



دانشگاه صنعتی شریف
دانشکده مهندسی مکانیک

پایان نامه کارشناسی ارشد
گرایش طراحی کاربردی

طراحی ربات فیلم برداری از اجسام متحرک

نگارش: احمد کلاهی دمیرچی درسی

استاد راهنما: دکتر حسن ظهور

آذر ۱۳۸۴

چکیده

در این پایان نامه رباتی تحت عنوان CamCrane برای فیلمبرداری از اجسام متحرک سریع در دامنه‌ای وسیع معرفی و طراحی گردیده است. ربات مذکور در واقع توسعه یافته مکانیزم‌های نیمه خودکار Camera Crane است، که در حال حاضر کاربرد وسیعی در صنعت فیلمبرداری دارد. مکانیزم‌های مورد استفاده بدلیل اینکه فاقد سیستم کنترلی و حتی فاقد نیرو محرکه داخلی هستند، و اغلب توسط خود فیلمبردار کنترل و هدایت می‌شود، دارای عملکرد نسبتاً ضعیفی مخصوصاً در فیلمبرداری از اجسام متحرک سریع می‌باشد. به همین منظور در این پایان نامه رباتی تحت عنوان CamCrane معرفی گردیده است، که بطور خودکار جسم متحرک مورد نظر را تشخیص داده و آن را دنبال می‌کند. هدف از این پایان نامه طراحی کامل ربات CamCrane می‌باشد که شامل سه قسمت اصلی طراحی مکانیزم، طراحی سیستم پردازش تصویر و پیاده سازی کنترلر می‌باشد. در طراحی مکانیزم سعی گردیده که مکانیزم‌های فعلی مورد استفاده بهینه گردد. کاهش وزن، افزایش سرعت عملکرد و طراحی عملگرهای مناسب برای حرکت درآوردن سریع ربات از جمله کارهای انجام شده بر روی مکانیزم بوده است. سیستم پردازش تصویر به منظور تشخیص جسم متحرک تحت تعقیب، طراحی و برنامه نویسی شده است. طراحی این سیستم شامل انتخاب قطعات سخت افزاری و پیاده سازی نرم افزار می‌باشد، که برای کاربرد ربات CamCrane تطبیق داده شده است. و در نهایت به پیاده سازی نرم افزار کنترلی و طراحی سخت افزار کنترلی پرداخته شده است. در شبیه سازی نرم افزار کنترلی برای کسب اطمینان از عملکرد آن از ایده بدیع لینک دو نرم افزار مدل ساز دینامیکی و طراح کنترلر استفاده شده و در طراحی سخت افزاری برای پیاده سازی کنترلر نیز از سخت افزارهای نوین تحت عنوان xPC استفاده گردیده است. به منظور تکمیل طراحی، مدلسازیها و شبیه سازیهای متعددی در طول پایان نامه انجام شده است.

کلمات کلیدی: CamCrane، xPC، مکانیزم، پردازش تصویر، مدلسازی، شبیه‌سازی، سیمولینک

فصل ۱ مقدمه	۱
۱-۱- آشنایی با Camera Crane و کاربردهای آن	۲
۱-۲- مروری بر کارهای انجام شده قبلی	۴
۱-۳- تعریف پروژه	۵
۱-۴- رئوس مطالب پایان نامه	۹
۱-۵- کارهای انجام گرفته در این پروژه	۱۰
فصل ۲ طراحی مکانیزم ربات	۱۲
۲-۱- مقدمه	۱۳
۲-۲- مشخصات فنی مکانیزم مورد نیاز	۱۳
۲-۳- طراحی ساختار کلی مکانیزم ربات	۱۴
۲-۳-۱- طراحی واگن و ریل	۱۴
۲-۳-۲- طراحی بازوی تلسکوپی	۱۹
۲-۳-۳- طراحی مفاصل لولایی عمود بر هم بازوی ربات	۲۲
۲-۳-۴- طراحی مفاصل لولایی عمود بر هم دوربین	۲۳
۲-۴- مدلسازی سه بعدی مکانیزم ربات	۲۴
۲-۵- تحلیل مکانیزم ربات	۲۵
۲-۵-۱- تحلیل‌های استاتیکی مکانیزم ربات	۲۶
۲-۵-۲- تحلیل‌های دینامیکی مکانیزم ربات	۲۸
فصل ۳ پردازش تصویر و بدست آوردن مختصات سه بعدی	۳۲
۳-۱- مقدمه	۳۳
۳-۲- طراحی کلی سیستم سخت‌افزاری پردازش تصویر	۳۴

۳۴	۱-۲-۳- دوربینهای تصویربرداری و کارت واسطه
۳۶	۲-۲-۳- کامپیوترهای پردازش تصویر
۳۶	۳-۲-۳- چارچوب کالیبراسیون
۳۷	۴-۲-۳- همزمان ساز دوربینها (سنکرونایزر)
۳۸	۳-۳- طراحی سیستم نرم افزاری پردازش تصویر
۳۹	۱-۳-۳- نرم افزار دریافت تصویر
۴۱	۲-۳-۳- طراحی و کدنویسی برنامه استخراج فریمهای فشرده شده از تصاویر (Deinterlacing)
۴۶	۳-۳-۳- کدنویسی برنامه پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ
۵۵	۴-۳- تئوری بازسازی مختصات سه بعدی
۵۵	۱-۴-۳- تکنیکهای بازسازی
۵۵	۱-۱-۴-۳- تکنیکهای بازسازی سنتی
۵۸	۲-۱-۴-۳- تکنیکهای بازسازی مجازی
۵۹	۳-۱-۴-۳- روش تبدیل خطی یا DLT
۶۶	۴-۱-۴-۳- موقعیت دوربینها و مختصات نقطه اصلی
۶۷	۵-۱-۴-۳- روش DLT بهبود یافته
۶۸	۵-۳- طراحی و کدنویسی نرم افزاری تبدیل مختصات سه بعدی
۷۰	۱-۵-۳- دریافت فایل های مختصات موقعیت توپ بر حسب پیکسل
۷۱	۲-۵-۳- انتخاب روش هموارسازی و بازسازی
۷۲	۳-۵-۳- دریافت خروجی ها
۷۳	۶-۳- نمونه ای از اجرای نرم افزار تهیه شده
۷۷	فصل ۴ تئوری کنترل پیش بین
۷۸	۱-۴- مقدمه
۷۹	۲-۴- استراتژی کلی کنترل پیش بین (MPC)

۸۱	۴-۲-۱- مدل فرآیند.....
۸۵	۴-۲-۲- مدل اغتشاش.....
۸۷	۴-۲-۳- بهینه سازی و تابع هدف.....
۹۰	۴-۳- مزایای کنترل پیش بین MPC.....
۹۲	۴-۴- مشکلات کنترل پیش بین MPC.....
۹۳	۴-۵- پیاده سازی کنترل پیش بین با مطلب.....
۹۵	فصل ۵ پیاده سازی کنترلر پیش بین برای ربات CamCrane.....
۹۶	۵-۱- مقدمه.....
۹۷	۵-۲- ایجاد ارتباط بین مطلب و نسترن.....
۱۰۰	۵-۳- مدلسازی کنترلر MPC برای ربات CamCrane در محیط سیمولینک.....
۱۰۴	۵-۴- شبیه سازی و بحث بر روی نتایج.....
۱۱۰	فصل ۶ طراحی سخت افزارهای کنترلی.....
۱۱۱	۶-۱- مقدمه.....
۱۱۲	۶-۲- ایجاد مدل Real time در محیط RTW برای سیستم کنترلی.....
۱۱۲	۶-۲-۱- معرفی RTW.....
۱۱۳	۶-۲-۲- ایجاد یک مدل Real Time.....
۱۱۳	۶-۲-۳- مراحل پروسه تولید کد.....
۱۱۶	۶-۲-۴- تولید S-function برای RTW.....
۱۱۷	۶-۲-۵- معرفی RTWin.....
۱۱۸	۶-۲-۶- ساختار RTWin.....
۱۲۰	۶-۲-۷- معرفی TLC.....
۱۲۱	۶-۲-۸- مفاهیم RTWin.....

۱۲۱.....	۹-۲-۶ RTWin محیط سخت افزاری
۱۲۳.....	۱۰-۲-۶ RTWin محیط نرم افزاری
۱۲۴.....	۳-۶ معرفی xPC Target و تولید کد قابل اجرا بر روی آن
۱۲۴.....	۱-۳-۶ معرفی xPC Target
۱۲۵.....	۲-۳-۶ ابزارهای مورد نیاز برای xPC Target
۱۲۷.....	۳-۳-۶ امکانات xPC Target
۱۲۹.....	۴-۳-۶ محیط سخت افزاری xPC Target
۱۳۲.....	۵-۳-۶ محیط نرم افزاری xPC Target
۱۳۳.....	۶-۳-۶ نحوه کنترل xPC Target
۱۳۵.....	۷-۳-۶ استفاده از شتابدهنده سیمولینک برای افزایش سرعت اجرایی مدل
۱۳۶.....	۸-۳-۶ استفاده از RTW برای تولید کد Real Time
۱۳۷.....	۹-۳-۶ استفاده از xPC Target برای تولید کد قابل اجرا بر روی سیستم
۱۳۸.....	فصل ۷ نتیجه گیری و پیشنهادات
۱۳۹.....	۱-۷ نتیجه گیری
۱۴۰.....	۲-۷ پیشنهادات
۱۴۲.....	مقاله مستخرج از پایان نامه
۱۵۴.....	مراجع
۱۵۸.....	پیوست
۱۵۹.....	پیوست ۱: سیمولینک و نحوه مدلسازی سیستمهای دینامیکی
۱۵۹.....	معرفی سیمولینک
۱۵۹.....	نحوه مدل سازی سیستمهای دینامیکی
۱۶۲.....	نحوه سیمولیشن سیستمهای دینامیکی

معرفی S-function.....	۱۶۷
پیوست ۲: مشخصات دوربین سرعت بالای Mikrotron مدل CAMMC1302.....	۱۶۸
پیوست ۳: مشخصات دوربین JVC مدل GR-DVL9800U.....	۱۷۱
پیوست ۴: جزئیات کدنویسی نرم افزار دریافت و پردازش تصویر.....	۱۷۳
پیوست ۵: جزئیات کدنویسی نرم افزار بازسازی مختصات سه بعدی.....	۱۸۴
پیوست ۶: معادلات دینامیکی ربات CamCrane به روش لاگرانژ.....	۱۸۹
پیوست ۷: مشخصات فنی موتور و گیربکس محرک واگن.....	۱۹۳
پیوست ۸: مشخصات فنی موتور و گیربکس محرک بازوی تلسکوپي.....	۱۹۸
پیوست ۹: مشخصات فنی موتور و گیربکس مفاصل لولایی عمود بر هم بازوی ربات.....	۲۰۱
پیوست ۱۰: مشخصات فنی موتور و گیربکس مفاصل لولایی عمود بر هم دوربین.....	۲۰۵

فهرست شکل‌ها

- شکل (۱-۱) نمونه‌ای از مکانیزم Camera Crane ۲
- شکل (۲-۱) نمونه‌ای از نحوه اتصال دوربین به بازو در مکانیزم Camera Crane ۳
- شکل (۳-۱) نمونه‌ای از مکانیزم Camera Crane با طول بازوی ۱۰۰ فوت ۳
- شکل (۴-۱) نمونه‌ای از کاربرد Camera Crane در فیلمبرداری از اجسام متحرک سریع ۴
- شکل (۱-۲) مکانیزم واگن و ریل ۱۵
- شکل (۲-۲) طرح ۳ بعدی سازه واگن ۱۶
- شکل (۳-۲) ابعاد و مشخصات فنی پروفیل راهنما ۱۶
- شکل (۴-۲) طرح ۳ بعدی رک سرتاسری ۱۷
- شکل (۵-۲) طرح ۳ بعدی پایه‌های ریل ۱۸
- شکل (۶-۲) طرح ارائه شده برای واگن ۱۸
- شکل (۷-۲) طرح بازوی تلسکوپی ۲۰
- شکل (۸-۲) طرح بازوی متحرک ۲۰
- شکل (۹-۲) طرح بازوی ثابت ۲۱
- شکل (۱۰-۲) طرح ۳ بعدی موتور و گیربکس محرک بازوی تلسکوپی ۲۱
- شکل (۱۱-۲) نمایی از مدل سه بعدی مکانیزم مفاصل لولایی عمود بر هم دوربین ۲۳
- شکل (۱۲-۲) مدل سه بعدی طرح اولیه مکانیزم ربات CamCrane ۲۵
- شکل (۱۳-۲) محیط انتخاب مشخصات جنس قطعات ۲۶
- شکل (۱۴-۲) نتایج تحلیل دینامیکی تنش‌های وان میز در بازوی تلسکوپی ۲۷
- شکل (۱۵-۲) نتایج تحلیل دینامیکی تنش‌های ایجاد شده در صفحه لولای بازوی تلسکوپی ۲۷
- شکل (۱۶-۲) نمودار شتاب زاویه‌ای بازو ۲۹
- شکل (۱۷-۲) نمودار سرعت زاویه‌ای بازو ۲۹
- شکل (۱۸-۲) نمودار سرعت خطی بازو ۳۰
- شکل (۱۹-۲) نمودار شتاب خطی بازو ۳۰

