



پایان نامه کارشناسی ارشد

دانشکده برق و رباتیک

عنوان :

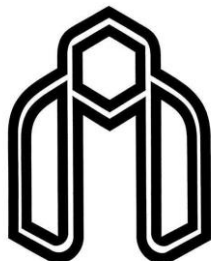
طراحی و ساخت ربات آزمایشگاهی اسکارا برای جابجایی

اجسام با استفاده از روش کنترلی گشتاور محاسباتی

استاد راهنما : آقای دکتر فاتح

ارائه دهنده : مجید صحتی یزدی

تابستان ۱۳۸۶



**SHAHROOD UNIVERSITY
OF TECHNOLOGY**

Department of Electrical and Robotic

Thesis submitted for

degree of Master of Science (M.sc) in Control Engineering

Ttitle:

**The Design and Constructing of a SCARA Robot Using Computed
Torque Control for pick and place**

Supervisor:

Dr. M.M. Fateh

Student:

Majid Sehati Yazdi

September 2007

تقدیم بہ

پدر عزیزم

روح پاکے مادر

تشکر و قدردانی

خداوندا داده ات را لطف ، نداده ات را حکمت و گرفته ات را امتحان می پندارم و در همه حال تو را شکر می کنم ، از استاد گرامی جناب آقای دکتر فاتح که با راهنماییها و نظارت خویش مرا در تدوین این پایان نامه یاری کرده‌اند سپاسگزاری نموده و موفقیت روزافزون ایشان را از خداوند منان خواستارم.

چکیده :

در حال حاضر اکثر مقالات علمی چاپ شده در ژورنالهای معتبر جهان مبتنی بر نتایج عملی اعمال تئوریهای کنترلی بروی سیستمهای مکانیکی واقعی است ، با توجه به اینکه یکی از مرسومترین سیستمهای مکانیکی غیر خطی که می توان اهم تئوریهای کنترلی را روی آن پیاده سازی کرد بازوهای ماهر می باشند و نیز اجرای چنین سیستمی در دانشگاه صنعتی شاهرود برای ایجاد بستری جهت طراحی و ساخت و کنترل نمونه های مشابه توسط تیمهای رباتیک متشکل از گرایشهای مکانیک ، الکترونیک ، رباتیک ، کنترل ضروری به نظر می رسيد ، تصمیم به ساخت ربات اسکارا که علاوه بر پیچیدگی ، ظاهری عامه پسند دارد گرفته شد . و پس از ساخت بدنه ، طراحی مدارات الکترونیکی و برقراری ارتباط آن با رایانه ، روش کنترلی گشتاور محاسباتی پیشنهادی خود را بروی آن اعمال کردیم . امید است اجرای این پروژه توانسته باشد ندای " ما می توانیم " را بار دیگر به گوش همگان برساند .

کلمه های کلیدی :

ربات اسکارا - طراحی وساخت ربات - الکترونیک ربات - مدارات واسطه - کنترل موقعیت ربات کنترل مفصل مستقل - کنترل گشتاور محاسباتی - کنترل تطبیقی - شناسایی سیستم

فهرست مطالب

فصل اول : مقدمه..... ۱

فصل دوم : ربات اسکارا..... ۴

۱-۲- مقدمه..... ۵

۲-۲- استانداردهای ساخت ربات اسکارا..... ۶

۱-۲-۲- خصوصیات ابعادی ربات نمونه Adept Cobra i800..... ۷

۲-۲-۲- طراحی بدنه ۹

۳-۲-۲- طراحی رابطها بوسیله نرم افزار Solid Work..... ۱۱

۳-۲- ساخت نمونه اولیه با اسفنج..... ۱۴

۴-۲- ساخت نمونه اولیه با تفلون CFC..... ۱۵

۵-۲- ساخت نمونه نهایی از جنس آلومینیوم ۱۶

۶-۲- کاربردهای ربات اسکارا ۱۸

فصل سوم : بررسی موتورهای DC مغناطیس دائم..... ۱۹

۱-۳- مقدمه..... ۱۹

۲-۳- چگونگی انتخاب موتورهای DC ۱۹

۳-۳- دینامیک موتورهای DC مغناطیس دائم..... ۲۱

۴-۳- اندازه گیری R_a ۲۳

۵-۳- اندازه گیری L_a ۲۴

- ۳-۶- اندازه گیری K_b ۲۶
- ۳-۷- اندازه گیری K_t ۲۷
- ۳-۸- اندازه گیری ضریب میرایی موتور B ۲۸
- ۳-۹- اندازه گیری ممان اینرسی J ۳۰
- ۳-۱۰- مونتاژ رابط ها ، موتورها ۳۱
- ۳-۱۰-۱- کاهش اثرات لقی بوسیله اورینگها و اصطکاک نامطلوب توسط بلبینگها ، پوشها ۳۲
- فصل چهارم : الکترونیک ربات** ۳۴
- ۴-۱- مقدمه ۳۴
- ۴-۲- چرا ATMEGA32 ؟ ۳۴
- ۴-۳- طراحی منبع تغذیه ربات ۳۷
- ۴-۴- طراحی و اجرای مدارهای راه انداز برای موتورها ۳۹
- ۴-۴-۱- راه انداز های پل H ترانزیستوری ۴۱
- ۴-۴-۲- چیپهای راه انداز ۴۳
- ۴-۵- کد گذارهای نوری ۴۷
- ۴-۵-۱- آشنایی با سنسورهای تشخیص موقعیت موجود در بازار ۴۷
- ۴-۵-۲- طراحی مدارات شمارنده مادون قرمز IR ۴۹
- ۴-۵-۳- ملاحظات عملی سنسورهای مادون قرمز ۵۱
- ۴-۵-۴- کدگذارهای دورانی ۵۲
- ۴-۵-۵- بررسی و کار با کدگذارهای نوری افزایشی شرکت Autonics ۵۳
- ۴-۶- زمان نمونه برداری ۵۵
- ۴-۶-۱- محاسبه سرعت و شتاب بازوی ماهر ۵۵
- ۴-۷- سیم بندی و حفاظت سیگنال ۵۶

۴-۷-۱- شیلد دار کردن سیمها..... ۵۶

فصل پنجم : بررسی نحوه اجرای ساختار سیستم ربات..... ۵۷

۵-۱- مقدمه..... ۵۷

۵-۲- اجرای ساختار سیستم بصورت یک مجموعه خارجی..... ۵۷

۵-۳- اجرای ساختار سیستم بصورت PC Interface..... ۵۸

۵-۳-۱- بررسی امکان ارتباط سریال ربات با رایانه..... ۶۰

۵-۴-۱- ارتباط SPI میکروکنترلرها با هم..... ۶۲

۵-۴-۱-۱- ارسال اطلاعات از رئیس به مرئوس ها..... ۶۳

۵-۴-۲- ارسال اطلاعات از مرئوس ها به رئیس..... ۶۴

۵-۵- مطالعه پروتکل ارتباط سریال نرم افزار مطلب در محیط Simulink..... ۶۵

۵-۵-۱- ارتباط سریال بین رئیس و رایانه..... ۶۶

۵-۵-۲- دریافت داده توسط RS232 Blocksets..... ۶۶

۵-۵-۳- ارسال داده توسط RS232 Blocksets..... ۶۷

۵-۵-۴- چگونگی ارسال و دریافت رئیس از RS232 Blocksets..... ۶۸

۵-۵-۵- جایگاه ویژه رئیس در دو پرتکل ارتباطی..... ۶۹

۵-۶- ارتباط رئیس با نرم افزار Visual Basic..... ۷۱

فصل ششم : سینماتیک و سینماتیک وارون SCARA..... ۷۴

۶-۱- مقدمه..... ۷۴

۶-۲- سینماتیک مستقیم و سینماتیک وارون..... ۷۵

۶-۳- نمایش دناویت هارتنبرگ و ربات اسکارا..... ۷۶

۶-۳-۱- محاسبه سینماتیک مستقیم اسکارا توسط MATLAB..... ۸۱

۸۲.....	۶-۳-۲- محاسبه سینماتیک وارون اسکارا توسط Visual Basic
۸۴.....	۶-۳-۳- سینماتیک مستقیم اسکارا و ربات لولایی ۲-رابط
۸۵.....	۶-۴- مدل سازی ربات اسکارا.....
۸۹.....	فصل هفتم : طراحی مسیر برای ربات.....
۸۹.....	۷-۱- مقدمه
۸۹.....	۷-۲- چرا طراحی مسیر در فضای مفصلی ؟.....
۹۰.....	۷-۲-۱- کمینه سازی زمان حرکت.....
۹۰.....	۷-۲-۲- ملاحظات عملی مسیرهای بنگ-بنگ.....
۹۱.....	۷-۳- طراحی مسیر های هموار.....
۹۴.....	۷-۳-۱- ناحیه مرده.....
۹۴.....	۷-۴- طراحی جدول بهره برای مقید کردن مسیر در جابجایی نقطه به نقطه.....
۹۸.....	فصل هشتم : کنترل ربات
۹۸.....	۸-۱- مقدمه.....
۹۸.....	۸-۲- کنترل نقطه به نقطه.....
۱۰۱.....	۸-۳- روش کنترل پیشخور.....
۱۰۳.....	۸-۴- روش کنترلی گشتاور محاسباتی.....
۱۰۵.....	۸-۵- گشتاور محاسباتی همراه با ترم پیشخور.....
۱۰۷.....	۸-۶- طرح عمومی کنترل تطبیقی بازوی ماهر.....
۱۱۲.....	۸-۷- اجرای روش کنترلی پیشنهادی.....

فصل نهم : نتایج شبیه سازی..... ۱۱۵

۹-۱- مقدمه..... ۱۱۵

۹-۲- کنترل پیشخور..... ۱۱۵

۹-۳- گشتاور محاسباتی تطبیقی..... ۱۱۹

۹-۴- گشتاور محاسباتی همراه با ترم پیشخور..... ۱۲۲

فصل دهم : نتیجه گیری..... ۱۲۵

۱۰-۱- نتیجه گیری..... ۱۲۵

۱۰-۲- بررسی مشکلات و موانع موجود..... ۱۲۸

۱۰-۳- فعالیتهای آینده..... ۱۲۸

فصل یازدهم : پیوستها..... ۱۳۰

پیوست ۱- برنامه میکرو کنترلرها..... ۱۳۰

پیوست ۲- برنامه Visual Basic..... ۱۵۹

پیوست ۳- مشخصات فنی چیپ ها..... ۱۶۶

منابع و مراجع..... ۱۶۷

فصل اول : مقدمه

کلمه ربات در سال ۱۹۲۱ توسط یک نمایش نامه نویس اهل چکسلواکی ارائه شد. وی در کتابی خدمتگزاران مکانیکی را توصیف نمود که قادر بودند تمامی کارهایی که یک انسان انجام می دهد انجام دهند در واقع ربات معادل کلمه کارگر در زبان چک می باشد.

برای کارهای خسته کننده یا خطرناک از رباتها استفاده میشود لزوما رباتها همیشه شبیه انسان نیستند مثلاً در یک خط تولید صنعتی رباتها به هیچ وجه شباهتی به انسان ندارند و هیچ لزومی هم ندارد .

در حال حاضر علم رباتیک یکی از جدیدترین علوم کاربردی و تحقیقاتی در بخشهای صنعتی ، خدماتی و پزشکی می باشد . آنچه مسلم است این که درک پیچیدگیهای مربوط به عملکرد رباتها فوق العاده سخت است و با ورود به میدان ساخت ربات این مسائل و پیچیدگیها بیشتر می شوند که درک آن نیاز به دانش در زمینه های مختلف از جمله مهندسی مکانیک ، مهندسی برق ، مهندسی رایانه و سایر علوم مربوط به آن دارد ، می توان گفت بیشتر بازوهای ماهر صنعتی در دسته بندی زیر قرار می گیرند :

۱- کروی

۲- اسکارا

۳- استوانه ای

۴- هنرمند

۵- کارتزین

امروزه رباتها کاربردهای متنوع و متعددی در صنعت، تجارت، پزشکی، اکتشاف و غیره دارند بنابراین از یک طرف آموزش رباتیک به صورت نظری و عملی وازطرف دیگر دستیابی به فن اوری طراحی و ساخت ربات از اهمیت فوق العاده ای برخوردار است . طراحی ، ساخت و کنترل یک ربات آزمایشگاهی می تواند در یادگیری تجربی آنچه را که در تئوری کنترل ربات آموخته ایم کمک موثری بنماید. روش گشتاور محاسباتی میدان جدیدی از تحقیق را در کنترل ربات گشوده ودر این خصوص مقالات ارزشمندی ارائه شده است . در این پایان نامه برای پیاده سازی این روش کنترلی از ربات اسکارا استفاده شده است .

با توجه به اینکه برای ساختن یک ربات دانشهای مختلفی بکار گرفته می شوند می توان به پیچیدگی و سختی کار ساخت از همان اول پی برد بدین ترتیب که علوم نظیر مکانیک برای طراحی بهینه بدنه اصلی رابطها و چگونگی اتصال مفاصل به یکدیگر کاهش لقی ها و در عین حال جلوگیری از افزایش اصطکاکها ، صلبیت و مسائل دیگر دخیل هستند که در فصل ۲ به بررسی و راههای رفع مشکلات وموانع موجود پرداخته ایم همچنین مسائل مربوط به انتخاب موتورها و اندازه گیری و محاسبه پارامترهای موتورهای ربات را در فصل ۳ دنبال می کنیم .

علم الکترونیک در قسمتهای مختلف و در جز به جز نقاط این پروژه به چشم می خورد از طراحی منبع تغذیه تا طراحی مدارات شمارنده و کد گذارهای نوری^۱ و چگونگی راه اندازی^۲ آنها ، راه اندازی درایو موتورها

^۱ Optical Encoder

^۲ Drive

مدارات فرمان دیجیتال و سایر ریزه کاریهای دیگر خود را نشان می دهند که در فصل ۴ به صورت مبسوط در مورد تمام این مسائل بحث و بررسی و طراحی صورت گرفته است .

علمی مانند کامپیوتر برای سیستمهای ارتباط رایانه ای و طراحی پانلهای فرمان واسطه در رایانه استفاده می شود . در مورد این مسائل و طراحیهای مدارهای واسطه رایانه ای ومحدودیتهای استفاده از برخی درگاهها^۳ از نظر نرخ دریافت و ارسال و.... در فصل ۵ بررسیها و طراحی های ارزشمندی شده است .

علم رباتیک که تمام مسائل مربوط به تغییر فضاها مدلهای دینامیکی ومعادلات حرکت بازوی ماهر وطراحی مسیرهای مختلف را در بر می گیرد نیز نقش مهمی را در اجرای موفقیت آمیز چنین طرحهایی ایفا می کند در فصل ۶ به بررسی سینماتیک مستقیم به روش زنجیرهای سینماتیکی و سینماتیک وارون اسکارا به دو روش و اشاره به حالت عمومی مدل دینامیکی اسکارا پرداخته ایم .

در فصل ۷ به بررسی مسائل مربوط به طراحی مسیرهای نرم برای حرکت نقطه به نقطه خواهیم پرداخت و مکانیزم محدود کننده های پویا را برای هموار کردن پاسخ کنترلر PD^۴ در عمل را بیان می کنیم .

و در نهایت پس از به هم پیوستن تمام این علوم ، علم کنترل برای کنترل کردن کل سیستم وارد عمل می شود بحثهای مربوط به کنترل نقطه به نقطه و کنترلر PD ، پیشرو^۵ و گشتاور محاسباتی^۶ و گشتاور محاسباتی تطبیقی^۷ واستفاده از الگوریتمهای تخمین پارامتر^۸ در بحث گشتاور محاسباتی در فصل ۸ به نظر شما خواهد رسید .

^۳ Ports

^۴ Proportional-Derivative (تناسبی- مشتقی)

^۵ Feed Forward

^۶ Computed Torque

^۷ Adaptive Computed Torque

^۸ Parameter Stimation Algorithm

درفصل ۹ به بررسی نتایج طرحهای بیان شده در فصل ۷ پرداخته ، مشکلات و موانع موجود در ساخت و ملاحظات عملی و مسائل مربوط به اجرای کنترلرها را بیان کرده دور نمایی از کارهای آینده را از نظرتان می گذرانیم .

در پیوست ۱ تمامی برنامه های مربوط به میکرو کنترلرها و ارتباط SPI^۹ میکرو کنترلرهای مرئوس با رئیس ارائه شده اند ، در پیوست ۲ برنامه های مربوط به ارتباط رئیس با Visual Basic^{۱۰} و در پیوست ۳ مشخصات همه چیپهای استفاده شده در این پروژه ارائه شده اند .

اصطلاحات :

در نقاط مختلف این متن از دو عنوان اختصاری به صورت لاتین استفاده شده است که مفهوم فارسی طولانی دارند این دو عنوان در زیر آمده است :

PWM : مدولاسیون پهنای پالس

SPI : ارتباط جانبی سریال با ریز پردازنده های دیگر

^۹ ارتباط سریال با پردازنده های جانبی (Serial Peripheral Interface)

^{۱۰} یکی از محیطها ی برنامه نویسی دیداری شرکت (Microsoft)

فصل دوم : ربات اسکارا

۱-۲- مقدمه

ربات اسکارا^{۱۱} یکی از بازوهای ماهر است که کاربردهای فراوانی در صنعت، آزمایشگاه واکتشاف دارد. مهمترین ونخستین کاربرد ربات اسکارا همان طور که از نام آن پیدا است مونتاژ کاری می باشد. ربات اسکارا یک ربات مونتاژ کار است و به خاطر ساختار دورانی- دورانی- کشویی که دارد ظاهری عامه پسند و پر کاربرد دارد. در شکل (۱-۲) چند نمونه ربات اسکارا را مشاهده می کنیم.

اگر چه از لحاظ نظری بازوهای مکانیکی ماهر سیستمهایی همه کاره و در موارد زیادی قابل بکارگیری هستند، اما در حالت کلی مسائل و ملاحظات اقتصادی باعث می شوند کاری که از بازوی ماهر انتظار می رود در طراحی مکانیکی سیستم اثر گذار باشد [۱]. بنابراین در طراحی باید مشخصه هایی چون اندازه بازوها،

¹¹ SCARA (Selective Compliant Articulated Robot for Assembly)

سرعت مورد نظر برای هر رابط ، و توانایی حمل بار توسط ربات را در نظر گرفت تمامی این ملاحظات بر کیفیت و اندازه فضای کاری ربات (بازوی ماهر) صلبیت سازه و دیگر خصوصیات اساسی ربات تاثیر مستقیم می گذارد ، در این طرح کاربردی سعی شده همه چیز بر اساس استاندارد های رباتهایی که تا کنون ساخته شده است انتخاب و طراحی گردد [۲] .



شکل ۱-۲- نمونه هایی از ربات اسکارا در آزمایشگاه و صنعت

۲-۲- استانداردهای ساخت ربات اسکارا

برای انتخاب ابعاد و مشخصه های مکانیکی ربات مانند طول رابطها ، اینرسی رابطها ، سرعت و شتاب رابطها ، فضای کار هر رابط ، شعاع دسترسی ربات ، حداکثر مقدار بار قابل حمل توسط ربات ، به عنوان نمونه مشخصه های ربات Adept Cobra i800 که در شکل (۲-۲) نشان داده شده است را بررسی می کنیم [۴].

Adept Cobra™ i800



شکل ۲-۲- ربات Adept Cobra i800

۲-۲-۱- خصوصیات ابعادی ربات نمونه Adept Cobra i800

۱. حداکثر شعاع دسترسی

80mm

۲. حداکثر بارگیری

2kg

۳. فضای کار رابطها

$$-105^{\circ} < \theta_1 < 105^{\circ}$$

$$-157.5^{\circ} < \theta_2 < 157.5^{\circ}$$

$$0 < d_3 < 210mm$$

$$-180^{\circ} < \theta_4 < 180^{\circ}$$

۴. سرعت رابطها

$$\dot{\theta}_1 = 360 \text{ deg/sec} \Rightarrow 2\pi \text{ rad/sec}$$

$$\dot{\theta}_2 = 720 \text{ deg/sec} \Rightarrow 4\pi \text{ rad/sec}$$

$$\dot{d}_3 = 1100 \text{ mm/sec} \Rightarrow 1.1 \text{ m/sec}$$

$$\dot{\theta}_4 = 1200 \text{ deg/sec} \Rightarrow \frac{20\pi}{3} \text{ rad/sec}$$

۵. قابلیت تکرار

$$XY = \pm 0.02mm$$

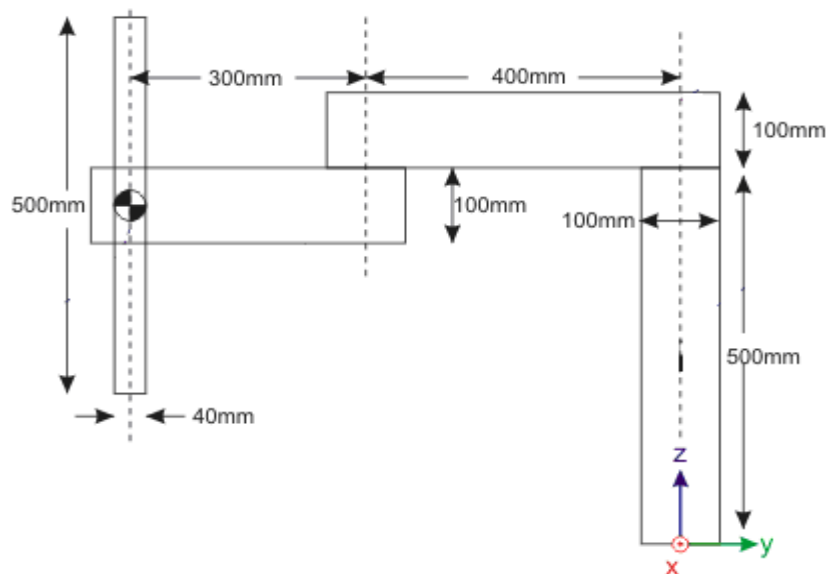
$$Z = \pm 0.003mm$$

$$\theta_4 = \pm 0.019^{\circ}$$

$$43kg$$

۶. وزن کل ربات

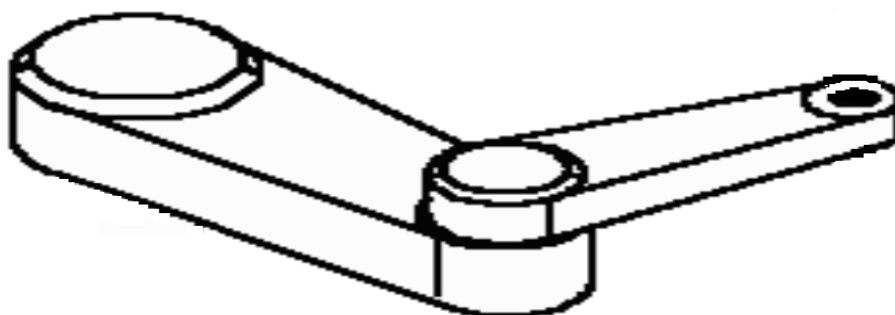
برای به دست آوردن مشخصات ساخت رابطهای ربات از اطلاعات عمومی گونه قدیمی ربات Adept i800 یعنی Adept i600 که در شکل (۳-۲) نشان داده شده است استفاده شد، با توجه به پیچیدگی های مکانیکی بدنه یک ربات تصمیم گرفته شد بدنه از یک فلز با صلبیت مناسب و جرم کم و در عین حال قابلیت تراش خوب ساخته شود که برای این منظور از آلومینیوم خالص استفاده شد و برای استقرار ستون اصلی ربات نیز صفحه ای دایره شکل از جنس آهن به عنوان پایه تعبیه شده که کل ربات روی آن قرار می گیرد این باعث شده که برای انتقال ربات نیاز به باز و بسته کردن پیچهای شاسی نداشته باشیم [۳] و [۲].



شکل ۲-۲- نمونه ای از ابعاد رابط های یک ربات اسکارا

۲-۲-۲- طراحی بدنه

از آنجا که ساخت ربات با ظاهری شبیه به آنچه در مرجع [۴] نتیجه چندین ماه تلاش مهندسين ماهر مکانیک است تصمیم گرفتیم برای اینکه بتوان به راحتی با بعضی از مسائل مکانیکی مانند مثلاً قرار دادن بلبرینگهای کف گرد بین نقطه اتصال موتور و رابط اول و نیز موتور و رابط دوم روبرو شویم مقاطع رابط های اول و دوم را با توجه به ابعاد شکل (۲-۲) به صورتی که در شکل زیر نمایش داده شده طراحی کنیم .



شکل ۲-۳- نمای بالایی رابط اول و دوم ربات اسکارا

ابعاد در نظر گرفته شده برای این ربات به صورت زیر می باشند :

۱- طول رابط ها

$$a_1 = 200mm$$

$$a_2 = 150mm$$

$$d_3 = 150mm$$

۲- وزن رابطها

$$m_0 = 10.1kg$$

$$m_1 = 1.240kg$$

$$m_2 = 0.810kg$$

$$m_3 = 0.615kg$$

۳- وزن ربات

$$\begin{aligned} m_0 + m_1 + m_2 + m_3 &= 10.1 + 1.24 + 0.810 + 0.615 \\ &= 13.215kg \end{aligned}$$

۴- حداکثر شعاع دسترسی

$$a_1 + a_2 = 200 + 150 = 350mm$$

۵- فضای کار رابطها

$$-125^\circ < \theta_1 < 125^\circ$$

$$-135^\circ < \theta_2 < 135^\circ$$

$$\text{با } 0 < d_3 < 150mm$$

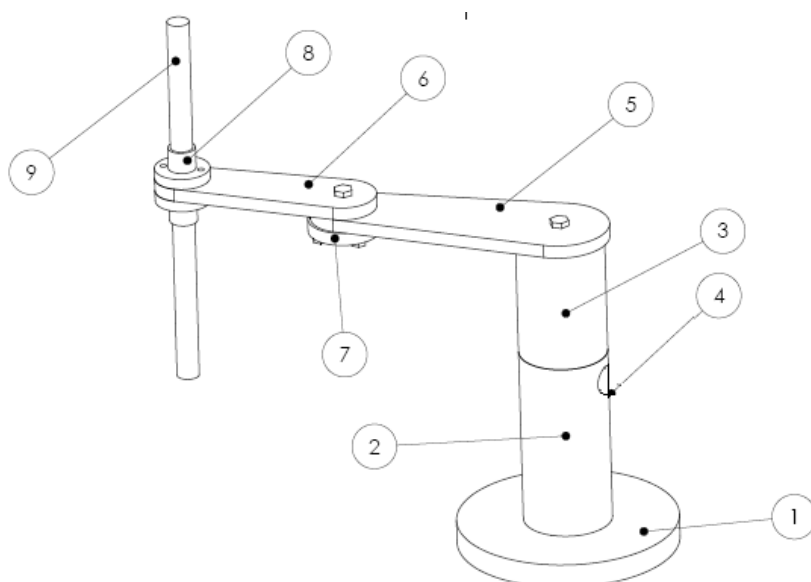
$$-180^\circ < \theta_4 < 180^\circ$$

توجه به مسائل اقتصادی در ساخت ، از آنجاکه قیمت فلز آلومینیوم نسبتا زیاد بوده و تراش و ساخت قطعات

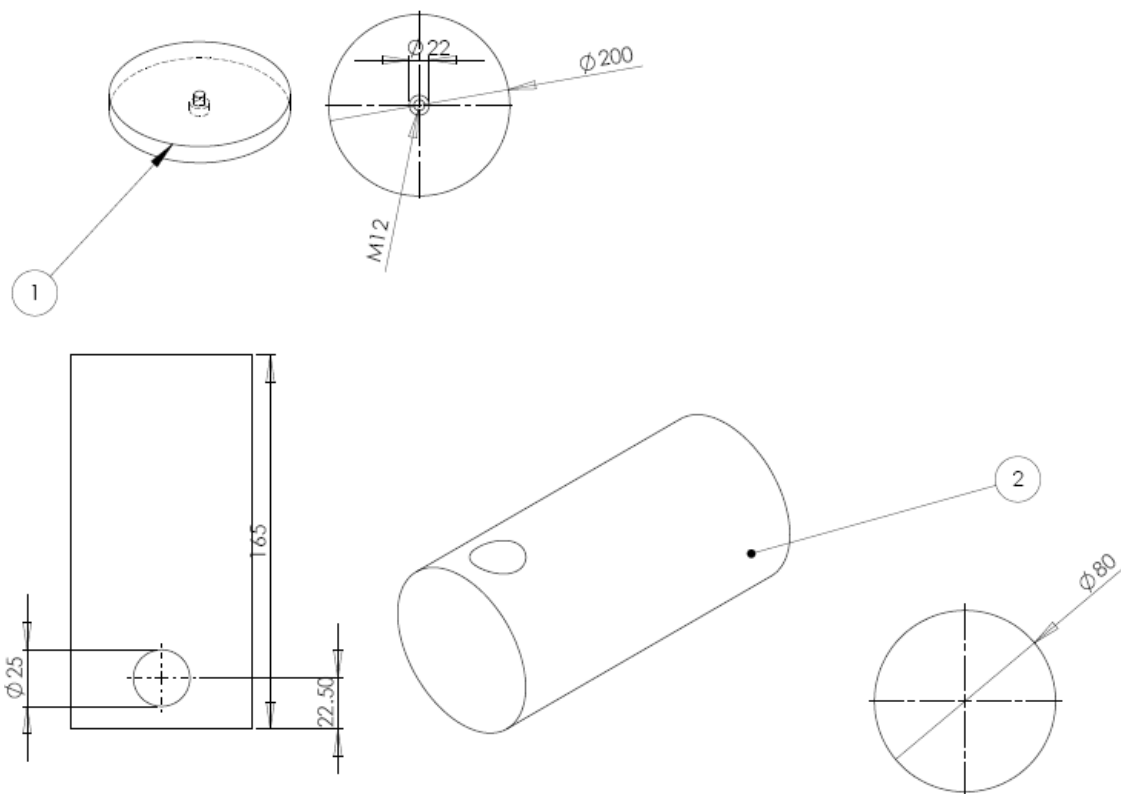
آلومینیومی مشکل و وقت گیر است باید پس از طراحی با نرم افزار^{۱۲} ابتدا یک یا دو نمونه آزمایشی برای باز شدن دید طراحی و در نظر گرفتن جزئیات اتصال قطعات به هم ساخته شود .

۳-۲-۲- طراحی رابطها بوسیله نرم افزار Solid Work

در شکل (۴-۲) شمایی از بدنه ربات به طور کامل مشاهده می شود نتایج طراحی هر بخش را در شکلهای (۴-۲)و(۵و۲)و(۶و۲) ملاحظه می شود روی هر شکل مقیاسها مشخص شده اند .



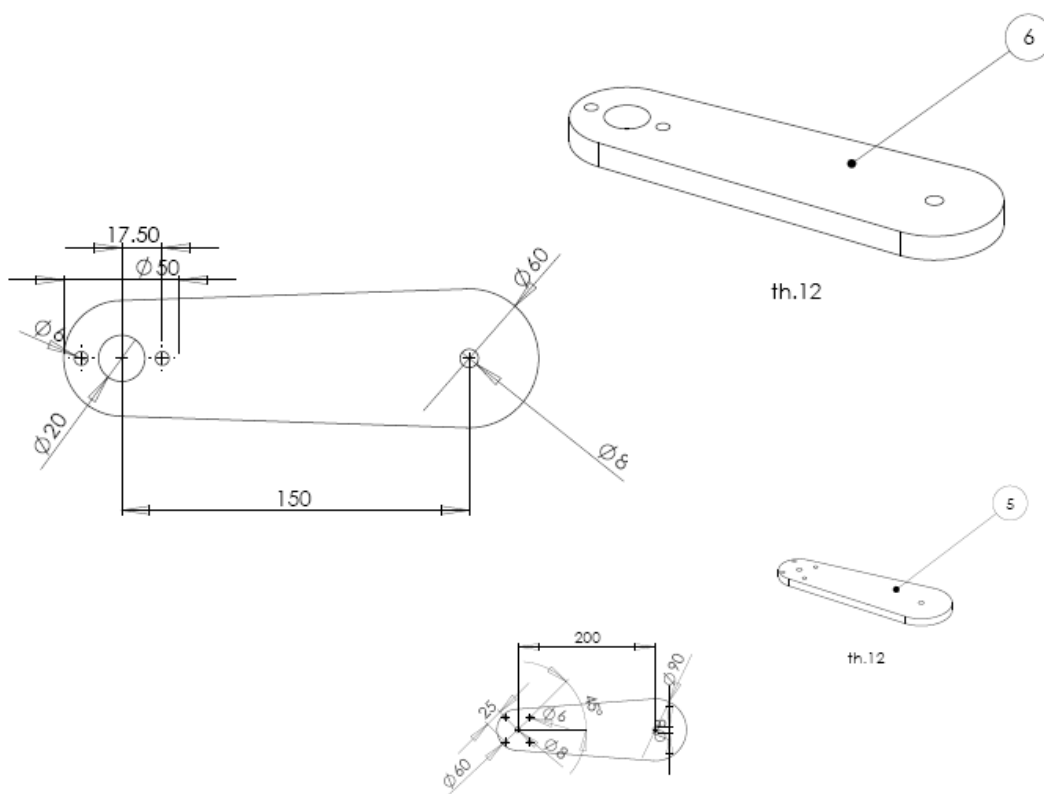
شکل ۴-۲- نمایی از طراحی بدنه اسکارا



شکل ۲-۵- دیسک نگهدارنده بدنه و نیمه پائین رابط صفر

با توجه به شکل (۲-۵) دیسک نگهدارنده دارای شعاع ۲۰۰ میلیمتر بوده جنس آن آهن و سوراخ ۱۲ موجود در مرکز آن برای اتصال رابط صفر می باشد ، همان طور که مشاهده می شود رابط صفر دو تکه در نظر گرفته شده که وظیفه قسمت پائین برای مهار کردن با دیسک از پائین و مهار کردن قسمت بالا که مکانیزم نگهداری موتور درون آن تعبیه شده است می باشد ، این قسمت دارای شعاع ۸۰ میلیمتر و ارتفاع ۱۶۵ میلیمتر می باشد .

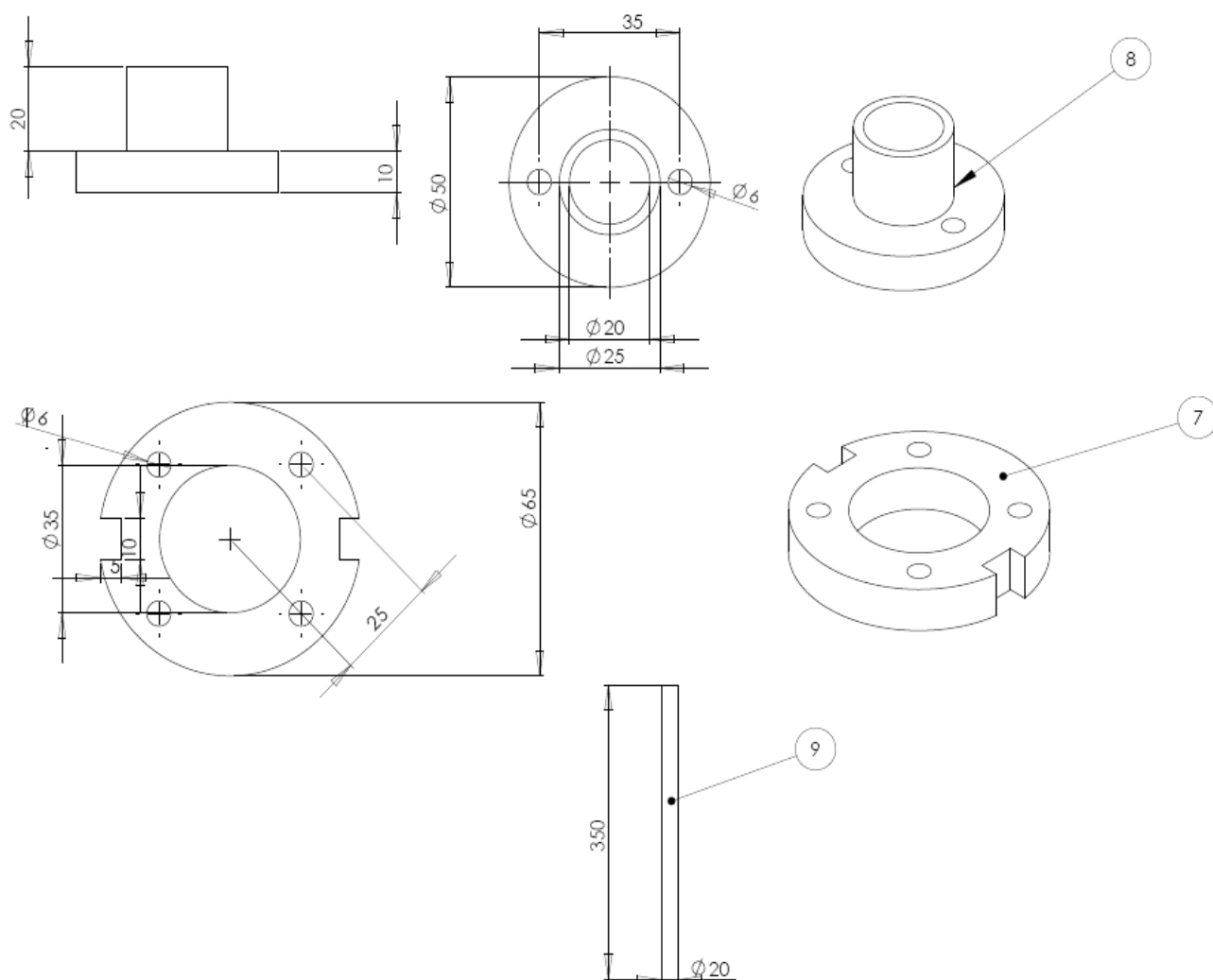
شکل (۲-۶) رابطهای طراحی شده اول و دوم را نشان می دهد ، برای رابط اول فاصله میان سوراخ تعبیه شده در ابتدای رابط برای اتصال به موتور ۱ و نیز سوراخ تعبیه شده در انتهای رابط یک برای اتصال به رابط ۲ مبین طول موثر رابط می باشد این طول با توجه به شکل ۲۰۰ میلیمتر می باشد و شعاع سوراخها ۸ میلیمتر می باشد ، این مشخصات برای رابط ۲ هم در شکل مشاهده می گردد که طول رابط را ۱۵۰ میلیمتر شعاع سوراخ ۱ برابر ۸ میلیمتر و سوراخ ۲ که از داخل آن رابط کشویی عبور می کند ۲۰ میلیمتر می باشد.



شکل ۶-۲- ابعاد طراحی شده برای رابطهای اول و دوم

شکل (۷-۲) را مشاهده کنید قسمتی که با ۷ مشخص شده است قطعه ایست که برای مهار کردن موتور ۲ بر اساس ابعاد این موتور طراحی شده است که از زیر به انتهای رابط اول با ۴ پیچ ۶ متصل شده است ۲ پیچ مغزی ۵ هم از کنار این قطعه موتور ۲ را مهار می کنند ، در قسمت ۸ قطعه ای که را نشان می دهد که ۲ نمونه از آن به صورت وارون روبروی هم در بالا و پائین انتهای رابط دوم به منظور مهار رابط کشویی با ۲ پیچ ۵ متصل می شوند . در قسمت ۹ شکل (۷-۲) رابط سوم مشاهده می شود طراحی مکانیزم حرکت این رابط کاملاً ابتکاری می باشد ، بدین ترتیب که با قرار دادن یک دنده شانه ای در جان رابط ۳ ، موتور ۳ می تواند با

اختلاف فاز مکانی ۹۰ درج این دنده را تحریک و رابط ۳ را درون ۲ مهار اشاره شده در قسمت ۸ شکل (۲-۷) حرکت دهد. ابعاد این رابطها با مقیاس میلیمتر در تمام شکلها ارائه شده است.



۲-۷- طراحی مهارموتور ۲ ومهارهای رابط ۳ و رابط ۳

۲-۳- ساخت نمونه اولیه با اسفنج

می دانیم برای طراحی و ساخت هر قطعه مکانیکی بخصوص قطعات پیچیده همچون قطعات بازوهای ماهر لازم است طراح پس از انجام محاسبات طراحی و مطالعه استانداردها یک یا چند نمونه به عنوان الگوی اولیه با جنس های ارزان قیمت تر بسازد ، دلیل این موضوع آنست که بعضی از محدودیتهای عملی فقط با وجود یک نمونه (لازم نیست نمونه خیلی دقیق باشد) ساده از طرح قابل درک و رویت می باشد .

در این طرح کاربردی نمونه اولیه از جنس اسفنج فشرده که ماده ای ارزان و در عین حال خوش برش و سبک است ساخته شد و از آن به دو نکته مهم به ترتیب زیر پی بردیم که در نمونه ثانویه رفع شد این دو علت بدین قرار هستند :

- محدودیت اندازه موتورها

- نحوه جایگذاری موتورها

دو نکته بالا که محدودیتهای زیادی را از جمله فضای کار موتور ، محفظه ای جهت تبادل حرارت ایجاد شده در موتور ، در نظر گرفتن مکانی برای گرفتن لقی موتورها وسایر مسائل جزئی مهم دیگر که بدون وجود چنین مدلی به سختی قابل روئیت و بررسی هستند .

اما این مدل جوابگوی تمام خواسته های ما از مدل اولیه نیست لذا در مرحله بعدی یعنی مدل ثانویه از تفلون استفاده کرده ایم .

۲-۴- ساخت نمونه ثانویه با تفلون CFC

بدلیل استحکام کم و عدم دقت در سوراخکاری و تراشهای ظریف نمونه ثانویه را از جنس تفلون که ماده ای شبیه پلاستیک اما نسوز و قابل تراش با دستگاه تراش و سوراخکاری است ساخته شد .

نکات و محدودیتهای اساسی و مهمی که از این مدل به دست آمد به قرار زیر است :

۱- انتخاب موتورها با احتساب محدودیتهای مکانی

۲- در نظر گرفتن مکانهای مورد نیاز برای سنسورهای تشخیص موقعیت

۳- مقابله با اصطکاک غیر قابل قبول توسط اورینگها و بلبرینگها

۴- ساخت یاتاقان و بوشهای کوچک برای مفصل کشویی

با توجه به مدل ثانویه و کمک هایی که از آن گرفتیم برای ساخت مدل نهایی آماده شدیم .

از آنجا که معادلات حرکتی ربات در شرایط صلب بودن ربات فوق العاده ساده تر به دست می آیند به همین دلیل چون مدل تفلونی از لحاظ صلبیت یا سفتی خواسته ما را برآورده نمی کند مدل نهایی را از جنس آلومینیوم ساختیم .

۲-۵- ساخت نمونه نهایی از جنس آلومینیوم

از آنجا که طراحی و ساخت ربات مبحث کاملاً تخصصی و بسیار سخت و پیچیده است در نظر داریم خواننده محترم را با تمامی مراحل طراحی و ساخت هر چند به صورت خلاصه و مختصر آشنا سازیم تا شاید بیشتر شاهد پایان نامه ای کاربردی در زمینه مباحث شیرین و جذاب ولی مشکل و طاقت فرسای رباتیک باشیم .

در این بخش تمامی کارهای انجام شده در بخشهای زیر را به صورت فشرده بیان کنیم .

