsim , Xilinx ISE

- [1] Krishna Vasudevan, G. Sridhara Rao, P. Sasidhara Rao, “Electrical Machines II, Speed control of Induction Machines,” *Indian Institute of Tehcnology Madras*, Nov.2014.
- [2] H. Shadabi, A. Rahnama Sadat, A. Pashaei, M.B.B. Sharifian, “Speed Control Of Linear Induction Motor Using DTFC Method Considering End-Effect Phenomenon,” *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, Issue 21, vol. 6, no. 4, pp. 75-81, Dec. 2014.
- [3] Almas, A.K. Bharadwaj, “Speed Control of Three Phase Induction Motor using Space Vector PWM Method,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 4, Issue 6, June. 2015.
- [4] K. Moshirvaziri and M. A. Amouzegar, “A Deep-Cutting-Plane Technique for Reverse Convex Optimization,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 4, pp. 1054-1060, Aug. 2011.
- [5] Gourambika, M. S. Aspalli, “Speed Control of Three Phase Induction Motor by VVVF method using G7/A-1000 Drive.” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 3, Issue 4, Sept. 2014.
- [6] Sujeet Kumar Soni and Anil Gupta, “Analysis of SVPWM Based Speed Control of Induction Motor Drive with using V/F Control Based 3 Level Inverter,” *International Journal of Scientific Engineering and Technology*, vol. 2, Issue 9, pp. 932-938, Sept. 2013.
- [7] Shilpa V. Kailaswar, Prof. R.A.Keswani, “Speed Control of Three Phase Induction Motor by V/f Method for Batching Motion System,” *International Journal of Engineering Research and Applications (IJERA)*, vol. 3, Issue 2, March -April 2013, pp.1732-1736.
- [8] T. Ramesh and A. K. Panda, “Direct flux and torque control of three phase induction motor drive using PI and fuzzy logic controllers for speed regulator and low torque ripple,” *Students Conference on Engineering and Systems (SCES)*, Allahabad, Uttar Pradesh, pp. 1-6, March. 2012.
- [9] Narongrit Pimkumwong, Amorn Onkrong, Tirasak Sapaklom, “ Modeling and Simulation of Direct Torque Control Induction Motor Drives via Constant Volt/Hertz Technique,” *International Conference on Advances in Computational*

- Modeling and Simulation, Elsevier Procedia Engineering*, Vol. 31, pp. 1211–1216, June. 2012.
- [10] Bin Wu A John Wiley & Sons, *High-Power Converters and AC Drives*. Wiley-IEEE Press, 2006.
- [11] F. Edwin, W. Xiao and V. Khadkikar, “Topology review of single phase grid-connected module integrated converters for PV applications,” *38th Annual Conference on IEEE Industrial Electronics Society (IECON)*, Montreal, pp. 821-827, Oct. 2012.
- [12] M. Andersen and B. Alvsten, “200 W low cost module integrated utility interface for modular photovoltaic energy systems,” *Industrial Electronics, Control, and Instrumentation, Proceedings of the 1995 IEEE IECON 21st International Conference on*, Orlando, FL, vol.1, pp. 572-577, Nov. 1995.
- [13] S. M. Chen, T. J. Liang, L. S. Yang and J. F. Chen, “A Boost Converter With Capacitor Multiplier and Coupled Inductor for AC Module Applications,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1503-1511, April. 2013.
- [14] D. Cruz and R. Demonti, “Grid Connected PV System Using Two Energy Processing Stages.” *Photovoltaic Specialists Conference, 2002. Conference Record of the Twenty-Ninth IEEE*, pp. 1649-1652, May. 2002.
- [15] S. Mekhilef, N. A. Rahim and A. M. Omar, “A new solar energy conversion scheme implemented using grid-tied single phase inverter,” *TENCON 2000. Proceedings*, Kuala Lumpur, vol. 3, pp. 524-527, Sept. 2000.
- [16] P. Wolfs and Quan Li, “An analysis of a resonant half bridge dual converter operating in continuous and discontinuous modes,” *Power Electronics Specialists Conference, IEEE 33rd Annual*, vol. 3, pp. 1313-1318, Nov. 2002.
- [17] Q. Li and P. Wolfs, “A Review of the Single Phase Photovoltaic Module Integrated Converter Topologies With Three Different DC Link Configurations,” *IEEE Transactions on Power Electronics*, vol. 23, no. 3, pp. 1320-1333, May. 2008.
- [18] N. Kasa, T. Iida, L. Chen, “Flyback Inverter Controlled by Sensorless Current MPPT for Photovoltaic Power System,” *IEEE Transactions on Industrial Electronics*, pp. 1145-1152, Aug. 2005.
- [19] A. C. Kyritsis, E. C. Tatakis and N. P. Papanikolaou, “Optimum Design of the Current-Source Flyback Inverter for Decentralized Grid-Connected Photovoltaic