- شکل (۱-۳) تصویر گرفته شده با استفاده از قابلیت سرعت بالای دوربین (به ترتیب از راست به چپ: ۵۰ فریم در ثانیه افقی، ۵۰ فریم در ثانیه عمودی با رزولوشن یک دوم و ۱۰۰ فریم در ثانیه با رزولوشن یک چهارم) ۳۵
- شکل (۲-۳) چارچوب کالیبراسیون نصب شده در زمین اسکی ۳۷
- شکل (۳-۳) استفاده از نور فلاش برای تشخیص فریم‌های معادل حاصل از دو دوربین ۳۸
- شکل (۴-۳) رابط اصلی نرم‌افزار دریافت تصویر از دوربین ۴۰
- شکل (۵-۳) انتخاب تعداد فریم‌های مورد نظر برای اخذ از دوربین ۴۰
- شکل (۶-۳) انتخاب بازه فریم‌های مورد نظر برای ذخیره سازی ۴۱
- شکل (۷-۳) مدهای مختلف تصویربرداری با دوربین JVC مدل GR-DVL 9800 ۴۲
- شکل (۸-۳) فریم Interlace (راست) و Deinterlace شده (چپ) ۴۳
- شکل (۹-۳) نمای کلی رابط نرم‌افزار Deinterlacing ۴۴
- شکل (۱۰-۳) منوی باز کردن یک فایل avi ۴۵
- شکل (۱۱-۳) منوی انتخاب ناحیه مورد نظر برای اجرای عملیات Deinterlacing ۴۵
- شکل (۱۲-۳) منوی کرکره‌ای تعیین مد سرعت تصویربرداری ۴۶
- شکل (۱۳-۳) نمای کلی رابط نرم‌افزار پردازش تصویر ۴۷
- شکل (۱۴-۳) کادر محاوره‌ای ثبت نام توپ ۴۸
- شکل (۱۵-۳) محل توپ (راست) و دکمه‌های تنظیم محل آن (چپ) ۴۹
- شکل (۱۶-۳) کادر محاوره‌ای مربوط به آخرین فریم بازه جستجوی توپ ۴۹
- شکل (۱۷-۳) نوار پیشرفت عملیات پردازش تصویر ۵۰
- شکل (۱۸-۳) مسیر توپ حین حرکت ۵۰
- شکل (۱۹-۳) کادر محاوره‌ای مربوط به بازه فریم‌هایی که قرار است ذخیره شود ۵۰
- شکل (۲۰-۳) ذخیره سازی اطلاعات به صورت فایل Excel ۵۱
- شکل (۲۱-۳) گزینه‌های موجود در منوی فایل ۵۱
- شکل (۲۲-۳) حالات مختلف نمایش مسیر موقعیت توپ ۵۲
- شکل (۲۳-۳) پانل کالیبراسیون دوبعدی ۵۲
- شکل (۲۴-۳) گزینه‌های Centroid و Manual Check ۵۳
- شکل (۲۵-۳) تنظیمات کرانه پائین و بالای شدت نور تصویر ورودی و خروجی ۵۳

- شکل (۳-۲۶) تنظیمات کرانه پائین و بالای تصویر تک رنگ خاکستری..... ۵۳
- شکل (۳-۲۷) تنظیمات کرانه‌های پائین و بالای تصاویر RGB..... ۵۴
- شکل (۳-۲۸) ترتیب قرارگیری مجموعه stereo-photogrammetric در حالی که محورهای اپتیکی با یکدیگر موازیند. فاصله دو دوربین از یکدیگر $1/200\text{ m}$ و فاصله کانونی $0/300\text{ m}$ است. شرایط هم‌راستایی و هم‌صفحه بودن برآورده شده است... ۵۶
- شکل (۳-۲۹) نگاشت بین فضای کاری و فضای تصویر..... ۶۰
- شکل (۳-۳۰) نقاط اصلی و نقاط گرهای..... ۶۱
- شکل (۳-۳۱) توالی عملکرد برنامه Transform برای بارگذاری فایل‌های مختصات..... ۷۱
- شکل (۳-۳۲) انتخاب روش بازسازی و هموارسازی..... ۷۱
- شکل (۳-۳۳) دستگاه ژيروسکوپ..... ۷۳
- شکل (۳-۳۴) نمودار مسیر بازسازی شده مارکرها..... ۷۴
- شکل (۳-۳۵) تغییرات سرعت زاویه‌ای بر حسب زمان برای حرکت بخش دوار ژيروسکوپ (نتایج آزمایش ۴)..... ۷۵
- شکل (۴-۱) استراتژی MPC..... ۸۰
- شکل (۴-۲) ساختار کلی MPC..... ۸۱
- شکل (۴-۳) سیستم واقعی با در نظر گرفتن نویز و اغتشاش..... ۸۵
- شکل (۴-۴) مدل سیستم با در نظر گرفتن نویز و اغتشاش..... ۸۶
- شکل (۴-۵) تاثیر α بر ردیابی سیگنال..... ۸۹
- شکل (۴-۶) ساختار کنترل سلسله مراتبی در یک سیستم پیشرفته صنعتی..... ۹۰
- شکل (۴-۷) بلوک و ورودیها و خروجیهای کنترلر MPC (مرجع [14])..... ۹۳
- شکل (۵-۱) پارامترهای ورودی و خروجی بلوک vNPlantBlock..... ۹۸
- شکل (۵-۲) ورودیها و خروجیهای سیستم کنترل پیش بین..... ۱۰۰
- شکل (۵-۳) مدل طراحی شده در سیمولینک برای کنترل ربات..... ۱۰۳
- شکل (۵-۴) نمودار موقعیت زاویه‌ای دوربین..... ۱۰۵
- شکل (۵-۵) نمودار موقعیت خطی دوربین..... ۱۰۵
- شکل (۵-۶) نمودار سرعت زاویه ای دوربین..... ۱۰۶
- شکل (۵-۷) نمودار سرعت خطی دوربین..... ۱۰۶
- شکل (۵-۸) نمودار شتاب زاویه ای دوربین..... ۱۰۷

شکل (۵-۹) نمودار شتاب خطی دوربین.....	۱۰۷
شکل (۵-۱۰) نمودار موقعیت خطی واگن.....	۱۰۸
شکل (۵-۱۱) نمودار سرعت خطی واگن.....	۱۰۸
شکل (۵-۱۲) نمودار شتاب خطی واگن.....	۱۰۹
شکل (۶-۱) فرآیند تولید کد توسط RTW.....	۱۱۳
شکل (۶-۲) مراحل کنترل پروسه تولید اتوماتیک کد.....	۱۱۴
شکل (۶-۳) پروسه تولید کد توسط RTW.....	۱۱۵
شکل (۶-۴) ارتباط سریال RS 232.....	۱۳۰
شکل (۶-۵) شبکه کامپیوترهای host و Target طبق پروتکل TCP/IP.....	۱۳۱

فهرست جداول

جدول (۱-۱) محصولات شرکت Starda.....	۵
جدول (۱-۲) مشخصات فنی رک و شماره قطعه پیشنهادی برای سفارش.....	۱۷
جدول (۲-۲) مشخصات فنی موتور الکتریکی انتخابی ساخت شرکت Schlenker Enterprises.....	۱۹
جدول (۳-۲) مشخصات فنی موتور الکتریکی انتخابی ساخت شرکت Schlenker Enterprises.....	۲۲
جدول (۴-۲) جزئیات فنی موتور انتخابی برای اتصال لولایی دوربین ساخت Original Motor.....	۲۴
جدول (۱-۳) مقایسه سرعت زاویه ای واقعی و محاسبه شده در آزمایش ژيروسکوپ.....	۷۴

فصل ۱

مقدمه

۱-۱- آشنایی با Camera Crane و کاربردهای آن

Camera Crane مکانیزمی است که به فیلمبردار امکان مانور و حرکت دوربین را در محدوده‌ای وسیع فراهم می‌کند و عموماً در صنعت فیلم سازی و تصویر برداری از صحنه های ورزشی کاربرد دارد شکل (۱-۱) نمونه‌ای از این مکانیزم را نشان می‌دهد که برای فیلم برداری از مسابقه فوتبال مورد استفاده قرار می‌گیرد. طول بازوی این مکانیزم در حدود ۱۵ متر می‌باشد و دوربین در انتهای بازو قرار دارد نمونه‌ای از نحوه اتصال دوربین به بازو در شکل (۱-۲) نشان داده شده است نحوه اتصال بگونه‌ایست که به فیلمبردار امکان می‌دهد تا دوربین را در هر موقعیت زاویه‌ای دلخواه قرار دهد. مکانیزم نشان داده شده در شکل (۱-۱) پایه ثابت بوده و بر روی خودرو نصب می‌شود. اینکار علاوه بر تسهیل امر حمل و نقل مکانیزم از عامل وزن خودرو بعنوان تکیه گاهی برای ثابت نگه داشتن بازو استفاده می‌کند. مکانیزمهای کوچکتری با طول بازوی کمتر از ۵ متر وجود دارد که قابلیت سوار شدن بر روی واگن های متحرک را دارا می‌باشند، که این موضوع باعث اضافه شدن یک درجه آزادی انتقالی به مکانیزم می‌شود.



شکل (۱-۱) نمونه‌ای از مکانیزم Camera Crane



شکل (۲-۱) نمونه‌ای از نحوه اتصال دوربین به بازو در مکانیزم Camera Crane

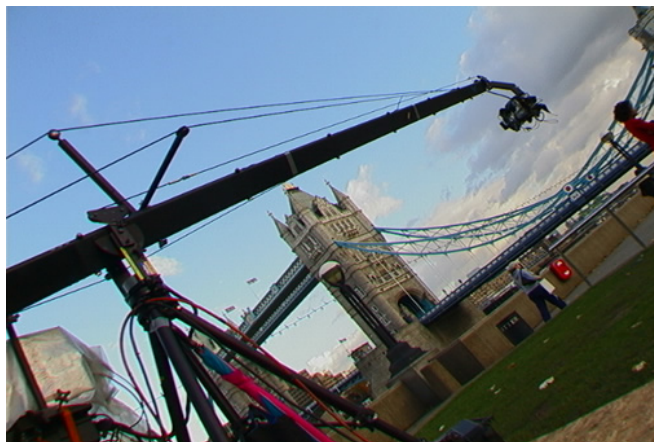
مکانیزمهای مورد استفاده برای Camera Crane چه از لحاظ وسعت ناحیه تحت پوشش و چه از لحاظ پیچیدگی و نحوه کنترل آنها بسیار متنوع می‌باشند. شکل (۳-۱) مکانیزمی را نشان می‌دهد که طول بازوی آن ۱۰۰ فوت می‌باشد و در صنعت سینما مورد استفاده قرار می‌گیرد. کلیه حرکات دوربین توسط اپراتور بصورت دستی کنترل می‌شود و سرعت حرکات دوربین معمولاً کند می‌باشد به همین خاطر نیازی به پیاده سازی سیستم های کنترل اتوماتیک برای تعقیب یک شی مورد نظر احساس نمی‌شود.



شکل (۳-۱) نمونه‌ای از مکانیزم Camera Crane با طول بازوی ۱۰۰ فوت

شکل (۴-۱) نمونه‌ای از کاربرد Camera Crane در فیلمبرداری از اجسام متحرک سریع را نشان می‌دهد. این مکانیزمها برای فیلمبرداری از مسابقات فوتبال تدارک دیده شده است و معمولاً در پشت دروازه

ها نصب می‌شود و برای فیلمبرداری از مسیر توپهایی که بسمت دروازه‌ها شلیک می‌شود مورد استفاده قرار می‌گیرد.



شکل (۱-۴) نمونه‌ای از کاربرد Camera Crane در فیلمبرداری از اجسام متحرک سریع

کل فرآیند فیلمبرداری از هر صحنه در اینگونه کاربردها معمولاً کمتر از چند ثانیه بطور می‌انجامد و بخاطر سرعت فوق العاده زیاد توپ انتظار می‌رود، که مکانیزم بتواند بسرعت خود را در پوزیشنهای مناسب قرار دهد این مکانیزمها بگونه‌ای طراحی شده‌اند که می‌توانند شرایط کاری تحت سرعتها و شتابهای بالا را تحمل کنند. در حال حاضر اکثر مکانیزمهایی که برای این منظور استفاده می‌شوند بصورت دستی و توسط فیلمبردار ماهر کنترل می‌شوند و اغلب صرفاً یک مکانیزم هستند که حتی نیروی مورد نیاز برای جابجایی دوربین را هم فیلمبردار تامین می‌کند.

۱-۲- مورو بر کارهای انجام شده قبلی

شرکتهای متعددی در سراسر دنیا در زمینه تولید Camera Crane فعالیت دارند که در این میان آمریکا و آلمان بیشترین سهم را دارند جدول زیر مشخصات تعدادی از محصولات شرکت Strada را نشان می‌دهد. که معمولاً برای محدوده کاری وسیع و سرعت پائین مورد استفاده قرار می‌گیرد طول بازوهای آن از ۵۷ تا ۱۱۲ فوت و حداکثر وزن دوربین قابل حمل از ۸۵ تا ۳۰۰ پوند می‌باشد که مقادیر بسیار بالایی

می‌باشد. جدول (۱-۱) محصولات این شرکت را نشان می‌دهد که محدوده کاری نسبتاً کمتری دارند ولی صلبیت مکانیزم آن معرف سرعت عمل بالای آنها می‌باشد.

جدول (۱-۱) محصولات شرکت Starda

Strada Camera Crane Specifications		Arm Reach	Lens Height With Dolly Risers	Lens Height Without Dolly Risers	Overall Arm Length	Max. Nose Load (Includes Camera & Remote Head)
100' Weight (with main chassis & weights).....	8500 lbs.					
85' Weight (with main chassis & weights).....	8000 lbs.					
72' Weight (with main chassis & weights).....	7500 lbs.					
Main chassis with steering.....	7'6" x 10'6"	100'	100'	75'	112'	85 lbs.
Pedestal to rear.....	12'	85'	85'	78'	97'	100 lbs.
Minimum operating space from pedestal placement.....	15'	79'	79'	73'	91'	125 lbs.
100' weight (with portable chassis & weights).....	6000 lbs.	72'	72'	68'	84'	200 lbs.
85' weight (with portable chassis & weights).....	6000 lbs.	65'	65'	62'	77'	225 lbs.
72' weight (with portable chassis & weights).....	5500 lbs.	58'	58'	55'	70'	250 lbs.
Portable chassis.....	7'6" x 8'6"	52'	52'	48'	64'	275 lbs.
Crane Operators (provided).....	2	45'	45'	42'	57'	300 lbs.

بطور کلی اکثر این مکانیزمها بصورت دستی کنترل می‌شود و فاقد سیستم کنترل اتوماتیک هستند بطوریکه مکانیزمهای با برد کم حتی فاقد نیرو محرکه هستند و خود فیلمبردار نیروی مورد نیاز برای حرکت درآوردن دوربین را تامین می‌کند.

۱-۳- تعریف پروژه

هدف از این پروژه طراحی رباتی برای فیلم برداری از اجسام متحرک سریع بویژه برای تعقیب مسیر توپ فوتبال می‌باشد که می‌تواند در محدوده‌ای با ابعاد ۵×۱۰×۵ متر حرکت داشته باشد. و در واقع نسل جدید مکانیزمهای Camera Crane محسوب می‌شود و ما آن را ربات CamCrane می‌نامیم. طراحی این ربات شامل چهار بخش اصلی زیر می‌باشد.

۱- طراحی مکانیزم ربات: مکانیزمی شش درجه آزادی که تمام زوایای دید مورد نیاز دوربین

فیلمبرداری را تامین کرده و بتواند محدوده مشخص شده (ابعاد ۵×۱۰×۵ متر) را پوشش دهد.

مسائل اصلی درگیر با طراحی این مکانیزم بشرح زیر می‌باشد که در طراحی آن مد نظر قرار گرفته است.