- تراشکاری
- فرزکاری
- سوراخکاری
- قلاویزکاری
- سوهان کاری
- جوشکاری آلومینیوم
- جلاکاری

برشهای اولیه تمامی قطعات اصلی یعنی صفحه دایره شکل زیرین از جنس آهن که حافظ تعادل ربات است ، رابط صفر ربات که محرک مفصل اول بالای آن است ، سیلندر و پیستون مفصل کشویی و کف تراشی ، رو تراشی ، تراش داخلی اجزا دایره شکل و کروی توسط دستگاههای تراش دستی و نیمه اتوماتیک انجام گرفته است .

ساخت رابط های اول و دوم به صورتی که در شکل زیر دیده می شود ماه عسل کار دقیق با دستگاه های فرز دستی می باشد ، از دیگر کارهای انجام شده توسط دستگاه های فرز کف تراشی اجزا با اشکال هندسی غیر دایره ای می باشد مثلا رابط های اول و دوم می باشد .

سوراخکاری و قلاویزکاری اجزا مختلف نیز از کارهای بسیار حساس و ظریف است که نیاز به دقت و تجربه کافی در این زمینه دارد ، عمل سوراخکاری افقی توسط مرغک و تراش و سوراخکاری عمودی بوسیله دریل های ثابت و پایه دار و عمل قلاویزکاری توسط قلاویزهای دستی سه مرحله ای انجام گرفته است .

برای به دست آوردن نتیجه مطلوب از انواع اتصالات (پیچ و رزوه ، پیچ و مهره ، مغزی و ...) لازم است که تمام سطوح مشترک و غیر مشترک صاف و سیقلی و کاملاً یکدست یا باصطلاح ساعت باشند این موضوع نشان دهنده ضرورت سوهان کاری و کف تراشی را نشان می دهد ، کف تراشی با دستگاه تراش و دستگاه فرز و سوهان کاری برای سوراخهای خیلی کوچک با سوهانهای پلاتینی انجام گرفته است .

جوش آلومینیوم برای شکافهای احتمالی سوراخهای اشتباه و نا هماهنگی های احتمالی در زوایای مختلف به همراه تراش نقش لکه گیری را ایفا می کنند .

در مرحله آخر برای زیبایی ظاهری ربات از جلا کاری استفاده شد که آلومینیوم را از حالت کدری درآورده و آنرا شفاف می کند .

۲-۶- کاربردهای ربات اسکارا

با توجه به اطلاعات ارائه شده در [۴] ، این ربات و نمونه های مشابه آن می تواند برای منظوره های زیر در صنعت و آزمایشگاه استفاده شود :

- مونتاژ کاری مکانیکی در کارگاههای مختلف

- حمل مواد و قطعات در خطوط تولید
- پشتیبانی و تعمیر نگهداری ماشین آلات
- بسته بندی در مقیاسهای کوچک
- پیچ و مهره کاری
- و بسیاری از کارهای دیگر که نیاز به اتوماسیون^{۱۳} با دقت و سرعت بالا دارند .

^{۱۳} Automation

فصل سوم بررسی موتورهای DC مغناطیس دائم

۳-۱- مقدمه

امروزه در سیستمهای رباتیکی از محرکهای الکتریکی، بادی، و هیدرولیکی استفاده می شود تمام این محرکها اثرات دینامیکی مخصوص به خود را دارند، با توجه به اینکه اخیرا در سیستمهای ربات برای تولید گشتاور تعمیم یافته τ بیشتر از موتورهای مغناطیس دائم استفاده می گردد لذا به بررسی دینامیک این محرکها می پردازیم.

در بازار قطعات الکترونیک ایران به دست آوردن قطعات بعد از طراحی کار بسیار دشواری است به همین دلیل ابتدا باید از تامین قطعه مطمئن شد بعد اقدام به طراحی کرد.

موتورهای DC که بعنوان محرکها در این پروژه استفاده شده اند موتورهای هستند که برای کاهش سرعت و افزایش گشتاور مجهز به جعبه دنده هستند و مشخصات آنها در زیر آمده است:

۳-۲- چگونگی انتخاب موتورهای DC

برای انتخاب موتور مربوط به هر رابط لازم است مقادیر حداکثر بعضی شتاب رابطها را مشخص کنیم برای این منظور لازم است حداکثر سرعت مورد نیاز برای هر رابط و حداقل زمان مورد نظر برای طی یک مسیر

نمونه را مشخص کنیم ، بهتر است این مقادیر را بیشتر از مقادیر واقعی مثلا ۱,۵ برابر آنها باشد تا در صورت بروز خطا در محاسبات ، موتور انتخابی ضعیف تر از آنچه نیاز است نباشد .

طراحی هندسی رابطها در محیط Solid Work به صورت شکل‌های ارائه شده در فصل دوم و همچنین داشتن مقادیر جرم رابطها این امکان را می دهد که بتوانیم مقادیر اینرسی رابطها را توسط این نرم ابزار به دست آوریم .

روش کار بدین ترتیب است که ابتدا مقدار حداکثر سرعت مورد نظر برای رسیدن به وسط مسیر نمونه ای درجه دو ونیز حداقل زمان پیمایش این مسیر را تعیین و سپس از روی نقطه شروع و نقطه وسط مسیر حداکثر شتاب حرکت هر رابط را به دست می آوریم با توجه به رابطه (۱-۳) می توانیم حداکثر گشتاور مورد نیاز برای موتور هر رابط را به دست آوریم (پیشنهاد می شود ابتدا فصل ۷ که مربوط به طراحی مسیر است مطالعه شود).

$$\tau = I\alpha \quad (1-3)$$

که در آن α شتاب دورانی و I اینرسی رابط را مشخص می کند .

در نرم افزار Solid Work برای اینرسی رابطها اطلاعات مختلفی به صورت $I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xy}$

ارائه می گردد که از میان آنها I_{ii} ها مبین اینرسی حول محور i می باشد که برای رابط های اول و دوم تنها از I_{zz} آنها استفاده می شود این مقادیر در زیر ارائه شده است :

$$\begin{aligned} I_{zz1} &= 0.058 \frac{kgm}{rad} \\ I_{zz2} &= 0.014 \frac{kgm}{rad} \\ m_3 &= 0.615 kg \end{aligned} \quad (2-3)$$

حال چنانچه مقادیر حداکثر سرعت و حداقل زمان پیمایش مسیر نمونه درجه دوم برای پیمایش ۱۸۰ درجه یا π رادیان را به صورت زیر تعریف می کنیم :

$$\omega_{1max} = 15rpm = \frac{\pi}{2} \frac{rad}{s}, t_{1min} = 10s$$

$$\omega_{2max} = 20rpm = \frac{2\pi}{3} \frac{rad}{s}, t_{2min} = 10s$$

(۳-۳)

$$\begin{aligned}
\alpha_{1\max} &= \frac{2\omega_{1\max}}{t_{1\min}} = \frac{\pi}{10} \frac{rad}{s^2} \\
\alpha_{2\max} &= \frac{2\omega_{2\max}}{t_{2\min}} = \frac{4\pi}{30} \frac{rad}{s^2} \\
a_{3\max} &= \frac{2v_{3\max}}{t_{3\min}} = \frac{0.05}{5} \frac{m}{s^2}
\end{aligned}
\tag{۴-۳}$$

با توجه به مقادیر حداکثر شتاب می توانیم طبق روابط زیر حداکثر گشتاور هر رابط را به دست آوریم :

$$\begin{aligned}
\tau_{1\max} &= I_{1zz} \alpha_{1\max} = 0.58 \times \frac{\pi}{10} = 1.79 \frac{kgm}{s^2} \Rightarrow \tau_{1\max} = 0.179 \text{ N.m} \\
\tau_{2\max} &= I_{2zz} \alpha_{2\max} = 0.14 \times \frac{4\pi}{30} = 0.079 \frac{kgm}{s^2} \Rightarrow \tau_{2\max} = 0.079 \text{ N.m} \\
F_3 &= m_3 a_{3\max} = 0.615 \times 0.01 = 0.006 \frac{kgm}{s^2} \Rightarrow F_{3\max} = 0.006 \text{ N.m}
\end{aligned}
\tag{۵-۳}$$

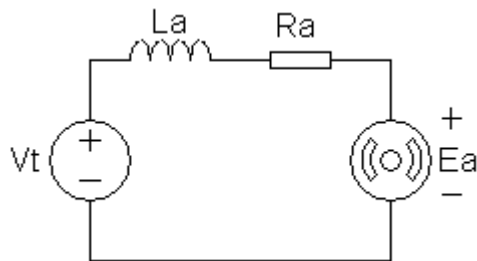
لذا موتورهای را به صورت زیر انتخاب می کنیم :

$$\begin{aligned}
V_1 &= 12V, I_{a1} = 900mA, \omega_1 = 15Rpm, r_1 = \frac{1}{250}, \tau_1 = 20N.cm \\
V_2 &= 12V, I_{a2} = 400mA, \omega_2 = 60Rpm, r_2 = \frac{1}{96}, \tau_2 = 10N.cm \\
V_3 &= 6V, I_{a3} = 200mA, \omega_3 = 60Rpm, r_3 = \frac{1}{82}, \tau_3 = 3N.cm \\
V_4 &= 6V, I_{a4} = 200mA, \omega_4 = 60Rpm, r_4 = \frac{1}{82}, \tau_4 = 3N.cm
\end{aligned}
\tag{۶-۳}$$

۳-۳- دینامیک موتورهای DC مغناطیس دائم

موتورهای DC مغناطیس دائم اصولاً بر این اساس کار می کنند که یک فلز هدایتگر جریان درون یک میدان مغناطیسی ثابت که توسط استاتور موتور که به شکل آهنربا می باشد قرار می گیرد و نیروی $F = \phi I$ به آن وارد می شود چون میدان مغناطیسی استاتور ثابت است می توان شار مغناطیسی استاتور را ثابت در نظر گرفت این شار که به صورت شعاعی است با عبور جریان از رتور باعث ایجاد گشتاور روی آن شده و رتور را می چرخاند در قسمتهای بعدی این بخش به بررسی جزئیات این موضوع می پردازیم.

به طور کلی برای یک موتور DC مغناطیس دائم می توان مدل مداری به صورت شکل (۳-۱) در نظر گرفت:



شکل ۳-۱- موتور DC تحت ولتاژ نامی

در این مدل ۲ پارامتر ثابت و ۳ متغیر به قرار زیر وجود دارند :

V_t : ولتاژ ترمینال موتور

E_a : ولتاژ ضد محرکه القایی

I_a : جریان آرمیچر

R_a : مقاومت اهمی سیم پیچهای آرمیچر

L_a : ضریب خود القایی (اندوکتانس) آرمیچر

K_t : ثابت گشتاور

K_b : ثابت ضد محرکه القایی

J_m : ممان اینرسی آرمیچر

J_l : ممان اینرسی بار و گیربکس

B_m : ضریب میرایی آرمیچر

B_l : ضریب میرایی بار

n : نسبت سرعت خروجی به ورودی گیربکس

W_a : سرعت زاویه ای آرمیچر

W_g : سرعت زاویه ای گیربکس

R : مقاومت اهمی معادل مسیر و آرمیچر

T_e : گشتاور الکترومغناطیس

J : ممان اینرسی تحویلی به محور آرمیچر

B : ضریب میرایی معادل تحویلی به محور آرمیچر

به منظور تعیین مدل ریاضی موتور DC مغناطیس دائم نیاز است که پارامترهای R_a ، L_a شناسایی شوند .

این شناسایی منوط به انجام یک سری آزمایشهای تکراری و ساده می باشند که به صورت زیر هستند :

۳-۴- اندازه گیری R_a

برای شناسایی مقاومت سیم پیچهای رتور ، را در حالت قفل قرار داد. و همزمان جریان R_a و ولتاژ V_t موتور را اندازه گیری کنیم تا جاییکه جریان موتور نزدیک به جریان نامی موتور شود حال با اندازه گیری ولتاژ می توان از تقسیم این ولتاژ به جریان در حالت رتور قفل مقاومت R_a را به صورت زیر به دست آورد.

$$R_a = \frac{V_t}{I_a} \quad (۷-۳)$$

$$R_{a1} = \frac{V_{t1}}{I_{a1}} = \frac{2.9}{0.480} = 6\Omega$$

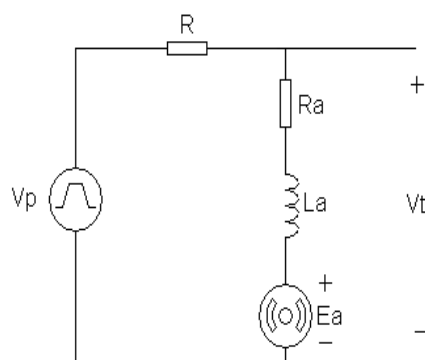
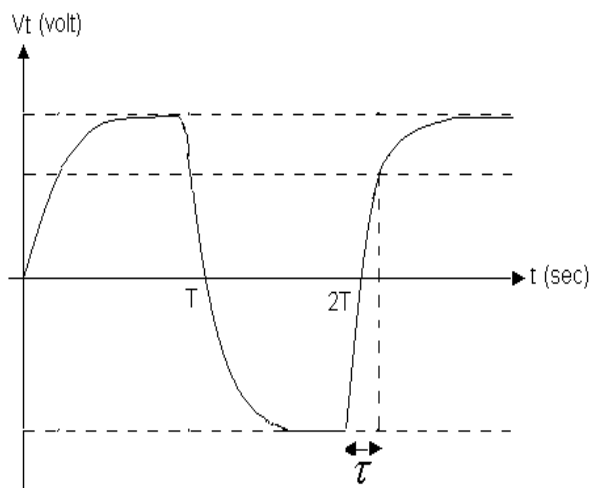
$$R_{a2} = \frac{V_{t2}}{I_{a2}} = \frac{4.77}{0.232} = 20.5\Omega$$

$$R_{a3} = \frac{V_{t3}}{I_{a3}} = \frac{1.4}{0.102} = 13.72\Omega$$

$$R_{a4} = \frac{V_{t4}}{I_{a4}} = \frac{1.4}{0.102} = 13.72\Omega$$

۳-۵- اندازه گیری L_a

برای اندازه گیری ضریب خود القایی یا همان اندوکتانس موتور نیاز به مداری داریم تا در فواصل مشخص رفتار القایی مدار را مشاهده کنیم . یکی از روشهای موجود اعمال پالسهای با فرکانس هایی در محدوده ۵-۲۰ هرتز به دو سر موتور می باشد که رتور آن قفل بوده و در نتیجه $E_a = 0$ می باشد . در این حالت باید یک مقاومت R که بهتر است حداقل ۱۰ برابر R_a انتخاب گردد . برای راحتتر و دقیقتر کردن مراحل شناسایی با موتور و منبع تولید پالس سری کرد . برای این منظور مدار شکل (۳-۲) پیشنهاد می شود . در این حالت ولتاژ دو سر موتور را روی اسیلوسکپ مشاهده می کنیم که نموداری به صورت شکل (۳-۳) دارد .



شکل ۳-۲- موتور DC متصل به پالس ژنراتور

شکل ۳-۳- نمودار ولتاژ دو سر موتور موتور Vt

این نمودار حاوی اطلاعات مفیدی در مورد ثابت زمانی مدار می باشد از آنجا که ثابت زمانی این مدار مدت زمان رسیدن ولتاژ موتور به ۰,۶۳ مقدار نهایی آنست که از روی شکل بالا به دست خواهد آمد با توجه به معادلات ثابت زمانی مدار آن است که از روی شکل بالا به دست خواهد آمد با توجه به معادلات ثابت زمانی مدار $R-L$ سری به راحتی L_a به صورت زیر به دست می آید :

$$L_a = \tau(R_a + R) \quad (۸-۴)$$

در این موتورها چون ثابت زمانی الکتریکی که همان نسبت اندوکتانس سیم پیچهای روتور به مقاومت اهمی این سیم پیچها می باشد بسیار کوچکتر از ثابت زمانی مکانیکی یا نسبت اینرسی محرکه و سیستم انتقال نیروی چرخ دنده ای به ضریب اصطکاک های ناشی از یاتاقانها و جاروبکها می باشد لذا می توان از L_a صرفنظر کرد که این مهم خیلی محاسبات را آسانتر می کند زیرا مرتبه سیستم دینامیکی را کاهش می دهد . معمولا مقدار L_a خیلی کوچک و قابل صرفنظر کردن است .

$$V_1 = 12V, V_{0.63} = 5 \times 1.51V, \tau_1 = 0.1 \times 2 \times 10^{-3} mS, R_{a1} = 6\Omega, R = 0\Omega$$

$$L_{a1} = 2 \times 10^{-4} \times 6 = 1.2mH$$

$$V_2 = 12V, V_{0.63} = 5 \times 1.51V, \tau_2 = 0.2 \times 2 \times 10^{-3} mS, R_{a2} = 20.5\Omega$$

$$R = 0\Omega, L_{a2} = 4 \times 10^{-4} \times 20.5 = 8.3mH$$

$$V_3 = 6, V_{0.63} = 2 \times 1.6V, \tau_3 = 0.1 \times 5 \times 10^{-4} mS, R_{a3} = 13.72\Omega, R = 0\Omega$$

$$L_{a3} = 5 \times 10^{-5} \times 13.72 = 0.69mH$$

$$V_4 = 6V, V_{0.63} = 2 \times 1.6V, \tau_4 = 0.1 \times 5 \times 10^{-4} mS, R_{a4} = 13.72\Omega, R = 0\Omega$$

$$L_{a4} = 5 \times 10^{-5} \times 13.72 = 0.69mH$$

۳-۶- اندازه گیری K_b

$E_a = K_b W_a$ است. با یک آزمایش ساده هنگامی که ولتاژ نامی دو سر موتور قرار داده شود $V_t = V_{nominal}$ آنگاه جریان موتور I_a و سرعت موتور از روی شفت گیربکس W_g نیز اندازه گیری می شود حال W_a به راحتی با ضرب نسبت تبدیل چرخ دنده های موتور به گیربکس n به دست می آید W_a, I_a, V_t را داریم به راحتی K_b محاسبه می شود.

(۳-۹)

$$W_a = n.W_g$$

$$E_a = K_b.W_a$$

$$V_t = R_a I_a + E_a$$

(۳-۱۰)

$$K_b = \frac{E_a}{W_a} = \frac{V_t - R_a I_a}{W_a}$$

$$V_{t1} = 12V, I_{a1} = 0.1A, R_{a1} = 6\Omega, w_{g1} = 17.3rpm = 452.9rads^{-1}$$

$$K_{b1} = \frac{V_{t1} - R_{a1}I_{a1}}{w_{g1}} = \frac{12 - 0.6}{452.9} = 0.025 V.S/rad$$

$$V_{t2} = 12V, I_{a2} = 0.045A, R_{a2} = 20.5\Omega, w_{g2} = 34rpm = 341.8rads^{-1}$$

$$K_{b2} = \frac{V_{t2} - R_{a2}I_{a2}}{w_{g2}} = \frac{12 - 0.923}{452.9} = 0.032 V.S/rad$$

$$V_{t3} = 6V, I_{a3} = 0.012A, R_{a3} = 13.72\Omega, w_{g3} = 42rpm = 219.9rads^{-1}$$

$$K_{b3} = \frac{V_{t3} - R_{a3}I_{a3}}{w_{g3}} = \frac{6 - 0.165}{219.9} = 0.026 V.S/rad$$

$$V_{t4} = 6V, I_{a4} = 0.012A, R_{a4} = 13.72\Omega, w_{g4} = 42rpm = 219.9rads^{-1}$$

$$K_{b4} = \frac{V_{t4} - R_{a4}I_{a4}}{w_{g4}} = \frac{6 - 0.165}{219.9} = 0.026 V.S/rad$$

۳-۷- اندازه گیری K_t

برای محاسبه ثابت گشتاور باید از قانون توانهای مکانیکی و الکتریکی در موتور ها استفاده کنیم ،

$$\tau_e = K_t I_a \quad (۱۱-۳)$$

توان مکانیکی حاصله از این گشتاور به صورت زیر است :

(۱۲-۳)

$$P = \tau_e W_a$$

و معادل الکتریکی این توان به صورت :

$$P = E_a I_a$$

(۱۳-۳)

در این راستا چون داریم :

$$\tau_e W_a = E_a I_a \Rightarrow K_t I_a W_a = K_b W_a I_a$$

(۱۴-۳)

لذا $K_t = K_b$ می باشد .

$$K_{t1} = K_{b1} = 0.025 V.S / rad$$

$$K_{t2} = K_{b2} = 0.032 V.S / rad$$

$$K_{t3} = K_{b3} = 0.026 V.S / rad$$

$$K_{t4} = K_{b4} = 0.026 V.S / rad$$

۳-۸- اندازه گیری ضریب میرایی موتور B

بدون هیچ مقدمه ای معادلات دینامیکی موتور و بار را به صورت زیر می نویسیم :

$$\tau_e = K_t I_a$$

(۱۵-۳)

$$E_a = K_b W_a$$

$$\tau_e = J \ddot{q} + B \dot{q} + r \tau_l = J \dot{W}_a + B W_a + r \tau_{\text{مقاومت}}$$

$$V = R_a I_a + E_a$$

$$J = J_m + n^2 J_l$$

$$(۱۶-۳)$$

$$(۱۷-۳)$$

$$(۱۸-۳)$$

$$(۱۹-۳)$$

$$(۲۰-۳)$$

با توجه به اینکه در مراحل قبلی ثابتهای R_a, L_a, K_b, K_t را به دست آوردیم برای به دست آوردن B کافی است روی موتور در سرعت ثابت آزمایش زیر را انجام دهیم :

$$I_a = \frac{V - E_a}{R_a} = \frac{V - K_b W_a}{R_a} \quad (۲۱-۳)$$

$$\tau_e = K_t I_a = \frac{K_t}{R_a} (V - K_b W_a) \quad (۲۲-۳)$$

$$\frac{K_t}{R_a} (V - K_b W_a) = (Js + B) W_a$$

$$V = \frac{R_a}{K_t} (Js + (B + K_b)) W_a$$

$$V = \left(\frac{R_a}{K_t} Js + \left(\frac{BR_a}{K_t} + K_b \right) \right) W_a$$

$$(۲۳-۳)$$

چنانچه W_a ثابت باشد آنگاه مشتق آن یعنی $sW_a = 0$ خواهد شد و B به راحتی محاسبه خواهد شد :

(۲۴-۳)

با گرفتن تبدیل لاپلاس از (۵-۴) با فرض پله بودن $V(s)$ خواهیم داشت :

$$V = \left(\frac{BR_a}{K_t} + K_b \right) W_a$$

$$V(s) = \frac{V}{s}$$

$$V = \left(\frac{R_a J}{R_a B + K_t K_b} s + 1 \right) W_a(s) \quad (۲۵-۳)$$

$$B = \left(\frac{V}{W_a} - K_b \right) \frac{K_t}{R_a} \quad (۲۶-۳)$$

$$V_{t1} = 12V, R_{a1} = 6\Omega, w_{g1} = 452.9 \text{ rads}^{-1}, K_{b1} = K_{t1} = 0.025 \text{ VS/Rad}$$

$$B_1 = \left(\frac{V_{t1}}{w_{g1}} - K_{b1} \right) \frac{K_{t1}}{R_{a1}} = 8.3 \times 10^{-6} \text{ Kg.m/Rad.S}$$

$$V_{t2} = 12V, R_{a2} = 20.5\Omega, w_{g1} = 341.8 \text{ rads}^{-1}, K_{b2} = K_{t2} = 0.032 \text{ VS/Rad}$$

$$B_2 = \left(\frac{V_{t2}}{w_{g2}} - K_{b2} \right) \frac{K_{t2}}{R_{a2}} = 3.1 \times 10^{-6} \text{ Kg.m/Rad.S}$$

$$V_{t3} = 6V, R_{a3} = 13.72\Omega, w_{g3} = 219.9 \text{ rads}^{-1}, K_{b3} = K_{t3} = 0.026 \text{ VS/Rad}$$

$$B_3 = \left(\frac{V_{t3}}{w_{g3}} - K_{b3} \right) \frac{K_{t3}}{R_{a3}} = 2.43 \times 10^{-6} \text{ Kg.m/Rad.S}$$

$$V_{t4} = 6V, R_{a4} = 13.72\Omega, w_{g4} = 219.9 \text{ rads}^{-1}, K_{b4} = K_{t4} = 0.026 \text{ VS/Rad}$$

$$B_4 = \left(\frac{V_{t4}}{w_{g4}} - K_{b4} \right) \frac{K_{t4}}{R_{a4}} = 2.43 \times 10^{-6} \text{ Kg.m/Rad.S}$$

۳-۹- اندازه گیری ممان اینرسی J

از رابطه (۳-۲۵) خواهیم داشت :

$$W_a(s) = \frac{K_t V}{R_a J s^2 + (R_a B + K_t K_b) s} \quad (۳-۲۷)$$

بنابراین :

$$W_a(t) = \frac{K_t V}{R_a B + K_t K_b} \left(1 - e^{-\frac{t}{\left(\frac{R_a J}{R_a B + K_t K_b} \right)}} \right) \quad (۳-۲۸)$$

$$K_m = \frac{K_t V}{R_a B + K_t K_b} \quad , \quad \varphi = \frac{R_a J}{R_a B + K_t K_b} \quad (۳-۲۹)$$

$$W_a(t) = K_m (1 - e^{-\frac{t}{\varphi}}) \quad (۳-۳۰)$$

در این حال با به دست آوردن ۳ نمونه سرعت در $t = 0, t = t_1, t = \infty$ براحتی φ و در نتیجه J به دست خواهد آمد .

در این بخش چگونگی شناسایی و محاسبه پارامترهای ثابت موتورهای DC مغناطیس دائم اعم از

R_a, L_a, K_b, K_t, J, B از نظرتان گذشت .

$$K_{m1} = \frac{K_{t1}V_1}{R_{a1}B_1 + K_{t1}K_{b1}} = 480 \quad , \quad w_{a1}(t) = 480(1 - e^{\frac{-t}{\varphi}})$$

$$t_1 = 25ms, w_a(t_1) = 400 \Rightarrow \varphi_1 = \frac{-t_1}{\ln(1 - \frac{400}{480})} = 0.014$$

$$\varphi_1 = \frac{R_{a1}J_1}{R_{a1}B_1 + K_{t1}K_{b1}} \Rightarrow 0.014 = \frac{6}{625.2 \times 10^{-6}} J_1 \Rightarrow J_1 = 1.46 \times 10^{-6} \frac{kgm}{rad}$$

$$K_{m2} = \frac{K_{t2}V_2}{R_{a2}B_2 + K_{t2}K_{b2}} = 375 \quad , \quad w_{a2}(t) = 375(1 - e^{\frac{-t}{\varphi}})$$

$$t_1 = 5ms, w_a(t_1) = 312.5 \Rightarrow \varphi_2 = \frac{-t_1}{\ln(1 - \frac{312.5}{375})} = 0.0027$$

$$\varphi_2 = \frac{R_{a2}J_2}{R_{a2}B_2 + K_{t2}K_{b2}} \Rightarrow 0.0027 = \frac{20.5}{1024.1 \times 10^{-6}} J_2 \Rightarrow J_2 = 0.135 \times 10^{-6} \frac{kgm}{rad}$$

$$K_{m3} = \frac{K_{t3}V_3}{R_{a3}B_3 + K_{t3}K_{b3}} = 220 \quad , \quad w_{a3}(t) = 220(1 - e^{\frac{-t}{\varphi}})$$

$$t_1 = 15ms, w_a(t_1) = 192 \Rightarrow \varphi_3 = \frac{-t_1}{\ln(1 - \frac{192}{220})} = 0.0073$$

$$\varphi_3 = \frac{R_{a3}J_3}{R_{a3}B_3 + K_{t3}K_{b3}} \Rightarrow 0.0073 = \frac{13.72}{709.2 \times 10^{-6}} J_3 \Rightarrow J_3 = 0.372 \times 10^{-6} \frac{kgm}{rad}$$

$$J_4 = J_3 = 0.372 \times 10^{-6} \frac{kgm}{rad}$$

۳-۱۰- مونتاژ رابط ها و موتورها

پس از طراحی و ساخت رابطها بر طبق استانداردهایی که بیان شد نوبت به مونتاژ اولیه بعضی از قسمتهای ربات می رسد . از آنجا که با توجه به نیازها و غنی نبودن بازار توانستیم موتور گیربکسهای تقریبا مناسبی از لحاظ گشتاور و سرعت انتخاب کنیم البته شایان ذکر است که چون گیربکسهای تمامی این موتورها از نوع خطی بدون اختلاف فاز مکانی (غیر حلزونی) بودند در خروجی این گیربکسها لقی های مختصری دیده می شوند که باعث بوجود آمدن خطا می شود . لقی و خلاصی چرخنده ها یکی از مشکلات بزرگ در سیستمها ی مکانیکی می باشد ، ترم لقی ترمی غیر خطی و غیر قابل کنترل می باشد تنها راه رهایی از مشکلات ناشی از لقی ، تهیه موتور گیر بکسهای با کیفیت بالا و بدون لقی است. به عنوان یک راه حل ساده برای کاهش اثرات لقی می توان راه حل زیر را ارائه کرد :

۳-۱۰-۱- کاهش اثرات لقی بوسیله اورینگها و اصطکاک توسط بلبرینگ ها و بوش ها

چنانچه برای کاهش لقی افزایش اصطکاک را در سیستم بپذیریم می توان با استفاده از ایجاد اصطکاکهای یکنواخت عمودی در سیستم دوار بوسیله اورینگها اثرات این لقی را تا حدی کاهش داد اما به هر صورت باز هم خطای ناشی از لقی خود را نشان خواهد داد، دراین پروژه گیربکس رابط دوم دارای لقی بود لذا از این تکنیک استفاده کرده ایم ب طوری که در قسمت انتقال نیروی موتور روی محیط بیرونی قسمت واسط دو شیار ایجاد کردیم و دو اورینگ درون آنها جایگذاری شد این اورینگ ها در یک محیط روغن اندود اصطکاکی یکنواخت در حرکت رابط دوم بوجود آوردند که منجر به کاهش مختصری در لقی شد . اما برای رهایی از این مشکل اساسی راهی جز انتخاب موتور-گیربکسهای مرغوب نداریم .

پس از مونتاژ پایه ، رابط اول ، رابط دوم و سوم در مدل نهایی دیدیم که یک سری اصطکاک ناشی از مشکلات هم محوری در سیستم نمایان گشته است که نتیجه لقی موتورها در مکان تعبیه شده برای آنها و لقی گیربکس کوپل شده به موتورها و نیز اصطکاکات ناشی از برخورد سطوح رابطها و سطوح مهار کننده

موتورها به همدیگر می باشد. با یک مکانیزم پیچیده مکانیکی توسط بلبرینگهای درون گرد و کف گرد توانستیم لقی های ناشی از اتصال رابطها (با لقی شافت گیر بکس متفاوت است) و اصطکاک های هم محور با شافت گیر بکس را به طور چشمگیری کاهش دهیم ، نمونه ای از بلبرینگها و پوشه های بکار رفته در بدنه اصلی ربات در شکل (۴-۴) نشان داده شده است.



شکل ۳-۴- بلبرینگهای کف گرد و تو دلی



شکل ۳-۵- نمایی از ربات اسکارا SCARA (تکمیل شده)

فصل چهارم الکترونیک ربات

۴-۱- مقدمه

در این بخش که از جذاب ترین و دقیق ترین بخشهای طراحی ربات می باشد سعی داریم کلیه جزئیات مربوط به طراحی الکترونیک ربات را بررسی کنیم ، ابتدا در مورد چگونگی انتخاب ریز پردازنده های ربات

تحت عنوان چرا ATMEGA32؟، می پردازیم سپس در مورد طراحی ابتدایی ترین نیاز سیستمهای الکترونیکی یعنی منبع تغذیه مطالبی بیان می کنیم به دنبال آن در مورد راه اندازی محرکها به روش PWM و انواع طراحیها و مدارات راه انداز ترانزیستوری و چیپهای راه انداز و طرز کار آنها مباحثی مطرح می گردد و نیز در انتها چگونگی استفاده و راه اندازی سنسورهای تشخیص موقعیت را بررسی کرده و مسائل مربوط به طراحی مدارات الکترونیک تشخیص موقعیت مبتنی سنسورهای مادون قرمز و همچنین گونه های صنعتی آنها برای می پردازیم .

۴-۲- چرا ATMEGA32 ؟

با توجه به چند وظیفه ای^{۱۴} بودن ریز پردازنده این ربات باید برای انجام وظایف مختلف مانند ارتباطهای خارجی چون سریال ، SPI و یا باس موازی و نیز عملیات شمارش وقایع خارجی و پاسخ به وقفه های^{۱۵} مختلف خارجی و داخلی مانند وقفه اتمام کار مبدل آنالوگ به دیجیتال^{۱۶} داخلی و همچنین تولید پالسهای PWM برای راه اندازی موتورها و سایر وظایف یک میکرو کنترلر از کنترلی استفاده شود که علاوه بر امکانات بالا در مصالحه قیمت و کیفیت شرایط مذکور را احراز کند برای داشتن قابلیتهای بالا این کنترلر باید حداقل دارای امکانات زیر باشد :

- داشتن ۳ وقفه خارجی
- مبدل آنالوگ به دیجیتال داخلی
- کانتر ۸ بیتی برای شمارش وقایع ناشی از عملکرد سنسورهای موقعیت
- تایمر ۱۶ بیتی برای به دست آوردن زمان و سرعت حرکت رابطهای ربات
- برقراری ارتباط با پروتکلهای سریال و SPI

^{۱۴} Multi Task

^{۱۵} External Interrupts

^{۱۶} Analog to Digital Converter (A/D)

و امکانات معمول دیگر برای این منظور از چیپ ATMEGA32 که تواناییها و قابلیتها آن به صورت خلاصه در زیر مطرح شده و از طرفی تمام نیازهای ما را بر آورده می کند استفاده شده است شمای ظاهری این کنترلر در شکل (۱-۴) آمده است.

■ ۱- تعداد ۱۳۱ دستور با کارایی بالا که معمولا در یک کلاک سیکل انجام میشوند

■ ۲- فرکانس کار 1-16 MHZ

■ ۳- حافظه قابل برنامه ریزی FLASH ROM (32kb)

■ ۴- حافظه SRAM (2kb)

■ ۵- حافظه EEPROM (1024 byte)

■ ۶- دو عدد تایمر کانتر ۸ بیتی و یک تایمر کانتر ۱۶ بیتی

■ ۷- تعداد ۴ کانال PWM

■ ۸- تعداد ۸ کانال A/D ۱۰ بیتی

■ ۹- مقایسه کننده آنالوگ داخلی

■ ۱۰- دو کانال تفاضلی با کنترل گین ۲۰۰/۱۰/۱

■ ۱۱- ارتباط با پروتکل ۲ سیمه TWO WIRE

■ ۱۲- ارتباط با پروتکل ۱ سیمه ONE WIRE

■ ۱۳- ارتباط SPI (in system programming)

■ ۱۴- ارتباط سریال SPI (slave/master)

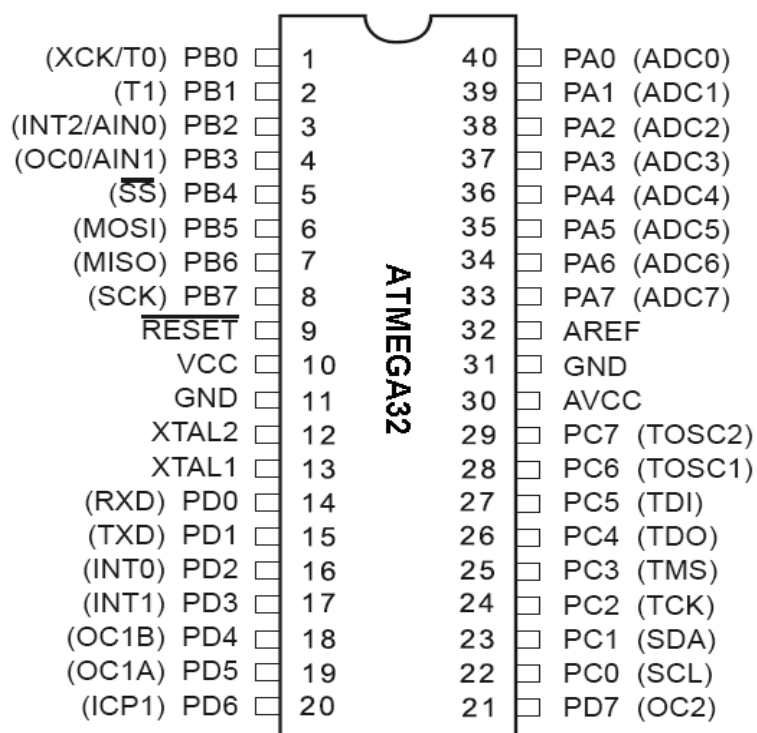
■ ۱۵- قابل برنامه ریزی USART

■ ۱۶- دارای اسیلاتور مجزا RTC (real time clock)

■ ۱۷- ولتاژ کاری ۲,۷ الی ۵,۵ ولت

■ ۱۸- تعداد ۳۲ خط ورودی و خروجی

■ ۱۹- عملکرد ثابت و توان مصرفی پایین



شکل ۴-۱- میکرو کنترلر ATMEGA23

۴-۳- طراحی منبع تغذیه ربات

در بخش طراحی و ساخت قسمت‌های الکترونیکی هر پروژه ای مهمترین و اصلی ترین بخش که باید قبل از هر بخش دیگر طراحی شود منبع تغذیه سیستم است زیرا عملکرد دقیق یک سیستم رابطه مستقیم با تغذیه مناسب آن سیستم دارد .

در این پروژه ۳ سطح ولتاژ مختلف نیاز داریم زیرا موتورهای انتخابی برای این پروژه ۱۲ و ۶ ولتی هستند و از طرفی برای قسمت دیجیتال سیستم کنترل به ولتاژ ۵ ثابت احتیاج داریم اما با یک پیش بینی ساده چون می خواهیم از تکنیک PWM برای راه اندازی موتورها استفاده کنیم لذا می توان تنها به ۲ سطح ولتاژ ۱۲ و ۵ اکتفا کرد .

اما مسئله اصلی قابلیت جریان دهی منبع تغذیه است . به صورت سرانگشتی اگر تمامی موتورها جریان حداکثر خود را بکشند داریم :

$$I_1 + I_2 + I_3 + I_4 + I_d =$$

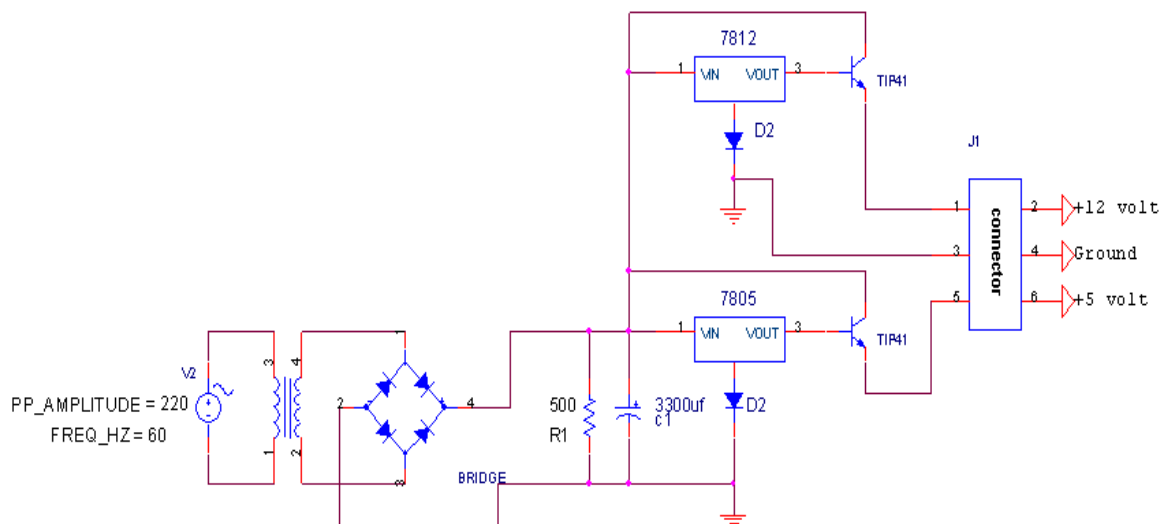
$$900 + 400 + 300 + 300 + 500 = 2400ma \quad (1-4)$$

یعنی در بحرانی ترین شرایط این منبع تغذیه باید ۲/۴ آمپر جریان دهی داشته باشد با در نظر گرفتن ملاحظات منبع تغذیه ۱۲ ولت ۲ آمپر و ۵ ولت ۱ آمپر را طراحی کرده ایم .

در عملی ترانس‌هایی که به منظور استفاده برای ورودی یکسوسازها استفاده می شوند قبل از یکسوسازی و فیلتر شدن باید ولتاژ خروجی آنها V_{ac} تقریباً ۲۵٪ بیشتر از خروجی مطلوب ایده آل V_{dc} است یعنی:

$$V_{ac} = 1.25V_{dc} \quad (2-4)$$

منبع تغذیه طراحی شده برای این پروژه با توجه به نیازهای ولتاژ و جریان مذکور که در مورد آنها بحث شد در شکل (۲-۴) نشان داده شده است .



شکل ۴-۲- مدار الکتریکی طراحی شده برای منبع تغذیه ربات

با توجه به شکل ولتاژ ۲۲۰ ولت متناوب توسط ترانسفرماتور کاهنده (۱۶ ولت، ۳ آمپر) به ۱۶ ولت متناوب کاهش یافته سپس توسط پل دیودی به صورت تمام موج یکسو شده و توسط خازن ۳۳۰۰ میکرو فاراد و مقاومت ۵۰۰ اهم یک وات به ولتاژ ۱۶ ولت یکسو شده می‌رسیم، با توجه به اینکه در مدار فرمان و قدرت در بیشترین حالت مصرف به ولتاژهای ۵ ولت ۱ آمپر و ۱۲ ولت ۲ آمپر نیاز داریم و با توجه به اینکه خروجی رگولاتورهای ۷۸۰۵ و ۷۸۱۲ در عمل ۱۰۰ میلی آمپر است لذا باید خروجی این رگولاتورها تقویت جریان شوند، با توجه به شکل در می‌یابیم که در طبقه زیرین در خروجی ۷۸۰۵ ولتاژ ۵ با جریان ناچیز داریم این ولتاژ ۵ ولت با جریان ناچیز می‌تواند ترانزیستور نیمه قدرت

TIP41 را روشن و به اشباع می‌برد چون کلکتور آن به ولتاژ ۱۶ ولت با جریان بالا متصل است در نتیجه هنگامی که ترانزیستور به اشباع می‌رود $V_{BE} = 0.7$ شده ولتاژ ۵ با جریان دهی بالا در خروجی که از امیتر TIP41 گرفته شده خواهیم داشت برای درک این مطلب با اعمال KVL در مسیر دیود ۱ و رگولاتور ۷۸۰۵ و پیوند بیس - امیتر TIP41 خواهیم داشت :

$$-V_{D1} - 5 + V_{BE1} + V_{o1} = 0$$

(۳-۴)

با توجه به اینکه $V_{BE1} = V_{D1}$ لذا داریم :

$$\begin{aligned} V_{o1} &= 5 + V_{D1} - V_{BE1} \\ V_{o1} &= 5^v \end{aligned} \quad (4-4)$$

برای طبقه دوم این منبع یا تغذیه بخش قدرت هم روال به همین ترتیب است با اعمال KVL در مسیر دیود ۲ و رگولاتور ۷۸۱۲ و پیوند بیس - امیتر $TIP41$ خواهیم داشت :

$$-V_{D2} - 12 + V_{BE2} + V_{o2} = 0 \quad (5-4)$$

با توجه به اینکه $V_{BE2} = V_{D2}$ لذا داریم :

$$\begin{aligned} V_{o2} &= 12 + V_{D2} - V_{BE2} \\ V_{o2} &= 12^v \end{aligned} \quad (6-4)$$

با توجه به اینکه ولتاژ از امیتر ترانزیستور $TIP41$ قدرت گرفته می شود جریان خروجی منبع از مسیر کلکتور - امیتر و ولتاژ آن از مسیر بیس - امیتر تامین می شود این یک طرح جدید بوده و در عمل توانسته است به خوبی نیازهای مربوط به مسائل منبع تغذیه ربات را برای ما فراهم کند .