- Systems,” in *IEEE Transactions on Energy Conversion*, vol. 23, no. 1, pp. 281-293, March 2008.
- [20] J. Lai, “Power Conditioning Circuit Topologies,” *IEEE Industrial Electronics Magazine*, pp. 24-34, June 2009.
- [21] T. Shimizu, K. Wada, and N. Nakamura, “Flyback-type single-phase utility interactive inverter with power pulsation decoupling on the dc input for an ac photovoltaic module system,” *IEEE Trans. Power Electron*, vol. 21, no. 5, pp. 1264-1272, Sep. 2000.
- [22] S. Harb, H. Hu, N. Kutkut, I. Batarseh and Z. J. Shen, "A three-port Photovoltaic (PV) micro-inverter with power decoupling capability," *Applied Power Electronics Conference and Exposition (APEC), 2011 Twenty-Sixth Annual IEEE*, Fort Worth, TX, pp. 203-208, March 2011.
- [23] M. Kamil Microchip’s “Grid-Connected Solar Micro Inverter Reference Design Lowers the Cost and Improves the Efficiency of Solar Power System,” Sept. 2011.
- [24] D. C. Martins and R. Demonti, “Interconnection of a photovoltaic panels array to a single-phase utility line from a static conversion system,” *Power Electronics Specialists Conference, PESC IEEE 31st Annual*, Galway, pp. 1207-1211, vol. 3, Aug. 2000.
- [25] J. Beristain, J. Bordonau, A. Gilabert and G. Velasco, “Synthesis and modulation of a single phase DC/AC converter with high-frequency isolation in photovoltaic energy applications,” *Power Electronics Specialist Conference, PESC '03. 2003 IEEE 34th Annual*, pp. 1191-1196, vol.3, Aug. 2003.
- [26] K. C. A. de Souza, M. R. de Castro and F. Antunes, “A DC/AC converter for single-phase grid-connected photovoltaic systems,” *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, pp. 3268-3273, vol.4, Nov. 2002.
- [27] S. Yatsuki, K. Wada, T. Shimizu, H. Takagi and M. Ito, “A novel AC photovoltaic module system based on the impedance-admittance conversion theory,” *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual*, Vancouver, BC, pp. 2191-2196, vol. 4, June. 2001.
- [28] A. Trubitsyn, B. J. Pierquet, A. K. Hayman, G. E. Gamache, C. R. Sullivan and D. J. Perreault, “High-efficiency inverter for photovoltaic applications,” *IEEE Energy Conversion Congress and Exposition*, Atlanta, GA, 2010, pp. 2803-2810, Sept. 2010.

- [29] S. K. Sahoo, A. Ramulu, S. Batta and S. Duggal, "Performance analysis and simulation of three phase voltage source inverter using basic PWM techniques," *Sustainable Energy and Intelligent Systems (SEISCON 2012), IET Chennai 3rd International on*, Tiruchengode, pp. 1-7, Dec. 2012.
- [30] N. I. Raju, M. D. Shahinur Islam, A. Ahsan Uddin, "Sinusoidal PWM Signal Generation Technique for Three Phase Voltage Source Inverter with Analog Circuit & Simulation of PWM Inverter for Standalone Load & Micro-grid System," *Department of EEE*, vol. 3, no. 3, Jun. 2007.
- [31] A. Bermak, "A CMOS imager with PFM/PWM based analog-to-digital converter," *Circuits and Systems, ISCAS 2002. IEEE International Symposium on*, pp. IV-53-IV-56, vol. 4, May. 2002.
- [32] S. M. M. Islam and G. M. Sharif, "Microcontroller based sinusoidal PWM inverter for photovoltaic application," *Developments in Renewable Energy Technology (ICDRET), 2009 1st International Conference on the*, Dhaka, pp. 1-4, Dec. 2009.
- [33] J. J. Zacharias and J. M. Conrad, "A survey of digital signal processing education," *Proceedings 2007 IEEE SoutheastCon*, Richmond, VA, pp. 322-327, March. 2007.
- [34] F. A. Ramirez, M. A. Arjona and C. Hernandez, "A Space-Vector PWM Voltage-Source Inverter for a Three-Phase Induction Motor Based on the dsPIC30F3011," *Electronics, Robotics and Automotive Mechanics Conference, CERMA '09*, Cuernavaca, Morelos, pp. 429-434, Sept. 2009.
- [35] Ian Kuon, Jonathan S. Rose, and Russell Tessier, "FPGA Architecture: Survey and Challenges," *Foundations and Trends in Electronic Design Automation*, vol. 2, Issue 2, Feb. 2008.
- [36] M. Curkovic, K. Jezernik and R. Horvat, "FPGA-Based Predictive Sliding Mode Controller of a Three-Phase Inverter," in *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 637-644, Feb. 2013.
- [37] P. Banerjee, "An overview of a compiler for mapping MATLAB programs onto FPGAs," *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific*, pp. 477-482, Jan. 2003.
- [38] W. Miller and K. Owyang, "Designing a high performance FPGA-using the PREP Benchmarks," *WESCON/'93. Conference Record*, San Francisco, CA, pp. 234-239, Sept. 1993.