الف) سازه بازوی ربات: در طراحی سازه بازو مسائلی چون بهینه سازی برای حداقل وزن سازه، مقاومت در مقابل نیروهای ضربه‌ای ناشی از شتابهای تند شونده و کند شونده، مسائل مربوط به ارتعاش و نوسان سازه در محدوده شرایط کاری، میزان خمش سازه در مقابل نیروهای وارده و مخصوصاً مودهای نرمال سازه در نظر گرفته شده است.

ب) مکانیزم تلسکوپی: این مکانیزم امکان حرکت شعاعی برای دوربین را فراهم می‌کند. که می‌تواند با حداکثر سرعت ۲ متر بر ثانیه باز یا بسته شود.

ج) مکانیزم حرکت افقی واگن روی ریل: این مکانیزم می‌تواند با حداکثر سرعت 4 m/s واگن و کلیه متعلقات نصب شده روی آن را جابجا کند. حداکثر شتاب این مکانیزم 4 m/s^2 و طول مسیر ریل گذاری شده 5 m می‌باشد.

د) اتصالات لولایی عمود بر هم: دو سری از این اتصالات مورد نیاز می‌باشد اولی برای اتصال بازوی ربات به واگن مورد استفاده قرار می‌گیرد که موتورهای نصب شده روی آن می‌توان شتاب 1 rad/s^2 و حداکثر سرعت 1 rad/s را تامین کنند. اتصالات لولایی دوم روی دوربین نصب می‌شود و امکان جهت‌گیری‌های دلخواه را به دوربین می‌دهد.

۲- پردازش تصویر و بدست آوردن مختصات سه بعدی توپ: هدف از این بخش تولید اطلاعات فیدبک از مختصات سه بعدی توپ برای سیستم کنترلی می‌باشد در بین انواع روشهای موجود برای بدست آوردن مختصات سه بعدی توپ در زمین فوتبال، روش پردازش تصویر مناسب شناخته شد [3,4,5,7] سایر روشها (مثل نصب سنسور روی توپ، روشهای الکترومغناطیسی و ...) دارای محدودیتهایی در قوانین بازی و نیز عدم دقت کافی هستند. روش پردازش تصویر هیچکدام از این معایب را ندارد تنها مشکل این روش نیاز به نرم افزار مطمئن و حرفه‌ای در این زمینه است که بسیار گران قیمت می‌باشد بطور کلی بخش پردازش تصویر و بدست آوردن مختصات سه بعدی توپ شامل دو قسمت مجزا می‌باشد.

الف – دریافت، پردازش و بدست آوردن مختصات دو بعدی توپ

ب – ترکیب اطلاعات مختصات دو بعدی همه دوربینهای نصب شده در ورزشگاه و بدست آوردن

اطلاعات مختصات سه بعدی توپ

در طرح پیشنهادی این پایان نامه دوربینهای متعددی در محلهای مناسبی از ورزشگاه نصب می شود که هر کدام می تواند محدوده مشخصی از زمین بازی را در دید خود داشته باشد. کارهای بهینه سازی متعددی در رابطه با پوزیشن و نحوه نصب آنها در مراجع ۱،۲،۳،۴ آمده است بگونه ای که در هر لحظه حداقل دو دوربین توپ را در تصویر داشته باشد که بتوان از روی تصاویر حاصله از آنها اطلاعات سه بعدی را استخراج نمود.

یکی از اساسی ترین پارامترهای تاثیرگذار در نرم افزار پردازش تصویر قابلیت اجرای Real time آن است که با توجه به حجم عملیات فوق العاده بالای الگوریتمهای پردازش تصویر رسیدن به این هدف بسیار مشکل می باشد. [۸، ۹ و ۱۰] سایر پارامترهایی که وجود آنها به قابلیت این نرم افزار می افزاید عبارتند از: فیلتر کردن نویزهای موجود در تصاویر (شرایط متنوع جوی مثل هوای بارانی یا مه گرفتگی) قابلیت هوش مصنوعی (برای تشخیص قسمت آفتابی و سایه ورزشگاه و حرکت تدریجی آنها و یا تغییرات در روشنایی ورزشگاه، وجود چند توپ در زمین و) استفاده از اطلاعات سایر دوربینها برای یافتن موقعیت توپ در تصویر و ...

۳ – نرم افزار سیستم کنترلی:

نرم افزار سیستم کنترلی اطلاعات سه بعدی مختصات توپ را بعنوان ورودی گرفته و دستورات کنترلی

برای عملگرها تولید می کند تا همواره تصویر توپ را در دوربین داشته باشد

کنترلر مورد استفاده برای این منظور کنترلر پیش بین هست که کاربردهای متنوعی در صنعت دارد.

بطور کلی ورودیهای سیستم کنترلی عبارتند از:

الف) مختصات سه بعدی توپ

ب) پارامترهای دلخواهی که توسط کارگردان یا فیلمبردار تعیین می‌شود و عمدتاً زاویه دید دوربین یا تنظیمات لنز و مسیر حرکت آن را تحت تاثیر قرار می‌دهد.

دستورات کنترلی تولید شده توسط سیستم کنترلی نیز بشرح زیر می‌باشد:

الف) عملگرهای واگن

ب) حرکات زاویه‌ای اتصالات لولایی عمود بر هم بازو

ج) عملگر مربوط به مکانیزم تلسکوپی

د) حرکات زاویه‌ای اتصالات لولایی عمود بر هم دوربین

ه) تنظیمات لنز دوربین

سیستم کنترلی مورد استفاده علاوه بر اینکه باید بصورت Real time قابل اجرا باشد باید حداقل ۲۰ دستور هر در ثانیه صادر کند تا بتواند نسبت به حرکات سریع توپ عکس العمل نشان دهد در ضمن قابلیت نصب و اجرا بر روی xPC target را نیز داشته باشد.

پیاده سازی و تست نرم افزار کنترلی در محیط سیمولینک نرم افزار مطلب و بکمک جعبه ابزار RTW انجام شده است جعبه ابزار RTW محیطی برای تست سیستمهای کنترلی بصورت Real time می‌باشد که بدون نیاز به سخت افزارهای جانبی این کار را انجام می‌دهد. و قابلیت تولید کدهای Real time و کدهای قابل اجرا بر روی سخت افزار را دارا می‌باشد [۸].

۴ - سخت‌افزار سیستم کنترلی

پروژه پردازش تصویر بصورت Real time به همراه نرم‌افزار کنترلی به محاسبات بسیار زیادی نیاز دارد که از عهده سخت‌افزارهای معمولی خارج است کامپیوترهای امروزی سرعت محاسباتی بسیار بالایی دارند ولی سیستم عاملهای متداول مثل ویندوز از کاربرد مورد نیاز ما فاصله زیادی دارد و جوابگوی نیاز ما نمی‌باشد [۹]. برای حل این مشکل نرم‌افزار مطلب کرنلی (سیستم عاملی بسیار خلاصه و بهینه) را تدارک دیده که علاوه بر فراهم ساختن سرعت محاسباتی بسیار بالا امکان تبادل اطلاعات با سخت‌افزارها را نیز آسان نموده است. این کرنل بر روی سخت‌افزاری تحت عنوان xPC target نصب و اجرا می‌شود [۱۰].

۴-۱- رئوس مطالب پایان‌نامه

فصول پایان‌نامه به ترتیب مراحل طراحی و فراهم ساختن پیش نیازهای فصول بعدی تدارک دیده شده است. بطوریکه هر فصل با مقدمه‌ای در رابطه با هدف فصل و ارائه خلاصه‌ای از مطالب پرداخته شده در آن شروع می‌شود. سپس مراحل مربوط به طراحی آن بطور کامل توضیح داده می‌شود و در نهایت نتایج شبیه‌سازی‌های انجام شده و یا نتیجه‌گیری ارائه می‌شود.

پس از ارائه مقدماتی در رابطه با پروژه و کاربردهای آن، در فصل دوم طراحی مکانیزم ربات مورد بحث قرار می‌گیرد. مشخصات فنی مکانیزم، طراحی قسمت‌های مختلف مثل واگن و ریل، بازوی تلسکوپی، مفاصل و انتخاب عملگرهای مکانیزم بطور کامل توضیح داده می‌شود مدلسازی سه بعدی مکانیزم در محیط نرم‌افزار Working Model 4D و تحلیل دینامیکی آن توسط نرم‌افزار MSC Visual Nastran در انتهای این فصل ارائه شده است.

در فصل سوم به بحث سخت‌افزاری و نرم‌افزاری پردازش تصویر و بدست آوردن مختصات سه بعدی پرداخته شده است پس از ارائه طرح سخت‌افزارهای مورد نیاز قسمت‌های مختلف نرم‌افزار پردازش تصویر شامل بخشهای دریافت تصویر، Deinterlacing، سنکرونیزاسیون، پردازش تصویر و بدست آوردن مختصات دو بعدی توپ و در نهایت بخش نرم‌افزار سه بعدی ساز بطور کامل توضیح داده شده است. توضیحات کامل

مربوط به کدهای نوشته شده و عناوین توابع و وظایف آنها در ضمیمه آمده است در انتهای این فصل نیز نمونه‌ای از عملیات پردازش تصویر و بدست آوردن مختصات سه بعدی ارائه شده است.

فصل چهارم به تئوری کنترل پیش بین پرداخته است. پس از ارائه مقدمه‌ای در رابطه با این نوع کنترلرها، تئوری کنترل پیش بین بصورت خلاصه توضیح داده شده است سپس مزایا و معایب و علت انتخاب این نوع کنترلر برای پیاده سازی سیستم کنترلی ربات توضیح داده شده است و در انتها توضیحاتی در رابطه با امکانات نرم افزار Matlab® برای پیاده سازی این نوع کنترلرها ارائه شده است.

پیاده سازی کنترلر پیش بین برای ربات طراحی شده در فصل پنجم توضیح داده می‌شود در ابتدای این فصل شیوه‌ای جدید برای مدلسازی دینامیکی و کنترل سیستمها بکمک دو نرم افزار معتبر Matlab® و MSC Visual Nastran ارائه شده است و نحوه لینک کردن این دو نرم افزار توضیح داده شده است سپس مقدمه‌ای در رابطه با سیمولینک و نحوه مدلسازی سیستمهای دینامیکی و امکانات نرم افزاری مورد نیاز برای پیاده سازی سیستم کنترلی ارائه شده است و در انتهای فصل نتایج شبیه سازی و بحث برروی نتایج آنها آمده است.

فصل ششم در رابطه با طراحی سخت افزارهای کنترلی می‌باشد این فصل با ارائه مقدمه‌ای در رابطه با مباحث نرم افزاری درگیر با تولید کدهای Real time (شامل ایجاد مدل Real time، مراحل پروسه تولید کد، معرفی RTW و ساختار آن و محیطهای سخت افزاری و نرم افزاری RTW) شروع می‌شود سپس مباحث مربوط به سخت افزار xPC target (شامل معرفی xPC، ابزارها و امکانات، محیط سخت افزاری و نرم افزاری RTW و نحوه کنترل آن، استفاده از شتاب دهنده برای بهینه سازی سرعت اجرایی مدل و نحوه استفاده از xPC target برای تولید کد قابل اجرا برروی سیستم) توضیح داده شده است.

۱-۵- کارهای انجام گرفته در این پروژه:

بطور کلی طرح ربات CamCrane طرحی جدید می‌باشد که در راستای تکامل مکانیزمهای Camera Crane ارائه می‌شود ولی با توجه به پیچیدگیهای طرح، تاکنون به مرحله ساخت کامل نرسیده است هدف ما

در این پایان‌نامه ارائه طرحی کامل برای ربات فیلم‌بردار از اجسام متحرک می‌باشد علاوه بر این نتایج شبیه‌سازیهای کامپیوتری نیز برای اطمینان از عملکرد صحیح طرح ارائه شده است و قسمت پردازش تصویر و نرم‌افزار کنترلی بطور کامل طراحی و پیاده‌سازی شده است و برای حالت‌های خاص مورد تست قرار گرفته است و قسمت‌های سخت‌افزاری پروژه شامل طراحی مکانیزم و سخت‌افزار کنترلی فقط در حد طراحی و ارائه شبیه‌سازیها ارائه شده است. بطور خلاصه اهم کارهای انجام گرفته در این پایان‌نامه به ترتیب فصول بشرح زیر می‌باشد.

ارائه طرحی بهینه برای مکانیزم ربات CamCrane و سیمولیشن کامپیوتری و تحلیل دینامیکی آن
برنامه‌نویسی نرم‌افزار پردازش تصویر و محاسبه مختصات سه بعدی جسم متحرک در محیط مطلب
(بیش از هفت هزار خط کد مطلب)

بررسی انواع کنترلرهای مناسب صنعتی و انتخاب و پیاده‌سازی کنترلر MPC برای ربات
ارائه شیوه‌ای برای کنترل سیستم‌های دینامیکی از طریق ارتباط بین نرم‌افزارهای مطلب و نستر
بررسی تکنولوژی جدید کنترل کامپیوتری با xPC Target و ارائه طرحی دقیق برای پیاده‌سازی
سیستم کنترلی بر روی xPC

فصل ۲

طراحی مکانیزم ربات

۲-۱- مقدمه

مکانیزم انتخاب شده برای CamCrane یک مکانیزم شش درجه آزادی است که می‌تواند دوربین فیلمبرداری را در تمام پوزیشن‌ها (فضای کاری این ربات ناحیه‌ای به وسعت $5 \times 10 \times 15$ متر می‌باشد) با هر موقعیت زاویه‌ای و جهت‌گیری دلخواه قرار دهد.

پس از بررسی انواع مکانیزم‌های موجود مکانیزمی با طرح کلی زیر مورد تأیید قرار گرفت.

(۱) کل مجموعه مکانیزم ربات بر روی واگنی نصب می‌شود که این واگن بر روی ریلی بطول 5 متر قرار دارد و امکان حرکات انتقالی را فراهم می‌سازد.

(۲) بازوی ربات مکانیزمی تلسکوپ‌می‌باشد که طول کلی آن 5 متر می‌باشد که سه متر آن بازوی اصلی و 2 متر نیز بازوی تلسکوپ‌می‌باشد و بر اساس دستورات کنترلی دوربین می‌تواند در فاصله شعاعی 3 تا 5 متر پوزیشن‌دهی شود.

(۳) دو اتصال لولایی عمود بر هم، دو درجه آزادی دورانی برای بازوی ربات فراهم می‌کند.

(۴) مشابهاً دو اتصال لولایی عمود بر هم دو درجه آزادی دورانی برای دوربین نصب شده در انتهای بازو را فراهم می‌کند.

۲-۲- مشخصات فنی مکانیزم مورد نیاز

بطور خلاصه مشخصات فنی مکانیزم مورد نیاز به شرح زیر می‌باشد.