۴-۴- طراحی و اجرای مدارهای راه انداز برای موتورهای

همانطور که می دانیم برای راه اندازی موتورهای DC در حال حاضر روشهای گوناگونی وجود دارد مثلا راه اندازهای خطی و راه اندازهای PWM و... در این میان چون استفاده از راه اندازهای خطی نیاز به طراحیهای چند طبقه و از طرفی برای سیستم چپگرد و راستگرد موتور نیاز به منابع تغذیه متقارن به صورت شکل (۳-۴) دارند لذا از لحاظ هزینه و طراحی مقرون به صرفه نیستند ، اخیرا در اکثر طرحهای صنعتی برای

کنترل دور موتورهای از تکنیک PWM استفاده می شود به طوری که در یک محدوده فرکانسی مشخص مثلاً ۱۰۰ الی ۵۰۰ هرتز یک پالس قدرت با فرکانس ثابت و چرخه کار^{۱۷} کنترل شده به موتور اعمال می گردد، با کنترل این چرخه کار می توان به راحتی توان اعمالی به موتور و در نتیجه سرعت موتور را کنترل کرد برای استفاده از این تکنیک هم می توان از طراحیهای ترانزیستوری استفاده کرد و هم این راه اندازها به صورت چیپهای فراوانی در بازار موجودند تعدادی از آنها عبارتند از :

L298 : راه انداز دو موتور DC

LMD18200 : راه انداز یک موتور DC

UCN3951 : راه انداز دو موتور DC

754410 : راه انداز دو موتور DC

L293B : راه انداز دو موتور DC

L293D : راه انداز دو موتور DC

و نیز راه اندازهای پل H که با ترانزیستورهای PNP,NPN قابل ساخت بوده و پیش پا افتاده هستند اما اینکه چیپهای موجود در بازار جوابگوی نیاز ما هستند یا خیر نیاز به بررسی دارد .

چنانچه موتورهایی با جریانهای بیشتر از ۳ آمپر داشته باشیم باید برای خرید چیپهای راه انداز هزینه ای معادل حداقل ۳۰۰۰۰ تومان متحمل شویم یا خود را برای طراحی پلهای H ترانزیستوری با ترانزیستورهای با بهره بالا و همچنین توان بالا آماده کنیم بعضی از این ترانزیستورها به قرار زیرند که در شروع به کار این پروژه راه انداز با جریان دهی ۶ آمپر تحت ولتاژ ۶۰ ولت را با موفقیت با استفاده از آنها ساختیم :

^{۱۷} Duty Cycle

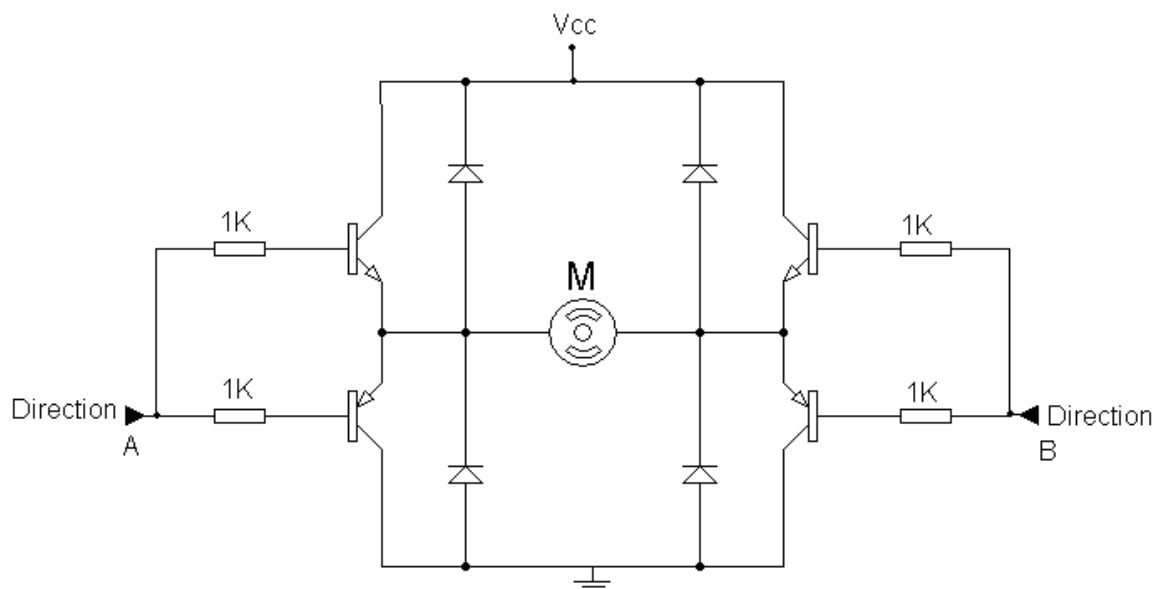
TIP147 - TIP146- TIP145- TIP127- TIP126- TIP125 : PNP

TIP142- TIP141- TIP140- TIP122- TIP121- TIP120 : NPN

بعضی از راه اندازهای گفته شده دارای ورودی جهت ، پالس PWM و ترمز و بعضی دیگر دارای ورودی ترمز نمی باشند .

۴-۱-۴ راه اندازهای پل H ترانزیستوری

پلهای H یا همان H-Bridge معمولترین راه اندازهای موتورهای DC هستند در این راه اندازها از ۴ ترانزیستور استفاده شده می شود که بسته به مشخصه های مختلف بکار می روند در این حالت یا تمام ترانزیستورها p هستند یا همه n هستند یا دو تا n و دو تا p هستند که معمولترین آنها گونه آرایشی است که از دو ترانزیستور n و دو ترانزیستور p استفاده می کند شماتیک این مدار در شکل (۴-۱) نشان داده می شود .



شکل ۴-۳- شماتیک مدار راه انداز با دو ترانزیستور n و دو ترانزیستور p

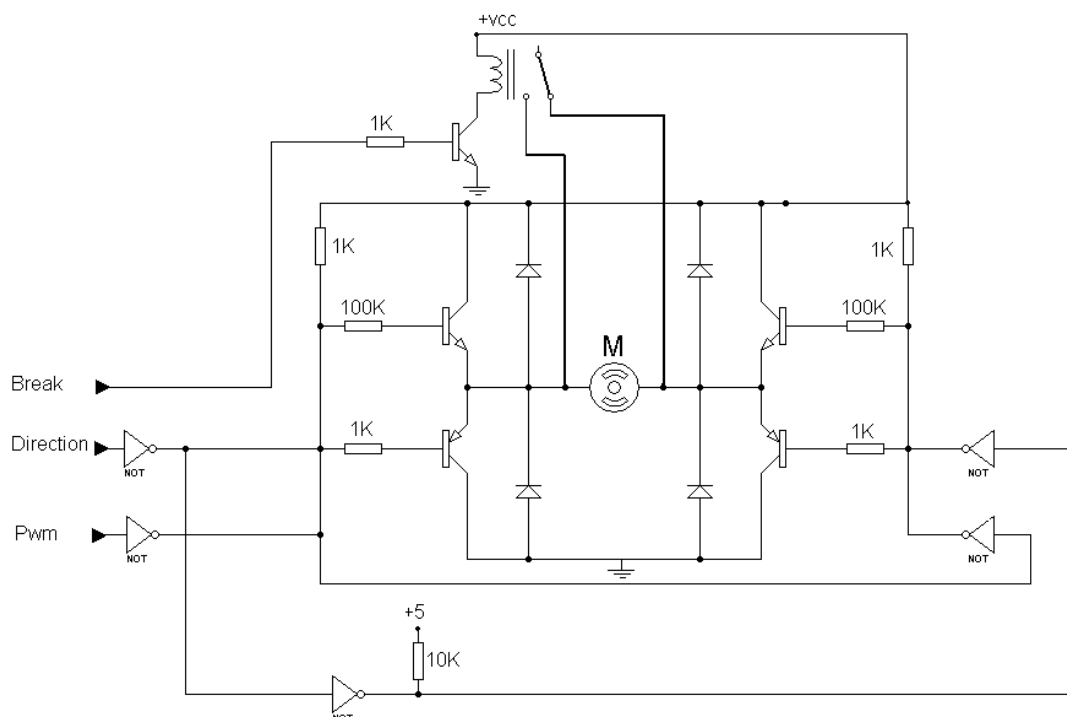
در این حالت با توجه به حالت موجود در پایا نه های A,B واکنش و جهت حرکت موتور به صورت زیر می باشد .

A	B	جهت حرکت
0	0	بی حرکت
0	1	ساعتگرد
1	0	پاد ساعتگرد
1	1	بی حرکت

این مدار ساده تنها تغییر جهت چرخش را برای ما در حالتی که تغذیه مدار به طور دائم روشن است را برای ما فراهم می کند اما این برای ما کافی نیست . ما به مداری نیاز داریم که هم بتوان جهت چرخش موتور را تغییر داد هم توان اعمال شده به آنرا تغییر داد و هم در صورت لزوم بتواند به صورت آنی موتور را متوقف کند برای این منظور مدار نشان داده شده در شکل (۴-۴) را طراحی کرده و پیشنهائی می کنیم این مدار همانند چیپهای راه انداز که در بخش بعد به آنها اشاره خواهیم کرد دارای ۳ ورودی ، Direction ، PWM ،

Break می

باشد .

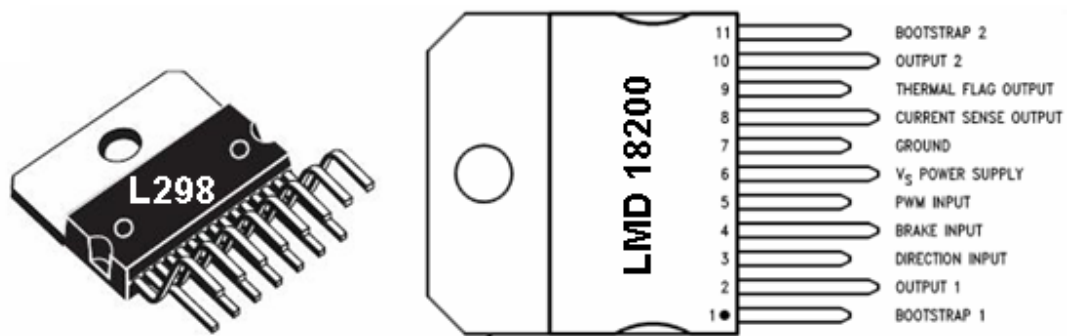


شکل ۴-۴- راه انداز ترانزیستوری کامل

۴-۴-۲- چپهای راه انداز

چپهای راه انداز پلهای H بسته بندی شده با امکاناتی اضافی همچون ورودی PWM و نیز ورودی جهت چرخش و ورودی دیگری جهت ترمز ناگهانی موتورهای DC تعبیه شده است از میان این چپها چپ L298 مرسومترین است این چپ یک Dual Driver برای ۲ موتور DC می باشد که می تواند حداکثر جریان ۱ آمپر را در ولتاژ ۳۰ ولت به بار تحویل دهد این راه انداز دارای ورودیهای PWM, DIRECTION مجزا برای هر موتور می باشد اما دارای ورودی جهت ترمز نمی باشد ولی به سادگی مانند آنچه در پلهای H ترانزیستوری مشاهده کردیم می توان ترمز هم به آن اضافه کرد .

در کنار این چپ، چپ دیگری با نام LMD18200 وجود دارد که آن هم یک Dual Driver است و توانایی عبور حداکثر ۳ آمپر جریان در ولتاژ ۴۸ ولت را دارا می باشد و دارای ورودیهای PWM, DIRECTION, BREAK مجزا برای هر موتور نیز می باشد اما قیمت آن تقریباً ۱۰ برابر چپ L298 می باشد ما در این پروژه از این چپ استفاده کرده ایم که به خوبی پاسخگوی نیاز ما بوده است شمای ظاهری این چپ به صورت نمایش داده شده در شکل (۴-۵) می باشد .



شکل ۴-۵- نمای ظاهری چپهای راه انداز

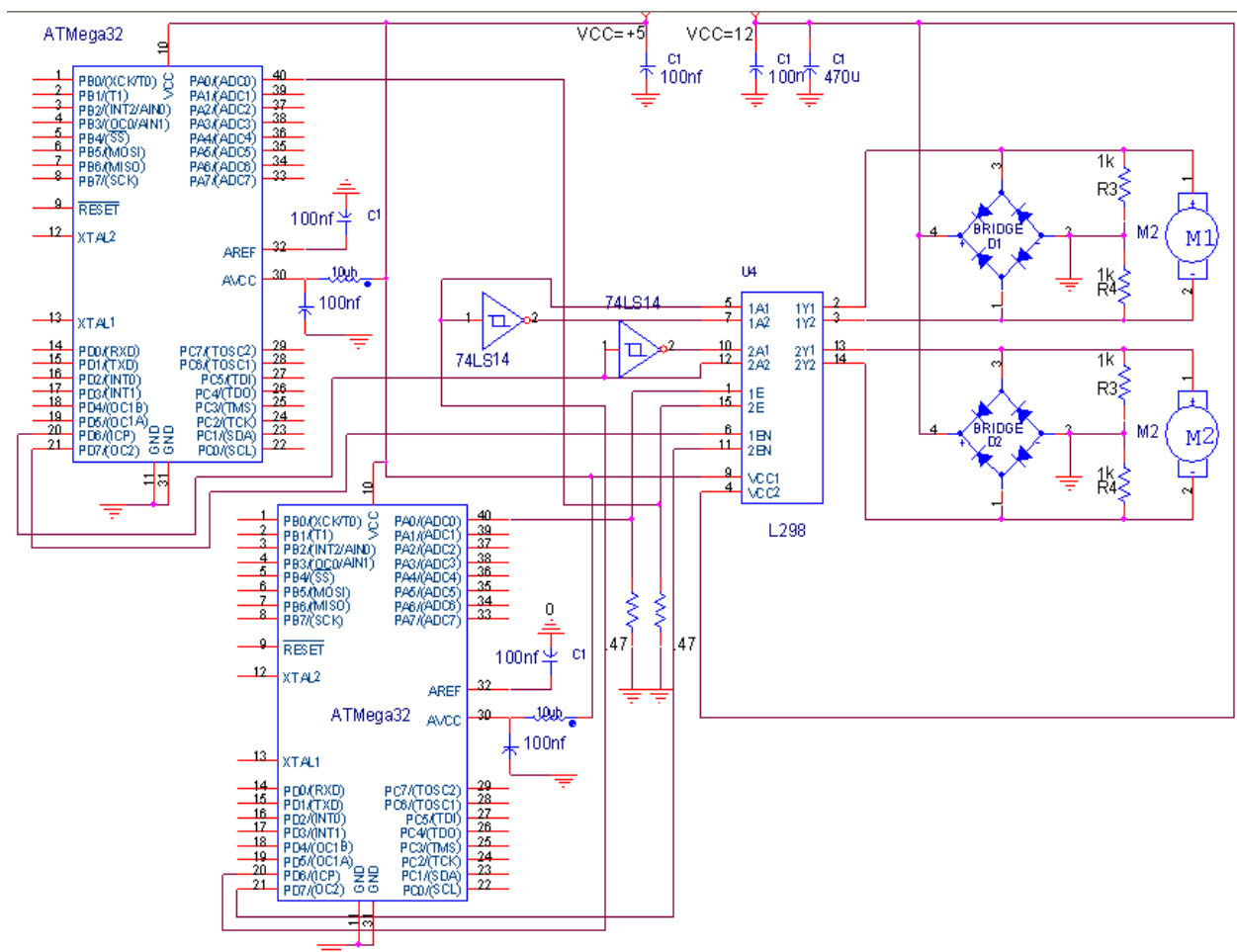
در این پروژه از راه انداز L298 استفاده شده است که یک راه انداز دوگانه^{۱۸} است که دارای ورودی هایی جهت PWM می باشد مدار راه اندازی L298 به صورت کامل در شکل (۴-۶) نمایش داده شده است ، با توجه به اینکه این چیپ دارای دیودهای هرز گرد برای جلوگیری از ضربات ولتاژی ناشی از سیم پیچهای رتور موتورها نمی باشد گذاشتن دیودهای ۱ تا ۸ الزامیست همان طور که در شکل دیده می شود نقش این راه اندازها تقویت سیگنال PWM می باشد برای این منظور دو ورودی جهت و فعال کننده آن را به پینهای ۲۰ و ۲۱ (PD6,PD7) میکرو متصل می کنیم که پین ۲۱ همان Ocr2^{۱۹} یا پایه خروجی PWM تایمر ۲ میکرو کنترلر می باشد .

با توجه به شکل (۴-۶) یکی از امکانات ویژه این راه انداز امکان سنس جریان کشیده شده توسط موتور می باشد به این ترتیب که این چیپ یک فیدبک جریانی از جریان موتور روی پایه های ۱ و ۱۵ (۱ برای موتور ۱ و ۱۵ برای موتور ۲) ایجاد می کند چنانچه مایل به استفاده از این قابلیت نباشیم این پایه ها را به زمین متصل می کنیم ، اما از آنجا که برخورد بازوی ماهر با موانع احتمالی باعث اضافه جریان^{۲۰} می شود و این موضوع هم برای موتور و هم مکانیک ربات مضر است لازم است سیستمی برای آشکار سازی برخورد بازوی ماهر با موانع احتمالی طراحی کنیم به شکل (۴-۶) توجه کنید .

^{۱۸} Dual Driver

^{۱۹} Timer2 PWM Output (pind.7 or pin[21] of TMEGA32)

^{۲۰} Over Current



شکل ۴-۶- مدار راه انداز L298 برای کنترل توان ۲ موتور

می توان این کنترل جریان را به راحتی توسط تبدیل فیدبک جریان موجود در پایه های ۱۵ و ۱۵ راه اندازه ولتاژ توسط قرار دادن مقاومتهای ۰,۴۷ اهم ۱ وات میان هر یک از پایه های ۱۵ و ۱۵ تا زمین به ولتاژ قابل اندازه گیری برای مبدل های آنالوگ به دیجیتال میکروکنترلر مربوط به هر موتور تبدیل کرد سپس با اندازه گیری این ولتاژ می توان تغییرات جریان موتور را با نسبت ۰,۴۷ به دست آورد و تصمیمات لازم را در مواقع اتفاق اضافه جریان اتخاذ کرد.

موضوع دیگر بحث سیستم ترمز برای این راه انداز هاست ترمزی که می تواند در کسری از ثانیه موتور را

۴-۵- کد گذارهای نوری

این پروژه شامل دو رابط دوار و یک رابط کشویی می باشد . در ابتدا نظر بر این بود که از کدگذارهای دورانی^{۲۲} برای آشکار سازی موقعیت و سرعت همه رابطها استفاده گردد اما پس از ملاحظات و بررسی های عملی طرح تصمیم گرفته شد تا برای سنجش موقعیت رابط سوم یا همان رابط کشویی و همچنین میج از مدارات شمارنده مادون قرمز استفاده کنیم، بدین ترتیب که می توان با شمارش عبور دنده های دنده خروجی محرک سوم که مستقیماً با دنده کشویی در ارتباط است استفاده کرد این تغییر شمارشگر با گام چرخ دنده خروجی و گام چرخدنده کشویی نسبتی ساده داشته و به راحتی می توان تغییر موقعیت عمودی رابط کشویی را آشکار سازی کرد .

۴-۵-۱- آشنایی با سنسورهای تشخیص موقعیت موجود در بازار

در بازار قطعات الکترونیک ایران به دست آوردن قطعات بعد از طراحی کار بسیار دشواری است به همین دلیل ابتدا باید از تامین قطعه مطمئن شد بعد اقدام به طراحی کرد .

یکی از راههای عملی و عمومی تشخیص موقعیت برای هر حرکتی اعم از خطی و دورانی استفاده از فرستنده و گیرنده های مادون قرمز می باشد ، گونه های مختلفی از این سنسورها با مشخصه های متفاوت در بازار یافت می شوند که قیمتهای مناسبی هم دارند .

نوع خاصی از این سنسورها که می توان از آنها برای تشخیص موقعیت استفاده کرد بدین ترتیب است که هر دو زوج فرستنده و گیرنده در یک بسته روبروی هم قرار گرفته اند که به وقفه دهنده های نوری^{۲۳} مشهورند

^{۲۲} Rotary Shuft Encoders

^{۲۳} Optical-Interrupter

برای این پروژه مناسب بود اما از آنجا که جهت چرخش هم برای ما مهم است و نیز دقتی در حد یک درجه در دور یا ۳۶۰ پالس در دور از کد گذار انتظار داشتیم این باعث شد به فکر تهیه دیسکهای شکاف دار ظریف برای بالا بردن حد تفکیک و مدارات لازم برای آشکار سازی موقعیت و جهت حرکت دورانی بیافتیم و چون سه عدد کدگذار لازم بود و دست ساز هر کدام با مدارهای جانبی که برای آنها طراحی کردیم شکل ناموزون و بزرگ داشتند بر آن شدیم که از کد گذارهای محوری^{۲۴} موجود در بازار که تمامی این خواسته ها را بر آورده می کنند استفاده کنیم .

سنسورهای اندازه گیری موقعیت بررسی شده را می توان به سه دسته زیر تقسیم کرد :

- کدگذار نوری افزایشی^{۲۵}
- کدگذار نوری مطلق^{۲۶}
- ریزالور تک فاز^{۲۷}

کار با گونه افزایشی بسیار ساده است . این کدگذار از یک یا چند صفحه دوار دایره شکل تشکیل یافته است و دارای ۱-۳۶۰ شکاف متقارن در روی محیط خود است دو سنسور مادون قرمز گیرنده و فرستنده در مقابل هم قرار دارند به طوری که این شکاف فضای بین آنها را پر کند با چرخش این دیسک سنسور گیرنده پالسی در خروجی خود ایجاد می کند که فرکانس این پالس به تعداد شکافها و سرعت چرخش دیسک وابسته است .

^{۲۴} Shuft Encoder

^{۲۵} Incremental Optical Encoder

^{۲۶} Absolute Optical Encoder

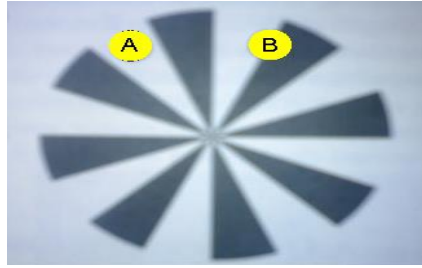
^{۲۷} Single Phase Resolver

پس واضح است که با کوپل کردن شافت هر محرک دواری از جمله موتور DC با شافت این کدگذارها که به کد گذارهای محوری هم معروفند می توان موقعیت خروجی محرک دوار را به دست آورد .

۴-۵-۲- طراحی مدارات شمارنده مادون قرمز IR

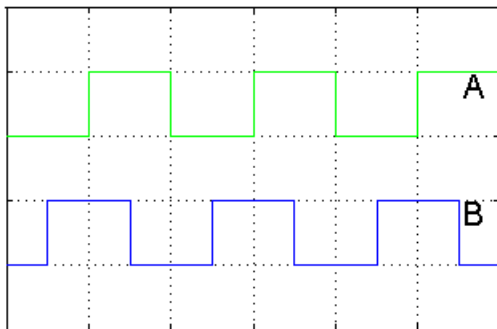
از آنجاکه لازم است همزمان با حرکت رابط موقعیت و جهت نمونه برداری شوند باید از ۲ آرایش زوج گیرنده و فرستنده که نسبتا پیچیده هستند استفاده کرد برای این منظور مدارات پیچیده ای می توان طراحی کرد بسته به مشخصه های دیود گیرنده و ترانزیستور گیرنده می باید مقادیر المانها تغییر کند [۵] ، مدار طراحی شده زیر در عمل پاسخ خوبی داده است در این طرح ۲ زوج سنسور در فاصله ای کم در کنار هم به گونه ای تنظیم شده اند که پالسهای خروجی آنها دقیقا ۹۰ درجه با هم اختلاف فاز دارند بدین ترتیب اگر خروجی زوج اول A را به یک پین از یک میکرو کنترلر و خروجی زوج دوم B را به پین دیگری متصل کنیم و اطلاعات این دو بیت را بخوانیم یک عدد ۲ بیتی خواهیم داشت که بیت بالای آن A و بیت پایین آن B می باشد در این حالت چنانچه چرخش ساعتگرد باشد رشته ۲ بیتی زیر از راست به چپ تکرار می گردد , 11 , 01 , 10 , 00 و چنانچه چرخش پادساعتگرد باشد این رشته دقیقا برعکس تکرار می شود یعنی به صورت , 00 , 01 , 11 , 10 در کنار این تشخیص جهت حرکت می توان به طور مداوم اطلاعات مربوط به موقعیت و سرعت را با اتصال زوج A به یک کانال شمارنده و استفاده از یک کانال تایمر به دست آورد این تنظیم موقعیت زوج A,B در شکل (۴-۸)

نشان داده شده است [۵] .

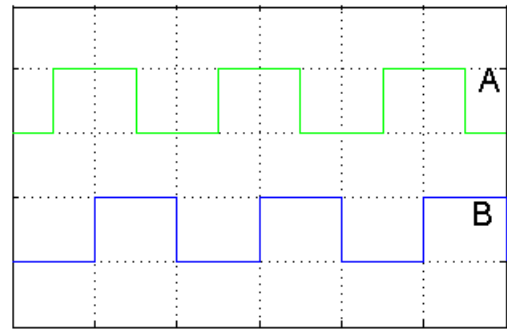


شکل ۴-۸- طرز قرار گرفتن زوج سنسورهای IR

در این حالت پالسهای خروجی گیرنده های زوجهای A,B در حالت ساعتگرد به صورت شکل (۴-۹) و در حالت پادساعتگرد به صورت شکل (۴-۱۰) می باشد .

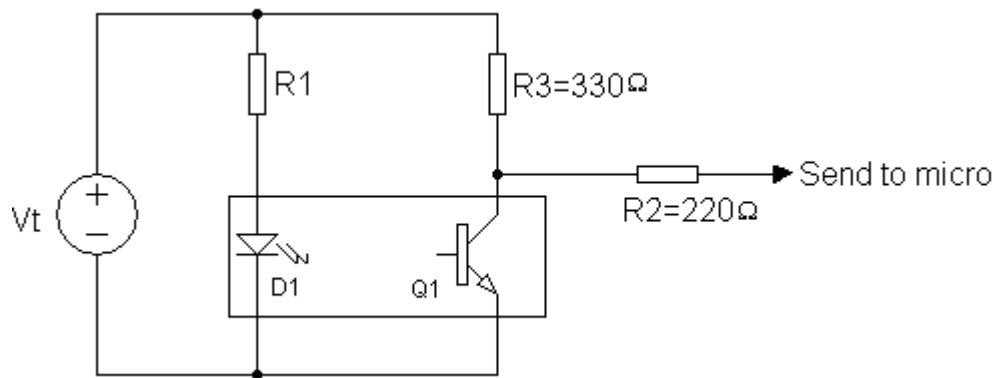


شکل ۴-۱۰- خروجی در حالت پادساعتگرد



شکل ۴-۹- خروجی در حالت ساعتگرد

این زوج سنسورها که به صورت بسته بندی شده با ابعاد و آرایشهای گوناگون موجودند به صورت نشان داده شده در شکل (۴-۱۱) می باشد تنها نکته این است که مقاومت R1 باید به گونه ای انتخاب شود که از دیود مادون قرمز فرستنده جریانی بین ۲۰ mA تا ۲۵ mA عبور کند .

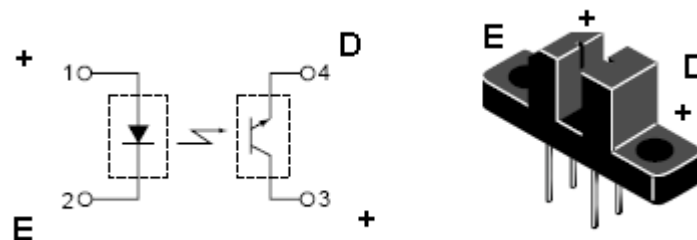


شکل ۴-۱۱- طریقه راه اندازی کردن سنسورهای مادون قرمز

$$R_1 = \frac{5}{20 \times 10^{-3}} = 250 \Omega \cong 220 \Omega$$

پس از محاسبه داریم :

شمای ظاهری و پایه بندی این سنسورها به صورت شکل (۴-۱۲) است .

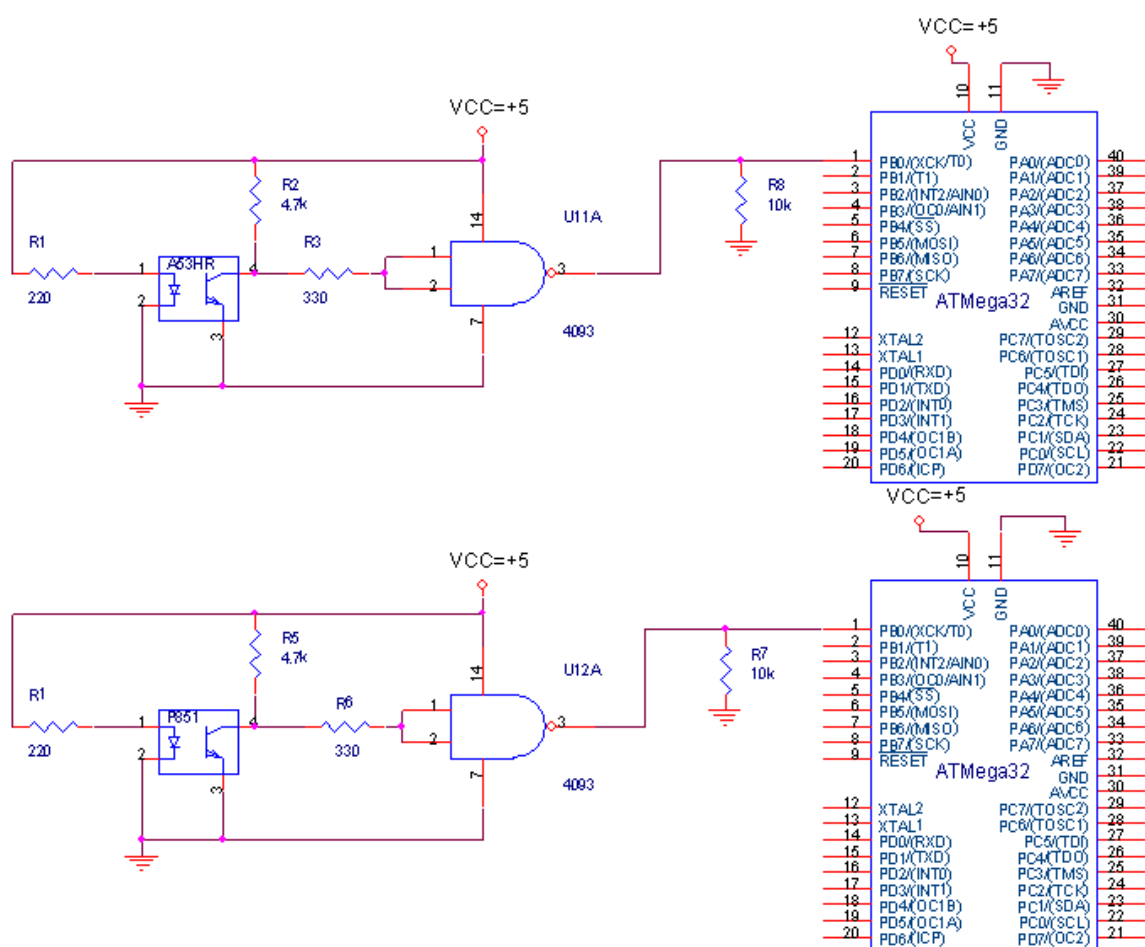


شکل ۴-۱۲-شمای ظاهری زوج سنسورهای مادون قرمز

۴-۵-۳- ملاحظات عملی سنسورهای مادون قرمز

در عمل این سنسورها در تحریکهای فرکانس بالا شکل موجهای با فرمهای غیر قابل شمارش برای شمارنده های دیجیتال دارند برای رفع این مشکل پیشنهاد می شود خروجی این سنسورها را همان طور که در شکل

(۴-۱۳) نشان داده شده توسط یک گیت^{۲۸} منطقی مانند 4093 ویا 74hc14تریگر کرد تا در فرکانسهای بالای تحریک شکل موجهای آنها از حالت مربعی خارج نشود .



شکل ۴-۱۳- نمودار مدارهای مادون قرمز برای رابط ۳ و موج ربات

۴-۵-۴- کدگذارهای دورانی

این قطعات که ظاهری شبیه موتور دارند و یک شافت هم در خروجی خود دارند برای تشخیص موقعیت در سیستمهای مبتنی بر محرکهای دوار به کار می روند به طوری که این شافت و شافت محرک باید به صورت

^{۲۸} Gate

هم مرکز با هم جفت^{۲۹} شوند. این قطعات پر کاربرد در رنجهای مختلفی از نقطه نظرهای زیر در بازار یافت می شوند:

۱- ولتاژ کاری

۲- تعداد پالس خروجی در هر دور

۳- تعداد کانال

۴- ساین بدنه

ولتاژهای استاندارد کاری ۵، ۱۲، ۲۴ ولت می باشد.

تعداد پالسهای خروجی استاندارد ۱۰۰، ۳۶۰، ۱۰۰۰، ۵۰۰، ۳۶۰، ۲۵۰ می باشد.

تعداد کانالها ۱، ۲، ۳ می باشد.

ساینهای استاندارد با قطر مقطع دایره ای ۳، ۴، ۵ میلیمتر می باشد.

اساس کار این قطعات کاملاً با سنسورها یکی است فقط دقت اندازه گیری در آنها خیلی بالا تر از سنسورها است.

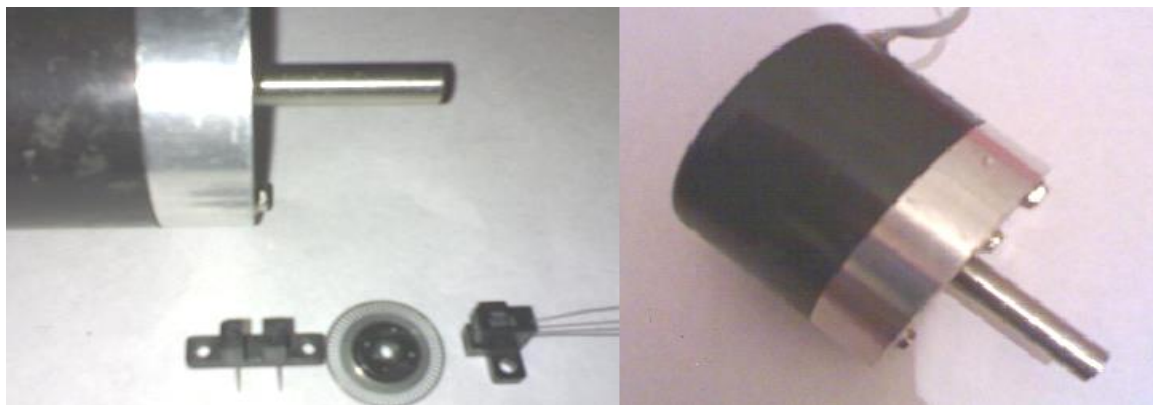
۴-۵-۵- بررسی و کار با کدگذارهای نوری افزایشی شرکت Autonics

کدگذارهای نوری دورانی شرکت‌های Delta, Fatek, Autonics ... با حد تفکیک‌های مختلف با ظاهری شبیه به موتور در بازار ایران موجود می باشند.

در شکل (۴-۱۴) نمونه‌هایی از سنسورهای مادون قرمز که در یک بسته بندی مناسب برای اهدافی چون تشخیص موقعیت به کار میروند مشاهده می شود و نیز در میان دو نوع سنسور دیسکی با ۳۲ شکاف وجود دارد که همان طور که گفته شد با قرار دادن این دیسک شکافدار در دهانه تعبیه شده میان دو سنسور

^{۲۹} Couple

مادون قرمز و چرخش آن پالسی در خروجی سنسور گیرنده مشاهده خواهند شد که با پردازشی ساده مبین مقدار جابجایی می باشد ، و نیز یک کدگذارهای نوری افزایشی با دقت ۰/۱ درجه یا به عبارتی ۳۶۰۰ پالس در دور مشاهده می شود همانطور که گفته شد این کد گذارها ظاهری شبیه به موتور دارند .



۴-۱۴- نمونه هایی از سنسورهای مادون قرمز و دیسک شکافدار و Incremental Optical Encoder

در این پروژه از کدگذارهای نوری شرکت Autonics کره استفاده شده ظاهر ساده ، عملکرد و کاربرد آسان و قیمت مناسب مهمترین دلایل استفاده از آنها می باشد و حد تفکیک آنها ۳۶۰ پالس در دور معادل دقت ۱ درجه در دور می باشد .

این محصول دارای ۵ سیم به ترتیب زیر است :

۱- خروجی پالس مربوط به کانال یک

۲- خروجی پالس مربوط به کانال دو

۳- تغذیه +۵ ولت

۴- شیلد یا پوسته

۵- زمین

خروجیهای ۱ و ۲ با هم ۹۰ درجه اختلاف فاز دارند که این باعث می شود ما بتوانیم در هر لحظه با

سرکشیهای مناسب جهت چرخش را از روی ترتیب تغییر سطوح پالسهای مشاهده شده در خروجی-های ۲ و ۱ تشخیص دهیم .

در این ربات برای رابطهای چرخشی اول و دوم از شافت انکدر که به صورت مستقیم روی مفصل قرار گرفته اند استفاده شده و برای بازوی کشویی و محور چرخاننده مچ از زوج سنسورهای مادون قرمز P581 و H21A1 سنسورهای مادون قرمز دابل ماوسهای قدیمی استفاده شده است . چگونگی نصب این سنسورها روی ربات در شکلهای (۳-۴) و (۴-۴) مشاهده می شود .

۴-۶- زمان نمونه برداری

به طور کلی در این پروژه زمان نمونه برداری بر اساس دریافت اطلاعات جدید از سنسورهای تشخیص موقعیت تعیین می شود بدین ترتیب که هر بار وقفه مربوط به تغییر موقعیت رابط به میکروکنترلر مربوطه می رسد تایمری که در با تغییر موقعیت قبلی رابط به کار افتاده بود متوقف می شود و این مقدار به عنوان Δt یا همان زمان نمونه برداری^{۳۰} در نظر گرفته می شود.

۴-۶-۱- محاسبه سرعت و شتاب بازوی ماهر

سرعت و شتاب رابط های ربات را می توان با فرمولهای ساده دیفرانسیل طوری محاسبه کرد که با ریز پردازنده ها به سادگی قابل پیاده سازی است فرض کنید که $f(k)$ سیگنال نمونه برداری شده موقعیت رابط و Δt زمان نمونه برداری باشد در این صورت مشتق زمانی رابط یا همان سرعت رابط با تقریب بسیار خوبی به صورت رابطه (۷-۴) محاسبه می شود :

$$f'(k) = \frac{f(k) - f(k-1)}{\Delta t} \quad (7-4)$$

^{۳۰} Sample Time

محاسبه شتاب حرکت رابط دوم نیز به همین ترتیب است لذا مشتق زمانی سرعت رابط یا همان شتاب رابط

به صورت رابطه (۸-۴) می باشد :

$$f''(k) = \frac{f'(k) - f'(k-1)}{\Delta t} \quad (۸-۴)$$

در پیوست ۱ برنامه مربوط به استفاده از این روش در میکرو کنترلرها ارائه شده است .

۷-۴- سیم بندی و حفاظت سیگنال

از هر سنسور یک رشته سیم حامل پالس با محدوده فرکانسی مشخص می گذرد از طرفی به هر موتور هم ۲ رشته سیم که حاوی پالسهای با توان بالاتر و فرکانسی در محدوده ۲۰۰-۴۰۰ هرتز می باشند می گذرد این رشته سیمها همه در کنار هم قرار گرفته و به صورت مجموعه از کنترلر و راه انداز به سمت رابطها هدایت می شوند . یکی از مسائل بسیار مهم که عدم توجه به آن یا بی دقتی نسبت به ملاحظات مربوط به آن ، تمام دقت حسگرها را زیر سوال می برد این موضوع حفاظت رشته سیمهایی است که حامل موجهای مربعی ، سینوسی ، دنداناره ای ، مثلثی و یا در کل هر شکل موج پریودیک می باشند .

۷-۴-۱- شیلد^{۳۱} دار کردن سیمها

می دانیم اطراف یک سیم حامل جریان میدان مغناطیسی ایجاد می شود چنانچه این میدان قطع و وصل شود می تواند تا محدوده شعاع میدان مغناطیسی خود روی میدانهای مغناطیسی دیگر و همچنین بر روی سطوح ولتاژ هادیهای مجاور تاثیرات نا مطلوب می گذارد . بهترین راه مقابله با این مشکل شیلد دار کردن هادیها می باشد ، سیمهای شیلد دار کابلهایی می باشند که هسته مرکزی کابل همان هادیهای حامل سیگنالهای پریودیک هستند که با لایه ای از فویل یا آلومینیوم شیلد یا محافظت می شوند یک سر شیلد

^{۳۱} Shield

باید به سطح ولتاژ منفی مدار یا زمین^{۳۲} متصل و سر دیگر آن آزاد باشد، در این حالت نویز ناشی از برهم کنش میدانهای مغناطیسی توسط این لایه از بین می رود، دقت شود که چنانچه دو سر شیلد زمین شود نویز در حلقه قرار گرفته و اطلاعات کماکان دستخوش اغتشاش و تغییرات نامطلوب خواهند ماند.

فصل پنجم: بررسی نحوه اجرای ساختار سیستم ربات

۵-۱- مقدمه

ساختار سیستم در این طرح را می توان به چند صورت پیاده سازی کرد:

روش اول: اینکه تمامی اعمال کنترلی ربات را توسط چندین میکروکنترلر که با هم در یک شبکه قرار می گیرند انجام داد.

روش دوم: در این روش می توان سیگنالهای مطلوب را توسط رایانه فراهم کرده و به میکروکنترلر از طریق پورت سریال ارسال کرد و میکرو تنها محاسبات کنترلی را انجام دهد[۳].

۱- استفاده از محیط نرم افزار MATLAB

۲- استفاده از محیط نرم افزار VISUAL BASIC

روش سوم: هم اینست که رایانه علاوه بر تولید سیگنال بخشی از محاسبات کنترلی را هم انجام دهد.

^{۳۲} Erth

در بخش انتخاب پانل کنترل مطالعات جذابی به خصوص برای ارتباط ربات با رایانه و استفاده از فضای نرم افزار مطلب و نرم افزار قوی Visual Basic انجام گرفت .