- [39] S. Harris and D. Harris, "Inexpensive student-assembled FPGA / microcontroller board," *2005 IEEE International Conference on Microelectronic Systems Education (MSE'05)*, pp. 101-102, June 2005.
- [40] R. Babsuka, H. Verbruggen, "Neuro-Fuzzy methods for nonlinear system identification," *Annual Reviews in Control*, vol. 27, Issue 1, pp. 73-85, 2003.
- [41] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," in *IEEE Transactions on Computers*, vol. 40, no. 12, pp. 1320-1336, Dec. 1991.
- [42] Luis Alberto Torres Salomao, Hugo Gámez Cuatzin, Juan Anzures Marín and Isidro I. Lázaro Castillo, "Fuzzy-PI Control, PI Control and Fuzzy Logic Control Comparison Applied to a Fixed Speed Horizontal Axis 1.5 MW Wind Turbine," *Proceedings of the World Congress on Engineering and Computer Science, WCECS 2012, San Francisco, USA*, vol. 2, Oct. 2012.
- [43] H. Hu, S. Harb, N. Kutkut, I. Batarseh, and Z. J. Shen, "A review of power decoupling techniques for microinverters with three different decoupling capacitor locations in PV systems," *IEEE Trans. Power Electron*, vol. 28, no. 6, pp. 2711–2726, Jun. 2013.
- [44] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May/Jun. 1993.
- [45] R. W. Zhou and C. Quek, "A pseudo outer-product based fuzzy neural network and its rule-identification algorithm," *Neural Networks, IEEE International Conference on*, Washington, DC, pp. 1156-1161 vol. 2, June. 1996.
- [46] S. W. Tung, C. Quek and C. Guan, "SaFIN: A Self-Adaptive Fuzzy Inference Network," in *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1928-1940, Dec. 2011.
- [47] Kiam Heong Ang, G. Chong and Yun Li, "PID control system analysis, design, and technology," in *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559-576, July 2005.
- [48] Y. H. Ji, D. Y. Jung, J. H. Kim, T. W. Lee and C. Y. Won, "A current shaping method for PV-AC module DCM-flyback inverter under CCM operation," *Power Electronics and ECCE Asia (ICPE & ECCE), 2011 IEEE 8th International Conference on*, Jeju, pp. 2598-2605, June. 2011.

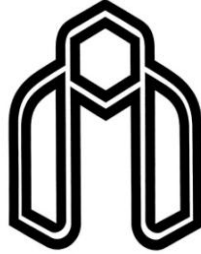
- [49] R. C. Dugan, T. S. Key and G. J. Ball, "Distributed resources standards," *IEEE Industry Applications Magazine*, vol. 12, no. 1, pp. 27-34, Jan.-Feb. 2006.
- [50] H. Hu, S. Harb, N. H. Kutkut, Z. J. Shen and I. Batarseh, "A Single-Stage Microinverter Without Using Electrolytic Capacitors," *IEEE Transactions on Power Electronics*, vol. 28, no. 6, pp. 2677-2687, June. 2013.
- [51] J. Alfeu, S. Filho and E. Ruppert, "Tuning PI regulators for IM direct torque control using complex transfer function," *Industry Applications (INDUSCON), 2010 9th IEEE/IAS International Conference on*, Sao Paulo, pp. 1-6, Nov. 2010.
- [52] R. Mishra, S. P. Singh, D. Singh, B. Singh and D. Kumar, "Investigation of transient performance of VSI-Fed IM drives using Volts/Hz and vector control techniques," *Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on*, Allahabad, pp. 1-6, Dec. 2012.
- [53] P. M. Menghal and A. J. Laxmi, "Dynamic modeling, simulation & analysis of induction motor drives," *Science Engineering and Management Research (ICSEMR), 2014 International Conference on*, Chennai, pp. 1-7, Nov. 2014.

Abstract

Along with advances in semiconductor industry such as transistors and thyristors, rising pricing in new drivers with high power medium voltage motor exploded. Semiconductors are widely used in petrochemical, mining, steel and metal industry, transportation and other industries used to save electrical energy, increase productivity and improve product quality. Among common drivers in industry there are inverters used in single-phase and three-phase mode in order to control motor speed. The researchers in this field has always been considered to design correct to avoid disturbance and prevent energy waste. Due to the many benefits of inverters, selected switching technique has a significant role in reducing or eliminating harmonics of the inverter output voltage and switching loss reduction.

In this thesis a new efficient architecture is provided to implement correctly the speed motor control section on FPGA. In order to reduce distortion and energy dissipation of the designs that have been made in this field the pulse width modulation method is used to select the frequency of the carrier wave. The new design is such that the correctly switching of three-phase inverter is provided to optimize system performance. Circuit is simulated on Spartan3 and due to the selected hardware volume, the output quickly reaches final response.

Keywords: Three phase inverter, Sinusoidal pulse width modulation, Proportional integral controller, Anti windup controller, Fuzzy controller, Field Programmable Gate Array.



Shahrood University of Technology
Electrical and Robotics Engineering department

Design and simulation of a three-phase inverter using FPGA

Tan'naz Tabasi

Supervisor:

Dr. Seyed Ali Soleimani

February 2016