مکانیزمی شش درجه آزادی

فضای کاری ربات ناحیه‌ای به وسعت $5 \times 10 \times 15$ متر

مقاومت در مقابل نیروهای ضربه‌ای ناشی از شتابهای تند شونده و کند شونده

حداکثر سرعت باز یا بسته شدن مکانیزم تلسکوپ‌می 2 m/s حداکثر شتاب آن 2 m/s^2

حداکثر سرعت مکانیزم حرکت افقی واگن روی ریل 4 m/s و حداکثر شتاب آن 4 m/s^2

طول مسیر ریل گذاری شده 5 m

حداکثر شتاب زاویه‌ای مکانیزم اتصالات لولایی عمود بر هم 1 rad/s^2 و حداکثر سرعت زاویه‌ای

آن 1 rad/s را تامین کنند.

۲-۳- طراحی ساختار کلی مکانیزم ربات

همانطور که در بالا توضیح داده شد ساختار کلی مکانیزم ربات از چهار بخش زیر تشکیل شده

است:

الف) طراحی واگن و ریل

ب) طراحی بازوی تلسکوپی

ج) طراحی مفاصل لولایی عمود بر هم بازوی تلسکوپی

د) طراحی مفاصل لولایی عمود بر هم دوربین

در ادامه به ارائه جزئیات طرح می‌پردازیم.

۲-۳-۱- طراحی واگن و ریل

واگن در واقع پلت فرم بازوی ربات محسوب می‌شود و تمام عملگرها و موتورهای محرک اصلی

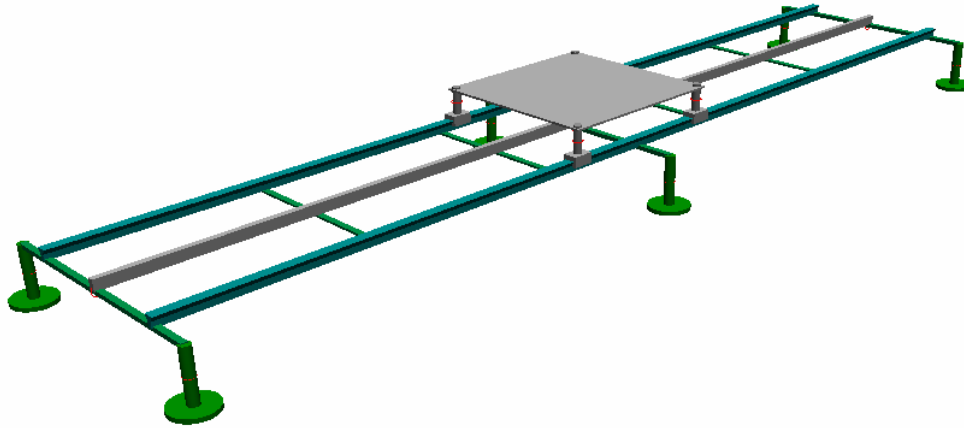
ربات بر روی آن قرار می‌گیرد مکانیزم مفاصل لولایی عمود بر هم بازو، به همراه سیستم محرک آن بر

روی واگن نصب می‌شود. برای حرکت درآوردن خود واگن نیز از یک موتور و گیربکس استفاده شده

است که واگن را روی ریل حرکت در می‌آورد. ریل دارای طول پنج متر بوده و از دو پروفیل راهنما

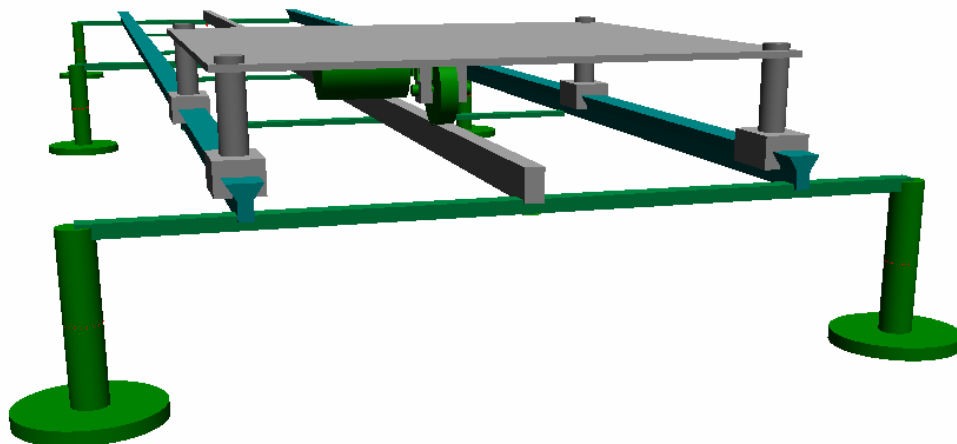
(برای نگه داشتن واگن) و یک رک (برای حرکت درآوردن واگن) تشکیل شده است. وزن کل واگن (به

همراه بازو، دوربین، عملگرها و موتورها) 120Kg می‌باشد و ابعاد آن 80 cm × 80 cm انتخاب شده است شکل (۱-۲) واگن را روی ریل نشان می‌دهد.



شکل (۱-۲) مکانیزم واگن و ریل

همانطور که مشاهده می‌شود پروفیل ریل‌های راهنما بگونه‌ای است که واگن را کاملاً مقید به حرکت در راستای ریل می‌کند و مانع هر گونه لرزش و حتی واژگون شدن واگن تحت شرایط خاص می‌شود. رک سرتاسری در بین ریل‌های راهنما با چرخنده‌ای در زیر واگن در تماس است که برای به حرکت درآوردن واگن در نظر گرفته شده است. نیروی مورد نیاز برای حرکت در آوردن این چرخنده توسط یک موتور DC به همراه گیربکس تأمین می‌شود. سازه واگن بگونه‌ای طراحی شده است که علاوه بر تأمین فضای مناسب برای نصب انواع مکانیزم‌های مورد نیاز روی آن از استحکام کافی و وزن سبک نیز برخوردار است. (شکل (۲-۲)) به همین ترتیب سازه نگهدارنده ریلها و رک نیز بگونه‌ای طراحی شده که صلبیت مورد نیاز را برای ریلها تأمین کند. همانطور که در شکل (۱-۲) مشاهده می‌کنید کل مجموعه ریلها روی شش پایه قرار گرفته است که بر روی این پایه‌ها وزنه‌هایی قرار داده می‌شود تا مانع حرکت و لرزش ریل شود.

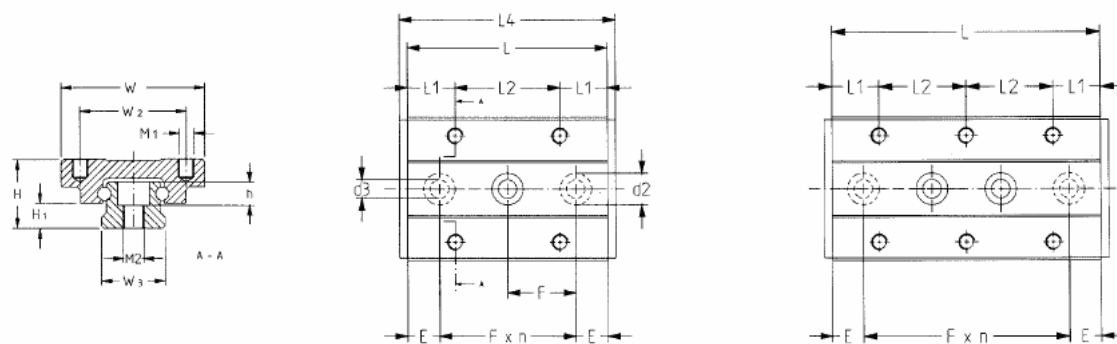


شکل (۲-۲) طرح ۳ بعدی سازه واگن

مشخصات فنی قطعات مورد استفاده در مجموعه ریل و واگن بشرح زیر می باشد:

۱- پروفیل راهنما بطول ۵ متر، دو عدد از جنس فولاد ضد زنگ با ابعاد و مشخصات فنی نشان

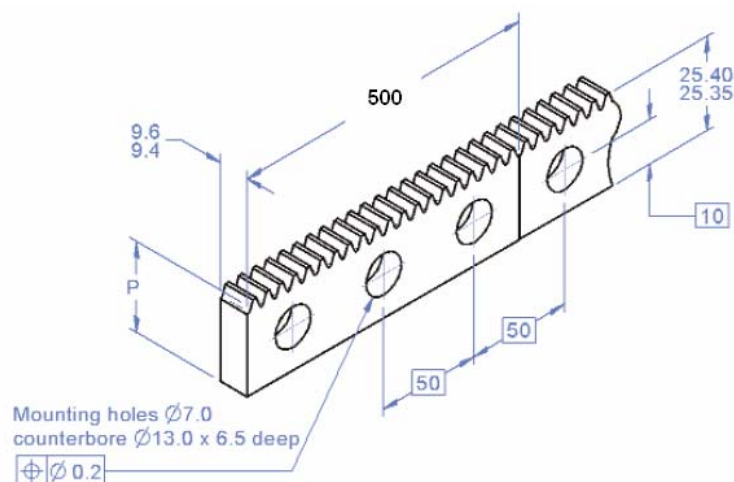
داده شده در شکل (۳-۲)



Type	W	W ₂	W ₃	L ₂	M1 x depth	d3 x d2 x h	H	H ₁	M ₂	F
mm										
LZM HS 7	68	48	28	32	M8 x 10	10 x 16 x 10	32	9	M10	60

شکل (۳-۲) ابعاد و مشخصات فنی پروفیل راهنما

۲- رک سرتاسری بطول 5 متر با طرح شکل (۲-۴) و مشخصات فنی ارائه شده در جدول (۲-۱)



شکل (۲-۴) طرح ۳ بعدی رک سرتاسری

جدول (۲-۱) مشخصات فنی رک و شماره قطعه پیشنهادی برای سفارش

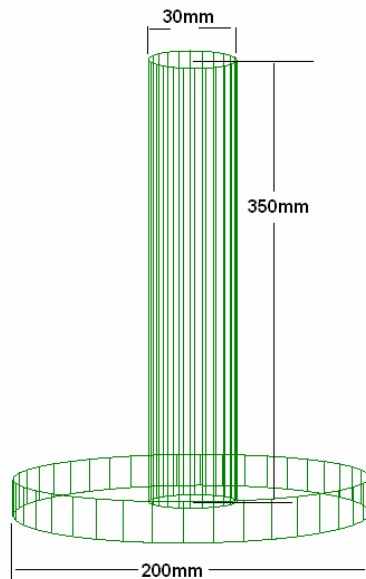
Part Number	Circular Pitch (mm)	Pitch Height (P)	Rack Thrust (N)
RH79-25M-300	2.5	24.601 24.576	130*

* Quoted rack thrust based on meshing with a 76 tooth pinion ($\phi 60.47\text{mm}$).

۳- تسمه‌هایی بطول 1.2m و با سطح مقطع $5 \times 1\text{cm}$ به تعداد 4 عدد

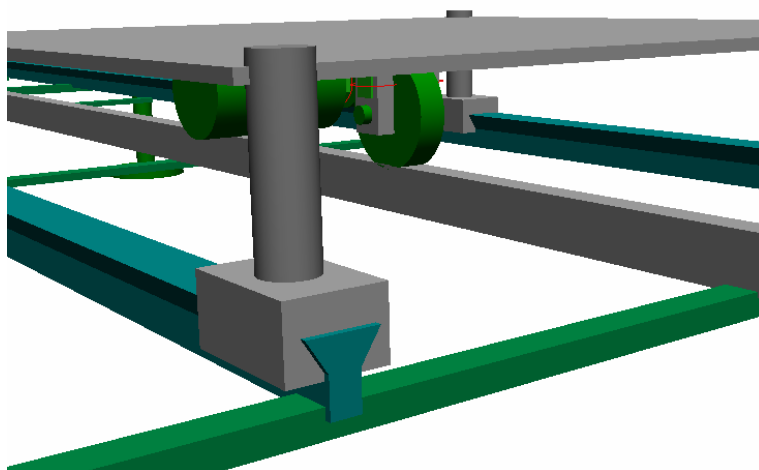
۴- تسمه‌هایی بطول 1.2m و با سطح مقطع $5 \times 2\text{cm}$ به تعداد سه عدد .

۵- پایه‌هایی با شکل هندسی زیر به تعداد شش عدد



شکل (۵-۲) طرح ۳ بعدی پایه‌های ریل

طرح ارائه شده برای واگن بصورت شکل (۶-۲) می‌باشد. واگن توسط چهار پایه روی ریل قرار می‌گیرد بر روی این چهارپایه صفحه‌ای قرار گرفته که وزن تمام سازه ربات بر آن اعمال می‌شود در زیر این صفحه موتور و گیربکس محرک واگن قرار دارد و در روی آن موتور گیربکس محرک بازوی ربات نصب شده است جزئیات طرح واگن بشرح زیر می‌باشد:



شکل (۶-۲) طرح ارائه شده برای واگن

۱- پایه‌های واگن: این پایه‌ها در شیار پروفیل راهنمای ریل قرار می‌گیرد و فقط امکان حرکت در راستای ریل را اجازه می‌دهد.

۲- صفحه واگن: ابعاد این صفحه 85×85 سانتی‌متر است و ضخامت آن 4mm می‌باشد جنس آن از آلایژ آلومینیم می‌باشد.

۳- موتور و گیربکس محرک واگن: کل اینرسی اعمال شده بر روی واگن 120Kg می‌باشد که برای اعمال شتاب $4m/s^2$ به نیروی 480N نیاز دارد با فرض سرعت ماکزیمم 4m/s توان موتور مورد نیاز 1920W خواهد بود از آنجا که این توان، حداکثر توان مورد نیاز است و فقط برای لحظاتی کوتاه اعمال می‌شود پس می‌توان از موتور الکتریکی DC با توان 1.5KW برای این منظور استفاده کرد. مشخصات فنی موتور الکتریکی انتخابی ساخت شرکت Schlenker Enterprises بشرح زیر می‌باشد.