۵-۲- اجرای ساختار سیستم به صورت یک مجموعه خارجی

چنانچه بخواهیم تمامی اعمال کنترلی ربات را توسط چندین میکروکنترلر که با هم در یک شبکه قرار می گیرند انجام دهیم با مشکلات زیر روبرو می شویم :

۱- برای یک بازو این روش آزمایش شد در این فرایند یک تایمر و یک کانتر برای محاسبات همزمان موقعیت و سرعت از روی کدگذار لازم است که نیاز به یک میکروکنترلر مجزا دارد و همچنین به دو تایمر برای تولید موقعیت ، سرعت و شتاب مطلوب نیاز است و نهایتاً یک میکروکنترلر مادر که اطلاعات را به صورت همزمان از میکروکنترلر مولد مسیرهای مطلوب و میکرو کنترلر نشان دهنده مسیر طی شده توسط ربات گرفته و محاسبات کنترلی را که شامل روابط پیچیده در پیاده سازی با زبان اسمبلی ، C می باشد انجام دهد این کار برای یک رابط دو میکروکنترلر برای دو رابط پنج میکروکنترلر و برای ۳ رابط ۷ میکروکنترلر لازم دارد که ایجاد همزمانی بین تولید و جمع آوری مسیرهای مطلوب و مسیر جاری ربات کاری فوق العاده مشکل می شود به همین دلیل سراغ راهی رفتیم که به ظاهر ساده تر بود اما در عمل با مشکلاتی روبرو شد در بخش بعدی به آن اشاره می شود .

۲- با توجه به اینکه روش کنترلی مورد استفاده در این پروژه روش گشتاور محاسباتی می باشد در

عمل محاسبه معادلات حرکت ربات ۲ رابط ساده به صورت

$$\tau_i = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

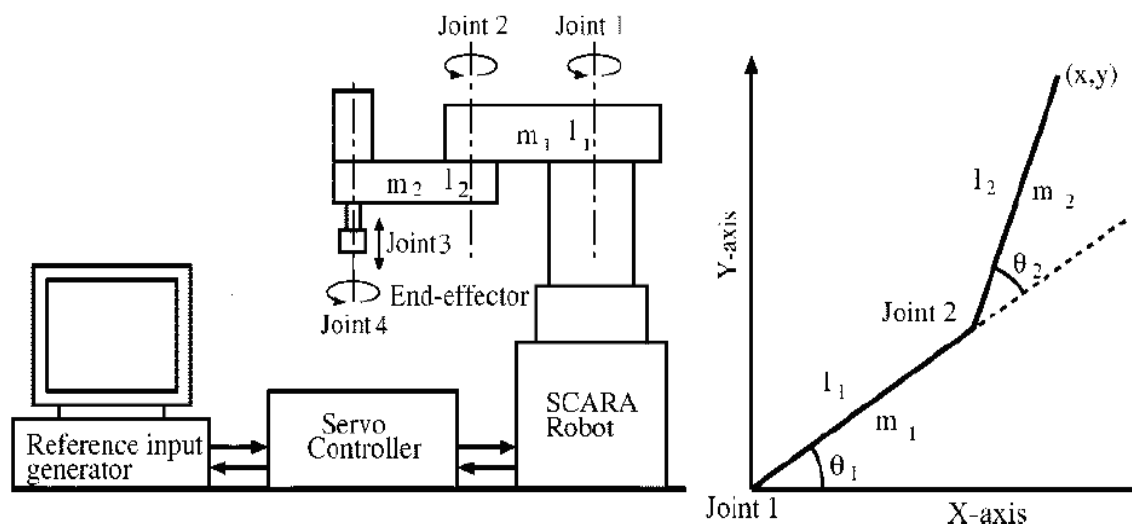
بطور همزمان برای یک کنترلر مبتنی بر میکرو پروسوسور کاری فوق العاده مشکل و پیچیده است
تصمیم گرفتیم که از نرم افزار قدرتمند مطلب برای انجام یک سری عملیات کنترلی پیچیده در کنار میکرو
کنترلرها استفاده کنیم .

۵-۳- اجرای ساختار سیستم به صورت PC Interface

در این روش همانطور که در شکل مشاهده می شود از رایانه تنها جهت تولید سیگنالهای مرجع استفاده
می گردد و سپس این اطلاعات با یک زمانبندی مشخص به میکروکنترلر رئیس^{۳۳} ارسال می گردد از طرف
دیگر با تاخیری کوتاه اطلاعات مربوط به مسیر جاری ربات هم از کنترلرهای مربوط به جمع آوری آنها
دریافت می گردد و نهایتاً سیگنال پر ارزش خطای ردیابی موقعیت و سرعت به دست آمده جهت انجام
محاسبات کنترلی در کنترلر مادر به همراه سیگنالهای موقعیت مطلوب مورد استفاده قرار می گیرند .

ساختار سیستم در این روش به صورت شماتیک در شکل (۵-۱) نشان داده شده است . فضای کنترلی برای
ربات هم می تواند فضای مفصلی باشد هم دکارتی اما با وجود اینکه درک مسیر دکارتی برای همه ساده تر
است ولی چون مسائل مربوط به سینماتیک وارون وارد محاسبات می شود تمام محاسبات و الگوریتمهای
کنترلی را در فضای مفصلی انجام خواهیم داد، البته در پانل فرمان نهایی که در انتها خواهیم دید این امکان
فراهم شده تا هم بتوان دستورات را در فضای دکارتی و هم در فضای مفصلی صادر کرد .

^{۳۳} Master



شکل ۵-۱- نمای شماتیکی از اجرای پانل کنترلی به روش دوم

۵-۳-۱- بررسی امکان ارتباط سریال ربات با رایانه

امکان ارتباط سریال رایانه با هر وسیله جانبی که سازگار با پروتکل‌های استاندارد برای ارتباط با رایانه داشته باشد امکان پذیر است از آنجا که تا کنون از نرم افزار قدرتمند MATLAB در شبیه سازی مسائل و فرایندهای کنترلی استفاده کرده ایم بر آن شدیم که بررسیهایی برای استفاده از این نرم افزار برای انجام محاسبات کنترلی در قسمت رایانه استفاده کنیم و همین موضوع باعث شد وارد مقوله شیرین ارتباط فضای MATLAB با دنیای خارج از رایانه شویم .

در این طرح رایانه و نرم افزار مطلب به عنوان تولید کننده سیگنالهای مطلوب حرکتی $\ddot{q}_d, \dot{q}_d, q_d$

بوده از طرفی با دریافت سیگنالهای خطای $e, \dot{e}, e = q - q_d$ توسط کنترلر طراحی شده سیگنالهای فرمان مورد نیاز برای محرکها را فراهم می کند ، اما نظر به اینکه حجم محاسبات و داده های انتقالی و از طرفی دقت مورد نظر برای محاسبات همگی مسائلی هستند که باعث شدند که برای نرخ انتقال اطلاعات اهمیت ویژه ای قائل شویم و طبق آزمایشاتی که برای یک رابط با چنین ارتباطی داشتیم این روش هم مناسب نمی باشد.

از آنجاکه برای به دست آوردن موقعیت و سرعت یک محرک دوار نیاز به یک کانال شمارنده و یک کانال تایمر داریم و نیز در میکروهای سری ATMEGA^{۳۴} حداکثر تا ۳ کانال تایمر، شمارنده موجود است و از طرفی برای ساختن سیگنالهای PWM ورودی راه اندازهای DC برای حرکت محرکها نیز نیاز به یک کانال PWM داریم در نتیجه بر آن شدیم که برای هر محرک از یک ATMEGA 32 استفاده کنیم یعنی برای هر رابط یک میکرو کنترلر و در نتیجه برای ۴ رابط از ۴ میکرو کنترلر استفاده می کنیم ، وظیفه این کنترلرها فقط دریافت اطلاعات مربوط به موقعیت و سرعت رابط و انتقال آن به میکرو کنترلر مادر است و متقابلاً دریافت اطلاعات مربوط به سیگنال PWM از میکرو کنترلر مادر و ارسال آن به راه اندازهاست .

در این طرح برای تنظیمات زمانبندی و از دست نرفتن اطلاعات در مسیر ارسال به رایانه از یک ATMEGA 32 به عنوان کنترلر مادر واسطه میان رایانه و ۴ میکرو کنترلر ATMEGA 32 استفاده شده است ، وظیفه این کنترلر دریافت اطلاعات مربوط به هر رابط از میکرو کنترلر مربوط به آن و ارسال آن به Simulink نرم افزار مطلب در رایانه و متقابلاً دریافت اطلاعات مربوط به پالس PWM مورد نیاز محرک هر رابط از Simulink نرم افزار مطلب و ارسال آنها به میکرو کنترلر مربوط به آن رابط می باشد .

این ارتباط یک ارتباط پیچیده است ، برای اجرای آن از پروتکل های ارتباطی مختلفی مانند :

گذرگاه موازی^{۳۵} , SPI , USART^{۳۷} , UART^{۳۶} می توان استفاده کرد ، از میان آنها ارتباط SPI برای ارتباط میان میکرو کنترلرها و UART برای ارتباط کنترلر واسط با رایانه استفاده شده است دلایل این انتخاب در زیر آمده است .

پروتکل ارتباطی SPI که از شاهکارهای میکروهای سری AVR می باشد یک روش ارتباطی سریال ۳ سیمه با فرکانس کاری ۱/۲ کلاک سیستم می باشد ، از آنجاکه ارتباط موازی ۱ بیتی بسیار دقیق و سریع می باشد اما مشکلاتی از نظر محدودیت پایه های ورودی/خروجی میکرو کنترلرها وجود دارد به این ترتیب که برای

1 Avr Family Micro Controller

^{۳۵} Parallel Bus

^{۳۶} Universal Synchrone Receive and Teransmit

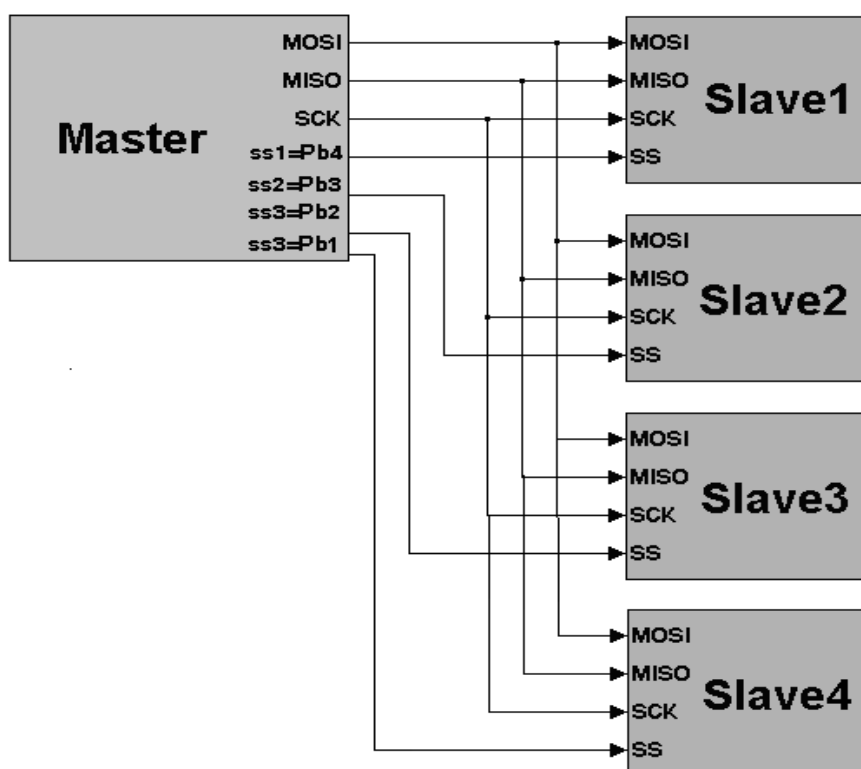
^{۳۷} Universal Synchrone and Asynchrone Receive and Teransmit

یک باس ۳ به ۱ که ۸ بیت را ارسال می کند برای کنترلر مادر ۲۴ پین برای دریافت داده و ۲۴ پین برای ارسال داده و ۲ پین برای آدرس دهی میکروهای مرنوس لازم است.

شایان ذکر است که چنانچه بخواهیم دریافت و ارسال اطلاعات را از روی یک خط مشترک انجام دهیم ممکن است اطلاعات تداخل کنند این محدودیت باعث شد از ارتباط موازی صرفنظر کنیم ، از طرفی چون میکروکنترلر مادر واسط توسط پروتکل UART با رایانه در ارتباط است از این ارتباط هم برای ارتباط میکروها با هم صرفنظر می کنیم .

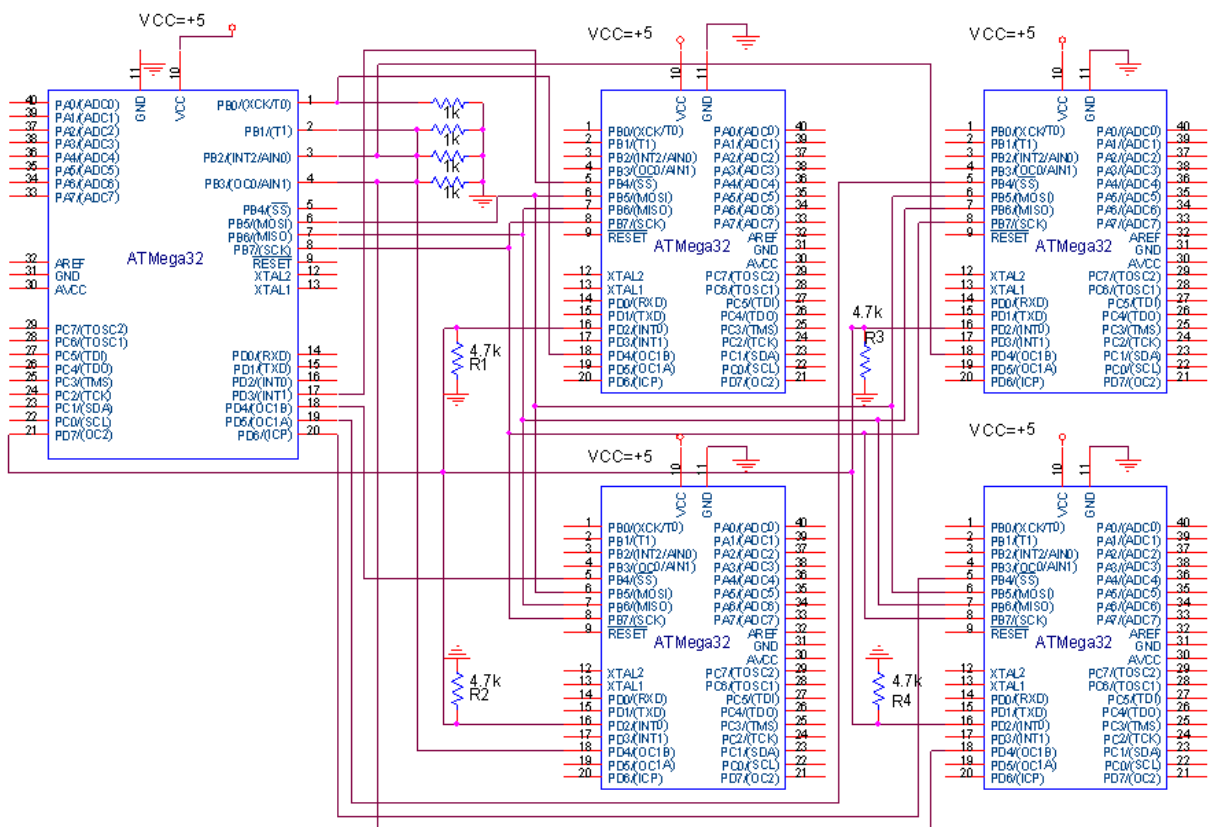
۵-۴- ارتباط SPI میکروکنترلرها با هم

در این حالت کنترلر رئیس دارای ۴ مرنوس S1,S2,S3,S4 می باشد شکل (۵-۲) بلوک دیاگرام ارسال و دریافت اطلاعات را در این پروتکل مشاهده می کنیم .



شکل ۵-۲- طریقه اتصال مرنوس, رئیس ها در SPI

در این حالت با فعال شدن پایه $\overline{SS}_1$ مربوط به هر مرئوس این کنترلر داده مربوط به خود را از روی خط MOSI^{۳۸} برمی دارد و داده ای که باید به رئیس ارسال کند را روی خط MISO^{۳۹} قرار می دهد. در این بین مهمترین موضوع زمانبندی ارسال و دریافت اطلاعات و انتخاب مرئوس ها می باشد که نیاز به برنامه نویسی قوی در این زمینه می باشد، سرعت انتقال اطلاعات در این پروتکل تا حدی بالاست که می توان کلاک ارسال سریال هر بیت را تا نصف کلاک پردازشگر افزایش داد.



شکل ۵-۳- مدار کامل ارتباط رئیس، مرئوس در ارتباط SPI

¹ Master Output Slave Input

با توجه به شکل (۳-۵) نحوه این ارتباط به صورت ترتیب زیر است :

۵-۴-۱- ارسال اطلاعات از رئیس به مرئوس ها

- ابتدا میکرو کنترلر رئیس پس از دریافت وقفه سریال از رایانه اطلاعات خود را باید به مرئوس ها ارسال کند .
- گزینه اول برای دریافت اطلاعات در میان مرئوس ها می باشد ابتدا طبق شکل (۴-۵) زیر پایه $\overline{SS}_1$ توسط پایه PD_3 میکرو کنترلر رئیس غیر فعال شده و به مرئوس ۱ وقفه ارتباط SPI می رسد به زیر برنامه دریافت خود می رود و اطلاعات را از رئیس دریافت می کند این جز از برنامه هم در شکل (۴-۵) آمده است .
- پس از اتمام ارسال اطلاعات به مرئوس ۱ PD_3 میکرو کنترلر رئیس فعال شده همین سیکل برای مرئوس ۲ و ۳ و ۴ توسط PD_4 و PD_5 و PD_6 اجرا خواهد شد.
- پس از اتمام ارسال به اطلاعات مرئوس ۴ میکرو کنترلر رئیس توسط پایه PD_7 که به کانال صفر وقفه هر چهار مرئوس متصل است وقفه می دهد و هر چهار مرئوس به طور همزمان شروع به اجرای فرمانهای کنترلی می کنند و نهایتا ربات به حرکت در می آید .

MASTER

```

Disable Interrupts
Reset Ss1
Spiout X , 1
Spiout Signteta1 , 1
X = 255
Spiout X , 1
Spiout Teta1 , 1
Set Ss1

Reset Ss2
Spiout X , 1
Spiout Signteta2 , 1
X = 255
Spiout X , 1
Spiout Teta2 , 1
Spiout X , 1
Spiout Signteta1 , 1
Set Ss2

Reset Ss3
Spiout X , 1
Spiout Signdist3 , 1
X = 255
Spiout X , 1
Spiout Dist3 , 1
Set Ss3

Reset Ss4
Spiout X , 1
Spiout Signteta4 , 1
X = 255
Spiout X , 1
Spiout Teta4 , 1
Set Ss4

Set Intall
Reset Intall

Enable Interrupts
Return

```

SLAVE's

```

S_isr:
Disable Interrupts
For C = 1 To 255
Spiout Pos(c) , 1
Spiout X , 1
Next C
For C = 1 To 255
Spiout Sign(c) , 1
Spiout X , 1
Next C

For C = 1 To 255
Locate 1 , 1
Lcd Pos(c)
Waitms 50
Next C
Wait 1
Cls

Reset Sm
Enable Interrupts
Timer1 = 0
Start Timer1
Return

```

شکل

۴-۵- جزئی از برنامه های انتقال اطلاعات از رئیس به مرئوس ها

۴-۵-۲- ارسال اطلاعات از مرئوس ها به رئیس

- هر میکرو کنترلر مرئوس پس از رسیدن محرک مربوطه اش (موتور مربوطه) به موقعیتی که برایش مشخص شده بود پایه PD_4 خود را فعال می کند.
- در همین حال میکرو کنترلر رئیس که پس از ارسال اطلاعات به مرئوس ها مشغول به سرکشی به پایه های PD_4 و PD_4 و PD_4 و PD_4 خود می باشد این فعال شدن پایه مرئوس تشخیص داده و سپس پایه مربوط به فعال کننده مرئوس یعنی $\overline{SS}_1$ را غیر فعال می کند.
- در این حالت مرئوس مربوطه شروع به ارسال اطلاعات به رئیس می کند پس از اتمام ارسال مرئوس مربوطه پایه PD_4 خود را غیر فعال می کند و از زیر برنامه وقفه SPI بیرون می آید و منتظر فرمان جدید از رئیس می ماند.
- در این حال رئیس هم پس از دریافت اطلاعات مرئوس مربوطه از زیر برنامه دریافت از آن مرئوس مربوطه بیرون آمده و دوباره مشغول سرکشی به مرئوس های دیگر می شود.

- پس از اتمام همه این مراحل باید ذکر شود که چنانچه فرمانی در این میان از سوی رایانه صادر گردد این سیکل دریافت اطلاعات بهم می خورد لذا بهتر است در زمان دریافت اطلاعات از مژغوس ها هیچ دستور عملی از رایانه به رئیس صادر نشود .

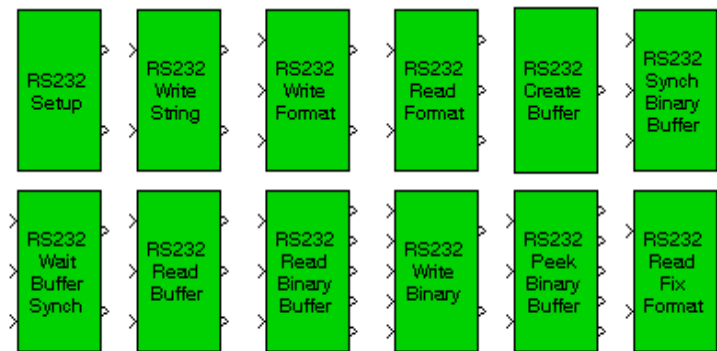
۵-۵- مطالعه پروتکل ارتباط سریال نرم افزار مطلب در محیط Simulink

در نرم افزار مطلبی که در دست است فایل های مربوط به پروتکل و Block Set های ارتباط سریال Simulink با محیط خارج وجود ندارد . بنابراین با جستجو در اینترنت و سایت Mathwork توانستیم بلوک های ارتباط سریال در سیمولینک مطلب را به دست آوریم راهنمای استفاده از این بلوک ها هم در همین سایت وجود دارد ، دسترسی به این بلوک ها با جستجو بر اساس نام RS232 به راحتی امکان پذیر است . با مطالعه این راهنما که ۶۰ صفحه است و چند روز تلاش بی وقفه می توان نتایج جذابی به دست آورد [۶].

ابتدا کار را با نمونه گیری از سرعت یک موتور DC و ارسال آن به Simulink و نمایش آن در Scope آغاز کردیم نتایج خوبی به دست آمد اما سرعت انتقال اطلاعات یک بایت در ۵ میلی ثانیه بود که این نرخ مسلما با افزایش اطلاعات کاهش می یابد ، لذا بر آن شدیم که از رایانه به عنوان سیگنال ژنراتور استفاده کنیم .

۵-۵-۱- ارتباط سریال بین رئیس و رایانه

از آنجاکه باید اطلاعات رئیس از طریق پورت سریال به محیط Simulink مطلب ارسال و از آن دریافت شود و با توجه به آنچه در گزارش قبلی به آن اشاره کردیم که هیچ بلوکی در Simulink برای ارتباط سریال وجود ندارد ، لذا در این پروژه بر آن شدیم که از RS232 Block Set های موجود در سایت Mathwork شکل (۵-۵) که پروتکل ارتباطی نسبتا پیچیده ای دارد استفاده کنیم.



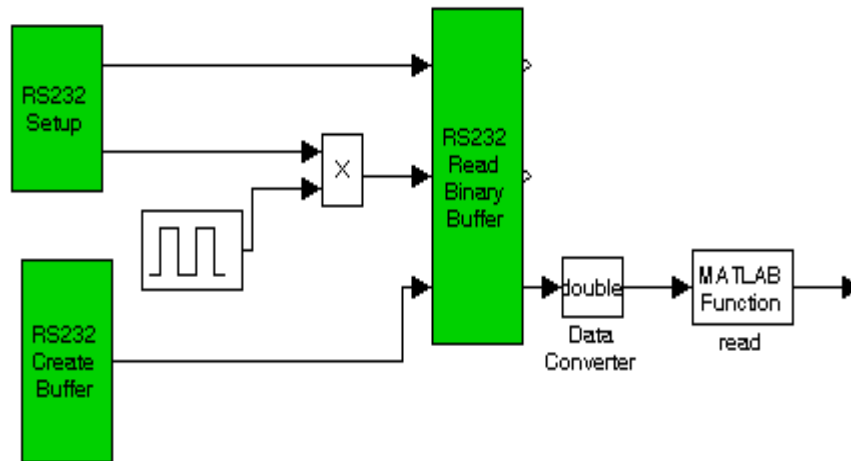
RS232 Blockset Library 1.2

شکل ۵-۵- بلوکهای ارسال و دریافت RS232

۵-۲-۵- دریافت داده توسط RS232 Blocksets

بدون پرداختن به جزئیات کارمی توان طریقه دریافت داده سریال توسط RS232 Blocksets در محیط simulink را به صورت زیر با فرض رعایت زمان بندی های لازم و انتخاب و اتصال درست اجزا به صورت شکل (۵-۶) به صورت زیر خلاصه کرد :

- ۱- در ابتدای کار بافر سریال خالی و حاوی مقدار صفر می باشد.
 - ۲- اگر رقم دریافتی منفی باشد مقدار ۴۵ در بافر سریال ظاهر می شود و گرنه (رقم اول داده ارسال شده + ۴۸) در بافر ظاهر می شود.
 - ۳- اگر در بافر عدد ۱۰ ظاهر شده باید دوباره برای دریافت رقم بعدی آماده باشیم.
 - ۴- اگر در بافر عدد ۱۳ ظاهر شده باشد یعنی دریافت داده ارسالی کامل شده است به ۲ می رویم .
 - ۵- (رقم بعدی داده ارسال شده + ۴۸) در بافر سریال قرار می گیرد به ۳ می رویم .
- حال باید تعداد رقمها را تا ظاهر شدن عدد ۱۳ در بافر سریال بشماریم و پس از ظهور ۱۳ در بافر رقمهای دریافتی را به ترتیب در توانهای مختلف ۱۰ ضرب کنیم تا عدد مربوط را به دست آوریم .



شکل ۵-۶- دریافت اطلاعات سریال توسط RS232 Binary Blockset

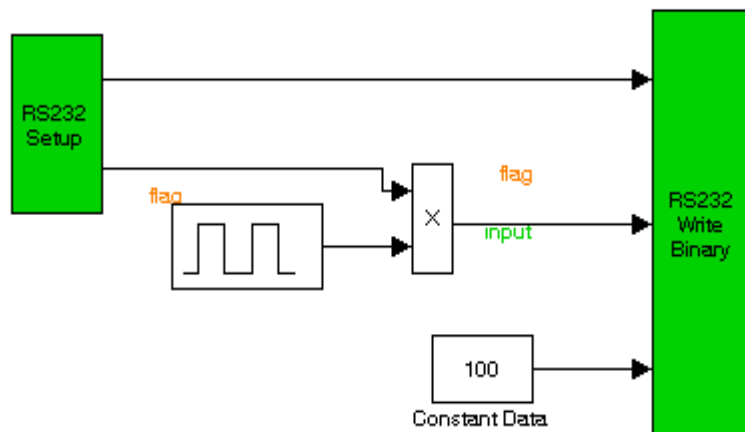
۵-۵-۳- ارسال داده توسط RS232 Blocksets

برخلاف پیچیدگی دریافت اطلاعات توسط RS232 Blocksets ارسال اطلاعات از Simulink بسیار آسان انجام می‌گیرد با رعایت زمان بندی های لازم و انتخاب و اتصال درست اجزا به صورت شکل (۵-۷) تنها کافیت در بلوک ارسال^{۴۰} نوع داده ارسالی را که یکی از اقلام زیر است انتخاب کنیم بعد داده گذاشته شده در ورودی بلوک به درگاه سریال ارسال می‌گردد .

- Bit
- Uint8^{۴۱}
- Int8
- Uint16
- Int16
- Uint32
- Int32
- Single
- Double

^{۴۰} Send

^{۴۱} Unsigned Integer Number



شکل ۵-۷- ارسال اطلاعات سریال توسط RS232 Binary Blockset

۵-۵-۴- چگونگی ارسال و دریافت رئیس از RS232 Blocksets

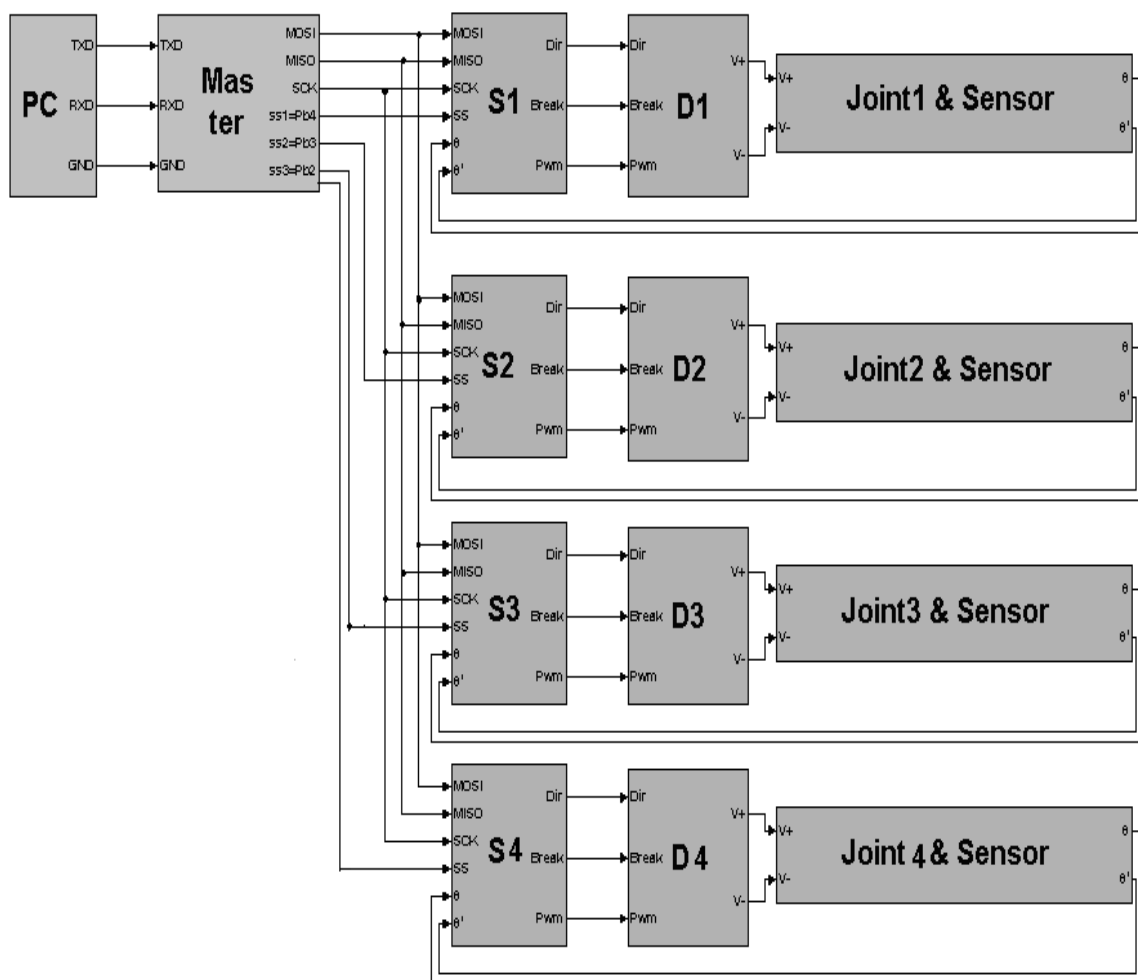
دریافت و ارسال اطلاعات توسط رئیس هم چندان مشکل نیست ارسال اطلاعات تنها با دستور چاپ^{۴۲} ولی دریافت اطلاعات کمی پیچیده تر است و نیاز به یک زمان بندی تجربی دارد ، هنگامی که اینترپت دریافت سریال فعال می گردد به محض دریافت اولین داده پرچم آن بالا رفته و به زیر برنامه دریافت از پورت سریال می رود در این زیر برنامه توسط دستور^{۴۳} Waitkey() منتظر می مانیم تا دریافت اطلاعات تکمیل گردد بعد دوباره پرچم را غیر فعال کرده و از زیر برنامه دریافت بیرون می رویم.

۵-۵-۵- جایگاه ویژه رئیس در دو پرتکل ارتباطی

^{۴۲} Print

^{۴۳} One of the MEGA AVR Instruction

از آنجاکه در این پروژه میکروکنترلر رئیس در دو پروتکل ارتباطی مختلف با دو سیستم مختلف ارتباط دارد زمان بندی دقیق و اصولی بعلاوه در نظر گرفتن نکات تجربی فوق العاده مهم است و شاید عدم دقت در این زمان بندی مجری چنین طرحهایی را ناامید کند ، بلوک دیاگرام سیستم حلقه بسته در حالت کلی در شکل (۵-۸) نشان داده شده است که در آن D_i همان راه انداز مربوط به بازوی i ام و S_i میکرو کنترلر مربوط به بازوی i ام می باشند .



شکل ۵-۸- بلوک دیاگرام سیستم در حالت کلی

در این پروژه ارسال و دریافت اطلاعات در کسری از ثانیه تقریباً ۵ms انجام می شود که اطلاعات دریافتی توسط رئیس از هر ۴ مرئوس شامل جهت ، موقعیت ، سرعت هر رابط می باشد که بعد توسط رئیس به رایانه ارسال می گردد پس از انجام محاسبات در رایانه اطلاعات مربوط به تحریک موتورهای مربوط به هر رابط که یک عدد ۱ بیتی (۰-۲۵۵) می باشد توسط رئیس از رایانه دریافت و بلافاصله به مرئوس مربوطه ارسال می گردد در مرئوس ها این اطلاعات کوانتیزه می شود و سپس به راه انداز مربوطه ارسال می گردد و آنها هم محرکها را فعال می کنند، این سیکل ارسال و دریافت اطلاعات در حدود ۵۰ms به طول می انجامد که با توجه به سرعتها و اطلاعات نمونه برداری شده توسط کدگذارها زمان مطلوبی نیست لذا در نهایت بر آن شدیم که برای ارتباط با رایانه، رایانه تنها تولید کننده سیگنالهای فرمان و مطلوب باشد .

برنامه های Simulink مربوط به ارتباط سریال رئیس و رایانه در محیط MATLAB با توضیحات لازم در پیوست ۳ آمده است .

۵-۶- ارتباط رئیس با نرم افزار Visual Basic

پس از تجربه ارتباط سریال با Simulink و درک این موضوع که آن نرخ انتقال برای ما کافی نیست تصمیم به ساختن یک پانل کنترل- فرمان کاربر پسند و عامه پسند گرفتیم و نتیجه مطالعات برای ارتباط سریال با MATLAB فقط یک تجربه بود .

فضای فرمان را در این محیط Visual Basic طراحی نموده در عین حال ارتباط رئیس با سایر مرئوس ها با همان پروتکل SPI می باشد مدار الکتریکی رئیس با پورت سریال به صورت شکل (۵-۹) می باشد.

در این شکل از چیپ MAX232 برای تطبیق سطوح ولتاژ TTL و سطوح ولتاژ پورت سریال که بین ۱۲و۱۲- می باشند استفاده شده است در این چیپ از ۴ خازن که نقش چند برابر کننده ولتاژ و تغییر سطوح ولتاژ را

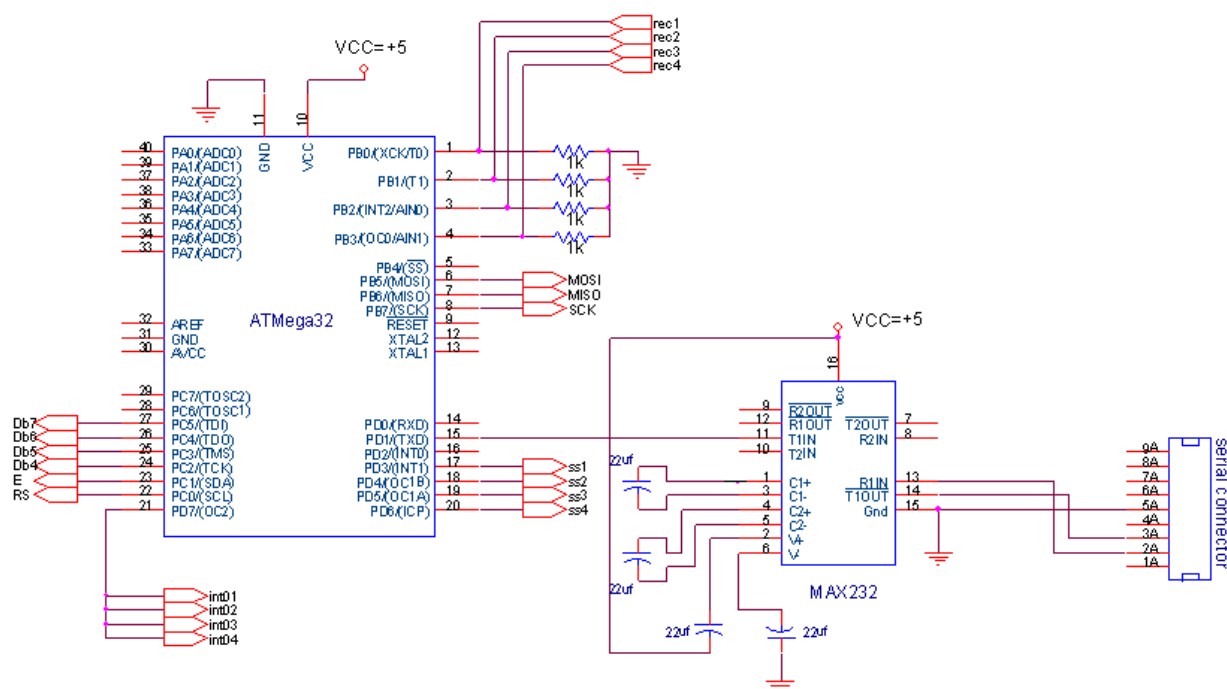
ایفا می کنند استفاده شده اند ، در این شکل تمام خطوط ارتباطی میکرو کنترلر رئیس به صورت باسهای جداگانه نمایش داده شده است :

۱- باس SPI (MISO-MOSI-SCK)

۲- باس ارسال فرمان انتخاب مرنوس ها برای ارسال فرمان به آنها $\overline{SS}_i$

۳- باس وقفه به مرنوس ها

۴- باس دریافت آمادگی مرنوس ها برای ارسال گزارش



شکل ۵-۹- مدارالکترونیکی ارتباط رئیس با پورت سریال و نمایش باسهای آن

شمایی از این فضای طراحی شده در این محیط در شکل (۵-۱۰) نمایش داده شده است که در این فضا هم امکان صدور فرمانهای مفصلی موجود می باشد و هم دکارتی با پر کردن جعبه متن^{۴۴} های بالا و فشردن کلید " Inverse Kinematic Transform " محاسبات جهت به دست آوردن متغیرهای مفصلی ربات اسکارا از

^{۴۴} Text Box

روی متغیرهای دکارتی آغاز می گردد و در جعبه متن های زیرین نمایش داده می شود سپس با فشردن کلید " Send to Micro " این اطلاعات مفصلی به رئیس ارسال شده و نهایتا به مرئوس های مربوطه ارسال شده و ربات شروع به حرکت می کند .

اما در جعبه متن های تحتانی امکان ارسال فرمانهای مستقیم مفصلی هم به چشم می خورد در این جا می توان فرمانها را به صورت مفصلی وارد و سپس با فشردن کلید " Send to Micro " این اطلاعات مفصلی به رئیس ارسال شده و نهایتا به مرئوس های مربوطه ارسال شده و ربات شروع به حرکت می کند . با توجه به اینکه مچ در حرکت ربات به سمت هدف دخیل نیست لذا متغیر مفصلی مربوط به چرخش مچ در هر دو روش فرمان به صورت مستقل وارد می شود .

در وارد کردن ورودیها باید از محدودیتهای سینماتیکی و مکانیکی (ساختاری) ربات اطلاعات کافی داشت این اطلاعات اولیه به صورت بر چسبهایی در بالای تمامی جعبه متن های ورودی نوشته شده است این محدودیتهای به صورت زیر هستند :

$$-125^{\circ} < \theta_1 < 125^{\circ}$$

$$-135^{\circ} < \theta_2 < 135^{\circ}$$

$$0 < d < 15 \text{ cm}$$

$$-180^{\circ} < \theta_4 < 180^{\circ}$$

محدودیتهای ساختاری:

محدودیتهای سینماتیکی :

$$-35 < X < 35 \text{ cm}$$

$$-35 < Y < 35 \text{ cm}$$

$$25 < X^2 + Y^2 < 1225$$

شكل

برنامه های کامل Visual basic در پیوست ۳ آمده است .

٦-١- مقدمة

14

ویژگیهای هندسی و وابسته به زمان حرکت را در بر می گیرد. و نیز رابطه بین این حرکتهای و نیروها و گشتاورهای ایجاد کننده آنها به علم دینامیک مربوط می شود.

به منظور ساده تر کردن مطالعه بازوهای مکانیکی ماهر که هندسه پیچیده ای دارند، به قسمتهای مختلف آنها چهارچوبهایی وصل می کنیم و سپس رابطه بین این چهارچوبها را به دست می آوریم. یکی از مسائلی که در مطالعه سینماتیک بازوهای مکانیکی ماهر مطرح است، چگونگی تغییر این چهارچوبها به هنگام حرکت بازو است. با توجه به اینکه کنترل ربات در فضای مفصلی انجام می گیرد لذا باید زوایای مطلوب رابطها به عنوان فرمان کنترلی به کنترلرها اعمال گردد ، اما از آنجا که از نظر فضای کار معمولا فرمانهای رباتها برای رسیدن به یک موقعیت خاص در فضای دکارتی تعریف می شوند مایلیم که فرمانهای دکارتی را به زوایای مفصلی تبدیل کنیم این تبدیل و حالت عکس آن مسئله اصلی مورد بحث در سینماتیک ربات می باشد .

۶-۲- سینماتیک مستقیم و سینماتیک وارون

مفهوم سینماتیک در حقیقت رابطه ای است که بین متغیرهای فضای مفصلی q و متغیرهای فضای دکارتی X برقرار می باشد و توسط این رابطه می توان موقعیت و جهت مجری نهایی را نسبت به موقعیت مفاصل و برعکس به دست آورد.

تعریف سینماتیک مستقیم :

محاسبه موقعیت و جهت مجری نهایی یا همان متغیرهای فضای دکارتی X بر حسب موقعیت مفاصل یا همان متغیرهای فضای مفصلی q است .

تعریف سینماتیک وارون :

محاسبه موقعیت مفاصل یا همان متغیر های فضای مفصلی q بر حسب موقعیت و جهت مجری نهایی متغیر های فضای دکارتی X است.

هر بازوی مکانیکی ماهر را می توان به صورت مجموعه ای از اجسام صلب در نظر گرفت که به وسیله مفصل به یکدیگر اتصال یافته اند. این اجسام را رابط می نامند، مفصل دو رابط مجاور را به هم متصل می سازد. سینماتیک مستقیم :

ماتریس تبدیل $T_{i-1 \times 4}^i$ که از طریق پارامترهای دناویت هارتنبرگ به دست می آید در حقیقت همان ماتریس سینماتیک مستقیم است که موقعیت دستگاه i ام را نسبت به دستگاه $i-1$ ام (d_{i-1}^i) و جهت دستگاه i ام را نسبت به دستگاه $i-1$ ام ($R_{i-1 \times 3}^i$) مشخص می کند و اگر برای هر مفصل ماتریس های تبدیل را به دست آورده و در هم ضرب کنیم در نهایت ماتریس T_0^n ماتریس سینماتیک مستقیم یک ربات با $n=3$ یعنی ۳ رابط به دست خواهد آمد.