جدول (۲-۲) مشخصات فنی موتور الکتریکی انتخابی ساخت شرکت Schlenker Enterprises

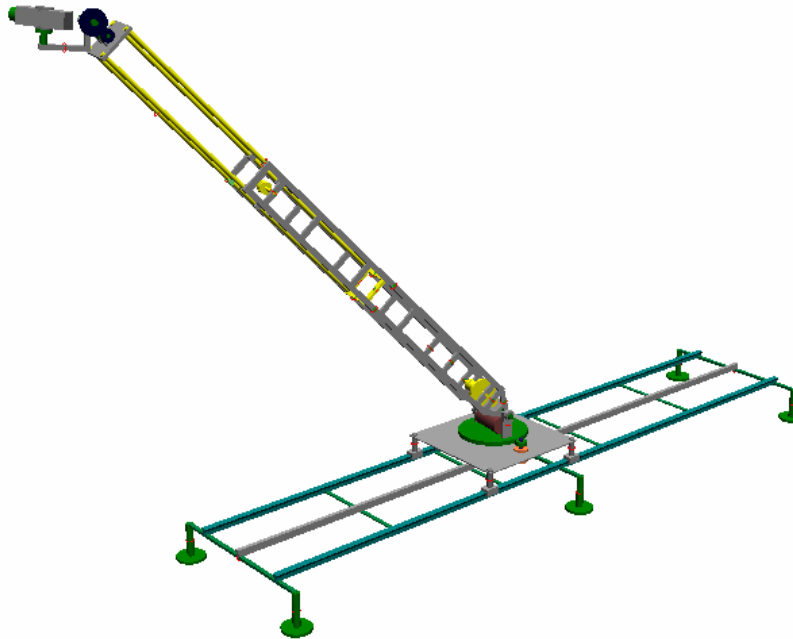
Reference Data			Current		Voltage	Speed	Kt	Frame	Mass
Motor Code	T100 Nm	Tpk Nm	Io Arms	Ipk Arms	Vrms	rad/s	Nm/Arms	mm	kg
508402	10.2	27.3	15.6	39.7	171	419	0.69	100	9.0

گیربکس مورد نیاز دارای نسبت تبدیل ۵ به ۱ می‌باشد که مشخصات فنی آن در ضمیمه ۷ آمده است.

۲-۳-۲- طراحی بازوی تلسکوپی

بازوی تلسکوپی مکانیزمی است که امکان پوزیشن‌دهی از فاصله شعاعی ۳ متر تا ۵ متر را برای دوربین فراهم می‌سازد این مکانیزم در شکل (۷-۲) نشان داده شده است. همانطور که در شکل مشاهده می‌کنید بازو از دو قسمت ثابت و متحرک تشکیل شده است و دوربین در انتهای قسمت متحرک نصب

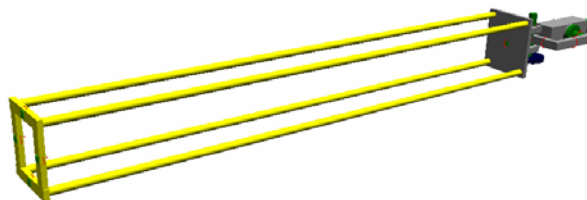
شده است برای به حرکت درآوردن قسمت متحرک از مجموعه موتور و گیربکس استفاده شده است که توسط کابلی می‌تواند بازوی متحرک را به سمت بیرون یا داخل حرکت دهد.



شکل (۷-۲) طرح بازوی تلسکوپی

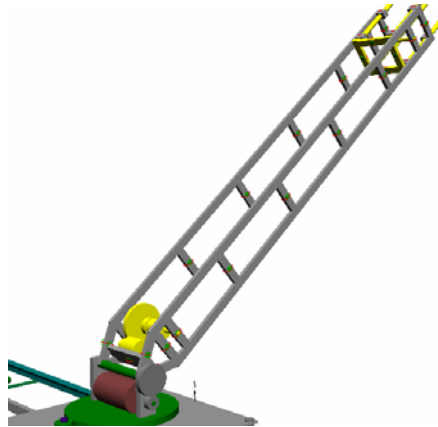
مشخصات فنی قسمت‌های مختلف بازوی تلسکوپی بشرح زیر می‌باشد.

۱- بازوی متحرک: از چهار میله به قطر خارجی 30 mm و قطر داخلی 26 mm و طول 250 cm تشکیل شده است (شکل (۸-۲)) که در داخل یاتاقانهای نصب شده روی انتهای بازوی ساکن قرار دارد. میله‌ها از جنس فولاد ضد زنگ آبکاری شده با کروم می‌باشد چهار میله در دو انتها به هم متصل می‌باشد.



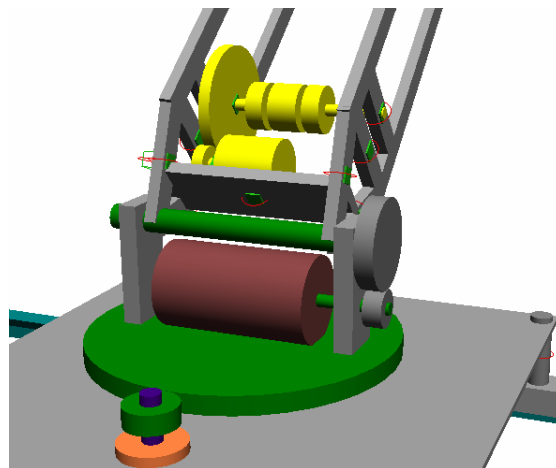
شکل (۸-۲) طرح بازوی متحرک

۲- بازوی ثابت: این بازو از یک سازه خرپایی تشکیل شده است طول آن سه متر و وزن آن 30 kg و از جنس فولاد ضد زنگ می باشد و بر روی مکانیزم لولاهای عمود بر هم روی واگن نصب شده است در نتیجه دارای دو درجه آزادی دورانی می باشد شکل (۹-۲) این سازه را نشان می دهد.



شکل (۹-۲) طرح بازوی ثابت

در طراحی این سازه حداکثر استحکام با حداقل وزن مد نظر بوده است. موتور و گیربکس محرک بازوی تلسکوپی نیز مطابق شکل (۲-۱۰) روی این سازه نصب می باشد که توسط دو رشته کابل (برای اعمال نیروی متقارن) به قسمت تلسکوپی متصل می باشد شکل زیر جزئیات بیشتر مکانیزم محرک بازوی تلسکوپی را نشان می دهد. جزئیات فنی قطعات انتخابی در ضمیمه ۸ آمده است.



شکل (۲-۱۰) طرح ۳ بعدی موتور و گیربکس محرک بازوی تلسکوپی

۲-۳-۳- طراحی مفاصل لولایی عمود بر هم بازوی ربات

شکل (۲-۱۰) تصویر سه بعدی از این مکانیزم را نشان می‌دهد همانطور که مشاهده می‌شود این مفاصل دو درجه آزادی دورانی افقی و عمودی را برای بازو فراهم می‌سازد مکانیزم حرکت دورانی قائم روی صفحه واگن نصب شده است که از دیسکی به قطر 50cm تشکیل شده و توسط یک موتور DC به حرکت در می‌آید مشخصات فنی موتور و گیربکس محرک این اتصالات به همراه جزئیات یاتاقان لغزشی کفی در ضمیمه ۹ آمده است.

با توجه به اینکه حداکثر شتاب زاویه‌ای بازو 1 rad/s^2 و حداکثر سرعت زاویه‌ای بازو rad/s است و با فرض توزیع یکنواخت وزن بازو در سرتاسر آن توان و مشخصات موتور DC مورد نیاز برای تأمین حرکت زاویه‌ای قائم بصورت جدول (۲-۳) می‌باشد.

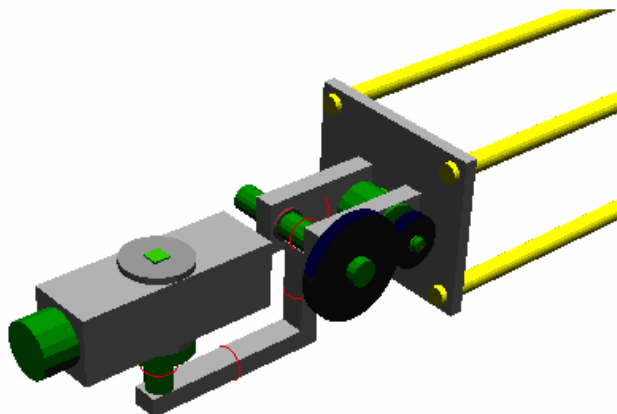
جدول (۲-۳) مشخصات فنی موتور الکتریکی انتخابی ساخت شرکت Schlenker Enterprises

Reference Data			Current		Voltage	Speed	Kt	Frame	Mass
Motor Code	T100 Nm	Tpk Nm	Io Arms	Ipk Arms	Vrms	rad/s	Nm/Arms	mm	kg
405302	5.0	13.5	5.5	14.1	183	314	0.96	85	4.0

مکانیزم حرکت دورانی افقی روی دیسک دوار قرار گرفته است همانطور که در شکل (۲-۱۰) مشاهده می‌کنید این مکانیزم از یک مجموعه موتور گیربکس، دو عدد چرخنده و یک شفت تشکیل شده است که کل سازه بازوی ربات روی این شفت نصب شده است توسط یاتاقانهای لغزشی از دو طرف مهار شده است مشخصات فنی موتور و گیربکس در ضمیمه ۹ آمده است. با توجه به فرضیات انجام شده بر روی حداکثر سرعت و شتاب زاویه‌ای بازو توان موتور DC مورد نیاز 800W برآورد می‌شود.

۲-۳-۴- طراحی مفاصل لولایی عمود بر هم دوربین

این مفاصل به منظور تأمین دو درجه آزادی دورانی برای دوربین طراحی شده است، که بر روی صفحه انتهایی بازوی تلسکوپی نصب شده است شکل (۱۱-۲) نمایی از مدل سه بعدی این مکانیزم را نشان می‌دهد.



شکل (۱۱-۲) نمایی از مدل سه بعدی مکانیزم مفاصل لولایی عمود بر هم دوربین

همانطور که در شکل مشاهده می‌شود مجموعه موتور و گیربکس نصب شده بر روی پلنت انتهایی بازوی تلسکوپی، حرکت دورانی افقی را برای دوربین فراهم می‌سازد. با توجه به اندازه نیروی وزن دوربین و بازوی نگهدارنده آن توان موتور محرک این مجموعه و مشخصات فنی آن مطابق ضمیمه ۱۰ می‌باشد.

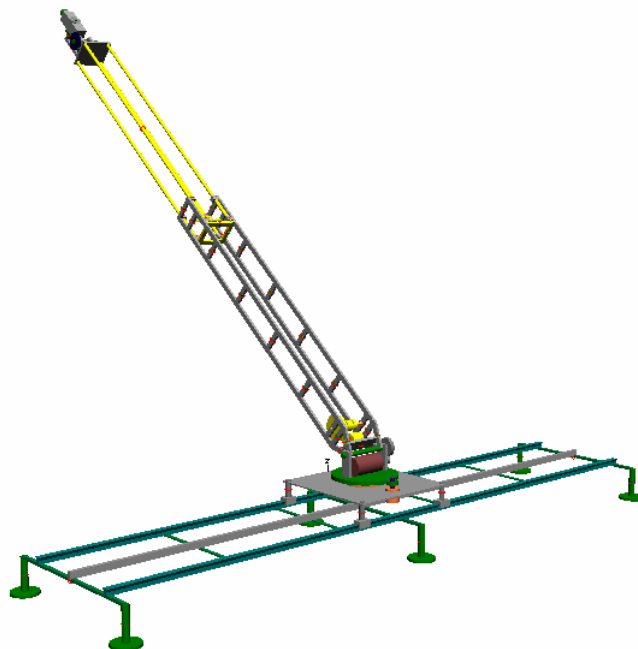
حرکت دورانی قائم بر دوربین نیز توسط یک مجموعه موتور گیربکس تأمین می‌شود. که بر روی بازوی نگهدارنده دوربین نصب شده است با توجه به اینکه تنها نیروی وارده به این مجموعه، نیروی وزن دوربین می‌باشد و حداکثر سرعت و شتاب زاویه‌ای مورد نیاز به ترتیب 5 rad/s و 2 rad/s^2 می‌باشد توان و جزئیات فنی موتور انتخابی که از نوع Hollow Rotary Actuator است، بشرح جدول (۴-۲) می‌باشد.

جدول (۴-۲) جزئیات فنی موتور انتخابی برای اتصال لولایی دوربین ساخت Original Motor

Motor Type		ALSTEP			
Type of Output Table Bearing		Deep-Groove Ball Bearings		Cross-Roller Bearings	
Permissible Torque ^{*2}	N·m lb·in	0.9 7.9		2.8 24	12 106
Maximum Static Torque ^{*3}	N·m lb·in	0.45 3.9		1.8 15.9	12 106
Inertial Moment ^{*4}	kg·m ² oz·in ²	4324 × 10 ⁻⁷ 24		2534 × 10 ⁻⁶ 139	15874 × 10 ⁻⁶ 870
Permissible Speed	r/min	200			
Gear Ratio		18 : 1			
Resolution ^{*5}		9000 P/R (Resolution Setting: 0.04°/pulse [500] [×1]) 18000 P/R (Resolution Setting: 0.02°/pulse [1000] [×1]) 90000 P/R (Resolution Setting: 0.004°/pulse [500] [×10]) 180000 P/R (Resolution Setting: 0.002°/pulse [1000] [×10])			
Repetitive Positioning Accuracy	sec	±15 (±0.004°)			
Lost Motion	min	2 (0.033°)			
Angle Transfer Error	min	4 (0.067°)			
Permissible Thrust Load	N lb	100 22		500 112	2000 450
Permissible Moment Load	N·m lb·in	2 17.7		10 88	50 440
Runout of Output Table Surface	mm in.	0.03 0.0012		0.015 0.0006	
Runout of Output Table Inner (Outer) Diameter	mm in.	0.03 0.0012		0.015 0.0006	
Parallelism of Output Table	mm in.	0.05 0.002		0.03 0.0012	
Degree of Protection		IP40 (IP20 for motor connector) (IEC60529)			
Mass of Actuator Unit	kg lb	0.5 1.1		1.2 2.64	2.6 5.72

۴-۲- مدلسازی سه بعدی مکانیزم ربات

طراحی اولیه مکانیزم ربات در محیط نرم افزار MSC Working Model® انجام شد. این نرم افزار محیط بسیار مناسبی برای طراحی و مدلسازی سه بعدی سیستمهای دینامیکی فراهم ساخته است و امکاناتی برای ایجاد انیمیشن از مکانیزمهای طراحی شده در اختیار کاربر قرار داده است. طراحی با این نرم افزار و قابلیت سیمولیشن و حتی تحلیل دینامیکی آن، این نرم افزار را از هر لحاظ برای ایجاد طرحهای اولیه مناسب ساخته است. شکل (۲-۱۲) مدل سه بعدی طرح اولیه مکانیزم ربات CamCrane که در محیط این نرم افزار طراحی شده است را نشان می دهد. به منظور حصول اطمینان اولیه از عملکرد صحیح مکانیزم، طرح تهیه شده در Working Model برای حالت های متنوع شرایط کاری سیمولیشن شده است که فایل های انیمیشن آن در CD ضمیمه پایان نامه موجود می باشد وجود این سیمولیشن ها از لحاظ ایجاد شهود اولیه از نحوه عملکرد مکانیزم، بارهای وارده به آن و حتی نحوه پیاده سازی ایده های کنترلی بر روی آن بسیار مفید می باشد.



شکل (۲-۱۲) مدل سه بعدی طرح اولیه مکانیزم ربات CamCrane

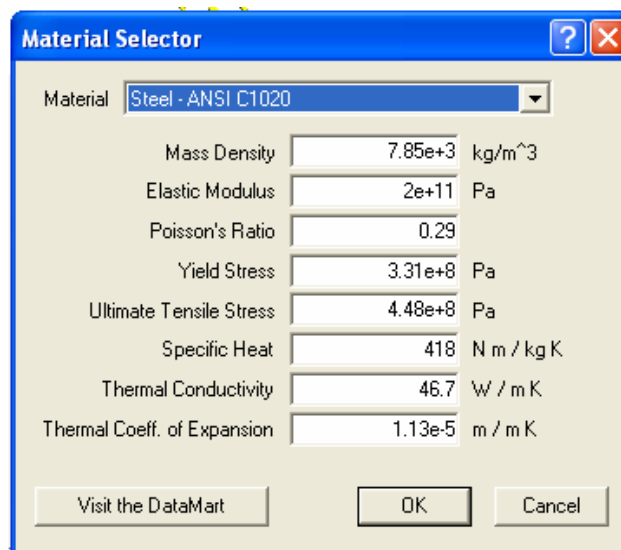
۲-۵- تحلیل مکانیزم ربات

مدلسازی سه بعدی انجام شده در محیط Working Model با یکسری تغییرات به محیط نرم افزار MSC Visual Nastran® منتقل شده است، تا تحلیلهای استاتیکی و دینامیکی بر روی آن انجام گردد. این تغییرات عمدتاً شامل تعریف دقیق نوع و نحوه اتصالات؛ وزن و جنس قطعات، نصب عملگرها و موتورها و مشخص ساختن توان آنها، ضرایب اصطکاک استاتیکی و دینامیکی نحوه برخورد و برهم کنش قطعات و حتی ضریب جاذبه زمین می باشد.

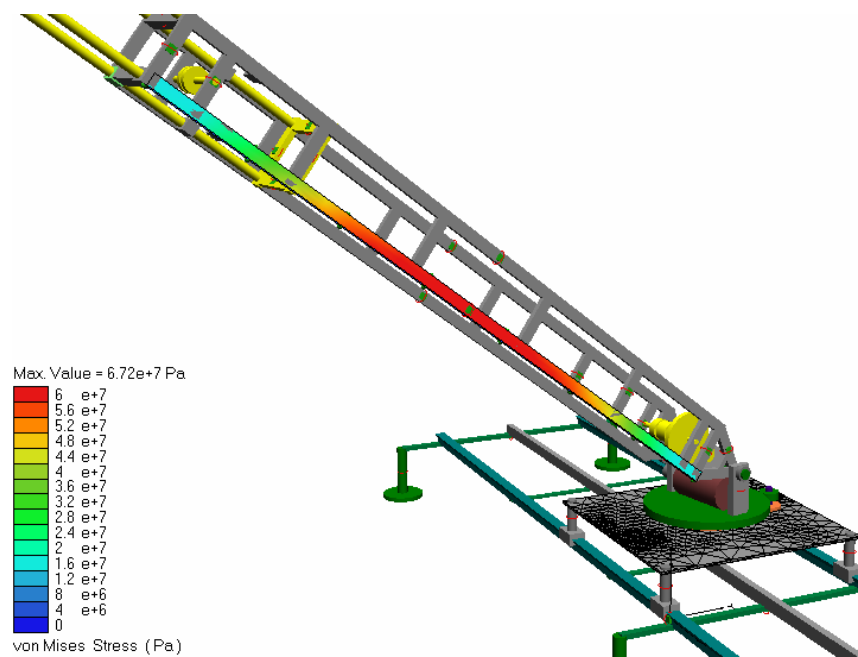
نرم افزار Visual Nastran در واقع توسعه یافته نرم افزار Working Model است که قابلیت تحلیلهای استاتیکی و دینامیکی را برای سیستم دینامیکی طراحی شده را دارد. در تحلیلهای استاتیکی می توان مقادیر تنش و کرنش در قسمتهای مختلف سازه را بدست آورده و در تحلیلهای دینامیکی سرعت و شتاب تک تک قطعات سازه به همراه نیروهای وارده به آنها را می توان تعیین کرد. و از مناسب بودن مشخصات فنی قطعات انتخاب شده مخصوصاً عملگرها اطمینان حاصل کرد.

۲-۵-۱- تحلیل‌های استاتیکی مکانیزم ربات

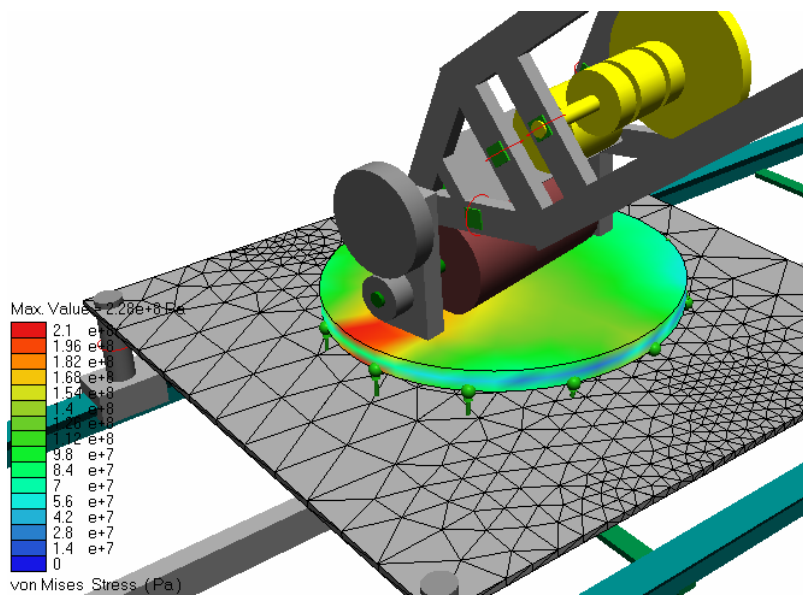
تحلیل‌های استاتیکی شامل بدست آوردن تنش و کرنش در قطعات مختلف مکانیزم ربات مستلزم مشخص ساختن جنس قطعات، نحوه مش‌بندی، قیدها و بارهای وارده به آن قطعه می‌باشد این تحلیل برای قسمتهایی از سازه ربات که بحرانی بنظر می‌رسیدند انجام شده است که نتایج آن در CD ضمیمه پایان نامه موجود می‌باشد بعنوان مثال برای بازوی تلسکوپی که مشخصات جنس آن بصورت جدول شکل (۲-۱۳) است نتایج تحلیل دینامیکی تنش‌های ایجاد شده در قسمتهای مختلف مکانیزم در شکل‌های (۲-۱۴) و (۲-۱۵) آمده است.



شکل (۲-۱۳) محیط انتخاب مشخصات جنس قطعات



شکل (۲-۱۴) نتایج تحلیل دینامیکی تنشهای وان میزز در بازوی تلسکوپی



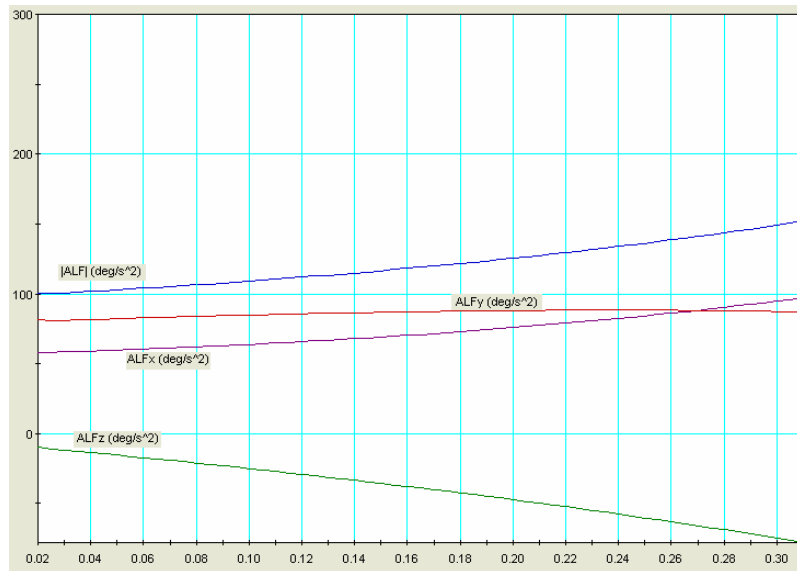
شکل (۲-۱۵) نتایج تحلیل دینامیکی تنشهای ایجاد شده در صفحه لولای بازوی تلسکوپی

۲-۵-۲- تحلیل‌های دینامیکی مکانیزم ربات

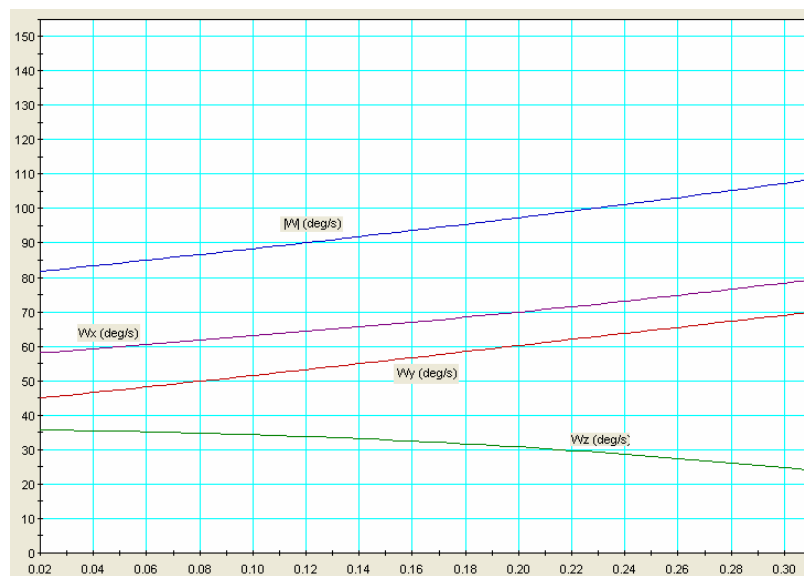
تحلیل‌های دینامیکی شامل بدست آوردن مقادیر سرعت، شتاب و نیروهای وارده به قسمت‌های مختلف سازه ربات می‌باشد که مستلزم مشخص ساختن پارامترهای مربوط به جنس قطعات، سرعت و شتاب، نیروهای خارجی وارد به آنها و نحوه عکس‌العمل قطعات نسبت به یکدیگر در هنگام برخورد (شامل ضرایب برخورد و اصطکاک) و پارامترهایی در رابطه با نحوه مدلسازی دینامیکی سیستم و بازه‌های زمانی سیمولیشن و ... می‌باشد.

تحلیل دینامیکی کل مجموعه سازه ربات برای چند حالت و ترکیب بحرانی از حرکات قسمت‌های مختلف ربات بطور کامل انجام شده است و نتایج بصورت نمودارهایی وابسته به زمان بدست آمده است. این نمودارها شامل نمودار، موقعیت، جهت، سرعت خطی و زاویه‌ای، شتاب خطی و زاویه‌ای، مومنت، میزان نیرو و گشتاور قیده‌ها، میزان جابجایی قیده‌ها، نیروهای تماس بین قطعات مختلف، ضربه و نیروی اصطکاک می‌باشد که برای هر قطعه با توجه به اهمیت پارامتر مورد تحلیل می‌توان از انواع این نمودارها استفاده کرد. بدلیل حجم بسیار بالای نتایج حاصل از تحلیل دینامیکی برای سازه ربات فیلمبردار، نتایج آن در CD ضمیمه پایان‌نامه بطور کامل آمده است و در اینجا بعنوان نمونه قسمتی از نتایج تحلیل دینامیکی برای بازوی ربات را ارائه می‌دهیم شبیه‌سازی دینامیکی برای حالتی انجام شده است که در آن واگن با شتاب 3 m/s^2 و سرعت 2 m/s حرکت می‌کند دوربین در موقعیت فاصله شعاعی 4 m قرار گرفته و هر دو اتصال لولایی آن دارای سرعت 3 rad/s و شتاب 1 rad/s^2 در جهت ساعتگرد را تجربه می‌کنند مکانیزم تلسکوپی هم با سرعت 1 m/s بسمت بیرون داخل می‌کند و شتاب آن 2 m/s^2 بسمت حرکت می‌باشد و بالاخره هر دو اتصال لولایی بازوی ربات نیز دارای سرعت 1 rad/s و شتاب 1 rad/s^2 از موقعیت اولیه صفر درجه می‌باشند.

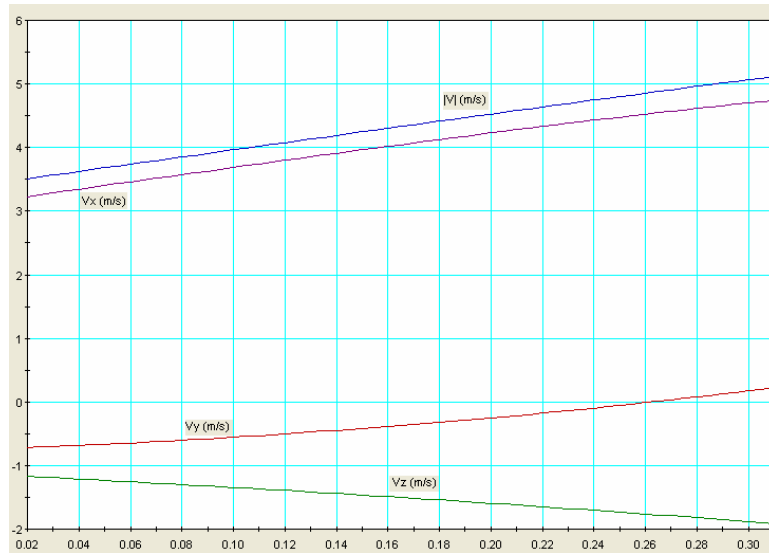
با توجه به شرایط اولیه ذکر شده در بالا شبیه‌سازی دینامیکی انجام شده و نتایج مربوط به سرعت و شتاب خطی و زاویه‌ای بازوی ربات در شکل‌های (۲-۱۶) تا (۲-۱۹) آمده است.