در اینجا برای ربات اسکارا با $n = 3$ بنا به مختصات بندی دناویت هارتنبرگ ماتریس T_0^n را به دست می آوریم.

۳-۶- نمایش دناویت هارتنبرگ و ربات اسکارا

یک دستورالعمل متداول برای تعیین دستگاه های مختصات در کاربردهای رباتیک، دناویت هارتنبرگ یا دستورالعمل D-H است. در این دستورالعمل هر ماتریس تبدیل $T_{i-1 \times 4}^i$ به صورت ضرب چهار تبدیل اساسی نشان داده می شود.

$$T_{i-1}^i = Rot_{z_{i-1}, \theta_i} \times Trans_{z_{i-1}, d_i} \times Trans_{x_i, a_i} \times Rot_{x_i, \alpha_i} \quad (1-6)$$

$$T_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۲-۶)$$

$$T_{i-1}^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۳-۶)$$

که در آن چهار کمیت θ_i ، a_i ، d_i و α_i پارامترهای رابط i و مفصل i هستند. پارامترهای ذکر شده به صورت زیر تعریف می شوند.

۱. a_i طول عمود مشترک (فاصله از Z_{i-1} تا Z_i در امتداد X_i)

۲. d_i فاصله بین مبدا و نقطه تقاطع محورهای x_i و z_{i-1} (فاصله از x_{i-1} تا x_i در امتداد Z_{i-1}).
هنگامی که مفصل i کشویی باشد d_i متغیر است.

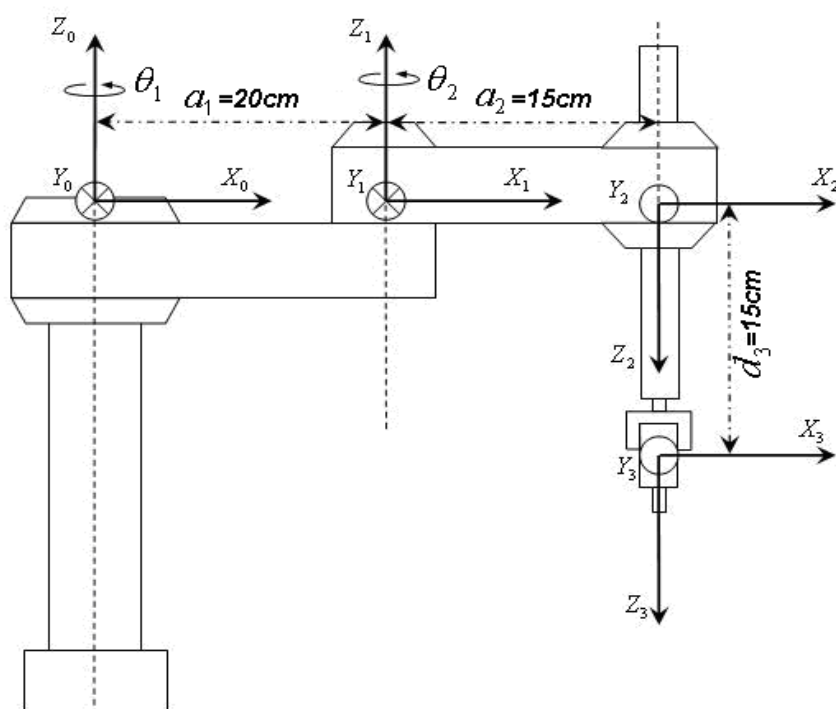
۳. α_i زاویه بین Z_{i-1} و Z_i که حول x_i در جهت راستگرد اندازه گیری می شود.

۴. θ_i زاویه بین محور X_{i-1} و x_i که حول محور Z_{i-1} در جهت راستگرد اندازه گیری می شود.
هرگاه مفصل i لولایی باشد θ_i متغیر است.

پارامترهای a_i و α_i پارامترهای ثابتی هستند که با هندسه رابط تعیین می شوند. a_i بیانگر طول لینک و α_i زاویه پیچش بین دو مفصل می باشد. یکی از دو پارامتر d_i و θ_i وقتی که مفصل حرکت می کند بسته به اینکه مفصل کشویی باشد یا لولایی تغییر می کنند.

برای مفاصل ربات ها دستگاه بندی ها را بر همین اساس تعیین می کنند که البته در نمایش DH باید نکته ای را در مورد جهت انتخاب دستگاه ها حتما رعایت کرد و آن این است که محور x_i دستگاه مربوط به

مفصل i ام باید عمود متقاطع بر محور Z_{i-1} دستگاه $i-1$ ام باشد. اصولاً در ربات ها برای رابط ها و مفصل ها اندیس در نظر گرفته می شود که معمولاً پایه ربات را رابط صفر و مجری نهایی را n گفته می شود. در ربات اسکارا که دارای ۳ مفصل می باشد این دستگاه بندی را در شکل (۱-۶) نشان داده شده است و پارامترهای سینماتیکی مربوط به آن در شکل (۲-۶) مشاهده می شود.



شکل ۱-۶- شماتیک ربات SCARA ۳-رابط

joint	d	a	θ	α
1	0	$a_1=20cm$	θ_1	0
2	0	$a_2=15cm$	θ_2	180
3	$d_3=15cm$	0	0	0

شکل ۲-۶- جدول پارامترهای سینماتیک اسکارا ۳-رابط

ماتریسهای انتقال یا سینماتیک مستقیم با توجه به (۱-۶) ، (۲-۶) به صورت زیر به دست می آیند :

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۴-۶)$$

$$T_1^2 = \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & -\cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۵-۶)$$

$$T_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۶-۶)$$

$$T_0^3 = T_0^1 \times T_1^2 \times T_2^3 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ \sin(\theta_1 + \theta_2) & -\cos(\theta_1 + \theta_2) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۷-۶)$$

که در آن ماتریس های دوران و انتقال مجری نهایی اسکارا به شرح زیر می باشند.

$$R_0^3 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 \\ \sin(\theta_1 + \theta_2) & -\cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (۸-۶)$$

$$d_0^3 = \begin{bmatrix} a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \\ a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \\ -d_3 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (۹-۶)$$

در ربات اسکارا متغیرهای مفصلی را تحت بردار q نمایش داده می شود. بنابراین به ازای هر مقداری از بردار q ، ماتریس تبدیلی به دست می آید که موقعیت و جهت مجری نهایی را به صورت عددی معلوم می کند.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ d_3 \end{bmatrix} \quad T_0^3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۱۰-۶)$$

با توجه به روابط (۶-۷) و (۶-۱۰) مساوی قرار دادن ستونهای چهارم آنها متغیرهای دکارتی به صورت زیر به دست می آیند:

$$X = a_2 \cos(\theta_1 + \theta_2) + a_1 \cos(\theta_1) \quad (۱۱-۶)$$

$$Y = a_2 \sin(\theta_1 + \theta_2) + a_1 \sin(\theta_1) \quad (۱۲-۶)$$

$$Z = -d_3 \quad (۱۳-۶)$$

سینماتیک وارون

همان طور که قبلا اشاره شد مفهوم سینماتیک وارون محاسبه مقادیر متغیرهای مفصلی از روی متغیرهای دکارتی می باشد. باید توجه داشت که درایه های ماتریس T_0^n تابعی از متغیرهای مفصلی هستند و برای محاسبه متغیرهای دکارتی با استفاده از ماتریس T_0^n و تشکیل دستگاه n معادله n مجهول غیر خطی از طریق معلوم بودن موقعیت و جهت مجری نهایی پی به مقداری متغیرهای مفصلی می بریم. اصولا حل یک چنین دستگاه معادلاتی بسیار مشکل می باشد مخصوصا برای ربات های بیش از ۳ مفصل. بنابراین در ربات

های ۶ مفصل که ۶ درجه آزادی دارند و از این ۶ مفصل، ۳ عدد مچ ربات و ۳ عدد بازوی آنرا تشکیل می دهند، از تکنیک خاصی به نام مجزا سازی سینماتیکی استفاده می شود.

در این تکنیک با معلوم بودن موقعیت مجری نهایی d_0^n و انحراف رابط آخر (مربوط به مچ)، متغیر های مفصلی مربوط به بازوی ربات را به دست می آورند که به سینماتیک موقعیت معروف است و با معلوم بودن ماتریس دوران R_0^n ، متغیر های مفصلی مربوط به مچ ربات را محاسبه می کنند که به سینماتیک جهت معروف است.

حال برای به دست آوردن سینماتیک وارون کافیت از روی معادلات (۶-۱۳) و (۶-۱۲) θ_1, θ_2 را به دست آوریم که نیاز به محاسبات پیچیده ای دارد وجود یا عدم وجود جواب، به فضای کاری بازوی مکانیکی ماهر بستگی دارد. به بیان ساده، فضای کاری حجمی از فضا است که مجری نهایی بازو قادر است آن را تحت پوشش قرار دهد. شرط وجود جواب آن است که نقطه تعیین شده به عنوان هدف، در محدوده فضای کاری باشد. گاهی بهتر است دو تعریف جداگانه از فضای کاری ارائه شود، فضای کاری ماهر و فضای کاری دسترسی پذیر. فضای کاری ماهر، حجمی از فضا است که مجری نهایی ربات می تواند با همه جهت گیری های خود به آن دسترسی پیدا کند. فضای کاری دسترسی پذیر، حجمی از فضا است که مجری نهایی می تواند حداقل در یک جهت گیری خود، به آن دسترسی یابد. روشن است که فضای کاری ماهر، زیر مجموعه فضای کاری دسترسی پذیر است [۷].

$$s = a_1^2 + a_2^2$$

$$q = a_1^2 - a_2^2$$

$$r^2 = X^2 + Y^2$$

$$\theta_1 = a \tan 2\left(\frac{Y}{X}\right) - a \tan 2\left[\frac{\sqrt{2rs - q^2 - r^2}}{r + q}\right]$$

(۱۵-۶)

$$\theta_2 = a \tan 2 \left[\frac{\sqrt{2rs - q^2 - r^2}}{r + q} \right] \quad (۱۶-۶)$$

$$d_3 = -Z$$

(۱۷-۶)

بدین ترتیب متغیرهای مفصلی لازم به دست می آیند تابع atan2() تابعی است که به علامت صورت و مخرج رابطه داخل پرانتز حساس بوده و دقیقاً ناحیه مربوطه را مشخص می کند .

این محاسبات با محاسبات منتجه از روش مرسوم دناویت هارتنبرگ [۸] برابر است برای ارتباط با MATLAB از برنامه زیر و در برنامه Visual Basic از نتایج بالا استفاده کرده ایم این برنامه ها به صورت زیر هستند:

۶-۳-۱- محاسبه سینماتیک مستقیم اسکارا توسط MATLAB

```
L1=link([0 1 0 0 0]);
```

```
L2=link([pi 1 0 0 0]);
```

```
L3=link([0 0 0 0 1]);
```

```
L4=link([0 0 0 .5 0]);
```

```
figure(1)
```

```
scara=robot({L1,L2,L3,L4},'SCARA');
```

```
s0=[pi/2,pi/2,1,0];
```

```
plot(scara,s0);
```

```
T=fkine(scara,s0)
```

```
q=ikine(scara,T)
```

```
Ts0=fkine(scara,s0)
```

```

figure(2)
scara1=robot({L1,L2,L3,L4},'SCARA1');
s1=[pi/2,pi/2,0,pi/2];
plot(scara1,s1);
T1=fkine(scara1,s1)
q1=ikine(scara1,T1)
T1_q1=fkine(scara1,s1)

```

در اینجا ابتدا ربات اسکارا توسط دستورات Link تعریف می شود بعد برای آن بردار متغیرهای مفصلی S_0, S_1 را تعریف می کنیم سپس دستور fkine ماتریس سینماتیک مستقیم T_i که شامل اطلاعات دوران و انتقال می باشد را به ما می دهد در قسمت بعد توسط دستور ikine سینماتیک معکوس T_i را که به صورت یک بردار شامل متغیرهای مفصلی می باشد به تحت عنوان q_i ارائه می کند سینماتیک معکوس بسته به کنترل موقعیت و جهت از روی هر یک از ماتریسهای دوران یا بردار انتقال قابل محاسبه می باشد.

۶-۳-۲- محاسبه سینماتیک وارون اسکارا توسط Visual Basic

```

Private Sub IKEN_Click()
sendcmd.Enabled = False
a2 = 0.2
a3 = 0.15
d1 = 0.15
s = (a2) ^ 2 + (a3) ^ 2
q = (a2) ^ 2 - (a3) ^ 2
x = 0.01 * Val(tx.Text)
y = 0.01 * Val(ty.Text)
z = 0.01 * Val(tz.Text)

```



```

theta = Val(tt.Text)
r = x ^ 2 + y ^ 2
xy.Text = 10000 * r
If r < 0.0025 Or r > 0.1225 Then
tx.Text = ""
ty.Text = ""
tet2.Text = ""
tet1.Text = ""
GoTo l1
End If
vy = ((2 * r * s) - (q ^ 2) - (r ^ 2)) ^ (0.5)
vx1 = r - s
vx2 = r + q
dd3 = d1 - z
d3 = Int(100 * dd3)
tt2 = (180 / 3.1416) * Atn(vy / (vx1 + 0.0001))
If vx1 < 0 Then tt2 = 180 + tt2
ttt1 = (180 / 3.1416) * Atn(y / (x + 0.0001))
If x < 0 And y > 0 Then ttt1 = 180 + ttt1
If x < 0 And y <= 0 Then ttt1 = ttt1 - 180
tt1 = (180 / 3.1416) * Atn(vy / (vx2 + 0.0001))
If vx2 < 0 Then tt1 = 180 + tt1
tt1 = ttt1 - tt1
tet2.Text = Int(tt2)
tet1.Text = Int(tt1)
If Int(tt2) > 135 Or Int(tt2) < -135 Or Int(tt1) > 125 Or Int(tt1) < -125 Then
ty.BackColor = &H8080FF
tx.BackColor = &H8080FF
End If
If d3 > 15 Or d3 < 0 Then
td3.Text = d3

```

```

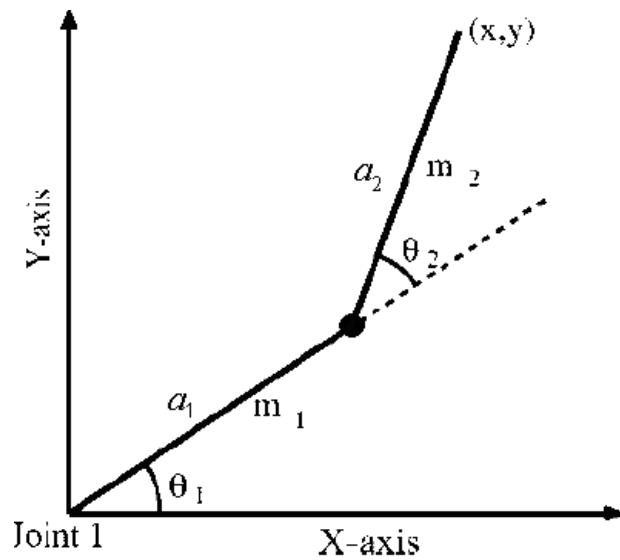
tz.BackColor = &H8080FF
End If
If theta > 180 Or theta < -180 Then
tet.Text = theta
tt.BackColor = &H8080FF
End If
If Int(tt2) > 135 Or Int(tt2) < -135 Or Int(tt1) > 125 Or Int(tt1) < -125 Or d3 > 15 Or d3 < 0 Or theta > 180 Or theta < -180 Then GoTo l1
t2 = Int(tt2)
td3.Text = d3(t1 = Int(tt1))
tet.Text = theta
tz.BackColor = &HFFFFFF
tt.BackColor = &HFFFFFF
tx.BackColor = &HFFFFFF
ty.BackColor = &HFFFFFF
sendcmd.Enabled = True
l1:
End Sub

```

۶-۳-۳- سینماتیک مستقیم اسکارا و ربات لولایی ۲-رابط

روش قبلی قدری پیچیده و مشکل به نظر می‌رسد اما پیشنهاد می‌کنیم که سینماتیک مستقیم را به صورت بدون واسطه از درک دقیق ارتباط میان حرکت اسکارا و بازوی ماهر ۲-رابط لولایی به دست آورد

با توجه به شکل (۶-۳) زیر به سادگی می‌توان متغیرهای دکارتی را از روی زوایای ۲ مفصل به دست آورد به ترتیب زیر :



شکل ۶-۳- نمودار حرکت در فضای مفصلی ربات لولایی ۲-رابط

با استفاده از قانون کسینوس ها خواهیم داشت :

$$\cos(\theta_2) = \frac{X^2 + Y^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (۱۵-۶)$$

$$\theta_2 = \cos^{-1}\left(\frac{X^2 + Y^2 - a_1^2 - a_2^2}{2a_1a_2}\right) \quad (۱۶-۶)$$

با استفاده از تبدیل cos به sin خواهیم داشت :

$$\sin(\theta_2) = \pm \sqrt{1 - \left(\frac{X^2 + Y^2 - a_1^2 - a_2^2}{2a_1a_2}\right)^2} \quad (۱۷-۶)$$

θ_1 و θ_2 از رابطه زیر به دست می آید :

$$\theta_2 = \tan^{-1} \left[\frac{\pm \sqrt{1 - \left(\frac{X^2 + Y^2 - a_1^2 - a_2^2}{2a_1a_2}\right)^2}}{\frac{X^2 + Y^2 - a_1^2 - a_2^2}{2a_1a_2}} \right] \quad (۱۸-۶)$$

$$\theta_2 = \tan^{-1} \left[\frac{Y}{X} \right] - \tan^{-1} \left[\frac{a_2 \sin(\theta_2)}{a_1 + a_2 \cos(\theta_2)} \right] \quad (۱۹-۶)$$

که این روابط دقیقا مانند روابط سینماتیک مستقیم اسکارا می باشد ، رابطه سینماتیکی رابط کشویی هم که در راستای محور Z حرکت می کند نیز به سادگی قابل حصول است :

$$Z = -d3 \quad (۲۰-۶)$$

برای مقادیر مثبت و منفی در رابطه (۱۸-۶) دسترسی به هر دو جواب آرنج بالا و آرنج پائین امکان پذیر است .

۴-۶- مدل سازی ربات اسکارا

در اینجا معادله های حرکت یک بازوی مکانیکی ماهر را بررسی می نماییم. این معادله ها، چگونگی ایجاد حرکت در ربات را به وسیله گشتاورهای اعمال شده توسط کار اندازها، یا اعمال نیروهای خارجی بر ربات نشان می دهد. درباره دینامیک مطالب زیادی بیان شده اند که برخی از این روشهای به دست آمده در دینامیک کاربردی بسیار مناسب در رباتیک دارند، به ویژه روشهایی که در آنها به ماهیت زنجیری بازوهای مکانیکی ماهر توجه شده است.

دو مسئله را در ارتباط با دینامیک بازوهای مکانیکی ماهر حل می کنیم. در مسئله اول با داشتن مشخصات θ ، $\dot{\theta}$ و $\ddot{\theta}$ مربوط به یک نقطه از مسیر، می خواهیم بردار گشتاورهای مفصلی τ را به دست آوریم. که رابطه های دینامیکی به کار رفته برای حل این مساله، در کنترل بازوهای مکانیکی ماهر سودمند هستند.

در مسئله دوم، چگونگی حرکت مکانیزم ربات در نتیجه اعمال مجموعه ای از گشتاور های مفصلی، محاسبه خواهند شد. به عبارت دیگر، مقادیر θ ، $\dot{\theta}$ و $\ddot{\theta}$ محاسبه می شوند. حل این مساله برای شبیه سازی ربات سودمند است.

با توجه به عمومیت ساختار مکانیکی ربات اسکارا و نزدیکی این ربات از نقطه نظر معادلات دینامیکی به معادلات ربات لولایی ۲- رابط تنها به ذکر معادلات ساده شده حرکت ربات می پردازیم [۹]:

متغیرهای زیر را تعریف می کنیم:

m_i : جرم رابط i ام

a_i : طول رابط مفصلی i ام

d_i : طول رابط کشویی i ام

C_2 : $\cos(\theta_2)$

S_2 : $\sin(\theta_2)$

b_i : ضریب اصطکاک رابط i ام با $i-1$ ام

θ_i : موقعیت رابط i ام

$\dot{\theta}_i$: سرعت رابط i ام

$\ddot{\theta}_i$: شتاب موقعیت رابط i ام

ماتریس اینرسی کلی اسکارا ۳- رابط به صورت زیر به دست آمده است:

$$D(\theta) = \begin{bmatrix} \left(\left(\frac{m_1}{3} + m_2 \right) a_2^2 + m_2 C_2 a_2 a_3 + \frac{m_2 a_3^2}{3} + m_3 (a_2^2 + 2a_2 a_3 C_2 + a_3^2) \right) & - \left(\frac{m_2 a_2 a_3 C_2}{2} + \frac{m_2 a_2^2}{3} + m_3 (a_2 a_3 C_2 + a_3^2) \right) & 0 \\ - \left(\frac{m_2 a_2 a_3 C_2}{2} + \frac{m_2 a_2^2}{3} + m_3 (a_2 a_3 C_2 + a_3^2) \right) & \left(\frac{m_2 a_2^2}{3} + m_3 a_3^2 \right) & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

(۲۱-۶)

ماتریس گشتاور جانب مرکز ، کوریولیس به صورت زیر به دست آمده که تا حد امکان ساده سازی شده است

:

$$C = -a_2 a_3 S_2 \begin{bmatrix} (m_2 + 2m_3) \dot{\theta}_1 \dot{\theta}_2 - \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_2^2 \\ \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_1^2 \\ 0 \end{bmatrix} \quad (۲۲-۶)$$

در نهایت با دخیل کردن دینامیک موتورهای معادله حرکت رابط اول به صورت زیر به دست می آید :

$$\begin{aligned} \tau_1 = & \left[\left(\frac{m_1}{3} + m_2 + m_3 \right) a_2^2 + (m_2 + 2m_3) a_2 a_3 C_2 + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right. \\ & \left. - \left[\left(\frac{m_2}{2} + m_3 \right) a_2 a_3 C_2 + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right] \ddot{\theta}_2 \right. \\ & \left. + b_1 (\dot{\theta}_1) - a_2 a_3 S_2 \left[(m_2 + 2m_3) \dot{\theta}_1 \dot{\theta}_2 - \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_2^2 \right] \right] \end{aligned} \quad (۲۳-۶)$$

معادله حرکت رابط دوم به صورت زیر به دست می آید:

$$\tau_2 = - \left[\left(\frac{m_2}{2} + m_3 \right) a_2 a_3 C_2 + \left(\frac{m_2}{3} + m_3 \right) a_3^2 \right] \ddot{\theta}_1 + b_2 (\dot{\theta}_1) + \left(\frac{m_2}{2} + m_3 \right) a_3^2 \ddot{\theta}_2 - a_2 a_3 S_2 \left(\frac{m_2}{2} + m_3 \right) \dot{\theta}_1^2 \quad (24-6)$$

و نهایتاً معادله حرکت رابط سوم به صورت زیر حاصل می گردد:

$$\tau_3 = m_3 \ddot{d}_3 - g m_3 + b_3 (\dot{d}_3) \quad (26-6)$$

در ربات اسکارا حرکت رابط سوم کاملاً مستقل از در رابط اول می باشد حال آنکه رابط سوم برای دو رابط اول حکم بار را دارد و کاملاً در حرکت آنها تاثیر گذار بوده و باعث ایجاد کوپلینگ در سیستم ربات می شود همانطور که می دانیم پدیده کوپلینگ باعث می شود تئوری های مربوط به کنترل مفصل مستقل به خوبی جوابگوی نیازهای کنترلی سیستم نخواهد بود .

در بخشهای بعدی مشاهده می کنید که چون در سیستم کنترل ربات از مدل دینامیکی ربات هیچ استفاده ای نکرده ایم ، لذا وارد بحث های تئوری به دست آوردن مدل دینامیکی و معادلات دقیق حرکت بازوی اسکارا نشده و تنها با ذکر مرجع معادلات متداول کلیه بازوهای استاندارد اسکارا را ارائه کرده ایم [۱۰] .

فصل هفتم طراحی مسیر برای ربات

۷-۱- مقدمه

حرکت رباتها به دو صورت نقطه به نقطه و رد گیری کنترل می شود در وضعیت نقطه به نقطه ربات موظف به تغییر مکان از نقطه اولیه به نقطه بعدی است در این حالت سرعت طی مسیر شتاب حرکت و نیز زمان پیمودن مسیر مشخص نشده و هیچ کنترلی روی مسیر حرکت و چگونگی پیمایش آن نیست.

از این حالت میتوان برای پیمودن فاصله بین دونقطه که هیچ مانعی بین آنها نیست استفاده کرد [۱۱] ، اما چنانچه بخواهیم مثلاً برای جلوگیری از برخورد با مانعی مجری نهایی را به یک مسیر مشخص مقید کنیم وارد بحثهای طراحی مسیر می شویم .

۷-۲- چرا طراحی مسیر در فضای مفصلی ؟

چون این پروژه با هدف حرکت نقطه به نقطه در فضای دکارتی یا ردگیری مسیر در فضای مفصلی پیشبرده شده است ، با طراحی مسیر در فضای مفصلی سروکار خواهیم داشت البته در حالت کلی شکل مسیرها در فضای مفصلی ساده تر از فضای دکارتی می باشد بدین ترتیب که یک مسیر ساده در فضای مفصلی در فضای

دکارتی توصیفی پیچیده خواهد داشت ، در مسیرهای دکارتی مشکلات گوناگونی در ارتباط با فضای کاری و نقاط تکین ایجاد می شود ، این مشکلات به صورت زیر دسته بندی می شوند :

- نقاط بینابینی غیر قابل دسترس
 - سرعت زیاد مفصل در نزدیکی نقاط تکین
 - دسترسی پذیری نقاط شروع و هدف در جوابهای مختلف
- لذا معمولا در صنعت از هر دو مسیر هم دکارتی و هم مفصلی استفاده می کنند به طوری که کنترلر به سرعت در می یابد که به دلیل محدودیتها و مشکلات بالا در مسیر دکارتی باید از مسیر مفصلی استفاده کند . معمولا توصیف مسیر در فضای مفصلی بسیار ساده است از طرفی چون هیچ تناظر یک به یک پیوسته ای میان فضای مفصلی و دکارتی برقرار نمی سازیم نقاط تکین مکانیزم ربات اشکالی در ردگیری مسیر ایجاد نمی کنند .

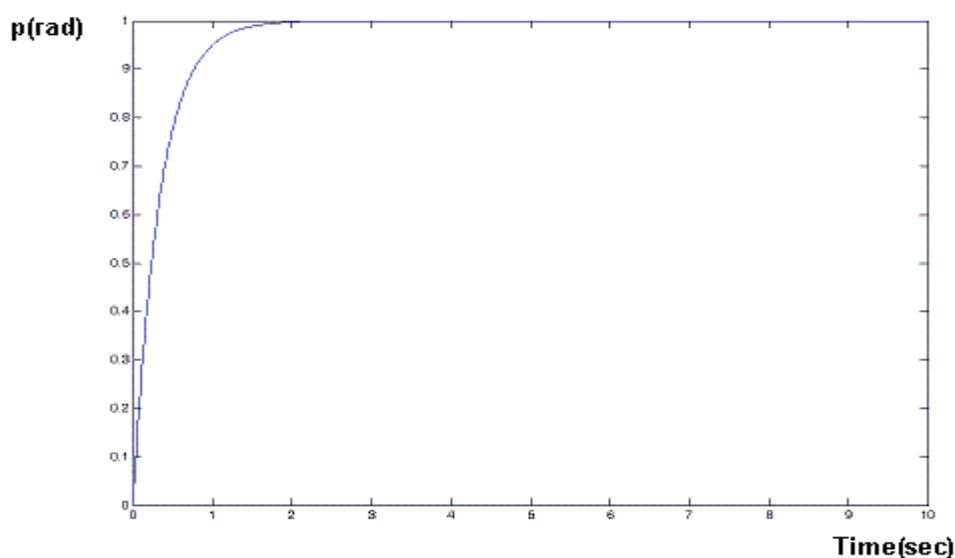
۷-۲-۱- کمینه سازی زمان حرکت

در این حالت با توجه به اینکه مایلیم زمان پیمایش کمینه شود ناچاریم که برای طی مسیر با حداکثر شیب سرعت را افزایش داده و به محض رسیدن به هدف با حداکثر شیب منفی سرعت را به صفر برسانیم از آنجا که معمولا حرکت رباتها باید هموار باشد ، حرکتهای خشن و ناگهانی مانند این نوع حرکتها ، باعث ساییدگی مکانیزم مکانیکی شده وبا ایجاد تشدید در آن ارتعاشات زیان آور پدید می آورد که این موضوع دقت و تکرارپذیری حرکت ربات را مختل می کند .

۷-۲-۲- ملاحظات عملی مسیرهای بنگ-بنگ

هنگامی که با بیشترین شتاب شروع به حرکت می کنیم و با بیشترین شتاب منفی توقف می کنیم یک مسیر بنگ-بنگ را به عنوان مسیر مورد نظر انتخاب کرده ایم چنین مسیرهایی به علت تنشهای فوق العاده مخربی که به سیستم مکانیکی ربات وارد می کنند باعث ایجاد لقی های نامطلوب و یا تشدید لرزشهای موجود در قسمتهای مکانیکی ربات می شوند که بسیار نا مطلوب هستند ما در طرح کنترلی خود از چنین مسیرهایی دوری می کنیم زیرا دقت و تکرار پذیری ربات را کاهش می دهند .

با توجه به اینکه کنترلر های PD در بهترین شرایط پاسخهای بحرانی می دهند این پاسخها هم تقریباً در زمره همین مسیرها بنگ-بنگ قرار می گیرند برای روشن شدن این موضوع شکل (۷-۱) را مشاهده کنید .

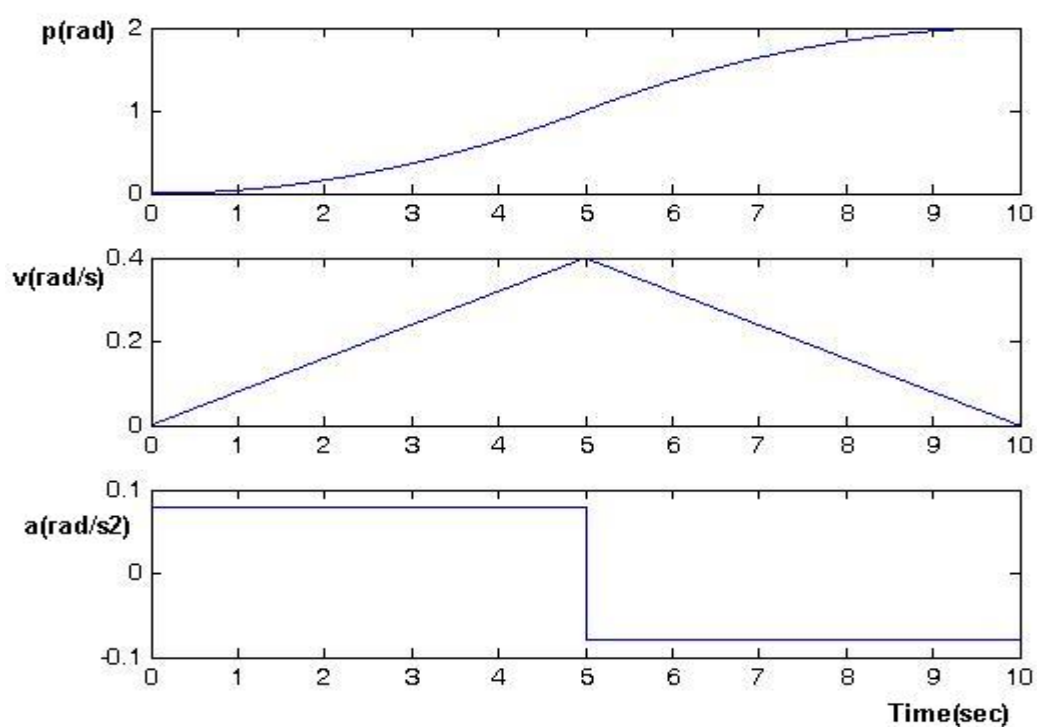


شکل ۷-۱- نمودار تغییر موقعیت مفصل تحت کنترل PD

۷-۳- طراحی مسیر های هموار

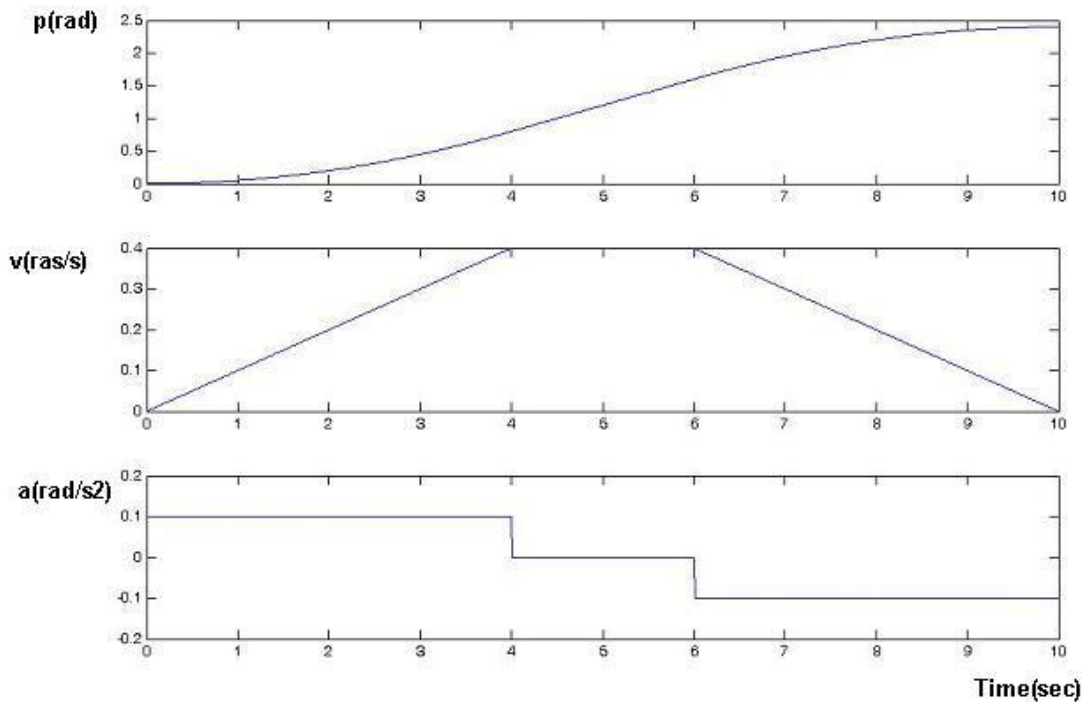
در سیستمهای رباتیکی بهترین نوع مسیر ، مسیرهای هموار هستند در اصطلاح مسیری را هموار می نامیم که تابع معرف آن حداقل دو بار مشتق پذیر باشد هر چه این مشتق پذیری بیشتر باشد مسیر هموارتر و در عین حال از نظر محاسباتی سنگین تر می شود ، مسیرهای درجه دوم که در آن هر رابط ربات با یک شتاب

ثابت $+a$ شروع به حرکت تند شونده کرده و سپس با رسیدن به وسط مسیر شتاب به $-a$ کاهش یافته و در نتیجه سرعت با شیبی منفی در حین نزدیک شدن به هدف به صفر می‌گراید مسیرهای عملی و در عمل قابل اجرا هستند شکل عمومی این مسیرها به صورت شکل (۷-۲) است. توجه به این نکته سودمند است که مسیرهای $LSPB^{۴۵}$ یا همان قسمتهای خطی با مسیر سهموی نیز زیر مجموعه‌ای از این مسیرها هستند شکل (۷-۳).



شکل ۷-۲- نمودار مسیر هموار مطلوب

^{۴۵} Linear Segment with Parabolic Blends



شکل ۷-۳- نمودار مسیر LSPB

این پروژه که با هدف Pick and Place و کنترل نقطه به نقطه در فضای دکارتی یا ردگیری مسیر در فضای مفصلی طراحی و ساخته شده است کنترلر PD نقش اصلی کنترل سیستم را ایفا می کند و ترم گشتاور محاسباتی و کنترل پیشخور تنها برای مقابله با اثرات غیر خطی ناشی از حرکت رابطها و نیز اغتشاشهای ورودی به سیستم اضافه شده اند در مجموع سیگنال فرمان یک سیگنال سریع و با جهش ناگهانی بود که بر روی اثرات مخرب آن بحث شد در این مرحله تصمیم گرفته شد تا یک Look up Table برای هموار سازی و مقید کردن سرعت و شتاب قبل از اعمال سیگنال فرمان تولید شده توسط PD کنترلر قرار گیرد این جدول بهره به صورت دقیق با در نظر گرفتن خواص غیر خطی محرکها مثل ناحیه مرده طراحی شده است که در ادامه به بررسی آن می پردازیم و سیگنال فرمان مناسبی از جهت همواری مسیر به ربات می دهد .

از آنجاکه منحنی گشتاور اعمالی به هر رابط ربات ارتباط مستقیم با توان اعمال شده به محرک هر رابط دارد و نیز با توجه به روابط موتورهای مغناطیس دائم این توان نیز با سرعت تولید شده توسط موتور متناسب می باشد لذا کافیسیت به شباهت و تناسب نمودار ولتاژ و سرعت موتورهای پی ببریم در این حالت هر نوع قیدی روی ولتاژ اثر مستقیم روی سرعت و شتاب حرکت موتور خواهد گذاشت .

۷-۳-۱- ناحیه مرده^{۴۶}

وجود ناحیه مرده از خواص ذاتی موتورهای می باشد بدین معنا که با اعمال هر ولتاژی از صفر تا ولتاژ معینی نباید انتظار حرکت داشته باشیم این موضوع در سطوح پایین ولتاژ یعنی کمتر از ۱۵٪ مقدار نامی مشاهده می گردد در چنین حالتی باید این ناحیه را با آزمایش در حالت بی باری و بار کامل اندازه گیری کرده و اثرات آنرا در طراحی کنترلر لحاظ کرد .

۷-۴- طراحی جدول بهره برای مقید کردن مسیر در جابجایی نقطه به نقطه

ما به راحتی در فضای مفصلی کار کنترل ربات را پیش برده ایم لذا با توجه به اینکه می توانیم تنها با دادن مقادیر هدف در فضای دکارتی برای انتقال مجری نهایی و جهت مجری نهایی ، سینماتیک وارون را حل کرده و مقادیر مورد نظر برای مفاصل به دست آورده به ربات اعمال می کنیم از آنجا که در این پروژه از استراتژی مفصل مستقل استفاده شده تمام ملاحظات و روابط مربوط به هموار کردن مسیر را به صورت پارامتری به دست آورده ایم که برای موتورهای مختلف با تعریف مقادیر جدید پارامترهای لازم قابل حصول می باشد .

^{۴۶} Dead Zone

برای هموار^{۴۷} کردن مسیر از تکنیک زیر استفاده می کنیم [۱۲]:

متغیرهای زیر را در نظر می گیریم:

$p_{\max}$ حداکثر مقدار سیگنال PWM

a_{start} سیگنال PWM آستانه حرکت

a_{stop} سیگنال PWM آستانه توقف

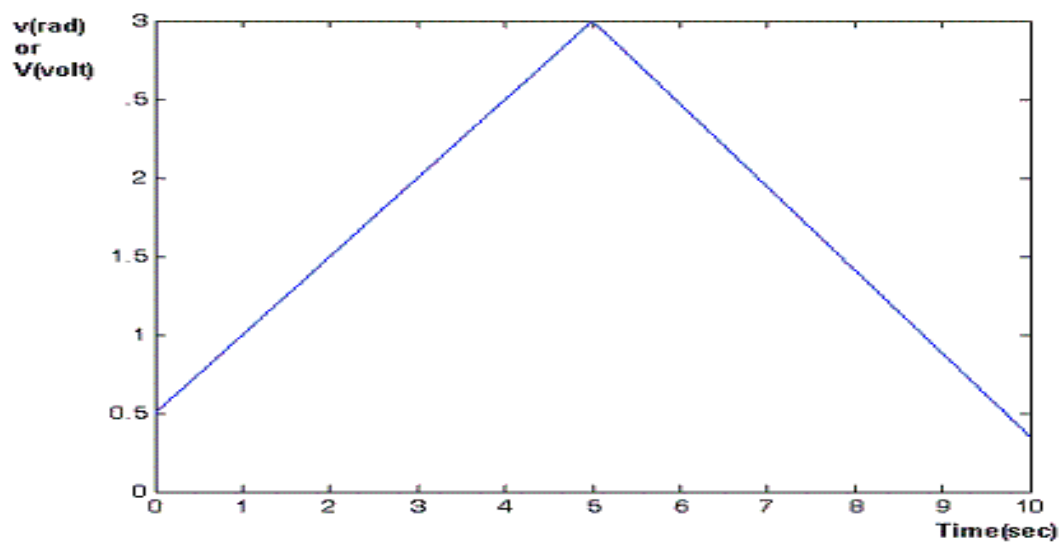
D : زاویه مطلوب مفصل

b_{start} : ضریب تعیین شروع

b_{stop} : ضریب تعیین توقف

C متغیر کمکی

نمودار سرعت شکل (۷-۴) را در نظر بگیرید:



^{۴۷} Smooth

شکل ۷-۴- نمودار مسیر سرعت بایاس شده

در ابتدا به علت وجود ناحیه مرده برای شروع حرکت سیگنال PWM آستانه حرکت به سیستم اعمال می گردد ، در اوج که به نیمه مسیر می رسد یک رابطه خطی بایاس شده با آستانه حرکت به صورت خواهیم داشت در اوج به سمت پایان نیز یک رابطه بایاس شده با متغیر کمکی c داریم و در نقطه پایانی هم یک رابطه بایاس شده با آستانه توقف خواهیم داشت :

در نظر داریم که مقدار پالس PWM در میکرو کنترلرهای MEGA AVR با توجه به عددی که در رجیستر کانال PWM مربوطه قرار می گیرد تغییر می کند این عدد بین ۰ تا ۲۵۵ می باشد که مقدار ۲۵۵ حداکثر ولتاژ تغذیه اعمال شده به راه انداز را با احتساب ولتاژی که روی آن افت می کند به موتور می دهد و مقدار صفر ولتاژ صفر را به موتور می دهد .

در شروع حرکت :

$$\begin{cases} a_{start} \\ \frac{D}{2} b_{start} + a_{start} = P_{max} \end{cases} \quad (۷-۱)$$

در مسیر شروع تا اوج :

در مسیر اوج تا پایان :

$$\begin{cases} Db_{stop} + c = a_{stop} \\ \frac{D}{2} b_{stop} + c = P_{max} \end{cases} \quad (۷-۲)$$

در لحظه توقف :

با حل ۲ دستگاه (۱-۷) و (۲-۷) خواهیم داشت :

$$b_{start} = \frac{2(P_{max} - a_{start})}{D} \quad (۳-۷)$$

$$\begin{cases} Db_{stop} + c = a_{stop} \\ -2 \times (\frac{D}{2} b_{stop} + c = P_{max}) \end{cases} \Rightarrow \begin{cases} c = 2P_{max} - a_{stop} \\ b_{stop} = \frac{2(P_{max} - a_{stop})}{D} \end{cases} \quad (۴-۷)$$

با استفاده از روابط (۱-۷) و (۲-۷) و نتایج به دست آمده از (۳-۷) و (۴-۷) قید سرعت برای محدوده های مختلف موقعیت محاسبه و با منطق زیر معلوم می شود [۱۲] :

موقعیت به اوج نرسیده است :

$$PWM = \theta b_{start} + a_{start} \quad (۵-۷)$$

موقعیت به اوج نرسیده است :

$$PWM = \theta b_{stop} + a_{stop} \quad (۶-۷)$$

با اعمال این محدودیت در هر لحظه به سیگنال تولید شده توسط کنترلر PD قید سرعت مورد نظر را اعمال کرده ایم ، لازم است که در هر بار که ربات عملیات جدیدی را می خواهد انجام دهد به جدولی که مقدار حداکثر سیگنال PWM را تغییر می دهد رفته و به صورت غیر همزمان این مقدار را انتخاب کند. این جدول برای همه رابطها شبیه به هم هستند فقط مقادیر آنها با هم متفاوتند برای رابط یک این جدول در شکل (۷-۵) نمایش داده شده است .

$D < 5^0$	$p_{max} = a_{start} + 2$
$D < 10^0$	$p_{max} = a_{start} + 4$
$D < 15^0$	$p_{max} = a_{start} + 6$
$D < 20^0$	$p_{max} = a_{start} + 8$
$D < 25^0$	$p_{max} = a_{start} + 10$
$D < 30^0$	$p_{max} = a_{start} + 12$
⋮	
⋮	
$D < 60^0$	$p_{max} = a_{start} + 26$
⋮	
⋮	
$D < 90^0$	$p_{max} = a_{start} + 52$
⋮	
$D < 135^0$	$p_{max} = a_{start} + 82$

شکل ۷-۵- جدول انتخاب مقادیر PWM

فصل هشتم کنترل ربات

۸-۱- مقدمه

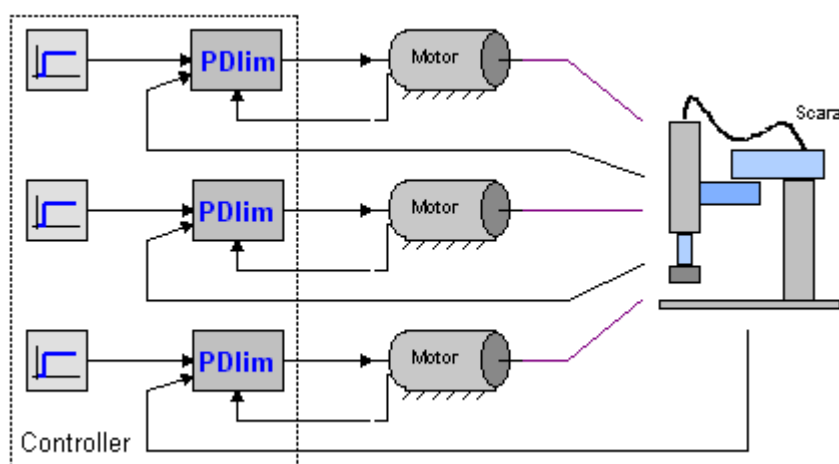
پس از بیان مسائل عملی مکانیک و الکترونیک و کامپیوتر ربات به بررسی روشهای کنترلی پیشخور به خصوص گشتاور محاسباتی و نیز گونه تطبیقی گشتاور محاسباتی می پردازیم . در این بخش ابتدا روش کنترل پیشخور را برای مقابله با اغتشاشات بیان می کنیم بعد از بررسی ضعف این روش در غلبه بر این

اغتشاشات ، روش کنترلی گشتاور محاسباتی را به کنترل پیشخور اضافه می کنیم در این حالت با توجه به تئوری این روش خواهیم دید که اغتشاش به طور قابل ملاحظه ای تضعیف می شود .

اما معلوم نبودن دینامیک و پارامترهای اغتشاش و همچنین پیچیدگی محاسبه باعث شد که بررسی هایی برای استفاده از تخمین پارامتر و کنترل تطبیقی برای تخمین یا تقریب اغتشاش انجام گیرد . در مطالب بالا منظور از اغتشاش ، گشتاور غیر خطی ناشی از حرکت بازوی ماهر است که سیگنال ورودی موتور را دستخوش تغییر می کند [۱۳] ، [۱۴] .

۸-۲- کنترل نقطه به نقطه

در حال حاضر در کاربردهای صنعتی کماکان کنترلرهای PD [۱۵] کنترل رباتها را بعهدہ دارند و برای کنترل موقعیت تمام روباتها به جز قسمتهایی که وارد میدان کنترل نیرو می شوند از این کنترلرها استفاده می شود . بلوک دیاگرام کنترل ربات با ساختار کلی RRP که اسکارا ، را هم شامل می شود در شکل (۸-۱) آمده است .



شکل ۸-۱- بلوک دیاگرام کنترل ربات اسکارا با ۳ درجه آزادی

هنگامی که به دنبال کنترل نقطه به نقطه هستیم و حرکت ربات خیلی سریع نیست یعنی از سیستم انتقال چرخ دنده ای با نسبت تبدیلهای کاهنده نسبتاً بزرگ استفاده کرده ایم کنترل کننده های PD می توانند نیازهای کنترلی را در این مرحله برآورده کنند. تزویج غیر خطی ناشی از نیروهای اینرسی

، کوریولیس ، جانب مرکز و گرانشی به صورت یک ترم مستقل اغتشاش با ضریب کاهنده r که همان نسبت تبدیل سیستم انتقال نیروی چرخ دنده ای است روی ورودی موتور قرار می گیرد و کنترلر را دچار مشکل می کند .

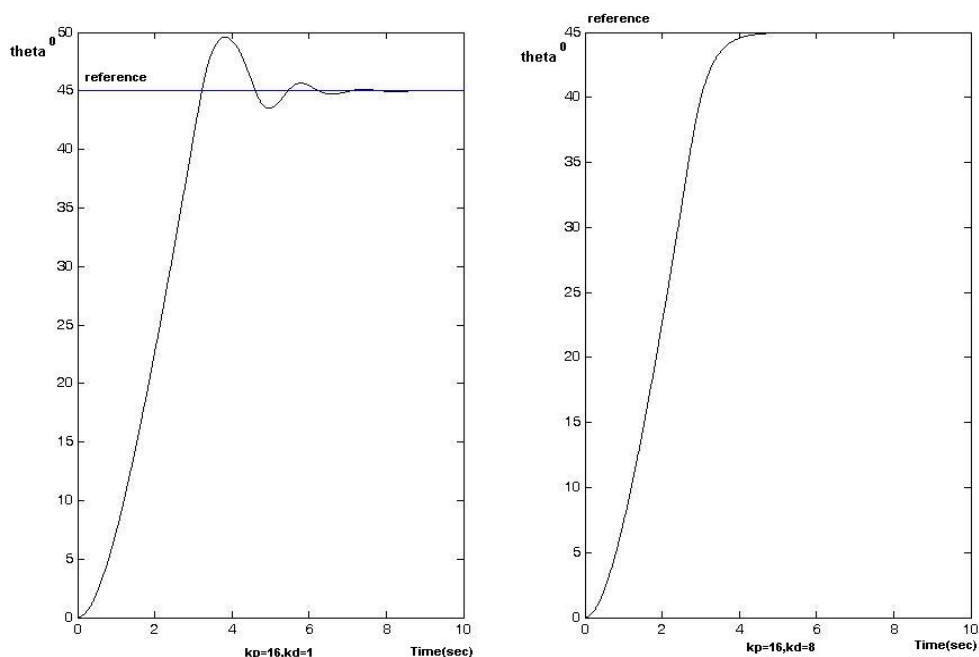
با توجه به خصوصیات عمومی ، کنترلر های PD چنانچه سیستم حلقه بسته پایدار باشد با انتخاب K_p بزرگ انتخاب شود خطای ماندگار به اغتشاشات ثابت کاهش می یابد اما از نقطه نظرهای عملی K_p بزرگ موجب اشباع محرک می شود .

معمولاً در سیستمهای رباتیکی عملی برای جلوگیری از آسیب های وارده به محرکها مانند اضافه جریان از محدودکننده های توان (جریان) قبل از محرکها استفاده می گردد لذا بزرگ انتخاب کردن ضرایب کنترل PD همیشه جوابگو نخواهد بود .

در سیستمهای رباتیکی مایلیم پاسخ کنترلر PD یک پاسخ میرایی بحرانی باشد یعنی $\zeta = 1$. این پاسخ سریعترین حرکت به سمت هدف و بدون فراجش^{۴۸} می باشد . از طرف دیگر وجود اغتشاشات ناشی از حرکت ، کنترل PD را با مشکل روبرو خواهند کرد . در قسمتهای بعدی این بخش خواهیم دید که برای محدود کردن شتاب و سرعت نامناسب در حرکت از محدودکننده های دینامیکی (محدود کننده هایی که سطوح برش آنها بسته به شرایط تغییر می کند) استفاده می کنیم و برای کاهش اثرات تزویجهای غیر خطی رابطها در حین حرکت از کنترلر گشتاور محاسباتی در سیستم کنترل استفاده خواهیم کرد [۱۶] ، [۱۷] .

^{۴۸} Over Shoot

در کنترلر های PD تغییر ضرایب معمولاً به روی سرعت پاسخ دهی ، فراجاهش و زمان نشست تاثیر میگذارند . نتایج شبیه سازی برای یکی از محرکها بدون حضور اغتشاش در شکلهای (۳-۸) و (۴-۸) مشاهده می شود :



شکل ۳-۸- کنترلر PD با ضرایب $K_p=16, k_d=8$ شکل ۴-۸- کنترلر PD با ضرایب $K_p=100, k_d=0$

در شکل (۳-۸) و (۴-۸) ملاحظه می گردد که K_p همیشه سرعت و فراجاهش را زیاد می کند و K_d معمولاً این حرکت را نرمتر کرده و زمان نشست را کمی بیشتر و فراجاهش و سرعت را کم می کند ضرایب کنترلر ها را به راحتی می توانیم برای داشتن پاسخ بحرانی به صورت $K_v = 2\sqrt{K_p}$ انتخاب کنیم [۱۸] .

۸-۳- روش کنترل پیشخور

فرض کنید می خواهیم یک موتور DC را کنترل مسیر کنیم و مسیر مطلوب نیز به صورت $q_d(t)$ باشد. تابع تبدیل یک موتور DC مغناطیس دائم به صورت زیر به دست می آید :

$$R_a I_a + L_a \dot{I}_a = V - V_b \quad (۱-۸)$$

$$V_b = K_b \dot{q}_m, \tau = K_t I_a \quad (۲-۸)$$

$$\tau = \frac{K_t}{R} V - \frac{K_t K_b}{R} \dot{q}_m \quad (۳-۸)$$

$$\tau = J_m \ddot{q}_m + B_m \dot{q}_m \quad (۴-۸)$$

$$J_m \ddot{q}_m + B_m \dot{q}_m + \frac{K_t K_b}{R} \dot{q}_m = \frac{K_t}{R} V \quad (۵-۸)$$

$$(۵-۸)$$

با صرفنظر از L_a در (۱-۸) داریم :

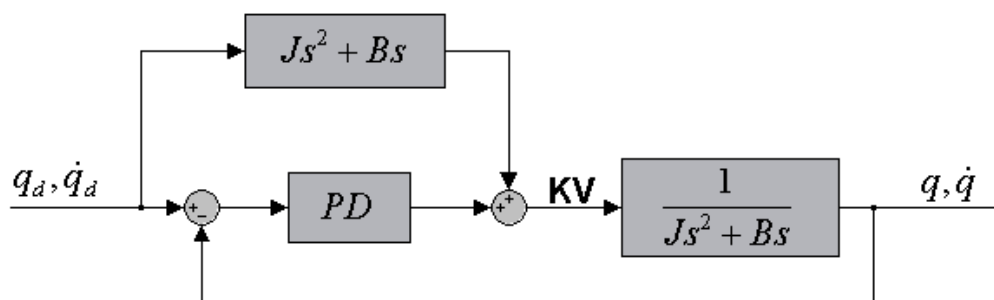
$$KV = J_m \ddot{q}_m + (B_m + \frac{K_t K_b}{R}) \dot{q}_m \quad (۶-۸)$$

$$KV = J \ddot{q}_m + B \dot{q}_m, J = J_m, B = B_m + \frac{K_t K_b}{R}$$

$$q_m(s) = \frac{K}{Js^2 + Bs} V(s) \quad (۷-۸)$$

$$(۸-۸)$$

بلوک دیاگرام کنترل پیشخور در شکل (۵-۸) آمده است :



شکل ۵-۸- دیاگرام بلوکی کنترل پیشخور

$$KV = J \ddot{q}_d + B \dot{q}_d - K_v \dot{e} - K_p e \quad (۹-۸)$$

$$KV = J \ddot{q}_m + B \dot{q}_m \quad (۱۰-۸)$$

$$J \ddot{q}_m + B \dot{q}_m = J \ddot{q}_d + B \dot{q}_d - K_v \dot{e} - K_p e$$

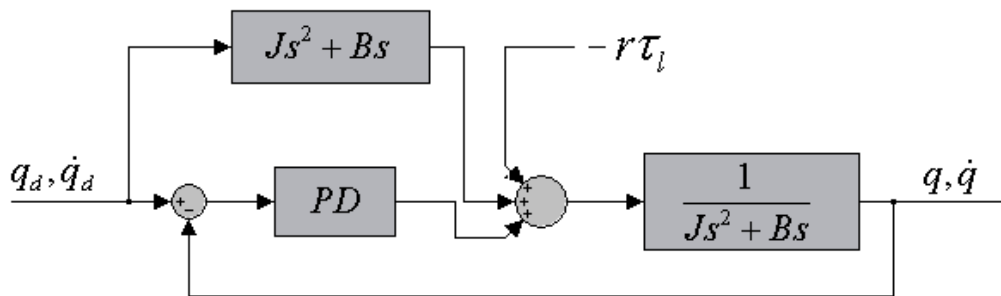
$$J \ddot{e} + (B + K_v) \dot{e} + K_p e = 0$$

که در روابط بالا $e = q_d - q$ یا همان خطای ردگیری می باشد با قرار دادن (۸-۱۰) در (۸-۹) خواهیم داشت :

$$(۸-۱۱)$$

چنانچه سیستم دارای اغتشاش نباشد با فرض پایداری سیستم حلقه بسته خطای ردگیری بطور مجانبی برای هر مسیر دلخواه $q_d(t)$ صفر می شود .

حال اگر اغتشاش $-r\tau_l$ را به صورت شکل (۸-۶) در نظر بگیریم که τ_l گشتاور بار موتور و r ضریب کاهش چرخ دنده می باشد . در این صورت به نا توانی روش کنترل پیشخور برای دفع اغتشاش پی خواهیم برد .



شکل ۸-۶- دیاگرام بلوکی کنترل پیشخور با وجود اغتشاش

$$KV - r\tau_l = J\ddot{q}_m + B\dot{q}_m \quad (۸-۱۲)$$

$$KV = J\ddot{q}_d + B\dot{q}_d - K_v\dot{e} - K_p e \quad (۸-۱۳)$$

با قرار دادن (۸-۱۲) در (۸-۱۳) داریم :

$$\begin{aligned} -r\tau_l &= J\ddot{e} + B\dot{e} + K_v\dot{e} + K_p e \\ J\ddot{e} + (B + K_v)\dot{e} + K_p e &= -r\tau_l \end{aligned} \quad (۸-۱۴)$$

$$\tau_l = 0$$

در این حالت با فرض پایداری سیستم حلقه بسته ، خطای ردگیری بطور مجانبی برای هر مسیر دلخواه $q_d(t)$ به صفر می گراید اگر و فقط اگر $\tau_l = 0$ باشد [۱۹] .

۸-۴- روش کنترلی گشتاور محاسباتی

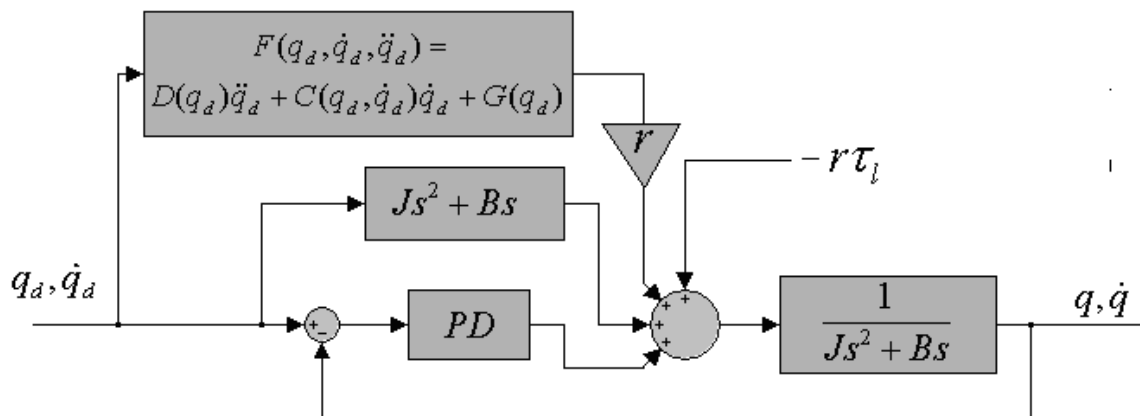
حال برای رفع این مشکل به بررسی دینامیک ربات در حالت واقعی می پردازیم در حقیقت گشتاور با τ_l همان دینامیک رابطهاست که به صورت ترم گشتاور اغتشاش وارد سیستم موتور می شود در حالت کلی این گشتاور به صورت زیر می باشد :

$$(۸-۱۵)$$

$$\tau_l = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$$

که ترم آخر مربوط به نیروهای گرانشی در دینامیک ربات اسکارا وجود ندارد این گشتاور در حقیقت همان لختی غیر خطی اتصالات و نیروهای کوریولیس و جانب مرکز و جاذبه ناشی از حرکت بازوهای ماهر می باشند که برای ما چندان معلوم نیست . یعنی از لحاظ ساختاری معلوم ولی از لحاظ پارامتری به اندازه کمی معلوم یا نامعلوم است یعنی در بیشتر مواقع بعضی از پارامترهای آن به طور تقریبی معلوم هستند [۲۰] ، [۲۱] ، [۲۲] .

بلوک دیاگرام روش گشتاور محاسباتی در شکل (۸-۷) آمده است [۲۳] :



شکل ۸-۷- دیاگرام بلوکی روش کنترلی گشتاور محاسباتی

چون τ_l از جنس گشتاور است روش کنترلی پیشخور بالا را گشتاور محاسباتی می نامیم در واقع ترم غیر خطی $\tau_l - r$ بوسیله گشتاور محاسبه شده پیشخور $F(q_d, \dot{q}_d, \ddot{q}_d)$ جبران می گردد .

از آنجا که وجود اغتشاش غیر قابل اجتناب در سیستمهای کنترلی است و روش گشتاور محاسباتی یکی از روشهای قدرتمند مقابله با اثرات مخرب اغتشاشات به سیستمها ست ، در این پروژه از این روش کنترلی که از زمینه های تحقیق فعال هم می باشد استفاده کنیم [۲۴] .

با توجه به شکل (۷-۸) چون روش گشتاور محاسباتی به صورت پیشخور وارد سیستم می شود آنرا در زمره روشهای کنترل پیشخور قرار داده اند . در این روش مقادیر موقعیت ، سرعت و شتاب مسیر مطلوب را در معادله دینامیکی قرار داده آنرا به سیستم ربات پیشخور می کنیم ، که این کار یعنی قرار دادن مسیر مربوط در معادلات دینامیکی ربات که از روش نیوتون- اوایلر یا اوایلر- لاکرانژ به دست آمده است نیازمند محاسبات پیچیده همزمان و غیر همزمان و حل معادلات سنگین ریاضی است [۲۵] ، [۲۶] در این مورد در مراجع [۲۷] و [۲] اطلاعات کلی وجود دارد که کافی نیست . در ظاهر روش گشتاور محاسباتی روشی ساده به نظر می رسد اما وقتی وارد جزئیات طرحهای کنترلی مبتنی بر گشتاور محاسباتی می -شویم مشکلات و پیچیدگی های اجرای آنها را درک می کنیم ، چون رابطه $F(q_d, \dot{q}_d, \ddot{q}_d)$ یک رابطه غیر خطی بسیار پیچیده است مشکلات محاسباتی زیادی برای محاسبه همزمان این رابطه بخاطر محدودیتهای ناشی از کمی حافظه و سرعت پردازشگر وجود دارد . لذا باید هر چه بیشتر معادلات دینامیکی بازوی ماهر را ساده کرد که همین موضوع میدانهای بزرگی از تحقیقات را در این زمینه باز کرده است [۲۸] و [۲۹] . این مشکلات به صورت زیر دسته بندی می شوند که اجرای این طرحها را هم بسیار مشکل کرده اند [۳۰] :

۱. در تمام تئوری های کنترل مبتنی بر مدل دینامیکی این مشکل وجود دارد که پیاده سازی آنها در عمل منوط به داشتن مدل دینامیکی دقیقی از ربات می باشد که آن هم در گرو شناسایی و محاسبه

دقیق پارامترهای مکانیکی بدنه و مدل سازی دقیق ربات می باشد اما می دانیم محاسبه دقیق این پارامترها و نیز ارائه مدل دقیق از ربات بسیار مشکل می باشد .

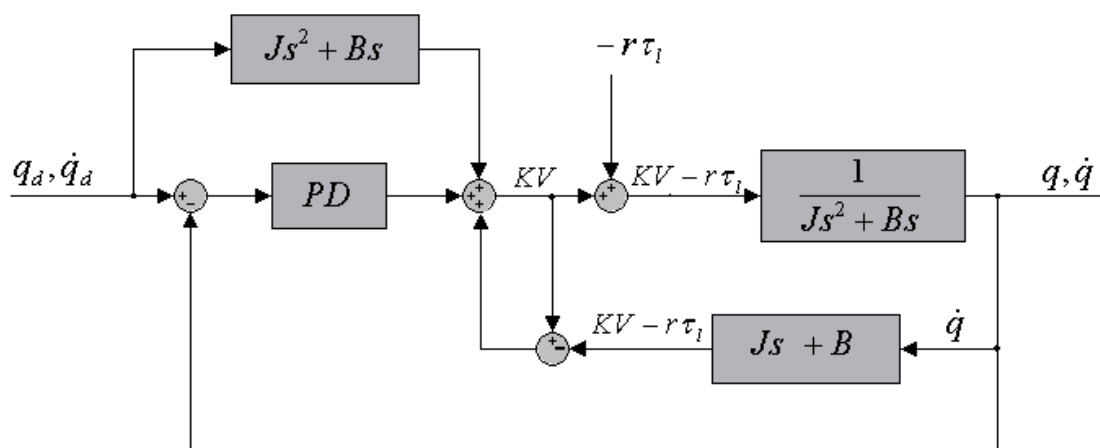
۲. با جایگذاری اطلاعات مربوط به مسیرهای مطلوب را در مدل دینامیکی ربات ، ترم گشتاور محاسباتی به صورت غیر همزمان^{۴۹} تولید می شود و به سیستم پیشخور می شود شکل (۸-۲) با توجه به اینکه مدل دینامیکی از ترمهای غیر خطی پیچیده تشکیل شده است محاسبه این ترمها بر اساس اطلاعات مسیر مطلوب برای میکرو کنترلرها بسیار سنگین است زیرا سرعت ریز پردازنده محدود می باشد .

شایان ذکر است که گونه تطبیقی طرحهای کنترلی گشتاور محاسباتی بسیار جالب و قدرتمند بوده و سیستم را در مقابل عدم قطعیت های پارامتری مقاوم می کند [۳۱] و [۳۲] .

۸-۵- گشتاور محاسباتی همراه با ترم پیشخور

یک روش جالب برای پوشش دادن این طرح روشی است که در نمودار بلوکی شکل (۸-۸) آمده است در این روش اگر توسط خروجی و عکس تابع تبدیل فرایند بتوان سیگنال فرمان مشتمل بر اغتشاش ورودی یا همان $KV - r\tau_l$ را آشکار سازی کرد می توان با محاسبه تفاضل آن و سیگنال فرمان KV ، گشتاور محاسبه شده τ_{CT} را تولید کرده و با سیگنال فرمان جمع کرده و سیگنال منتهی را به سیستم وارد کرد [۲۰] .

^{۴۹} Off-Line



شکل ۸-۸- دیاگرام بلوکی گشتاور محاسباتی همراه با ترم پیشخور

در این تکنیک با توجه به نتایج شبیه سازی چنانچه ضرایب B, J دارای خطای ۲۰٪ باشند خطای ردگیری در باند ۰,۰۰۱ واحد قرار می گیرد. همانطور که در شکل (۸-۸) مشاهده می شود سیستم فیدبک شده که در حقیقت τ_{CT} را فراهم می کند دقیقاً عملی شبیه به ترم $Js^2 + Bs$ انجام می دهد با این تفاوت که خود سیگنالهای $\dot{q}$ استفاده می کند در این تکنیک می توانیم با دقت خوبی اثرات سیگنال τ_l را توسط τ_{CT} خنثی کنیم بدین ترتیب که ورودی سیستم همان $KV + \varepsilon$ بماند که $\varepsilon = \tau_{CT} - \tau_l$ می باشد که این خطا هم در هر مرحله توسط کنترلر PD کاهش می یابد، در حالت کلی طرح ارائه شده پاسخ خوبی در حضور اغتشاشات ورودی می دهد تنها مشکل این طرح لزوم وجود $\ddot{q}$ یا مشتق $\dot{q}$ می باشد که از لحاظ اندازه گیری مشکل است اما می توان این مشکل را توسط فیلتر یا مشتق گیر تفاضلی حل کرد چنانچه خطای شناسایی ضرایب B, J تا مرز ۸۰٪ باشند خطاهای ردگیری در باند ۰,۰۰۱-۰,۰۰۱ خواهد بود.

مزیت این طرح نسبت به طرحهای بعدی که یک سیستم تطبیقی در جهت کاهش خطای ردگیری τ_l را خنثی می کند یا یک مکانیزم تخمین پارامتر که پارامترهای بردار P را تخمین زده و سپس τ_{CT} را که تقریبی از τ_l است را می سازد اینست که نه تنها به اندازه آنها پیچیده نیست بلکه محاسبات همزمان پیچیده ای هم ندارد.

۸-۶- طرح عمومی کنترل تطبیقی بازوی ماهر

خاصیت خطی بودن پارامترها نیز اولین بار توسط کریگ [۸] با پیشنهاد یک طرح کنترل تطبیقی که همگرایی کلی خطاهای ردگیری را نتیجه می داد استفاده شد. این طرح بر اساس طرح کنترل تطبیقی استاندارد برای سیستمهای خطی بیان شد که از ایده گشتاور محاسباتی بوسیله یک پیش تخمین از پارامترهای دینامیکی سیستم نیز استفاده می کرد و نهایتاً یک سیستم دینامیکی خطای ردگیری خطی را نتیجه می داد. از نظریه پایداری لیپانوف نیز برای به دست آوردن قانون تطبیق پارامتر به صورت تجدید بخش مربوط به گشتاور محاسباتی استفاده می شود.

کنترلر از دو بخش تشکیل شده است بخش اول کنترلگر خطی PD که با استفاده از تخمینهای پارامترهای سیستم با فرض کراندار ماندن سیگنالهای ورودی برای پایدار سازی ربات استفاده می شود و بخش دوم کنترلگر تطبیقی می باشد که برای کاهش خطای ردگیری در سیستم دینامیکی خطای منته به کار گرفته می شود این روشها معمولاً متکی به ۴ خاصیت مهم دینامیکی بازوهای ماهر می- باشند :

خاصیت (۱):

ماتریس لختی $D(q)$ مثبت معین متقارن بوده و $D(q)$ و $D^{-1}(q)$ به ازای تمام $q \in R^n$ به صورت یکنواخت محدودند.

خاصیت (۲):

برای هر درجه آزادی یک ورودی کنترل مجزا و مستقل وجود دارد.

خاصیت (۳):

تمام پارامترهای ثابت سیستم از جمله جرم بازوها - طول بازوها- اینرسی لحظه به صورت ضرایبی در توابع معلومی از متغیرهای سیستم و مشتقات اول و دوم آنها در نظر گرفته می شوند. با تعریف هر ضریب به صورت یک پارامتر مجزا یک رابطه خطی نتیجه می شود که آنرا می توان به صورت زیر نوشت:

$$D(q)\ddot{q} + c(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})P = \tau_l \quad (۱۶-۸)$$

آنجا که $Y(q, \dot{q}, \ddot{q})$ یک ماتریس $n \times r$ از توابع شناخته شده است که بازگرداننده^{۵۰} نامیده می شود و P یک بردار r بعدی از پارامترهاست [۳۲].

خاصیت (۴):

ماتریس N را به صورت روبرو تعریف می کنیم :

$$N(q, \dot{q}) = \dot{D}(q) + 2c(q, \dot{q}) \quad (۱۷-۸)$$

آنگاه $N(q, \dot{q})$ پاد متقارن و $-n_{kj} = n_{jk}$.

قانون کنترل تطبیقی به صورت رابطه (۱۸-۸) می باشد [۱۹] :

$$\tau_l = \hat{D}(q)(\ddot{q}_d - k_v \dot{e} - k_p e) + \hat{c}(q, \dot{q})\dot{q} + \hat{g}(q) \quad (۱۸-۸)$$

فرض می کنیم $\hat{D}, \hat{c}, \hat{g}$ از لحاظ فرم دقیقا همانند مقادیر حقیقی خود با پارامترهای $\hat{p}_1, \hat{p}_2, \dots, \hat{p}_r$

هستند بنابراین :

$$\hat{D}\ddot{q} + \hat{C}\dot{q} + \hat{G} = Y(q, \dot{q}, \ddot{q})\hat{P} \quad (۱۹-۸)$$

^{۵۰} Regressor

که $\hat{P} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_r]$ بردار پارامترهای تخمینی است .

با جایگذاری (۸-۱۸) در معادله دینامیکی حقیقی بازوی ماهر داریم :

$$\hat{D}\ddot{q} + c\dot{q} + g = \hat{D}(\ddot{q}_d - k_v\dot{e} - k_p e) + \hat{C}\dot{q} + \hat{G} \quad (۸-۲۰)$$

با جمع و تفریق کردن جمله $\hat{D}\ddot{q}$ در سمت چپ معادله (۸-۲۱) و استفاده از (۸-۱۹) داریم :

$$\begin{aligned} \hat{D}(\ddot{e} + k_v\dot{e} + k_p e) &= \tilde{D}\ddot{q} + \tilde{C}\dot{q} + \tilde{G} \\ &= Y(q, \dot{q}, \ddot{q})\tilde{P} \end{aligned} \quad (۸-۲۱)$$

که در آن $(\tilde{0}) = (0) - (\hat{0})$ لذا معادله دینامیکی خطا به صورت زیر خواهد شد :

$$\ddot{e} + k_v\dot{e} + k_p e = \hat{D}^{-1}Y\tilde{P} = \Phi\tilde{P} \quad (۸-۲۲)$$

می توان سیستم (۸-۲۲) را در فضای حالت بازنویسی کرد :

$$\dot{X} = AX + B\Phi\tilde{P} \quad (۸-۲۳)$$

که A ماتریس هرویتز و نیز (پایدار) می باشد :

$$A = \begin{bmatrix} 0 & I \\ -K_0 & -K_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad X = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (۸-۲۴)$$

در روش کریگ فرض شده که $\ddot{q}$ شتاب رابطهای ربات قابل اندازه گیری است و قانون تجدید طوری تصحیح شده که از کراندار بودن $\hat{D}^{-1}$ مطمئن با شیم با توجه به این دو فرض و قضیه لیاپانوف می توان برای به دست آوردن قانون به روز آوری به صورت زیر عمل کرد :

قانون تجدید را به صورت روبرو انتخاب می کنیم :

$$\ddot{\tilde{P}} = -\Gamma^{-1} \Phi^T B^T p X \quad (۲۵-۸)$$

از آنجا که $\Gamma = \Gamma^T > 0$ ، ماتریس مثبت معین متقارن یکتاست که با حل معادله لیاپانوف زیر:

$$A^T p + p A = -Q \quad (۲۶-۸)$$

برای ماتریس Q مثبت معین متقارن داده شده است به دست می آید .

تحت چنین شرایطی با محاسبه X از رابطه (۳۲-۸) معلوم می شود که به ازای $X \rightarrow 0, t \rightarrow \infty$ در حالی که کلیه سیگنالها کراندار باقی می مانند .

اثبات : تابع منتخب لیاپانوف را به صورت زیر در نظر بگیرید :

$$V = X^T p X + \tilde{P}^T \Gamma \tilde{P} \quad (۲۷-۸)$$

مشتق زمانی آن در امتداد مسیرهای (۲۳-۸) به صورت زیر است :

$$\dot{V} = -X^T Q X + 2\tilde{P}^T [\Phi^T B^T p X + \Gamma \tilde{P}] \quad (۲۸-۸)$$

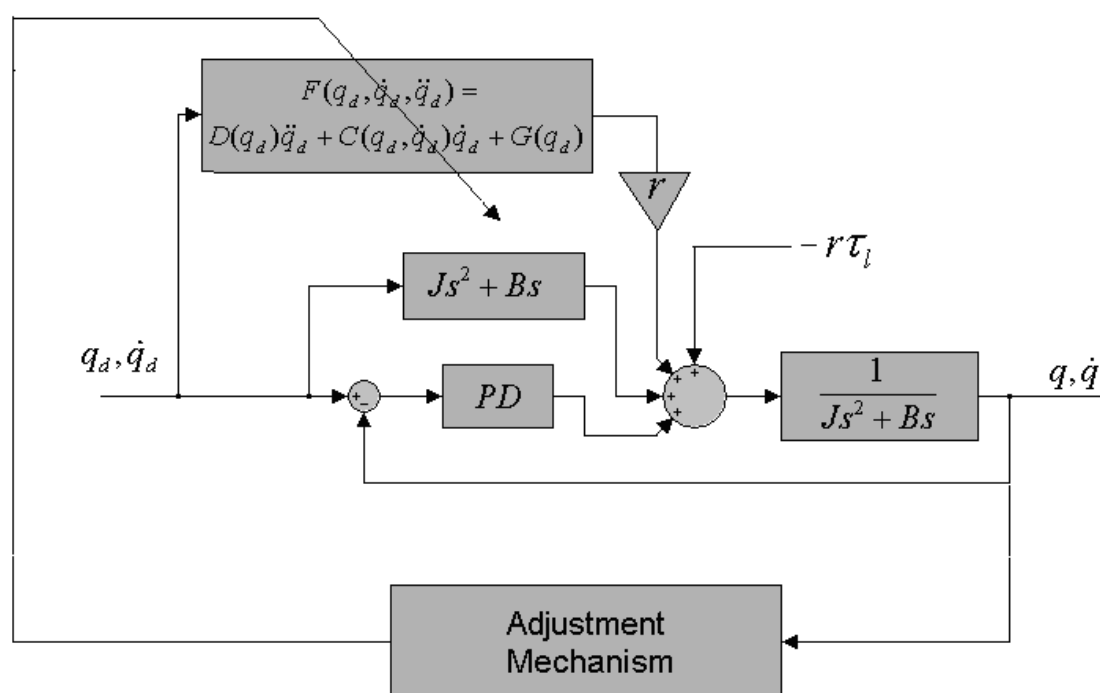
با استفاده از قانون تجدید پارامتر (۲۵-۸) رابطه بالا به صورت روبرو کاهش می یابد :

$$\dot{V} = -X^T Q X \leq 0 \quad (۲۹-۸)$$

لذا امتداد مسیرهای (۸-۲۳) منفی نیمه معین می گردد لذا بر اساس قضیه کرانداري و مجموعه همگرایی و بر اساس معیار پایداري لیپانوف سیستم پایدار و خطا به صفر همگرا می شود اگر $\ddot{V}$ کراندار باشد.

$$\begin{aligned}\ddot{V} &= -\dot{X}^T Q \dot{X} - X^T Q \dot{X} \\ &= -(X^T A^T + \tilde{P}^T \Phi^T B^T) Q \dot{X} - X^T Q (A \dot{X} + B \Phi \tilde{P}) \\ &= -X^T A^T Q \dot{X} - X^T Q A \dot{X} - 2\tilde{P}^T \Phi^T B^T Q \dot{X}\end{aligned}\quad (۸-۳۰)$$

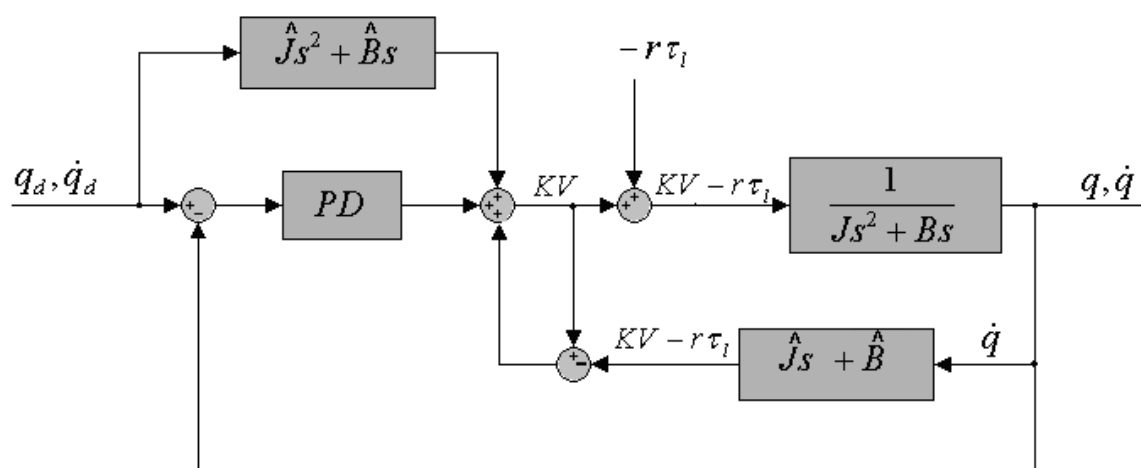
چون $X, \Phi, \tilde{P}$ کراندار هستند لذا $\ddot{V}$ کراندار و حالت سیستم یعنی بردار $X = [e \quad \dot{e}]^T$ که $e = q - q_d$ خطای ردیابی موقعیت می باشد به صفر همگرا می شود. چنانچه بتوانیم با هدف همگرایی خطای ردگیری به صفر در طرح مبتنی بر گشتاور محاسباتی توسط کنترل تطبیقی الگوریتمی به صورت دیاگرام بلوکی نشان داده شده در شکل (۸-۹) برای تضعیف اثرات اغتشاش بروی عمل ردگیری به دست آوریم، یک روش کاملاً مقاوم برای مقابله با اغتشاشات ورودی در سیستم های رباتیک به دست آورده ایم.



شکل ۸-۹- دیاگرام بلوکی طرح گشتاور محاسباتی تطبیقی

۸-۷- اجرای روش کنترلی پیشنهادی

روشی که در این پروژه بکار رفته یک روش بسیار موثر و ساده است که به خوبی می تواند اثرات اغتشاشات ناشی از حرکت رابطها را تضعیف کند ، شکل (۸-۱۰) را مشاهده کنید ، در این روش اگر بتوان سیگنال $KV - r\tau_l$ را آشکار سازی کرد با توجه به معلوم بودن KV ، گشتاور $r\tau_l$ را محاسبه کرده و به صورت پیشخور وارد سیستم می کنیم این ترم که آنر τ_{CT} می نامیم می تواند اثر $r\tau_l$ را در سیستم خنثی کند . [۱۰]



شکل ۸-۱۰- دیاگرام بلوکی روش کنترلی اجرا شده

در این تکنیک با توجه به نتایج شبیه سازی چنانچه ضرایب B, J دارای ۹۰٪ خطای شناسایی باشند خطای ردگیری در باند ۰,۰۰۱ واحد قرار می گیرد همانطور که در شکل (۸-۱۰) مشاهده می شود سیستم فیدبک شده که در حقیقت τ_{CT} را فراهم می کند دقیقاً عملی شبیه به ترم $Js^2 + Bs$ انجام می دهد با این تفاوت که خود سیگنالهای $\dot{q}$ استفاده می کند در این تکنیک می توانیم با دقت خوبی اثرات سیگنال τ_l را توسط τ_{CT} خنثی کنیم بدین ترتیب که ورودی سیستم همان $KV + \varepsilon$ بماند که $\varepsilon = \tau_{CT} - \tau_l$ می باشد که این خطا هم در هر مرحله توسط کنترلر PD کاهش می یابد ، در حالت کلی طرح ارائه شده پاسخ خوبی در حضور اغتشاشات ورودی می دهد تنها مشکل این طرح لزوم وجود $\ddot{q}$ یا مشتق $\dot{q}$ می باشد که از لحاظ اندازه گیری مشکل است اما به نظر می رسد بتوان این مشکل را توسط فیلتر یا مشتق گیر تفاضلی حل کرد . برای رفع این مشکل می توان از یک تقریب استفاده کرده و به جای سیگنال

پرنویز و پردردسرتاب از همان $KV + \varepsilon$ استفاده نمود این تقریب به صورت زیر است :

با توجه به شکل (۸-۱۰) متوجه می شویم که :

$$J^{-1}((KV + \varepsilon) - B\dot{\theta}) = \ddot{\theta} \quad (۸-۳۱)$$

در این حالت چنانچه سیستم درست عمل کند ترم $KV + \varepsilon$ به خوبی می تواند نقش $\ddot{\theta}$ را بازی کند

نتایج این دو حالت در بخش شبیه سازی آمده است .

با در نظر گرفتن چنانچه عدم قطعیت موجود در شناسایی J, B تا ۱٪ از مقادیر واقعی خود دور باشند خطاهای ردگیری در باند ۰,۰۰۱-۰,۰۱ خواهد بود .

مزیت این طرح نسبت به طرحهای بعدی که یک سیستم تطبیقی در جهت کاهش خطای ردگیری τ_I

را خنثی می کند یا یک مکانیزم تخمین پارامتر که پارامترهای بردار P را تخمین زده و سپس τ_{CT} را که تقریبی از τ_I است را می سازد اینستکه تنها به اندازه آنها پیچیده نیست بلکه محاسبات همزمان پیچیده ای هم ندارد [۳۴] .

نتایج شبیه سازی این طرح در بخش ۱۰ مشاهده می گردد در این روش بر خلاف اغلب روشهای گشتاور محاسباتی نیازی به دینامیک بازوی ماهر نیست تنها لازم است تقریبی از پارامترهای موتور را داشته باشیم که این پارامترها را در بخشهای گذشته شناسایی و محاسبه کرده ایم.

در این حالت می توان حدس زد که سیگنال ورودی به موتور هم مانند سیگنال خطا دچار لرزش^{۵۱} شده است که این حالت بسیار نامطلوب بوده و عملاً قابل اجرا نخواهد بود با یک تقریب می توان به جای سیگنال مشکل ساز شتاب از سیگنال $KV - r\tau_I$ استفاده کرد در این حالت همانطور که در شکلهای ارائه شده

^{۵۱} Chattreing

مشاهده می گردد لرزش از بین رفته و محرکها دچار لرزش نمی شوند و نیز خطا هم در محدوده قابل قبولی می باشد .