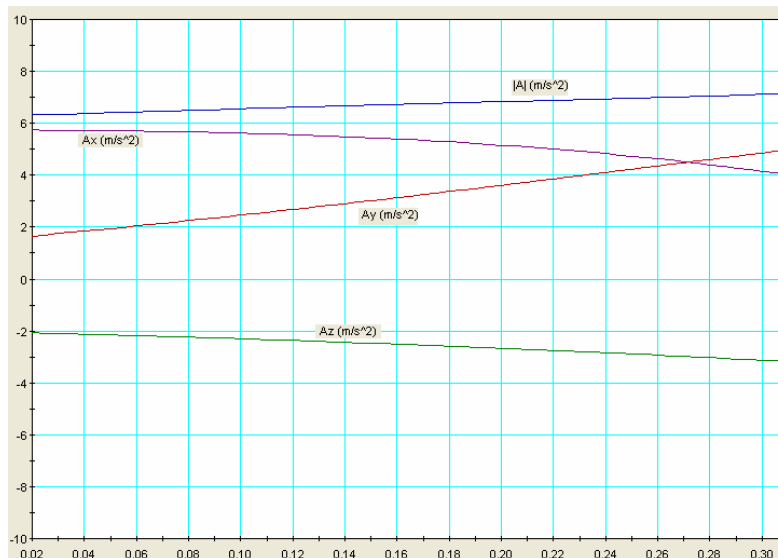
شکل (۲-۱۶) نمودار شتاب زاویه‌ای بازو



شکل (۲-۱۷) نمودار سرعت زاویه‌ای بازو



شکل (۱۸-۲) نمودار سرعت خطی بازو



شکل (۱۹-۲) نمودار شتاب خطی بازو

فصل ۳

پردازش تصویر و بدست

آوردن مختصات سه بعدی

هدف از پردازش تصویر تولید اطلاعات فیدبک از مختصات سه بعدی توپ برای سیستم کنترلی می‌باشد از میان انواع روشهای موجود برای بدست آوردن مختصات سه بعدی توپ در زمین فوتبال، روش پردازش تصویر مناسب شناخته شد برای این منظور دوربینهای متعددی در محلهای مناسبی از ورزشگاه نصب می‌شود که هر کدام می‌تواند محدوده مشخصی از زمین بازی را در دید خود داشته باشد. کارهای بهینه سازی متعددی در رابطه با پوزیشن و نحوه نصب آنها در مراجع ۱، ۲، ۳ و ۴ آمده است بگونه ای که در هر لحظه حداقل دو دوربین توپ را در تصویر داشته باشد که بتوان از روی اطلاعات دو بعدی حاصله از آنها اطلاعات سه بعدی را استخراج نمود.

یکی از اساسی‌ترین پارامترهای تاثیر گذار در نرم‌افزار پردازش تصویر قابلیت اجرای Real time آن است که با توجه به حجم عملیات فوق العاده بالای الگوریتمهای پردازش تصویر رسیدن به این هدف بسیار مشکل می‌باشد.

در این فصل به بحث سخت افزاری و نرم‌افزاری پردازش تصویر و بدست آوردن مختصات سه بعدی پرداخته شده است. در بخش نخست طرح سخت افزاری پیشنهادی و قطعات مورد نیاز ارائه می‌شود سپس قسمتهای مختلف نرم‌افزار پردازش تصویر شامل بخشهای دریافت تصویر، Deinterlacing، سنکرونیزاسیون، پردازش تصویر و بدست آوردن مختصات دو بعدی توپ توضیح داده می‌شود در ادامه بخش نرم‌افزار سه بعدی‌ساز بطور کامل توضیح داده شده است. و در نهایت نمونه‌ای از اجرای نرم‌افزار به همراه نتایج ارائه شده است. توضیحات کامل مربوط به کدهای نوشته شده و عناوین توابع و وظایف آنها در ضمایم ۲ و ۳ آمده است.

۳-۲- طراحی کلی سیستم سخت‌افزاری پردازش تصویر

برای ثبت و اندازه‌گیری حرکت توپ با استفاده از تصویربرداری ویدیویی، به سخت‌افزارهای متعددی نیاز است که در این قسمت مجموعه سخت‌افزارهای مورد نیاز معرفی و تشریح می‌شوند.

۳-۲-۱- دوربینهای تصویربرداری و کارت واسطه

انتخاب و تهیه دوربین‌ها در دو مرحله انجام شد. تعداد فریم‌ها در ثانیه (fps) برای هر دوربین از مشخصه‌های اصلی تصویربرداری محسوب می‌شوند. معیار انتخاب fps، سرعت حرکت توپ می‌باشد و هرچه سرعت حرکت بیشتر باشد، fps بیشتری مورد نیاز است. سرعت تصویربرداری در درجه نخست به نوع دوربین بستگی دارد.

با توجه به قیمت گزاف دوربین‌های سرعت بالا و به دلیل امکان تصویربرداری با اطمینان از عدم حذف فریم‌های تصویر در دوربین‌های دیجیتالی، بر روی استفاده از دوربین‌های دیجیتالی معمولی تمرکز صورت گرفت. البته دوربین‌های سرعت بالای مخصوص حرکات سریع با سرعت تصویربرداری بیش از ۱۰۰ فریم در ثانیه نیز شناسایی شدند. این دوربین‌ها مجهز به جمع‌آوری کننده فریم (Frame Grabber) هستند که قادر است به طور همزمان تصاویر چند دوربین را دریافت و ذخیره سازد. مشخصات فنی این دوربین‌ها در پیوست ۲ آمده است. کلیه دوربین‌های هندی‌کم و از جمله دوربین JVC مدل GR-DVL 320A از سیستم تصویربرداری PAL استفاده کرده و تصاویر را با سرعت ۲۵ فریم در ثانیه ثبت می‌کنند. نرخ تصویربرداری ۲۵ فریم در ثانیه پائین است. این امر به ویژه در تصویربرداری از حرکات سریع آشکارتر می‌شود [۱۲]. در چنین مواردی شکل توپ در تصاویر از حالت مدور خارج شده و به شکل بیضی کشیده در می‌آید. بنابراین به منظور فراهم‌سازی امکان تصویربرداری با سرعت‌های بالاتر و نیز از آنجا که برای

دستیابی به مختصات سه‌بعدی داشتن تصاویر از حداقل دو بعد نقطه هدف ضروری است، دوربین‌های دیجیتال JVC مدل GR-DVL 9800 انتخاب شدند. مشخصات فنی این نوع دوربین در پیوست ۲ آمده است. شایان ذکر است سرعت واقعی تصویربرداری برداشتن ۵۰ تصویر در ثانیه است و در دوربین‌ها به منظور صرفه‌جویی در فضای ذخیره‌سازی تصاویر، از تکنیک Interlacing استفاده می‌شود که این ۵۰ فریم را به ۲۵ فریم تبدیل می‌کند. با اعمال Interlacing، دو تصویر با رزولوشن کامل به یک تصویر با همان رزولوشن تبدیل می‌شوند و در عمل حجم ذخیره‌سازی به نصف می‌رسد. در صورت Deinterlacing فریم‌ها می‌توان به ۵۰ تصویر اولیه دست یافت. علاوه بر این دوربین‌های JVC مدل GR-DVL 9800 می‌توانند در مد با سرعت بالا، با سرعت ۵۰ و ۱۰۰ فریم در ثانیه به ترتیب با رزولوشن نصف و یک چهارم تصویربرداری کنند (شکل (۳-۱)). در نتیجه در صورت اعمال تکنیک Deinterlacing، عملاً فیلم‌هایی با سرعت ۱۰۰ تا ۲۰۰ فریم به دست خواهد آمد.



شکل (۳-۱) تصویر گرفته شده با استفاده از قابلیت سرعت بالای دوربین (به ترتیب از راست به چپ: ۵۰ فریم در ثانیه افقی، ۵۰ فریم در ثانیه عمودی با رزولوشن یک دوم و ۱۰۰ فریم در ثانیه با رزولوشن یک چهارم)

دوربین‌های دیجیتالی مجهز به درگاه ilink هستند که کاربر را قادر می‌سازد اطلاعات را طبق استاندارد IEEE1394 موسوم به Fire Wire به کامپیوتر انتقال دهد [۱۱]. برای انتقال تصاویر از دوربین به کامپیوتر از کارت واسطه IEEE 1394 Host Controller استفاده شد.

۳-۲-۲- کامپیوترهای پردازش تصویر

برای راه اندازی و استفاده از نرم افزار پردازش تصویر به شش دستگاه کامپیوتر سرعت بالا و ملزومات با مشخصات زیر نیاز است. شایان ذکر است استفاده از RAM بالا به دلیل حجم محاسبات و عملیات صورت گرفته در مراحل مختلف استفاده از نرم افزار پردازش تصویر ضروری است.

I-link RAM، Pentium 4، 3GHz Full Cash حداقل ۱Gb، دارای کارت واسط I-link

۳-۲-۳- چارچوب کالیبراسیون

کالیبراسیون با قرار دادن نقاط کنترلی مرجع و ثابت در یک فضای تعریف شده و تعیین مختصات سه بعدی آنها انجام می شود. از این اطلاعات برای تعریف یک سیستم مختصات فضایی مرجع استفاده می شود. با استفاده از این داده ها می توان موقعیت توپ در فضا و جهت حرکت را محاسبه کرد. برای به دست آوردن مختصات فضایی یک جسم متحرک در فضا، باید حداقل از دو دوربین استفاده نمود و از اطلاعات استخراج شده از این دو دوربین مختصات فضایی جسم بدست می آید. چون دستیابی به این مختصات به اطلاعات حاصل از دوربین ها وابسته است، برای به دست آوردن اطلاعات دقیق و درست، دوربین ها باید کالیبره شوند با این کار از اثرات غیرخطی بودن عدسی دوربین ها و تغییر ابعاد موقعیت توپ در اثر حرکت، بر مقادیر محاسبه شده اجتناب می شود. بدین منظور می توان از روش بازسازی تصاویر و روش تبدیل خطی مستقیم (DLT) استفاده نمود. معمولترین روش برای به دست آوردن ضرائب تبدیل، استفاده از یک چارچوب کالیبراسیون است که بسیار دقیق ساخته شده باشد. حداقل شش نقطه کنترلی در محلهای خاصی از فریم در نظر گرفته می شوند و سپس با استفاده از دستگاه اندازه گیری دقیق، مختصات فضایی دقیق آنها به دست می آید. پس از آن، با تصویربرداری از فریم توسط هر یک از این دوربین ها، ضرائب تبدیل

مشخص می‌شود و سپس می‌توان مختصات فضایی هر جسم متحرک را به دست آورد. مختصات سه‌بعدی توپ در ورزشگاه با تعیین مختصات دوبعدی حاصل از هر دوربین به دست می‌آید و سپس این دو مختصات با هم تلفیق شده و مختصات سه‌بعدی توپ بدست می‌آید. (شکل (۳-۲)).

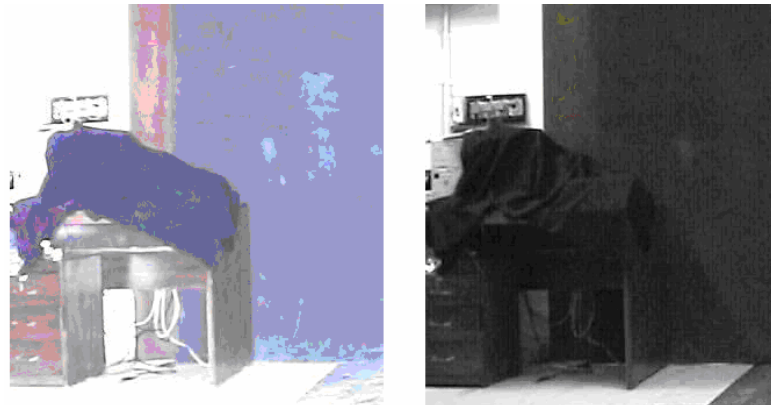


شکل (۳-۲) چارچوب کالیبراسیون نصب شده در زمین اسکی

۳-۲-۴- همزمان‌ساز دوربین‌ها (سنکرونایزر)

به دلیل استفاده از بیش از یک دوربین، لازم است به طریقی کار دوربین‌ها همزمان شود تا فریم‌های معادل از دیدهای مختلف قابل شناسایی باشند و بتوان داده‌های حاصل از آنها را به منظور دستیابی به مختصات سه‌بعدی ترکیب نمود. در این پروژه برای ایجاد همزمانی روشهای مختلفی مورد بررسی قرار گرفت. به علت عدم تعبیه قطعه خاصی در دوربین‌ها که بتوان با استفاده از آن شروع تصویربرداری توسط دو دوربین را همزمان ساخت، به ناچار روشهای محیطی به کار گرفته شد. در ابتدا سعی شد با استفاده از پالس شروع تصویربرداری، به کمک Remote Control دوربین‌ها هر دو دوربین را فعال نمود اما یافتن فاصله و زاویه مناسب با مشکل مواجه شد. در مرحله بعد روشن شدن یک منبع نورانی پیش از آغاز حرکت اصلی در کادر هر دو دوربین و ایجاد نوعی سنکرون تصویری مورد آزمایش قرار گرفت. به این منظور ابتدا روشن شدن فلاش استروبواسکوپ در مقابل دوربین‌ها آزموده شد اما به دلیل کوتاه بودن زمان نوردهی هر فلاش استفاده از این روش مناسب شناخته نشد. پس از آن یک بار از روشن کردن فلاش عکسبرداری و

بار دیگر روشن کردن پروژکتور پیش از شروع حرکت اصلی استفاده شد. مزیت فلاش عکس برداری روشن شدن آن تنها در یک فریم است که شناسایی فریم روشن را آسان می‌سازد (شکل (۳-۳))، البته این امر عیب این روش نیز به شمار می‌رود زیرا ممکن است فلاش در فاصله بین دو فریم یک دوربین روشن شود و در نتیجه در تصاویر حاصل از آن مشاهده نگردد.



شکل (۳-۳) استفاده از نور فلاش برای تشخیص فریم‌های معادل حاصل از دو دوربین

۳-۳- طراحی سیستم نرم‌افزاری پردازش تصویر

سیستم نرم‌افزاری پردازش تصویر شامل نرم‌افزارهای جداگانه‌ای برای دریافت تصویر از دوربین، استخراج فریم‌های فشرده شده از تصاویر، سنکرون کردن دوربین‌ها از طریق شناسایی فریم مبنا در هر یک، پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ از دید هر دوربین است. در این قسمت طراحی کلی نرم‌افزار پردازش تصویر آمده است. جزئیات کدنویسی قسمت‌های مختلف نرم‌افزار پردازش تصویر در پیوست ۴ قابل دسترسی است.