فصل نهم : نتایج شبیه سازی

۹-۱- مقدمه

نتایج شبیه سازی روشهای کنترلی به ترتیب زیر مورد بررسی قرار گرفته شده است .

- کنترل پیشخور
- گشتاور محاسباتی تطبیقی
- گشتاور محاسباتی همراه با ترم پیشخور

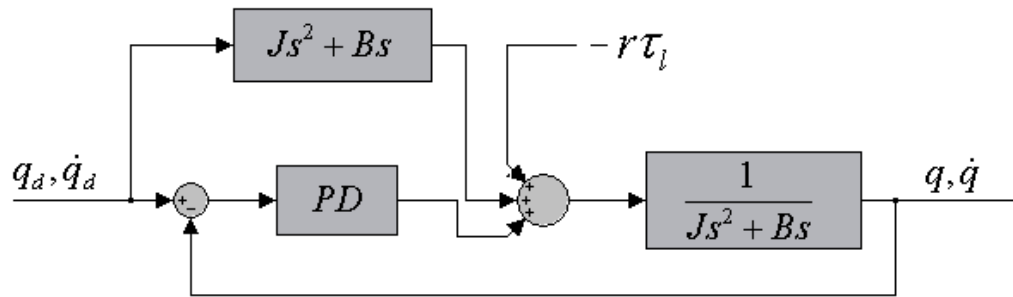
۹-۲- کنترل پیشخور

در روش کنترل پیشخور چنانچه سیستم دارای اغتشاش نباشد با فرض پایداری سیستم حلقه بسته خطای

ردگیری بطور مجانبی برای هر مسیر دلخواه $q_d(t)$ صفر می شود .

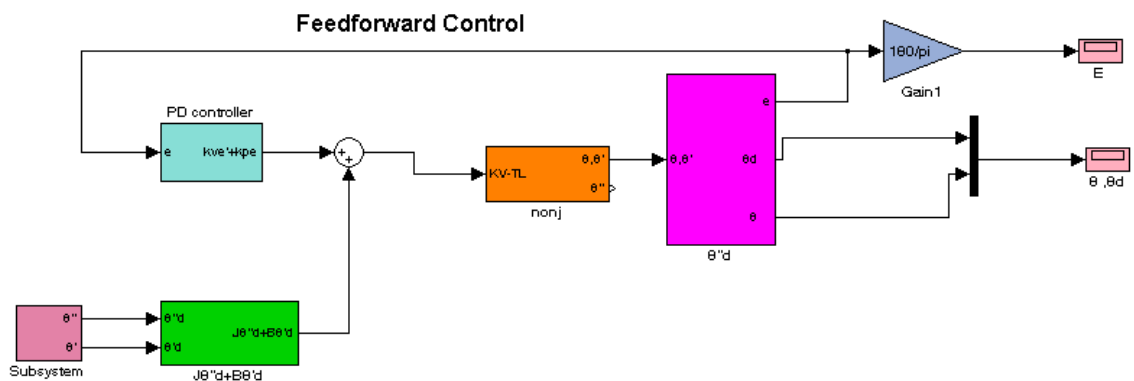
حال اگر اغتشاش پیشخور $r\tau_l$ را به صورت زیر در سیستم کنترل شکل (۹-۱) در نظر بگیریم به نا توانی

روش کنترل پیشخور برای دفع اغتشاش پیشخور پی خواهیم برد .

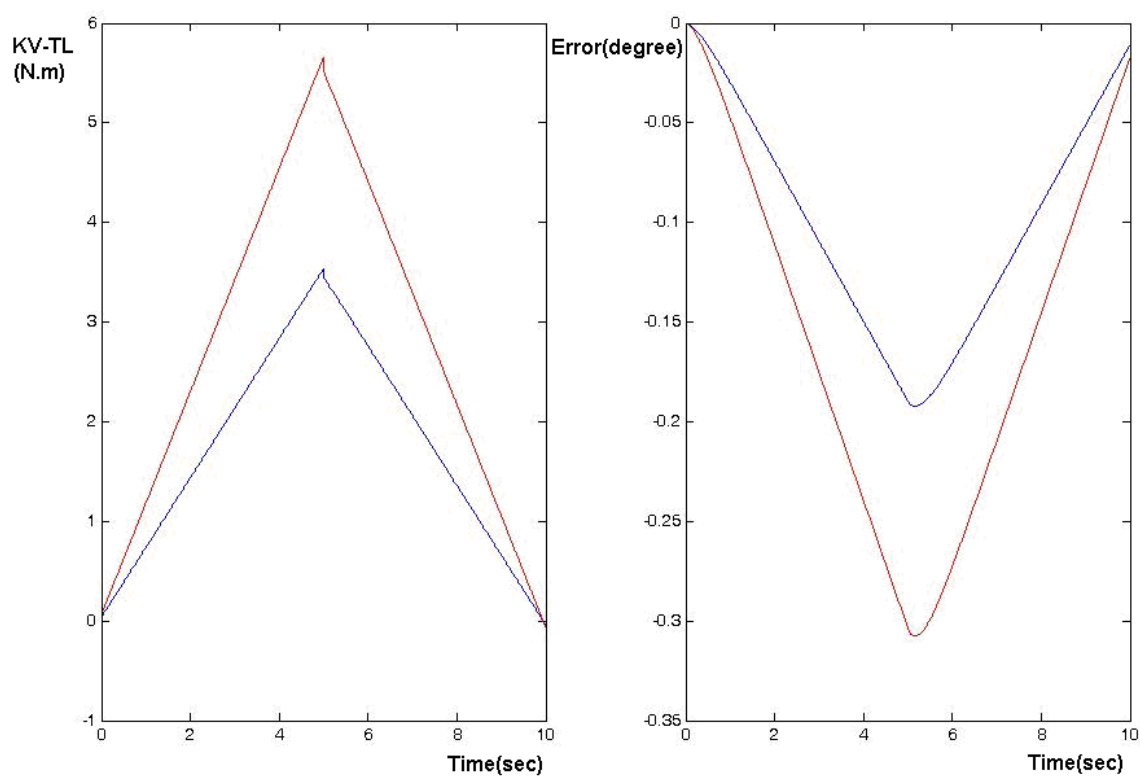


شکل ۹-۱- بلوک دیاگرام روش کنترل پیشخور همراه با ترم اغتشاش

در این روش چنانچه مقادیر شناسایی و محاسبه شده موتورها دارای ۹۰٪ خطا در شناسایی J, B باشند خطا در محدوده قابل قبولی قرار می گیرد برای زاویه های مطلوب زیر نتیجه نمودار خطا و سیگنال محرک در شکل (۹-۳) مشاهده می شود ، این شبیه سازی برای مسیرهای سهموی به صورت شکل (۹-۴) انجام شده که در شکل (۹-۲) مشاهده می شود.

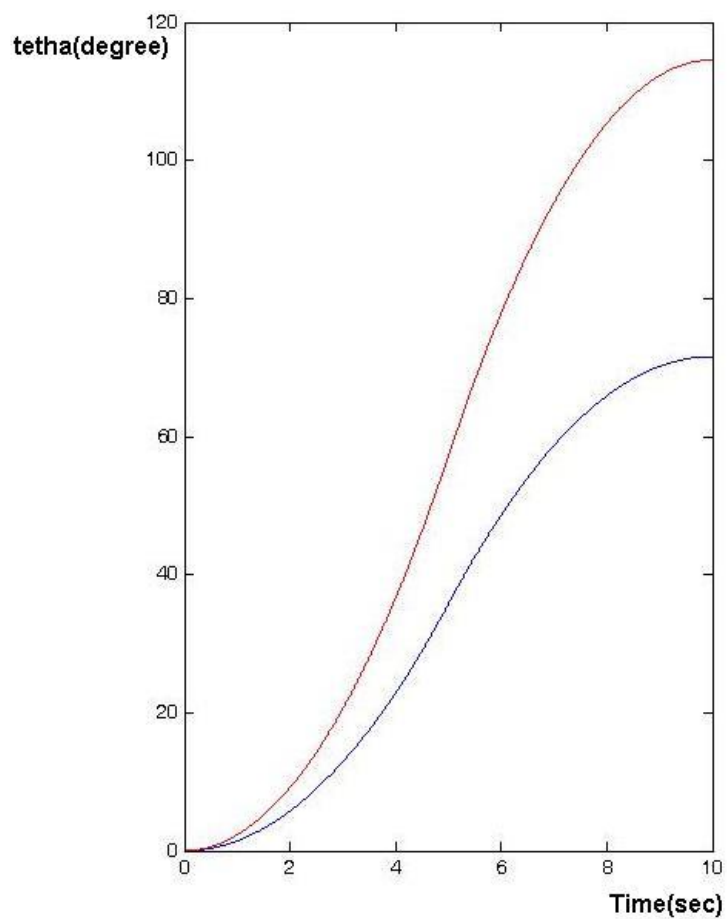


شکل ۹-۲- بلوک شبیه سازی شده کنترل پیشخور



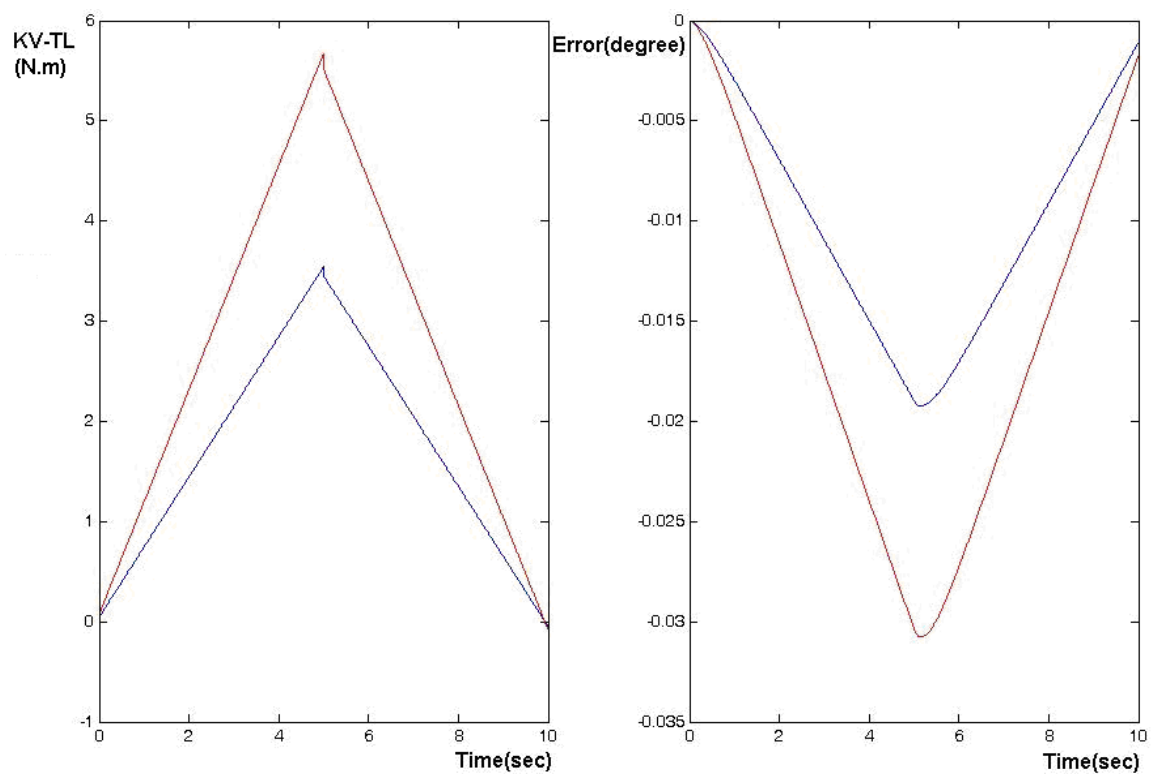
شکل ۹-۳- نمودار گشتاور ورودی و خطای تعقیب به ازای ۹۰٪ خطا در شناسایی J,B

چنانچه شبیه سازی را برای ۱۰٪ خطای تقریب در شناسایی J,B انجام دهیم نتایج خوبی به دست خواهد آمد . با توجه به اینکه سیستمهای مبتنی بر کنترل پیشخور با اضافه شدن اغتشاش به سیستم در مهار اغتشاش ناتوان می مانند این موضوع برای اغتشاش ناشی از حرکت رابطها در شکل (۹-۶) مشاهده می شود

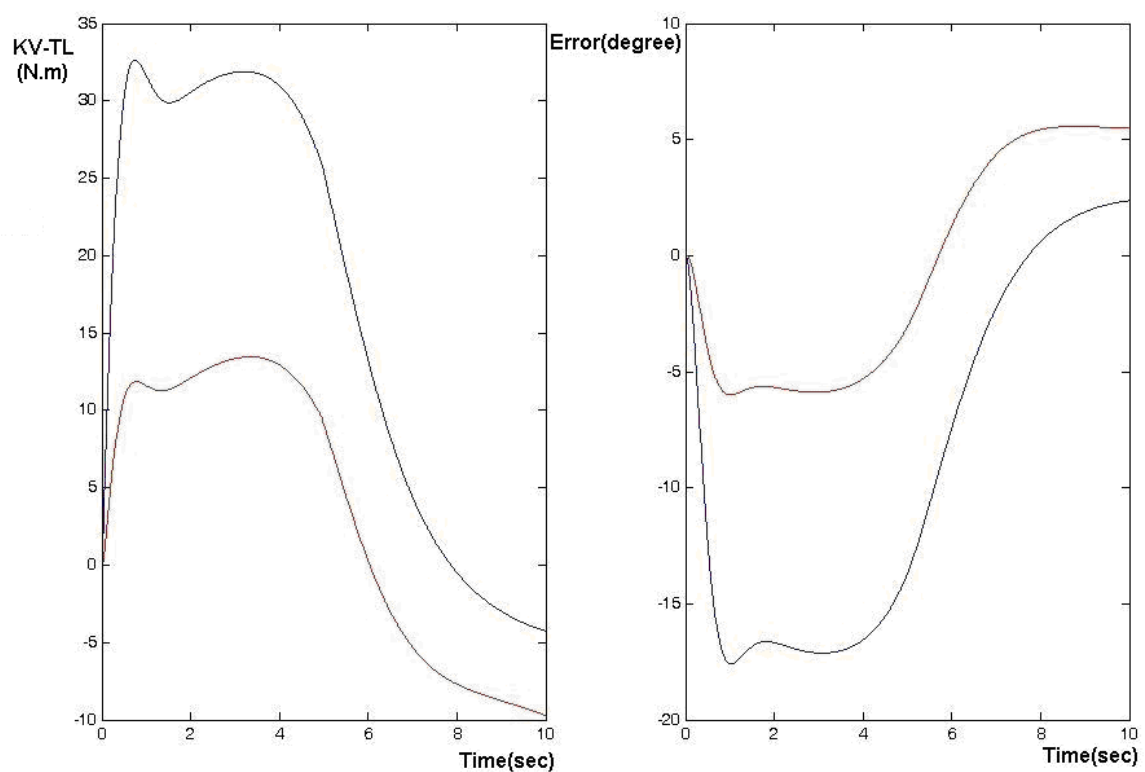


شکل ۹-۴- نمودار متغیرهای زاویه ای ۱ و ۲

نمودار گشتاور ورودی و خطای تعقیب به ازای ۱۰٪ خطا در شناسایی B, I در شکل (۹-۵) نشان داده شده که در مقایسه با شکل (۹-۳) تاثیر خطای تقریب را در پاسخ به وضوح دیده می شود .



شکل ۹-۵- نمودار گشتاور ورودی و خطای تعقیب به ازای ۱۰٪ خطا در شناسایی J,B

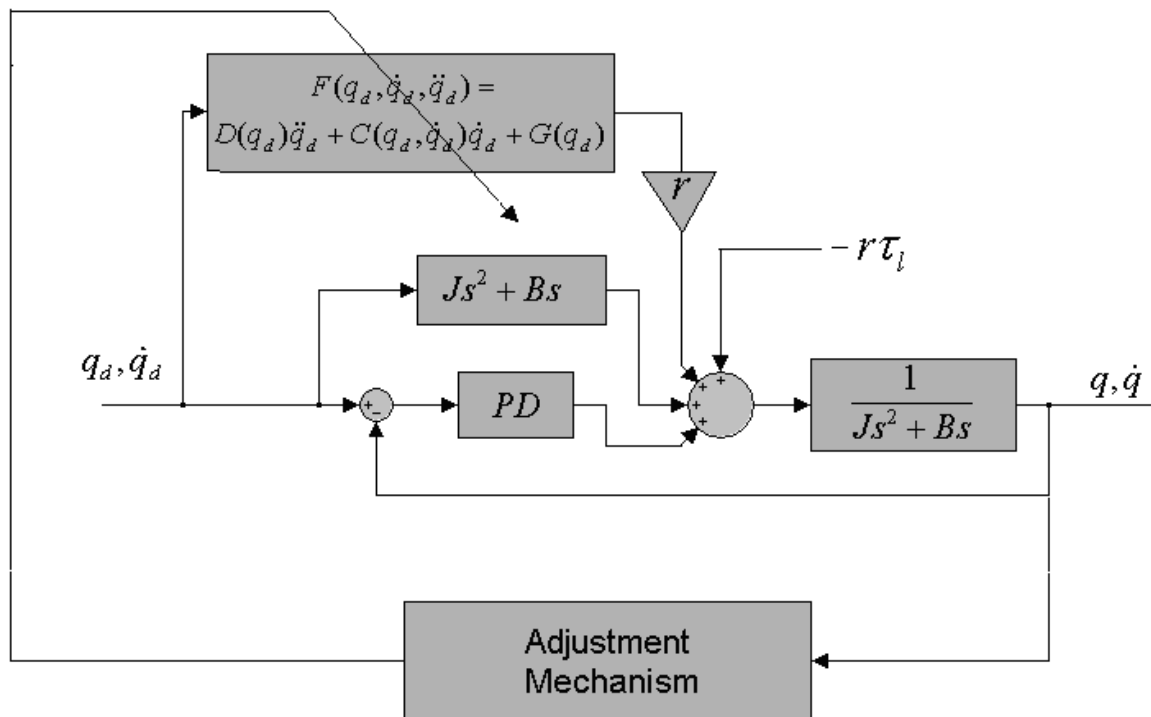


شکل ۹-۶- نمودار گشتاور ورودی و خطای تعقیب در حضور اغتشاش

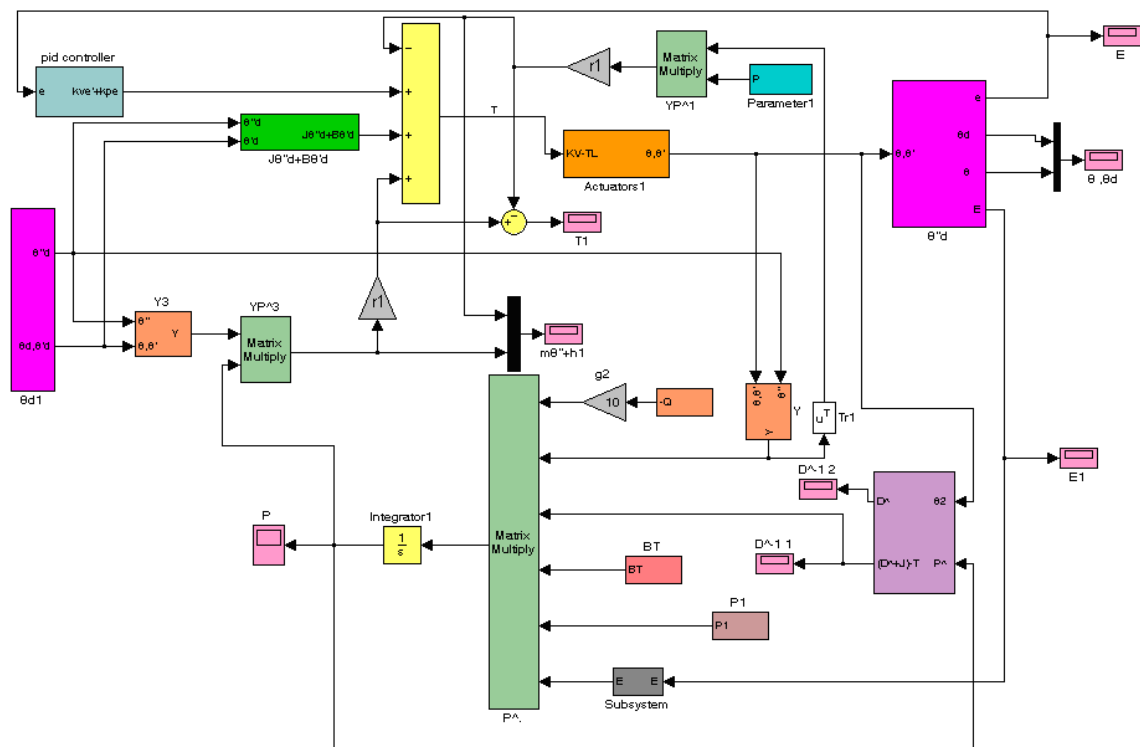
با توجه به ناتوانی این روش در حضور اغتشاش استفاده از روش گشتاور محاسباتی را پیشنهاد می کنیم.

۹-۳- گشتاور محاسباتی تطبیقی

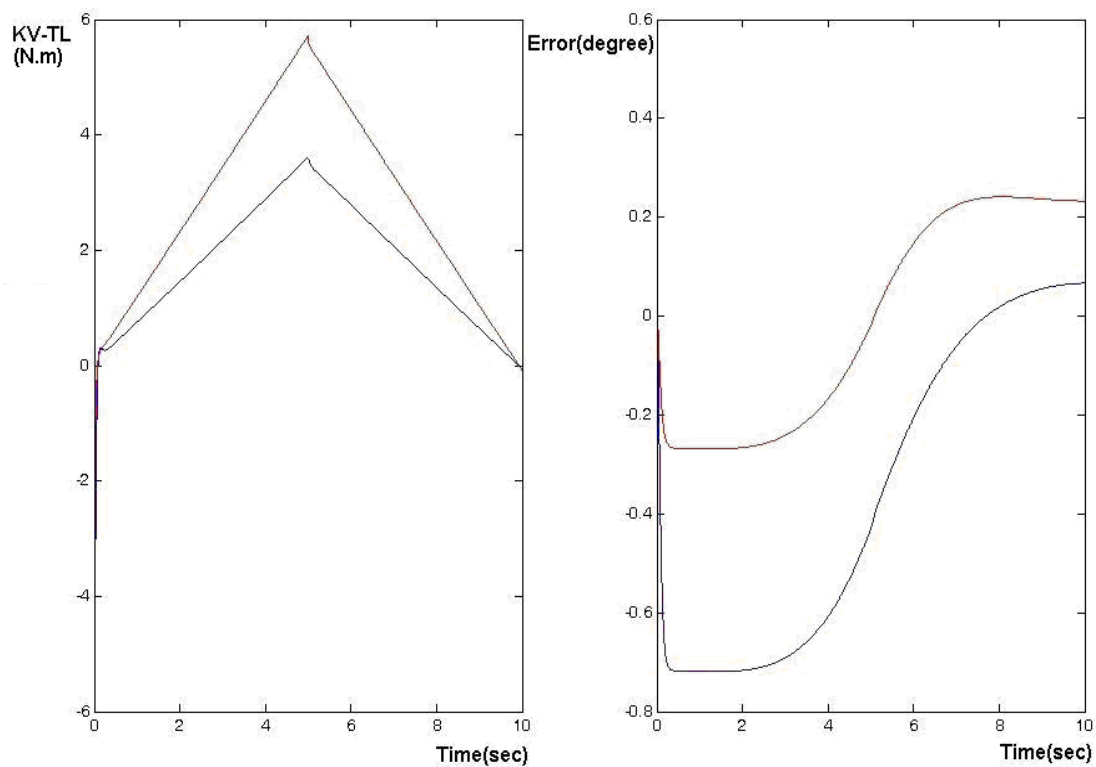
همانطور که در بخش ۷ به صورت کامل در مورد گونه تطبیقی گشتاور محاسباتی بحث شد توسط معیار پایداری لیاپانوف یک مکانیزم تجدید پارامتر به دست آوردیم که به صورت غیر مستقیم پارامترهای ترم گشتاور محاسباتی را به گونه ای که در شکل (۹-۷) مشاهده می شود به روز نموده است. چون این مکانیزم تجدید بر اساس کاهش خطای ردگیری شکل گرفته انتظار می رود که همراه با کاهش خطای ردگیری خطای پارامتری ناشی از عدم قطعیت‌های پارامتری در مدل دینامیکی محاسبه شده کاهش می یابد بلوک این سیستم در شکل (۹-۸) نشان داده شده است و خطای ردگیری به ازای عدم قطعیت پارامتری ۵۰٪ در مدل دینامیکی به صورت شکل (۹-۹) می باشد.



شکل ۹-۷- دیاگرام بلوکی طرح گشتاور محاسباتی تطبیقی



شکل ۹-۹- بلوک شبیه سازی شده گشتاور محاسباتی تطبیقی



شکل ۹-۹- نمودار گشتاور ورودی و خطای تعقیب (گشتاور محاسباتی تطبیقی)

همانطور که در شکل (۹-۹) مشاهده می گردد خطای تعقیب در حضور اغتشاش حداکثر ۰,۶ درجه یا

۰,۰۱ رادیان می باشد توسط این روش خطا را به اندازه قابل قبولی با حضور اغتشاش بهبود دادیم . یکی از مهمترین مسائلی که باید در طراحی سیستمهای کنترل ربات در نظر گرفته شود مسئله ولتاژ ورودی به موتورها است ، این ولتاژ با گشتاور ورودی به سیستم متناسب است ، در شرایط ایده ال کار کنترلر سیگنال ولتاژ یا گشتاور باید دارای خصوصیات زیر باشند :

- سطح سیگنال خارج از محدوده ولتاژ موتورها نباشد .

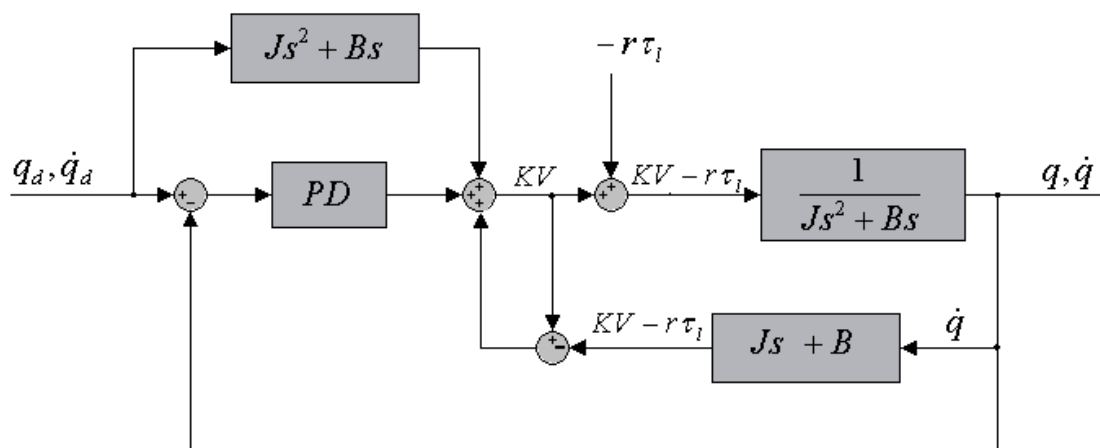
- سیگنال دارای لرزش با فرکانسهای بالا نباشند .

وقوع هر یک از دو حالت بالا یک نقص برای سیستم کنترل محسوب می شود بطوریکه ممکن است در شبیه سازی خطای ردگیری در محدوده مناسبی قرار گیرد اما این محدود ماندن خطا تنها در صورتی ارزش دارد که محرکها به اشباع نروند اگر این اتفاق رخ دهد باید از اشباع گرها^{۵۲} استفاده شود که خود باعث می شود که خطای سیستم بیشتر شود زیرا به ازای سیگنالهای بزرگ قبل از اشباعگر این کنترلر در ست کار میکند وقتی سیگنالها برش بخورند دیگر نباید انتظار جواب مطلوب قبل از قرار دادن اشباعگر را داشت ، از طرف دیگر وجود لرزش در سیگنالهای ورودی با فرکانسهایی نزدیک به مدهای مخفی فرکانس بالای می تواند باعث تحریک این مودها شود و چون مدل را بر اساس این مودهای مخفی به دست نیامده سیستم از کنترل خارج می شود.

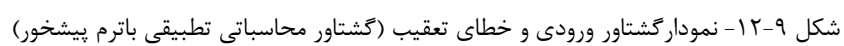
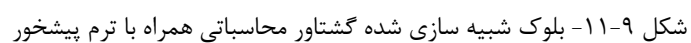
با توجه به سیگنال گشتاور که در شکل (۹-۹) دیده می شود این کنترلر بسیار عالی عمل کرده و با سیگنالهای معقولی عمل ردگیری را انجام می دهد .

۹-۴- گشتاور محاسباتی همراه با ترم پیشخور

در این روش گشتاور محاسباتی را به گونه ای که در شکل (۹-۱۰) مشاهده می شود محاسبه می کنیم و برای اجرای آن هیچ نیازی به مدل دینامیکی بازوی ماهر نیست و تنها لازم است پارامترهای J, B موتور هر رابط با دقت نه چندان زیادی شناسایی شوند بطوریکه در نتایج شبیه سازی شکل (۹-۱۲) مشاهده می شود در بدترین شرایط خطای شناسایی یعنی در نظر گرفتن ۹۰٪ خطای ردگیری در محدوده فوق العاده عالی قرار می گیرند بلوک شبیه سازی شده این سیستم کنترل در شکل (۹-۱۱) و برنامه اصلی اجرا شده این تئوری در انتهای این بخش آمده است .



شکل ۹-۱۰- دیاگرام بلوکی طرح گشتاور محاسباتی همراه با ترم پیشخور



در این روش همانطور که دیده می شود خطای تعقیب فوق العاده کوچک است و سیگنالهای گشتاور هم بدون نیاز به اشباع گر در محدوده قابل قبولی قرار دارند ، با توجه به اینکه پیاده سازی سیستمهای تطبیقی و الگوریتمهای تجدید پارامتر توسط میکرو کنترلرها کاری بسیار دشوار است روش زیر را که روش جدید و در عین حال قابل اجرا می باشد را به عنوان روش پیشنهادی به همراه روش گشتاور محاسباتی تطبیقی و کنترل پیشخور شبیه سازی کرده و نتایج شبیه سازی آنها را مقایسه کردیم با توجه به برتری این طرح از نقطه نظر خطای ردگیری ، محدود ماندن سیگنالهای ورودی به سیستم و نیز همانطور که در شکل (۹-۱۱) ملاحظه می شود این روش در عین سادگی نسبت به روشهای دیگر کاملاً عملی است بنابراین از این روش به عنوان روش کنترلی در این پروژه استفاده کردیم .

هدف از ساختن این ربات اجرای حرکات نقطه به نقطه در فضای دکارتی یا ردگیری در فضای مفصلی است لذا روش پیشنهادی را در عمل اجرا کرده و عملکرد ربات به صورت شهودی و تحلیلی نشان می دهد ترمهای پیشخور ، کنترلر PD و گشتاور محاسباتی کار خود را به خوبی انجام داده اند ، برنامه عملی نوشته شده برای این روش که سازگار با گونه های MEGA AVR می باشد در پیوست ۱ آمده است.

فصل دهم نتیجه گیری

۱۰-۱- نتیجه گیری

از آنجا که پیاده سازی عملی یک تئوری و گرفتن نتایج مطلوب عملی از آن بهترین راه استدلال بر صحت آن می باشد تصمیم گرفتیم ربات اسکارا را بسازیم و بررسی کنیم که آیا واقعا موتورها با فرمانهای ساده کنترلی واکنش درستی از خود نشان می دهند؟ آیا کنترل کننده PD واقعا می تواند پاسخ بحرانی برای کنترل موقعیت و سرعت یک موتور به ما بدهد؟ آیا برای ردگیری مسیرهای مفصلی در فضای مفصلی این کنترلر تمام نیازها ما را برآورده می کند؟ کنترلر پیشنهادی پیاده سازی شده در در عمل چگونه پاسخ می دهد؟ این خواسته ها در کنار هم منوط به بررسی و مطالعه تئوریهای مختلف در زمینه های سینماتیک و دینامیک ربات، الکترونیک و کنترل ربات و مدارهای واسط ربات و رایانه و نیز ارائه راه حلهای جدید برای رفع مشکلات احتمالی بود، در امتداد پیشبرد پروژه در بسیاری از مراحل مجبور به تعویض تئوری انتخاب شده یا روش مطالعه شده یا تعویض سخت افزار ساخته شده یا انتخابی شدیم اما تمام این مسائل و مشکلات باعث شد تا به تجربیات ارزشمندی دست پیدا کنیم.

۱. بررسی نتایج مربوط به شبیه سازیها نشان می دهد روش عمومی گشتاور محاسباتی که در بخش (۴-۸) بررسی شده روشی قدرتمند می باشد که توسط اطلاعات مربوط به مسیرهای مطلوب و مدل دینامیکی ربات ترم گشتاور محاسباتی را تولید و به سیستم پیشخور می کند. یکی از دلایل عملی نشدن این تئوری و دیگر تئوریهای مبتنی بر وجود مدل دینامیکی دقیق اینست که پیاده سازی این روش کنترلی در عمل منوط به داشتن مدل دینامیکی دقیقی از ربات می باشد که آن هم در

گرو شناسایی و محاسبه دقیق پارامترهای مکانیکی بدنه و مدل سازی دقیق ربات می باشد اما می دانیم محاسبه دقیق این پارامترها و نیز ارائه مدل دقیق از ربات بسیار مشکل می باشد .

۲. چنانچه فرض کنیم مشکل اول حل شده است در این مرحله لازم است اطلاعات مربوط به مسیرهای مطلوب را در مدل دینامیکی ربات جایگذاری کرده و ترم گشتاور محاسباتی را به صورت غیر همزمان^{۵۳} تولید کرده و به سیستم پیشخور کنیم اما از آنجا که مدل دینامیکی از ترمهای مثلثاتی پیچیده تشکیل شده است محاسبه این ترمها بر اساس اطلاعات مسیر مطلوب برای میکرو کنترلرها بسیار سنگین و تقریباً غیر قابل اجرا است .

برای حل مشکل اول یعنی مشکل محاسبه دقیق مدل دینامیکی ربات طرح پیشنهادی جدید خود را که از کنترل تطبیقی برای تقریب دینامیک ربات استفاده می کند تحت عنوان گشتاور محاسباتی تطبیقی ارائه کنیم . مزیت این روش نسبت به روش گشتاور محاسباتی اینست که در این روش بجای نیاز به مدل دینامیکی دقیق ربات می توان الگوریتم کنترلی را با مدلی همراه با عدم قطعیت های پارامتری شروع کرد ، این روش در الگوریتم تطبیق خود به صورت غیر مستقیم ترمهای $\hat{M}(\theta), \hat{h}(\theta, \dot{\theta})$ را در جهت کاهش خطای ردگیری به روز می کند لذا می توان از این ترمها به صورت مستقیم برای ساختن ترم گشتاور محاسبه شده استفاده کرد ، این روش در شبیه سازی نتایج خوبی داده است اما در مرحله پیاده سازی ملاحظات عملی نشان می دهد هرچند که این روش مشکل اول را با به روزکردن همزمان^{۵۴} مدل دینامیکی تقریبی حل می کند مشکل محاسبه دقیق مدل دینامیکی ربات اما همین همزمانی محاسبات ، مشکل دوم یعنی پیچیدگی محاسبات را که در بالا به آن اشاره شد تشدید می کند .

محدودیت سرعت پردازش برای انجام محاسبات این دو روش توسط میکروکنترلرها باعث شد طرح جدید خود را تحت عنوان گشتاور محاسباتی همراه با ترم پیشخور که در بخش (۵-۸) ارائه شد به عنوان طرح نهایی برای پیاده سازی عملی انتخاب کنیم در این طرح به مدل دینامیکی نیازی نداریم لذا مشکل اول را

^{۵۳} Off-Line

^{۵۴} On-Line

حل کرده ایم اما ترم گشتاور محاسباتی را به صورت همزمان توسط پارامترهای شناسایی شده موتورها یعنی J, B و سرعت ربات تولید می کنیم ، مکانیزم ساده ای که برای تولید ترم گشتاور محاسباتی به دست آمده نتیجه بررسی دقیق دینامیک موتور ها می باشد ، لازم به ذکر است که این سادگی باعث شده با وجود اینکه ترم گشتاور محاسباتی به صورت همزمان ساخته می شود اما مشکلات کمبود سرعت پردازش و نیز پیچیدگی محاسبات بوجود نیاید . این کنترلر با موفقیت روی ربات اجرا شد و نتایج شبیه سازی به صورت تحلیلی و نتایج عملی به صورت شهودی نشان می دهند طرح گشتاور محاسباتی ارائه شده طرح بسیار قدرتمند بوده که می تواند اغتشاشات ناشی از تاثیرات غیر خطی حرکت رابطها را بر عملکرد حلقه کنترل بدون نیاز به مدل دینامیکی ربات تا حد مطلوبی تضعیف کند این روش با تغییرات جزئی می تواند در زمره تکنیکهای مقاوم قرار گیرد ، مهمترین مزیت این روش همانطور که اشاره شد حل دو مشکلات مربوط به محاسبه دقیق مدل دینامیکی و محدودیت سرعت پردازشگرها می باشد .

اما در مورد ربات اسکارا می توان به ساختار تحلیلی جذاب آن اشاره کرد بازوی ماهری که ترم فوق العاده مشکل ساز نیروی گرانشی را دینامیک خود ندارد ، این ترم ، ترمی است که در خطی سازی فیدبکی سعی در تخمین زدن آن داریم و همین موضوع باعث بوجود آمدن خطاهای ناشی از تخمین در خطی سازی فیدبکی می شود ، از طرفی سینماتیک ملموس و شهودی آن درک ربات ارتباط فضاهای دکارتی و مفصلی را بسیار ساده می کند ، تشابه آن به ربات لولایی دو - رابط باعث شده که خیلی راحت بتوان تئوریهای ارائه شده روی ربات دو رابط لولایی را به ربات اسکارا تعمیم داد بنابراین ساختار اسکارا می تواند بهترین ساختار برای بررسی تئوریهای مختلف کنترلی باشد.

۱۰-۲- بررسی مشکلات و موانع موجود

بررسی مشکلات را در قالب این موضوع بیان می کنیم که یک تئوری آنجا ارزشمند می شود که در عمل بتوان آنرا تحقق داد ، ملاحظات عملی معمولاً باعث می شوند حتی تئوریهای جامع کنترلی غیر قابل اجرا شوند این مشکلات که در بخش نتیجه گیری به بعضی از آنها اشاره شد می توانند به صورت زیر باشند :

- صلب نبودن رابطهای ربات
- وجود لقی و خلاصی در چرخنده های جعبه دنده ها
- مشکلات ناشی از دقیق نبودن مدل دینامیکی
- محدودیت سرعت پردازشگرها

مشکل اول را با اعمال ملاحظات مکانیکی دقیق در موقع ساخت می توان از بین برد در ساخت این ربات با بررسی های مکانیکی این مشکل به حداقل رسید . مشکل دوم که معزل تمام رباتها است و می تواند دقت ردگیری و قابلیت تکرار ربات را بطور چشمگیری نامطلوب کند مبحث خلاصی چرخنده در جعبه دنده هاست این مشکل را نمی توان با تکنیکهای کنترلی و نرم افزار حل کردتنها راه مقابله با آن بکارگیری موتورهای با کیفیت عالی و با مکانیزم چرخ دنده ه ای حلزونی یا چرخ دنده های خطی با دقت بالا استفاده کرد . مشکلات دوم و سوم هم همانطور که به صورت مبسوط در نتایج آمد بوسیله طرح پیشنهادی گشتاور محاسباتی همراه با ترم پیشخور حل شده است .

۱۰-۳- فعالیتهای آینده

در نظر داریم در مراحل بعدی با طراحی یک سیستم گزارش دهنده که اطلاعات موقعیت ، زمان پیمایش و سرعت را ارائه می کند نتایج عملی را از حالت شهودی به عرصه تحلیلی آورده تا بتوان روشهای که نیاز بیشتری به تحلیل عددی نتایج دارند مانند طرح گشتاور محاسباتی تطبیقی پیشنهاد شده در بخش (۸-۶) و نیز گونه های مختلف تئوریهای کنترل غیر خطی ربات را روی این سیستم پیاده سازی کنیم .

پیوست ۱

برنامه میکرو کنترلرها :

رابط ۱:

```
"regfile "m32def.dat
```

```
crystal = 8000000
```

```
Config Spi = Hard , Interrupt = On , Data Order = Lsb , Master = No , Polarity = High , Phase = 0 ,  
Clockrate = 128
```

```
Config Int0 = Rising
```

```
Config Int1 = Rising
```

```
Config Timer0 = Counter , Edge = Falling
```

```
Config Timer1 = Timer , Prescale = 64
```

```
Config Timer2 = Pwm , Prescale = 64 , Pwm = On , Compare Pwm = Clear Up
```

```
Config Lcdpin = Pin , Db4 = Pinc.2 , Db5 = Pinc.3 , Db6 = Pinc.4 , Db7 = Pinc.5 , Rs = Pinc.0 , E = Pinc.1
```

```
Config Lcd = 16 * 2
```

```
Config Portd.6 = Output
```

```
Config Portd.5 = Output
```

```
Config Portd.0 = Output
```

```
Config Portd.1 = Output
```

```
Dim X As Byte , Ct As Single , Ct1 As Single , A As Word , B As Byte , C As Single , Q As Byte J As Single
```

```
Dim P As Integer , P2 As Byte , Pu As Word , Drift As Bit , W As Single , Ts As Single , Pc As Single
```

```
Dim Kp As Single , Kd As Single , E As Single , Edot As Single , El As Single , X0 As Byte , B As Single
```

Dim Astart As Byte , Astop As Word , Bstart As Single , Bstop As Single , Pwmmax As Byte , Start0 As Bit

Dim Sa As Integer , Sa1 As Byte , Sa2 As Byte , Sx As Byte , Give As Bit , Pold As Integer , Pnew As Integer

Dim Dc As Single , Pd As Single , Ff As Single , Kv As Single , Xx As Single , Ct1 As Single

J Const = 0.00000146

B Const = 0.0000083

Dirr Alias Portd.6

Brk Alias Portd.5

Sel2 Alias Portd.0

Di1 Alias Portd.1

Kp = 10

Kd = 1

Cls

" Lcd " SLAVE1

Cursor Off

Sa = 0

Reset Brk

Reset Give

Reset Start0

Reset Sel2

Pold = 0

Gosub Initialization

Enable Interrupts

Enable Int0

On Int0 Show_start

Enable Int1

On Int1 Backlash1

Enable Spi

On Spi S_isr

Spiinit

Enable Timer0

On Timer0 CO_isr

Do

If Start0 = 1 Then

Disable Interrupts

If P > 0 Then

Reset Dirr

Elseif P < 0 Then

Set Dirr

Else

Reset Start0

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Goto Ex1

End If

Gosub Start1

Counter0 = 245

Enable Interrupts

A = 0 : Give = 0

Gosub Main_loop

End If

:Ex1

Ocr2 = 0

Enable Interrupts

Loop

End

:S_isr

Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : X0 = 0

Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0

Disable Interrupts

Spiin Sa1 , 1

Do

Spiin Sx , 1

If Sx = 255 Then Exit Do

Loop

Spiin Sa2 , 1

Enable Interrupts

Return

Show_start:

Disable Interrupts

If Sa1 = 0 Then

Pnew = Sa2

Elseif Sa1 = 1 Then

Pnew = -sa2

Else

Locate 2 , 10

"Lcd "Error

End If

P = Pnew - Pold

Pold = Pnew

```

Cls
Lcd "p:" ; P ; "o:" ; Pold ; "n:" ; Pnew
Counter0 = 245
Set Start0
Enable Interrupts
Return

```

```

:Main_loop

```

```

Do

```

```

If Give = 0 Then

```

```

E = Pu - A

```

```

Edot = E - El

```

```

Edot = Edot / Ts

```

```

Pc = Kp * E

```

```

Dc = Kd * Edot

```

```

Pd = Pc + Dc

```

```

El = E

```

```

'CT=(J^s+b^)^8'

```

```

Ct1 = J * Acc

```

```

Ct = B * Vel

```

```

Ct = Ct1 + Ct

```

```

Ff = 0 'for point to point control feedforward statement is zero

```

```

Kv = Ff + Pd

```

```

Kv = Kv + Xx

```

```

Xx = Kv - ct

```

```

Gosub Dynamic_limiter

```

```

(B = Round(c

```

```

Ocr2 = B

```

```

Else

```

```

Goto Exit2

```

End If

Loop

:Exit2

Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : Reset Start0 : X0 = 0

Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0 : Reset Give

Counter0 = 245

Return

:Start1

(Pu = Abs(p

Set Drift

Astart = 69

Astop = 65

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Pwmmax = 150

Gosub Limit_pwm

A = 0

Return

:T1_isr

Stop Timer1

Cls

"Lcd "t1 ovf occur

Timer1 = 0

Start Timer1

Return


```

:CO_isr
Stop Timer1
Incr A
If A = Pu Then
A = 0
Ocr2 = 0
Set Brk
Waitms 200
Reset Brk
Set Give
If Dirr = 1 Then
Reset Di1
Else
Set Di1
End If
Set Sel2           'set int1 of joint 2 for overcome the backlash
Reset Sel2
End If
Ts = Timer1 * 8
Ts = Ts / 1000000
Timer1 = 0
vel0=vel
Vel = 1 / Ts       'degree/sec
Rad = Pi / 180
Vel = Vel / Rad    'rad/sec
Acc = Vel - vel0
Acc = Acc / Ts     'degree/sec2
Acc = Acc / Rad    'rad/sec2
Start Timer1

```

Toggle Drift

If Drift = 1 Then

Counter0 = 246

Else

Counter0 = 245

End If

Return

:Dynamic_limiter

a is current position'

If A < P2 Then

C = Bstart * Kv

C = C + Astart

Else

C = Bstop * Kv

C = C + Astop

End If

Return

:Initialization

Reset Dirr

Ocr2 = 65

Wait 5

Ocr2 = 0

Wait 1

Set Dirr

Ocr2 = 66

Counter0 = 0

:L1

If Counter0 < 200 Then Goto L1

Incr X0

Counter0 = 0

If X0 <= 5 Then Goto L1

X0 = 0

Counter0 = 0

:L2

If Counter0 < 124 Then Goto L2

Ocr2 = 0

Set Brk

Waitms 200

Reset Brk

Waitms 100

Return

:Limit_pwm

If Pu < 5 Then

Q = Astart + 0

W = 0.1 * Kv

(Pwmmax = Round(w

Pwmmax = Q + Pwmmax

Elseif Pu < 10 Then

Q = Astart + 2

W = 0.4 * Kv

(Pwmmax = Round(w

Pwmmax = Q + Pwmmax

Elseif Pu < 20 Then

Q = Astart + 3

W = 0.8 * Kv

(Pwmmax = Round(w

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 30$ Then

$Q = A_{start} + 4$

$W = K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 40$ Then

$Q = A_{start} + 5$

$W = 1.2 * K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 50$ Then

$Q = A_{start} + 6$

$W = 1.4 * K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 60$ Then

$Q = A_{start} + 7$

$W = 1.5 * K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 70$ Then

$Q = A_{start} + 8$

$W = 1.6 * K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 80$ Then

$Q = A_{start} + 9$

$W = 1.7 * K_v$

$(P_{wmmax} = \text{Round}(w$

$P_{wmmax} = Q + P_{wmmax}$

Elseif $P_u < 90$ Then

$Q = A_{start} + 10$

$W = 1.8 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Elseif $P_u < 120$ Then

$Q = A_{start} + 12$

$W = 1.9 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Elseif $P_u < 150$ Then

$Q = A_{start} + 14$

$W = 2 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Elseif $P_u < 180$ Then

$Q = A_{start} + 16$

$W = 2.2 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Elseif $P_u < 230$ Then

$Q = A_{start} + 18$

$W = 2.4 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Elseif $P_u < 260$ Then

$Q = A_{start} + 20$

$W = 2.6 * K_v$

$(P_{wmax} = \text{Round}(w$

$P_{wmax} = Q + P_{wmax}$

Else

```

Pwmmax = Astart + 10
Cls
"lcd "pwm overlap
End If
If Pwmmax > 120 Then Pwmmax = 120
If Z < .7 Then Pwmmax = 40

```

```

C = P / 2
(C = Abs(c
(P2 = Round(c
C = Pwmmax - Astart
C = C / P2
Bstart = C
A = 2 * Pwmmax
Astop = A - Astop
C = Pwmmax - Astop
C = C / P2
Bstop = C
Return

```

```

' end of program

```

رابط ٢:

```

"regfile "m32def.dat$
crystal = 8000000$
Config Spi = Hard , Interrupt = On , Data Order = Lsb , Master = No , Polarity = High , Phase = 0 ,
Clockrate = 128
Config Int0 = Rising
Config Int1 = Rising
Config Timer0 = Counter , Edge = Falling
Config Timer1 = Timer , Prescale = 64

```

Config Timer2 = Pwm , Prescale = 64 , Pwm = On , Compare Pwm = Clear Up

Config Lcdpin = Pin , Db4 = Pinc.2 , Db5 = Pinc.3 , Db6 = Pinc.4 , Db7 = Pinc.5 , Rs = Pinc.0 , E = Pinc.1

Config Lcd = 16 * 2

Config Portd.6 = Output

Config Portd.5 = Output

Config Portd.0 = Output

Config Portd.1 = Output

Dim X As Byte , Ct As Single , Ct1 As Single , A As Word , B As Byte , C As Single , Q As Byte J As Single

Dim P As Integer , P2 As Byte , Pu As Word , Drift As Bit , W As Single , Ts As Single , Pc As Single

Dim Kp As Single , Kd As Single , E As Single , Edot As Single , El As Single , X0 As Byte , B As Single

Dim Astart As Byte , Astop As Word , Bstart As Single , Bstop As Single , Pwmmax As Byte , Start0 As Bit

Dim Sa As Integer , Sa1 As Byte , Sa2 As Byte , Sx As Byte , Give As Bit , Pold As Integer , Pnew As Integer

Dim Dc As Single , Pd As Single , Ff As Single , Kv As Single , Xx As Single , Ct1 As Single

J Const = 0.00000135

B Const = 0.0000031

Dirr Alias Portd.6

Brk Alias Portd.5

Sel2 Alias Portd.0

Di1 Alias Portd.1

Kp = 10

Kd = 1.2

Cls

" Lcd " SLAVE2

Cursor Off

Sa = 0

Reset Brk

Reset Give

Reset Start0

Reset Sel2

Pold = 0

Gosub Initialization

Enable Interrupts

Enable Int0

On Int0 Show_start

Enable Int1

On Int1 Backlash1

Enable Spi

On Spi S_isr

Spiinit

Enable Timer0

On Timer0 CO_isr

Do

If Start0 = 1 Then

Disable Interrupts

If P > 0 Then

Reset Dirr

Elseif P < 0 Then

Set Dirr

Else

Reset Start0

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Goto Ex1

End If


```

Gosub Start1
Counter0 = 245
Enable Interrupts
A = 0 : Give = 0
Gosub Main_loop
End If
:Ex1
Ocr2 = 0
Enable Interrupts
Loop
End

```

```

:S_isr
Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : X0 = 0
Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0
Disable Interrupts
Spiin Sa1 , 1
Do
Spiin Sx , 1
If Sx = 255 Then Exit Do
Loop
Spiin Sa2 , 1
Enable Interrupts
Return

```

```

:Show_start
Disable Interrupts
If Sa1 = 0 Then
Pnew = Sa2

```

```

Elseif Sa1 = 1 Then
Pnew = -sa2
Else
Locate 2 , 10
"Lcd "Error
End If
P = Pnew - Pold
Pold = Pnew
Cls
Lcd "p:" ; P ; "o:" ; Pold ; "n:" ; Pnew
Counter0 = 245
Set Start0
Enable Interrupts
Return

```

```

:Main_loop
Do
If Give = 0 Then
E = Pu - A
Edot = E - El
Edot = Edot / Ts
Pc = Kp * E
Dc = Kd * Edot
Pd = Pc + Dc
El = E
'CT=(J^s+b^)^8'
Ct1 = J * Acc
Ct = B * Vel
Ct = Ct1 + Ct
Ff = 0

```

'for point to point control feedforward statement is zero

```

Kv = Ff + Pd
Kv = Kv + Xx
Xx = Kv - ct
Gosub Dynamic_limiter
(B = Round(c
Ocr2 = B
Else
Goto Exit2
End If
Loop
:Exit2
Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : Reset Start0 : X0 = 0
Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0 : Reset Give
Counter0 = 245
Return

```

```

:Start1
(Pu = Abs(p
Astart = 100
Astop = 95
Locate 2 , 1
" " Lcd
Locate 2 , 1
Lcd P ; " " ; Pold ; " " ; Pnew
Pwmmax = 200
Gosub Limit_pwm
A = 0
Return

```

```

:T1_isr
Stop Timer1
Cls
"Lcd "t1 ovf occur
Timer1 = 0
Start Timer1
Return

```

```

:CO_isr
Stop Timer1
Incr A
If A = Pu Then
A = 0
Ocr2 = 0
Set Brk
Waitms 200
Reset Brk
Set Give
If Dirr = 1 Then
Reset Di1
Else
Set Di1
End If
Set Sel2                                'set int1 of joint 2 for overcome the backlash
Reset Sel2
End If
Ts = Timer1 * 8
Ts = Ts / 1000000
Timer1 = 0
vel0=vel

```

```

Vel = 1 / Ts                                'degree/sec
Rad = Pi / 180
Vel = Vel / Rad                             'rad/sec
Acc = Vel - vel0
Acc = Acc / Ts                             'degree/sec2
Acc = Acc / Rad                             'rad/sec2
Start Timer1
Counter0 = 255
End If
Return

```

```

:Dynamic_limiter
a is current position'
If A < P2 Then
C = Bstart * Kv
C = C + Astart
Else
C = Bstop * Kv
C = C + Astop
End If
Return

```

```

:Initialization
Reset Dirr
Ocr2 = 100
Wait 5
Ocr2 = 0
Wait 1
Set Dirr

```

```

Ocr2 = 100
Counter0 = 0
:L1
If Counter0 <138 Then Goto L1
Ocr2 = 0
Set Brk
Waitms 200
Reset Brk
Waitms 100
Return

```

```

:Backlash
Disable Interrupts
If Start0 = 0 Then
Dir1 = Di1
Blash = Counter0
If Blash > 20 Then Goto Ex5
Counter0 = 0
If Dir1 = 0 Then
Set Dirr
Else
Reset Dirr
End If

```

```

:L4
Blashc = Blash / 1.6
(Blash = Round(blashc
Ocr2 = 100
If Counter0 <> Blash Then Goto L4
Ocr2 = 0
Set Brk

```

Waitms 200

Reset Brk

Wait 1

:Ex5

End If

Blash = 0

Counter0 = 255

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd Blash ; " " ; Dir1

Enable Interrupts

Return

:Limit_pwm

If Pu < 5 Then

Q = Astart + 0

W = 0.1 * Kv

(Pwmmax = Round(w

Pwmmax = Q + Pwmmax

Elseif Pu < 10 Then

Q = Astart + 2

W = 0.4 * Kv

(Pwmmax = Round(w

Pwmmax = Q + Pwmmax

Elseif Pu < 20 Then

Q = Astart + 3

W = 0.8 * Kv

(Pwmmax = Round(w

Pwmmax = Q + Pwmmax

Elseif $P_u < 30$ Then

$$Q = A_{start} + 4$$

$$W = K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 40$ Then

$$Q = A_{start} + 5$$

$$W = 1.2 * K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 50$ Then

$$Q = A_{start} + 6$$

$$W = 1.4 * K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 60$ Then

$$Q = A_{start} + 7$$

$$W = 1.5 * K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 70$ Then

$$Q = A_{start} + 8$$

$$W = 1.6 * K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 80$ Then

$$Q = A_{start} + 9$$

$$W = 1.7 * K_v$$

$$(P_{wmax} = \text{Round}(w$$

$$P_{wmax} = Q + P_{wmax}$$

Elseif $P_u < 90$ Then


```

Q = Astart + 10
W = 1.8 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 120 Then
Q = Astart + 12
W = 1.9 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 150 Then
Q = Astart + 14
W = 2 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 180 Then
Q = Astart + 16
W = 2.2 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 230 Then
Q = Astart + 18
W = 2.4 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 260 Then
Q = Astart + 20
W = 2.6 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Else
Pwmmax = Astart + 10

```

```

Cls
"lcd "pwm overlap
End If
If Pwmmax > 120 Then Pwmmax = 120
If Z < .7 Then Pwmmax = 40

```

```

C = P / 2
(C = Abs(c
(P2 = Round(c
C = Pwmmax - Astart
C = C / P2
Bstart = C
A = 2 * Pwmmax
Astop = A - Astop
C = Pwmmax - Astop
C = C / P2
Bstop = C
Return

```

```

' end of program

```

رابط ۳:

```

"regfile "m32def.dat$
crystal = 8000000$
Config Spi = Hard , Interrupt = On , Data Order = Lsb , Master = No , Polarity = High , Phase = 0 ,
Clockrate = 128
Config Int0 = Rising
Config Int1 = Rising
Config Timer0 = Counter , Edge = Falling
Config Timer1 = Timer , Prescale = 64
Config Timer2 = Pwm , Prescale = 64 , Pwm = On , Compare Pwm = Clear Up

```

Config Lcdpin = Pin , Db4 = Pinc.2 , Db5 = Pinc.3 , Db6 = Pinc.4 , Db7 = Pinc.5 , Rs = Pinc.0 , E = Pinc.1

Config Lcd = 16 * 2

Config Portd.6 = Output

Config Portd.5 = Output

Config Portd.0 = Output

Config Portd.1 = Output

Dim X As Byte , Ct As Single , Ct1 As Single , A As Word , B As Byte , C As Single , Q As Byte J As Single

Dim P As Integer , P2 As Byte , Pu As Word , Drift As Bit , W As Single , Ts As Single , Pc As Single

Dim Kp As Single , Kd As Single , E As Single , Edot As Single , El As Single , X0 As Byte , B As Single

Dim Astart As Byte , Astop As Word , Bstart As Single , Bstop As Single , Pwmmmax As Byte , Start0 As Bit

Dim Sa As Integer , Sa1 As Byte , Sa2 As Byte , Sx As Byte , Give As Bit , Pold As Integer , Pnew As Integer

Dim Dc As Single , Pd As Single , Ff As Single , Kv As Single , Xx As Single , Ct1 As Single

J Const = 0.000000372

B Const = 0.00000243

Dirr Alias Portd.6

Brk Alias Portd.5

Sel2 Alias Portd.0

Di1 Alias Portd.1

Kp = 2

Kd = 1

Cls

" Lcd " SLAVE3

Cursor Off

Sa = 0

Reset Brk

Reset Give

Reset Start0

Reset Sel2

Pold = 0

Gosub Initialization

Enable Interrupts

Enable Int0

On Int0 Show_start

Enable Int1

On Int1 Backlash1

Enable Spi

On Spi S_isr

Spiinit

Enable Timer0

On Timer0 CO_isr

Do

If Start0 = 1 Then

Disable Interrupts

If P > 0 Then

Reset Dirr

Elseif P < 0 Then

Set Dirr

Else

Reset Start0

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Goto Ex1

```

End If
Gosub Start1
Counter0 = 245
Enable Interrupts
A = 0 : Give = 0
Gosub Main_loop
End If
:Ex1
Ocr2 = 0
Enable Interrupts
Loop
End

```

```

:S_isr
Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : X0 = 0
Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0
Disable Interrupts
Spiin Sa1 , 1
Do
Spiin Sx , 1
If Sx = 255 Then Exit Do
Loop
Spiin Sa2 , 1
Enable Interrupts
Return

```

```

:Show_start
Disable Interrupts
If Sa1 = 0 Then

```

```

Pnew = Sa2
Elseif Sa1 = 1 Then
Pnew = -sa2
Else
Locate 2 , 10
"Lcd "Error
End If
P = Pnew - Pold
C = 3 * P
(P = Round(c
Pold = Pnew
Cls
Lcd "p:" ; P ; "o:" ; Pold ; "n:" ; Pnew
Counter0 = 255
Set Start0
Enable Interrupts
Return$

```

```

:Main_loop
Do
If Give = 0 Then
E = Pu - A
Edot = E - El
Edot = Edot / Ts
Pc = Kp * E
Dc = Kd * Edot
Pd = Pc + Dc
El = E
'CT=(J^s+b^)^8'
Ct1 = J * Acc

```

```

Ct = B * Vel
Ct = Ct1 + Ct
Ff = 0                                'for point to point control feedforward statement is zero
Kv = Ff + Pd
Kv = Kv + Xx
Xx = Kv - ct
Gosub Dynamic_limiter
(B = Round(c
Ocr2 = B
Else
Goto Exit2
End If
Loop
:Exit2
Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : Reset Start0 : X0 = 0
Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0 : Reset Give
Counter0 = 245
Return

```

```

:Start1
(Pu = Abs(p
Set Drift
Astart = 30
Astop = 20
Locate 2 , 1
"          " Lcd
Locate 2 , 1
Lcd P ; " " ; Pold ; " " ; Pnew
Pwmmax = 150
Gosub Limit_pwm

```

A = 0

Return

:T1_isr

Stop Timer1

Cls

"Lcd "t1 ovf occur

Timer1 = 0

Start Timer1

Return

:C0_isr

Stop Timer1

Incr A

If A = Pu Then

A = 0

Ocr2 = 0

Set Brk

Waitms 200

Reset Brk

Set Give

If Dirr = 1 Then

Reset Di1

Else

Set Di1

End If

Set Sel2 'set int1 of joint 2 for overcome the backlash

Reset Sel2

End If


```

Ts = Timer1 * 8
Ts = Ts / 1000000
Timer1 = 0
vel0=vel
Vel = 1 / Ts                                'degree/sec
Rad = Pi / 180
Vel = Vel / Rad                             'rad/sec
Acc = Vel - vel0
Acc = Acc / Ts                              'degree/sec2
Acc = Acc / Rad                             'rad/sec2
Start Timer1
Counter0 = 255
End If
Return

```

```

:Dynamic_limiter
a is current position'
If A < P2 Then
C = Bstart * Kv
C = C + Astart
Else
C = Bstop * Kv
C = C + Astop
End If
Return

```

```

:Initialization
Reset Dirr
Ocr2 = 30

```

```

Wait 5
Ocr2 = 0
Wait 1
Set Dirr
Ocr2 = 30
Counter0 = 0
:L1
If Counter0 < 200 Then Goto L1
Incr X0
Counter0 = 0
If X0 <= 5 Then Goto L1
X0 = 0
Counter0 = 0
:L2
If Counter0 < 124 Then Goto L2
Ocr2 = 0
Set Brk
Waitms 200
Reset Brk
Waitms 100
Return

```

```

:Limit_pwm
If Pu < 5 Then
Q = Astart + 0
W = 0.1 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 10 Then
Q = Astart + 2

```

$$W = 0.4 * Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 20$ Then

$$Q = Astart + 3$$

$$W = 0.8 * Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 30$ Then

$$Q = Astart + 4$$

$$W = Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 40$ Then

$$Q = Astart + 5$$

$$W = 1.2 * Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 50$ Then

$$Q = Astart + 6$$

$$W = 1.4 * Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 60$ Then

$$Q = Astart + 7$$

$$W = 1.5 * Kv$$

$$(Pwmmax = Round(w$$

$$Pwmmax = Q + Pwmmax$$

Elseif $Pu < 70$ Then

$$Q = Astart + 8$$

$$W = 1.6 * Kv$$

```

(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 80 Then
Q = Astart + 9
W = 1.7 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 90 Then
Q = Astart + 10
W = 1.8 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 120 Then
Q = Astart + 12
W = 1.9 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 150 Then
Q = Astart + 14
W = 2 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 180 Then
Q = Astart + 16
W = 2.2 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Elseif Pu < 230 Then
Q = Astart + 18
W = 2.4 * Kv
(Pwmmax = Round(w

```

```

Pwmmax = Q + Pwmmax
Elseif Pu < 260 Then
Q = Astart + 20
W = 2.6 * Kv
(Pwmmax = Round(w
Pwmmax = Q + Pwmmax
Else
Pwmmax = Astart + 10
Cls
"lcd "pwm overlap
End If
If Pwmmax > 60 Then Pwmmax = 60
If Z < .7 Then Pwmmax = 40

```

```

C = P / 2
(C = Abs(c
(P2 = Round(c
C = Pwmmax - Astart
C = C / P2
Bstart = C
A = 2 * Pwmmax
Astop = A - Astop
C = Pwmmax - Astop
C = C / P2
Bstop = C
Return

```

```

' end of program

```

```
"regfile "m32def.dat$
crystal = 8000000$
Config Spi = Hard , Interrupt = On , Data Order = Lsb , Master = No , Polarity = High , Phase = 0 ,
Clockrate = 128
Config Int0 = Rising
Config Timer2 = Pwm , Prescale = 64 , Pwm = On , Compare Pwm = Clear Up
Config Timer0 = Counter , Edge = Falling
Config Timer1 = Timer , Prescale = 64
Config Lcdpin = Pin , Db4 = Pinc.2 , Db5 = Pinc.3 , Db6 = Pinc.4 , Db7 = Pinc.5 , Rs = Pinc.0 , E = Pinc.1
Config Lcd = 16 * 2
Config Portd.6 = Output
Config Portd.5 = Output
Dim X As Byte , A As Word , B As Byte , C As Single , P As Integer , P2 As Byte , Pu As Word
Dim Astart As Byte , Astop As Word , Bstart As Single , Bstop As Single , Pwmmax As Byte , Start0 As
Bit
Dim Sa As Integer , Sa1 As Byte , Sa2 As Byte , Sx As Byte , Give As Bit , Pold As Integer , Pnew As
Integer
Dirr Alias Portd.6
Brk Alias Portd.5

Cls
" Lcd " SLAVE3
Cursor Off
Sa = 0
Reset Brk
Reset Give
Reset Start0
Pold = 0
Gosub Goto_home
```

Enable Interrupts

Enable Int0

On Int0 Show_start

Enable Spi

On Spi S_isr

Spiinit

Enable Timer0

On Timer0 CO_isr

Do

If Start0 = 1 Then

Disable Interrupts

If P > 0 Then

Reset Dirr

Elseif P < 0 Then

Set Dirr

Else

Reset Start0

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Goto Ex1

End If

Gosub Start1

Counter0 = 255

Enable Interrupts

A = 0 : Give = 0

Gosub Main_loop

End If

:Ex1

Ocr2 = 0

Enable Interrupts

Loop

End

:S_isr

Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0

Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0

Disable Interrupts

Spiin Sa1 , 1

Do

Spiin Sx , 1

If Sx = 255 Then Exit Do

Loop

Spiin Sa2 , 1

Enable Interrupts

Return

:Show_start

Disable Interrupts

If Sa1 = 0 Then

Pnew = Sa2

Elseif Sa1 = 1 Then

Pnew = -sa2

Else

Locate 2 , 10

"Lcd "Error


```

End If
P = Pnew - Pold
C = P / 5.625
(P = Round(c
Pold = Pnew
Cls
Lcd "p:" ; P ; "o:" ; Pold ; "n:" ; Pnew
Counter0 = 255
Set Start0
Enable Interrupts
Return

```

```

:Main_loop
Do
If Give = 0 Then
If A < P2 Then
C = Bstart * A
C = C + Astart
Else
C = Bstop * A
C = C + Astop
End If
(B = Round(c
Ocr2 = B
Else
Goto Exit2
End If
Loop

```

```

:Exit2

```

Give = 0 : P = 0 : Pu = 0 : Pnew = 0 : C = 0 : B = 0 : Reset Start0

Sa1 = 0 : Sa2 = 0 : Sa = 0 : Sx = 0 : P2 = 0 : A = 0 : Reset Give

Counter0 = 255

Return

:Start1

(Pu = Abs(p

Astart = 20

Astop = 17

Locate 2 , 1

" " Lcd

Locate 2 , 1

Lcd P ; " " ; Pold ; " " ; Pnew

Pwmmax = 60

Gosub Pass

A = 0

Return

:C0_isr

Incr A

If A = Pu Then

A = 0

Ocr2 = 0

Set Brk

Waitms 200

Reset Brk

Set Give

End If

Counter0 = 255

Return

```

:Goto_home
Set Dirr
Ocr2 = 20
Counter0 = 0
:L1
If Counter0 < 16 Then Goto L1
Counter0 = 255
Ocr2 = 0
Set Brk
Waitms 200
Reset Brk
Return

```

```

:Pass
If Pu < 5 Then
Pwmmax = Astart + 0
Elseif Pu < 10 Then
Pwmmax = Astart + 2
Elseif Pu < 20 Then
Pwmmax = Astart + 4
Elseif Pu < 30 Then
Pwmmax = Astart + 6
Elseif Pu < 40 Then
Pwmmax = Astart + 8
Elseif Pu < 50 Then
Pwmmax = Astart + 10
Elseif Pu < 60 Then
Pwmmax = Astart + 12
Elseif Pu < 70 Then
Pwmmax = Astart + 14
Elseif Pu < 80 Then

```

Pwmmax = Astart + 16

Elseif Pu < 90 Then

Pwmmax = Astart + 20

Elseif Pu < 120 Then

Pwmmax = Astart + 25

Elseif Pu < 150 Then

Pwmmax = Astart + 30

Elseif Pu <= 180 Then

Pwmmax = Astart + 35

Else

Pu = 60

End If

C = P / 2

(C = Abs(c

(P2 = Round(c

C = Pwmmax - Astart

C = C / P2

Bstart = C

A = 2 * Pwmmax

Astop = A - Astop

C = Pwmmax - Astop

C = C / P2

Bstop = C

Return

===== end of program ===== "

پیوست ۲

برنامه Visual Basic Interface :

Option Explicit

Dim x As Single, y As Single, z As Single, theta As Integer, d3 As Integer

Dim dd3 As Single, t1 As Single, tt2 As Double, t2 As Single, tt1 As Double, ttt1 As Double

Dim a2 As Single, a3 As Single, d1 As Single, r As Single, s As Single, q As Single

Dim vx1 As Single, vx2 As Single, vy As Double

Dim sign1 As Byte, sign2 As Byte, st1 As Byte, st2 As Byte

Dim sign4 As Byte, st3 As Byte, st4 As Byte

()Private Sub ang_Click

sendcmd.Enabled = False

((t1 = Int(Val(t1j.Text

((t2 = Int(Val(t2j.Text

```
((d3 = Int(Val(d3j.Text
```

```
((theta = Int(Val(wj.Text
```

```
If t2 > 130 Or t2 < -130 Or t1 > 125 Or t1 < -125 Then
```

```
t1j.BackColor = &H8080FF
```

```
t2j.BackColor = &H8080FF
```

```
End If
```

```
If d3 > 15 Or d3 < 0 Then
```

```
d3j.BackColor = &H8080FF
```

```
End If
```

```
If theta > 180 Or theta < -180 Then
```

```
wj.BackColor = &H8080FF
```

```
End If
```

```
If t2 > 130 Or t2 < -130 Or t1 > 125 Or t1 < -125 Or d3 > 15 Or d3 < 0 Or theta > 180 Or theta < -180  
Then GoTo l1
```

```
t1j.BackColor = &HFFFFFF
```

```
t2j.BackColor = &HFFFFFF
```

```
d3j.BackColor = &HFFFFFF
```

```
wj.BackColor = &HFFFFFF
```

```
sendcmd.Enabled = True
```

```
If t1 < 0 Then
```

```
sign1 = 1
```

```
(st1 = Abs(t1
```

```
Else
```

```
sign1 = 0
```

```
st1 = t1
```

```
End If
```

If t2 < 0 Then

sign2 = 1

(st2 = Abs(t2

Else

sign2 = 0

st2 = t2

End If

st3 = d3

If theta < 0 Then

sign4 = 1

(st4 = Abs(theta

Else

sign4 = 0

st4 = theta

End If

(MSComm1.Output = Chr\$(sign1

Timer1.Interval = 10

Timer2.Interval = 20

Timer3.Interval = 30

Timer4.Interval = 40

Timer5.Interval = 50

Timer6.Interval = 60

Timer1.Enabled = True

Timer2.Enabled = True

Timer3.Enabled = True

Timer4.Enabled = True

Timer5.Enabled = True

Timer6.Enabled = True

:11

End Sub

()Private Sub Form_Load

MSComm1.CommPort = 1

"MSComm1.Settings = "9600,n,8,1

MSComm1.InputMode = comInputModeText

MSComm1.PortOpen = True

reccmd.Enabled = False

End Sub

(Private Sub Form_Unload(Cancel As Integer

MSComm1.PortOpen = False

End Sub

()Private Sub clear_Click

"" = tx.Text

"" = ty.Text

"" = tz.Text

"" = tt.Text

"" = tet2.Text

"" = tet1.Text

"" = td3.Text

"" = tet.Text

"" = xy.Text

End Sub

(Private Sub end_Click(Index As Integer


```

sign1 = 0
st1 = 0
sign2 = 0
st2 = 0
st3 = 0
sign4 = 0
st4 = 0
(MSComm1.Output = Chr$(sign1
Timer1.Interval = 10
Timer2.Interval = 20
Timer3.Interval = 30
Timer4.Interval = 40
Timer5.Interval = 50
Timer6.Interval = 60
Timer7.Interval = 70
Timer8.Interval = 200
Timer1.Enabled = True
Timer2.Enabled = True
Timer3.Enabled = True
Timer4.Enabled = True
Timer5.Enabled = True
Timer6.Enabled = True
Timer7.Enabled = True
Timer8.Interval = True
End Sub

(Private Sub Image2_Click(Index As Integer

End Sub

()Private Sub sendcmd_Click

```

If t1 < 0 Then

sign1 = 1

(st1 = Abs(t1

Else

sign1 = 0

st1 = t1

End If

If t2 < 0 Then

sign2 = 1

(st2 = Abs(t2

Else

sign2 = 0

st2 = t2

End If

st3 = d3

If theta < 0 Then

sign4 = 1

(st4 = Abs(theta

Else

sign4 = 0

st4 = theta

End If

(MSComm1.Output = Chr\$(sign1

Timer1.Interval = 10

Timer2.Interval = 20

Timer3.Interval = 30

```
Timer4.Interval = 40
Timer5.Interval = 50
Timer6.Interval = 60
Timer7.Interval = 70
Timer8.Interval = 200
Timer1.Enabled = True
Timer2.Enabled = True
Timer3.Enabled = True
Timer4.Enabled = True
Timer5.Enabled = True
Timer6.Enabled = True
Timer7.Enabled = True
Timer8.Enabled = True
End Sub
```

```
()Private Sub Text4_Change
```

```
End Sub
```

```
()Private Sub Timer1_Timer
```

```
Timer1.Enabled = False
```

```
(MSComm1.Output = Chr$(st1
```

```
End Sub
```

```
()Private Sub Timer2_Timer
```

```
Timer2.Enabled = False
```

```
(MSComm1.Output = Chr$(sign2
```

```
End Sub
```

```
()Private Sub Timer3_Timer
```

```
Timer3.Enabled = False
```

```
(MSComm1.Output = Chr$(st2
```

```
End Sub
```

```

()Private Sub Timer4_Timer
Timer4.Enabled = False
(MSComm1.Output = Chr$(sign1
End Sub

```

```

()Private Sub Timer5_Timer
Timer5.Enabled = False
(MSComm1.Output = Chr$(st3
End Sub

```

```

()Private Sub Timer6_Timer
Timer6.Enabled = False
(MSComm1.Output = Chr$(sign4
End Sub

```

```

()Private Sub Timer7_Timer
Timer7.Enabled = False
(MSComm1.Output = Chr$(st4
End Sub

```

```

()Private Sub Timer8_Timer
Timer8.Enabled = False
MSComm1.PortOpen = False
SCARA.Hide
End
End Sub

```

```

()Private Sub trans_Click
sendcmd.Enabled = False
a2 = 0.2
a3 = 0.15
d1 = 0.15
s = (a2) ^ 2 + (a3) ^ 2
q = (a2) ^ 2 - (a3) ^ 2
(x = 0.01 * Val(tx.Text

```

```

(y = 0.01 * Val(ty.Text
(z = 0.01 * Val(tz.Text
(theta = Val(tt.Text
r = x ^ 2 + y ^ 2
xy.Text = 10000 * r
If r < 0.0025 Or r > 0.1225 Then
"" = tx.Text
"" = ty.Text
"" = tet2.Text
"" = tet1.Text
GoTo l1
End If

```

```

(vy = ((2 * r * s) - (q ^ 2) - (r ^ 2)) ^ (0.5
vx1 = r - s
vx2 = r + q
dd3 = d1 - z
(d3 = Int(100 * dd3

```

```

((tt2 = (180 / 3.1416) * Atn(vy / (vx1 + 0.0001
If vx1 < 0 Then tt2 = 180 + tt2

```

```

((ttt1 = (180 / 3.1416) * Atn(y / (x + 0.0001
If x < 0 And y > 0 Then ttt1 = 180 + ttt1
If x < 0 And y <= 0 Then ttt1 = ttt1 - 180

```

```

((tt1 = (180 / 3.1416) * Atn(vy / (vx2 + 0.0001
If vx2 < 0 Then tt1 = 180 + tt1
tt1 = ttt1 - tt1

```

(tet2.Text = Int(tt2

(tet1.Text = Int(tt1

If Int(tt2) > 130 Or Int(tt2) < -130 Or Int(tt1) > 125 Or Int(tt1) < -125 Then

ty.BackColor = &H8080FF

tx.BackColor = &H8080FF

End If

If d3 > 15 Or d3 < 0 Then

td3.Text = d3

tz.BackColor = &H8080FF

End If

If theta > 180 Or theta < -180 Then

tet.Text = theta

tt.BackColor = &H8080FF

End If

If Int(tt2) > 130 Or Int(tt2) < -130 Or Int(tt1) > 125 Or Int(tt1) < -125 Or d3 > 15 Or d3 < 0 Or theta > 180 Or theta < -180 Then GoTo l1

(t2 = Int(tt2

(t1 = Int(tt1

td3.Text = d3

tet.Text = theta

tz.BackColor = &HFFFFFF

tt.BackColor = &HFFFFFF

tx.BackColor = &HFFFFFF

ty.BackColor = &HFFFFFF

sendcmd.Enabled = True

:l1

End Sub

پیوست ۳

مشخصات فنی چیپهای استفاده شده در این پروژه در سایتهای ارائه شده در زیر موجودند :

<http://www.Datasheetcatalog.com>

<http://www.Datasheet.com>

DM74LS14 : HEX INVERTER WITH SCHMITT TRIGGER INPUTS

L298 :DUAL FULL-BRIDGE DRIVER

P521 : HIGH DENSITY MOUNTING PHOTOTRANSISTOR OPTICALLY COUPLED ISOLATORS

MAX232 : DUAL EIA-232 DRIVERS/RECEIVERS

GP1A53HR : OPIC PHOTOINTERRUPTER

HD4093 : HEX INVERTED NAND WITH SCHMITT TRIGGER INPUTS

<http://www.ATMEL.com>

ATMEGA32 , **ATMEGA16**

<http://www.Aoutonics.com>

ROTARY OPTICAL ENCODER /360/12-24/5

منابع و مراجع

- [1] A. Visioli, G. Legnani, "*on the Trajectory Tracking Control of Industrial SCARA Robot Manipulators*", IEEE Transactions on Industrial Electronics, Vol.49 ,No.1, February 2002 .
- [2] J. J. Craig, "*Adaptive Control of Mechanical Manipulators*", Addison –Wesely Publishing Co, Reading, MA, 1988.
- [3] M. Nakamura , S.R. Munasingle, S. Coto, N. Kyura, "*Enhanced Contour Control of SCARA Robot under Torque Saturation Constraint*", IEEE/ASME Transactions on mechatronics, Vol .5, No.4, December 2000 .
- [4] <http://www.Adept.com>, "Adept Cobra i800"
- [5] C. Dennis, O. Michael, "*Building Robot Drive Trains*" , McGraw-Hill, 1993.
- [6] <http://www.Mathwork/Rs232blockset/help.is?.ink>,
- [7] R. Agarwala, B.E, "*control of Robot integrated DMAW*", Bangalore University Chairman of Advisory Committee, July 1999.
- [8] K. J. Astrom, B. Wittenmark "*Adaptive Control* ", 2nd ed . Translated by M. T. H Beheshti 1998.
- [9] U. Gonzalez, Q. Li, "*Robotics : control, sensing, vision and Intelligence*", McGraw-Hill, international editions 1996.
- [10] C.H. Suh, C.W. Radcliffe, "*Kinematics and Mechanism Design*" , Wiley, New York, 1987.
- [11] C.Y. Kuo, S.P.T. Wang, "*Robust Position Control of Robotic Manipulator in Cartesian Coordinates*", IEEE Trans on Robotic and Automation, Vol. 7, No. 5, 1991.

[12] M. Nakamura, S.R. Munasinghe, S. Goto, N. Kyura ,” *Enhanced Contour Control of SCARA Robot Under Torque Saturation Constraint*”, IEEE/ASME Transaction on Mechatronics, VOL. 5, NO. 4, December 2000.

[13] H. Elloumi, M. Bordier, N. Maïzi, ” *An Encompassing Formalization Of Robust Computed Torque Schemes of Robot Systems*”, Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, December 12-15, 2005.

[14] M. Liu, ” *Computed Torque Control Scheme Based Adaptive Tracking for Robot Manipulators*”, IEEE International Conference on Robotics and Automation -0-7803-1965-6/95 \$4.00 © IEEE 1995.

[15] M. Uebel, I. Minis, K. Cleary, ” *Improved Computed Torque for Industrial Robots*”, Proceeding of the 1992 IEEE International Conference on Robotics And Automation nice, France-May 2002

[16] W. Khalil, F. Boyer, ” *An Efficient Calculation of Computed torque Control of Flexible Manipulators*”, IEEE International Conference on Robotics and Automation , 609-614, 1995 .

[17] T.D. Looke, M.M. Bayoumi, M. Farooq ,” *Simulation of Computed Torque Controllers for Flexible Manipulators*” , 91-645128/92 \$03.00 © IEEE, 1992.

[18] R.C. Dorf, ” *Modern Control System*”, Addison-Wesley, 1980.

[19] E. I. Rivin, ” *Mechanical Design of Robots*”, McGraw-Hill, St. Louis, 1988

[20] M. Uebel, I. Minis, K. Cleary, ” *Improved Computed Torque Control for Industrial Robot*” , Proceedings of the IEEE International Conference Robotics And Automation, Nice, France –May 1992

[21] S. Tzafestas , G. Stavrakakis, L. Dritsas, ” *Performance Analysis of the computed Torque Control Method For Robotic Manipulators with Fast Actuators*”, National Technical University of Athens, Zografou, Athens, Greece. 1998

[22] Jong Hyeon Park and Hoam Chung, ” *Hybrid Control for Biped Robots Using Impedance Control and Computed-Torque Control* ”, Proceedings IEEE, International Conference on Robotics & Automation Detroit, Michigan May 1999.

[23] A. Karim, D. Shusterman, M. Thompson, G.R. Widmann, "Input Signal Analysis for an Adaptive Computed Torque Control Scheme Applied To a 3-DOF Cylindrical Robot", 89CH2785-4/90/0000-0621\$01.00 © IEEE 1990.

[24] Middleton, G.C. Goodwin, "Adaptive computed torque control for rigid link manipulations" Systems and Control Letters 109-16 North-Holland 1988.

[25] C.H. An, C.G. Atkeson, J.D. Griffiths, J.M. Hollerbach, "Experimental Evaluation of Feedforward and Computed Torque Control", IEEE Transactions on Robotics and Automation, Vol.5, No 3, June 1989.

[26] Q. Li, A.N. POO, M. ANG, "An Enhanced Computed-Torque Control Scheme for Robot Manipulators with a Neuro-Compensator", 0-7803-2557-1/95 \$4.80 IEEE 1995.

[27] M. W. Spong, M. Vidyasagar, "Robot Dynamics and Control", John Wiley and Sons, Inc, New York, 1988.

[28] S. Arimoto, F. Miyazaki, "Stability and Robustness of P.I.D Feedback Control for Robot Manipulators of Sensory Capability," First Int Symp, Robotics Res, 1983.

[29] M.C. Good, L.M. Sweet, K.L. Strobel, "Dynamics Models for Control System Design of Integrated Robot and Drive System," ASME J. Dyn. Sys, Meas, and Cont, Vol.107, Mar 1985.

[30] A. Karim, G.K.F. Lee, "Model-Referenced Adaptive Computed Torque Control of a Cylindrical Robot Using Linear Compensation", Department of Electrical Engineering, Colorado State University, Fort Collins, CO 80523, U.S.A. 1997.

[31] S. Zuheir Tumei, "A Discrete Decentralized Hybrid Computed Torque-Model Referenced Adaptive Controller for Robotic Manipulators", 0-8186-2012-9/89/0000/044\$01.00 © IEEE 1989.

[32] T.W. Martin, E. Yaz, "Robot Manipulator Control Using Adaptive Computed Torque Technique", IEEE Conference Robotics and Automation, May 2002.

[33] P. Khosla, T. Kanade, "Parameter Identification Of Robot Dynamics", IEEE conf. on Decision and Control, Ft. Lauderdale, 1985.

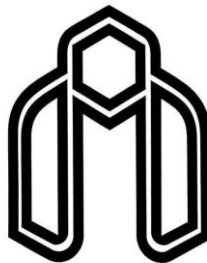
[34] F.D. Alonge, F.M. Raimondi, " *An adaptive control law for robotic manipulator without velocity feedback*" Dipartimento di Ingegneria dell'Automazione e dei Sistemi, University of Palermo, viale delle Scienze, 90128 Palermo, Italy Received 2 August 2001; accepted 12 October- 2002.

Abstract

Now a day the most scientific papers that published in authentic journals based on practical experiment of control theories on mechanical systems. In order to one of the current nonlinear mechanical system that we can implement control theory on it, is robot manipulator, furthermore accomplishment of this project has been necessary in Shahrood University of Technology to establish a good headstock for student of electronics, robotics and control to designing, constructing and control of same robots, so we decided to manufacture the SCARA robot and after built of its frame, design of electronic circuit and getting in touch with computer, applying our computed torque control method on it.

We hope to implement of this project could called every one "we can do"

Key words: SCARA robot, Design and manufacture of robot, PC interfacing, Position control, Computed torque control, Adaptive control



**SHAHROOD UNIVERSITY
OF TECHNOLOGY**

Department of Electrical and Robotic

Thesis submitted for

degree of Master of Science (M.sc) in Control Engineering

Ttitle:

**The Design and Constructing of a SCARA Robot Using Computed
Torque Control for pick and place**

Supervisor:

Dr. M.M. Fateh

Student:

Majid Sehati Yazdi

September 2007