به طور کلی بخش‌های زیر را می‌توان برای این نرم‌افزار در نظر گرفت:

۱ - دریافت تصویر از دوربین

۲ - استخراج فریم‌های فشرده شده از تصاویر

۳- پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ

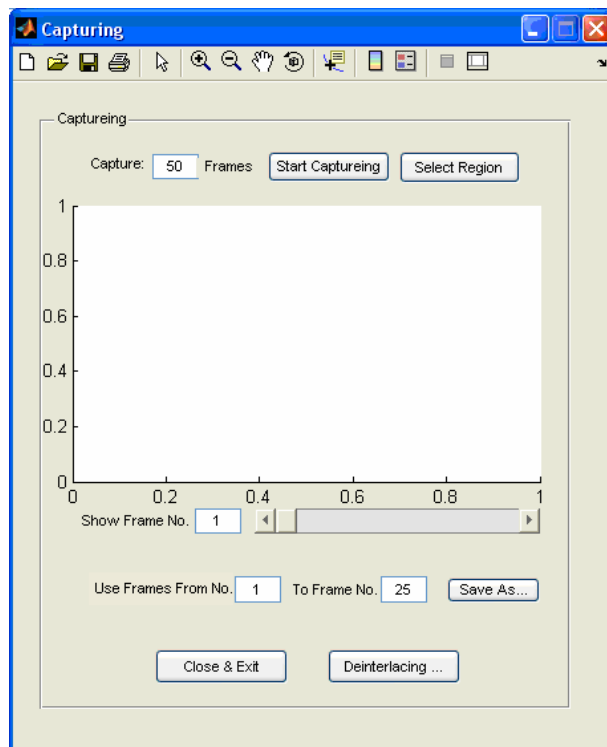
۴- سنکرون کردن دوربین‌ها از طریق شناسایی فریم مبنا در هر یک

۵- ترکیب اطلاعات دو دوربین و به دست آوردن مختصات سه بعدی موقعیت توپ

کلیه قسمت‌ها در محیط نرم‌افزار MATLAB® پیاده‌سازی و اجرا می‌شوند. علت انتخاب محیط نرم‌افزاری MATLAB برای پیاده‌سازی نرم‌افزار پردازش تصویر، به قابلیت‌های متعدد آن در محاسبات ماتریسی برمی‌گردد که امکان انجام محاسبات حجیم پردازش تصویر با سرعت بالا را فراهم می‌کند. وجود ابزارهایی برای کار با تصاویر ویدئویی و پردازش تصویر و امکان برقراری ارتباط با دوربین فیلمبرداری برای دریافت تصاویر از دوربین، همچنین وجود کتابخانه‌ای عظیم از انواع توابع مورد نیاز برای برنامه‌نویسی سبب شده است که از این نرم‌افزار به عنوان پلت فرمی برای پیاده‌سازی سیستم نرم‌افزاری پردازش استفاده شود. در ادامه به بخش‌های مختلف سیستم نرم‌افزاری پردازش می‌پردازیم.

۳-۳-۱- نرم‌افزار دریافت تصویر

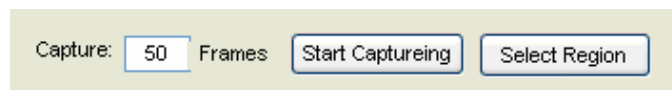
این برنامه برای دریافت تصاویر از دوربین و نیز برای انجام ویرایش‌های اولیه روی تصاویر حاصل از دوربین، تهیه شده است. نرم‌افزار MATLAB به کمک جعبه ابزار Image acquisition تصاویر را از طریق کارت I-Link از دوربین دریافت می‌کند که کاربر می‌تواند پس از انجام ویرایش‌های مورد نظر قسمتهایی از تصویر را که برای پردازش مورد نیاز است، انتخاب و save کند.



شکل (۳-۴) رابط اصلی نرم‌افزار دریافت تصویر از دوربین

روش کار با نرم‌افزار Capturing به صورت زیر است:

الف) با اجرای برنامه، پیش نمایشی از تصاویر گرفته شده از دوربین نشان داده می‌شود. با تعیین تعداد فریم‌های مورد نیاز و فعال سازی دکمه Start Capturing توسط کاربر، به تعداد فریم‌های مشخص شده، از لحظه فعال شدن این دکمه تصویر از دوربین اخذ می‌شود (شکل (۳-۵)).



شکل (۳-۵) انتخاب تعداد فریم‌های مورد نظر برای اخذ از دوربین

ب) کاربر می‌تواند پس از اخذ تصاویر از دوربین، به کمک اسلایدر موجود فریم‌های اخذ شده را مشاهده کند. سپس بازه فریم‌هایی را که لازم است پردازش تصویر روی آنها انجام شود انتخاب

کرده و با فعال کردن دکمه Save As... فریم‌های مورد نیاز Save در قالب فایلی با فرمت avi ذخیره خواهد شد (شکل (۶-۳)).



شکل (۶-۳) انتخاب بازه فریم‌های مورد نظر برای ذخیره سازی

۳-۳-۲- طراحی و کدنویسی برنامه استخراج فریم‌های فشرده شده از تصاویر

(Deinterlacing)

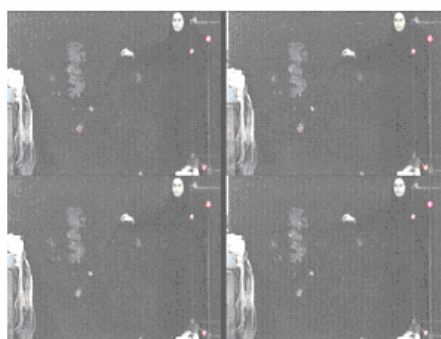
دوربین‌های JVC مدل GR-DVL 9800 دارای قابلیت تصویر برداری با سرعت‌های 50 Hz (با رزولوشن ۷۲۰*۵۷۶ پیکسل)، 100 Hz و 200 Hz (به ترتیب با رزولوشن نصف و یک چهارم) است. در دوربین‌ها به منظور صرفه‌جویی در فضای ذخیره سازی تصاویر، از تکنیک Interlacing استفاده می‌شود. در این روش دو تصویر با رزولوشن کامل به یک تصویر با همان رزولوشن تبدیل می‌شود و در عمل حجم ذخیره سازی به نصف می‌رسد. شکل (۷-۳) چهار نوع تصویر گرفته شده از دوربین را نشان می‌دهد.



100 Hz Vartical



100 Hz Horizontal



200 Hz



50 Hz Full

شکل (۷-۳) مدهای مختلف تصویربرداری با دوربین JVC مدل GR-DVL 9800

به عنوان مثال وقتی دوربین در حالت ۵۰ Hz و با رزولوشن ۷۲۰*۵۷۶ پیکسل قرار دارد در هر ثانیه ۵۰ فریم تصویربرداری می کند سپس هر دو فریم متوالی را با هم Interlace می کند تا به مقدار استاندارد ۲۵ فریم در ثانیه برسد به طوریکه هر یک از این فریم ها تا حدود زیادی دربردارنده اطلاعات دو فریمی است که تصویر نهایی از آن ساخته شده است و در واقع سطرهای فرد و زوج تصویر حاصله به ترتیب از فریم های اول و دوم گرفته شده است. بدیهی است که در این روش اطلاعات موجود در سطرهای زوج فریم اول و همچنین اطلاعات موجود در سطرهای فرد فریم دوم دور ریخته می شود به همین دلیل میزان اطلاعات موجود در تصویر حاصل از Interlacing کمتر از دو فریمی است که از آنها ساخته شده است.

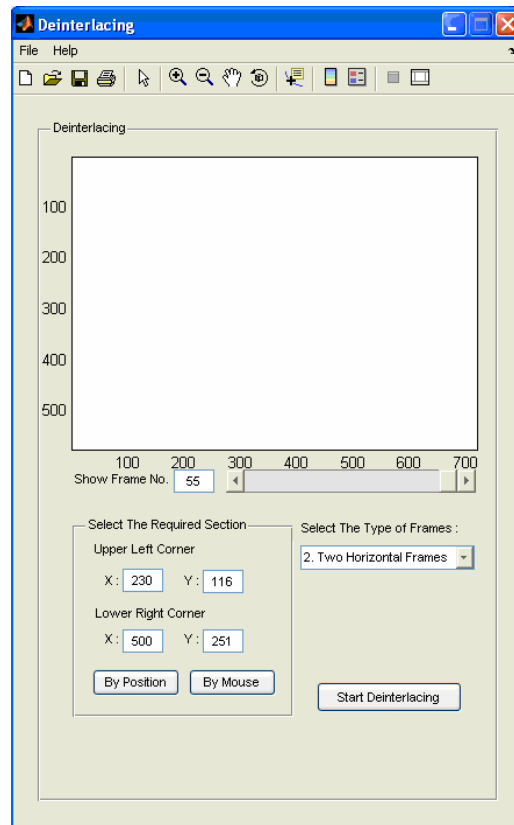
عکس عملیات Interlacing اصطلاحاً Deinterlacing نامیده می‌شود. در این روش فریم Interlace شده به دو فریم مجزا تقسیم می‌شود و در واقع فریم‌هایی که تشکیل دهنده تصویر Interlace شده هستند بازسازی می‌شوند. بدیهی است که فریم‌های بازسازی شده نصف رزولوشن عمودی فریم‌های اولیه را خواهند داشت. (یعنی هر فریم بازسازی شده بجای ۵۷۶ سطر فقط دارای ۲۸۸ سطر خواهد بود) که برای حل این مشکل می‌توان سطرهای زوج یا فرد حذف شده از هر فریم را با روش‌های میانمایی خطی، درجه دو یا سه بازسازی نمود تا کیفیت تصویر تا حد ممکن به تصویر اولیه نزدیکتر شود. در شکل (۳-۸) یک فریم Interlace و Deinterlace شده مشاهده می‌شود. تصویر سمت راست یکی از فریم‌های Interlace شده است که توسط نرم‌افزار Capturing از دوربین گرفته شده است و تصویر سمت چپ Deinterlace شده همان فریم می‌باشد که توسط نرم‌افزار Deinterlacing تهیه شده است.



شکل (۳-۸) فریم Interlace (راست) و Deinterlace شده (چپ)

دوربین‌های مورد استفاده دارای دو حالت تصویربرداری ۱۰۰ Hz و ۲۰۰ Hz نیز می‌باشد که در حالت ۱۰۰ Hz تصاویر Interlace شده به صورت دو تصویر عمودی یا افقی در کنار هم نشان داده می‌شوند و در حالت 200 Hz نیز به صورت چهار تصویر ارائه می‌شود. به طوری که ۲۰۰ فریم فیلمبرداری شده در هر ثانیه ابتدا Interlace شده و به ۱۰۰ فریم تبدیل می‌شود سپس این ۱۰۰ فریم با رزولوشن کامل به ۱۰۰ فریم با رزولوشن 1/4 تبدیل می‌شود و در نهایت هر چهار فریم

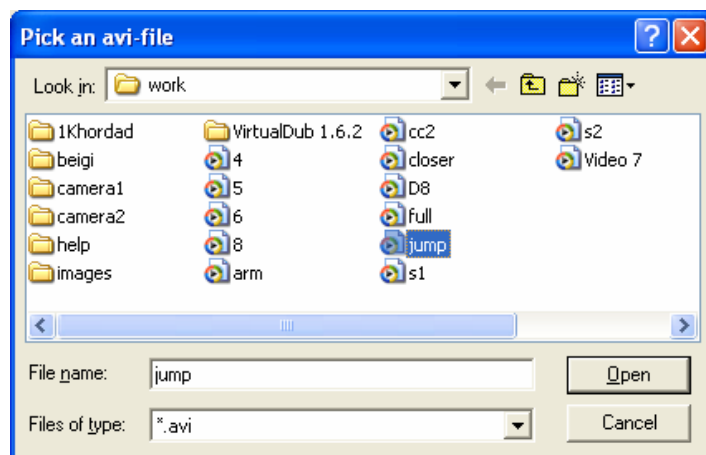
متوالی در یک فریم جایدهی می‌شوند بگونه ای که ۲۰۰ فریم در نهایت به ۲۵ فریم تبدیل می‌شود (شکل (۸-۳) پائین چپ). رابط برنامه نوشته شده برای Deinterlacing در شکل (۹-۳) آمده است.



شکل (۹-۳) نمای کلی رابط نرم‌افزار Deinterlacing

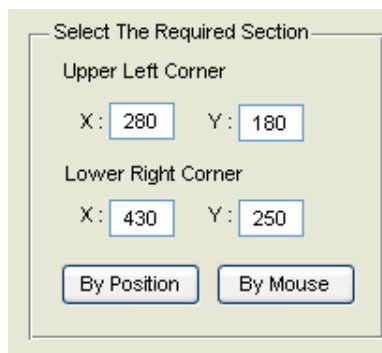
روش کار با این برنامه به صورت زیر است:

- ۱ - با انتخاب گزینه Open از منوی فایل می‌توان یک فایل با فرمت avi را باز کرد. فایل لود شده و آخرین فریم آن نشان داده می‌شود و شماره فریم در زیر تصویر در سمت چپ مشخص می‌گردد. کاربر می‌تواند با استفاده از اسلایدر تعبیه شده در زیر تصویر تک تک فریم‌ها را مشاهده کند.



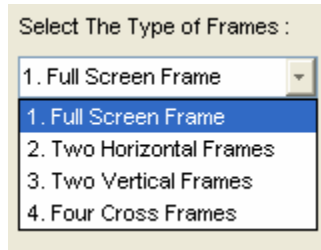
شکل (۳-۱۰) منوی باز کردن یک فایل avi

۲- به کمک ماوس و یا با دادن مقادیر مختصات، می‌توان ناحیه‌ای از تصویر را که لازم است Deinterlace شود در یکی از فریم‌ها انتخاب کرد.



شکل (۳-۱۱) منوی انتخاب ناحیه مورد نظر برای اجرای عملیات Deinterlacing

۳- کاربر می‌تواند با توجه به سرعت تصویربرداری (۵۰ هرتز تمام صفحه، ۱۰۰ هرتز با دو فریم عمودی، ۱۰۰ هرتز با دو فریم افقی یا ۲۰۰ هرتز با چهار فریم) یکی از چهار حالت مناسب را از منوی کرکره‌ای انتخاب کرده و سپس قسمت‌های متناظر با ناحیه انتخاب شده در قسمت (۲) را در بقیه فریم‌ها انتخاب کند.



شکل (۳-۱۲) منوی کرکره‌ای تعیین مد سرعت تصویربرداری

۴ - با انتخاب دکمه Start Deinterlacing محاسبات Deinterlacing آغاز می‌شود. پس از اتمام این عملیات می‌توان فایل حاصله را با فرمت avi ذخیره کرد.

۳-۳-۳- کدنویسی برنامه پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ

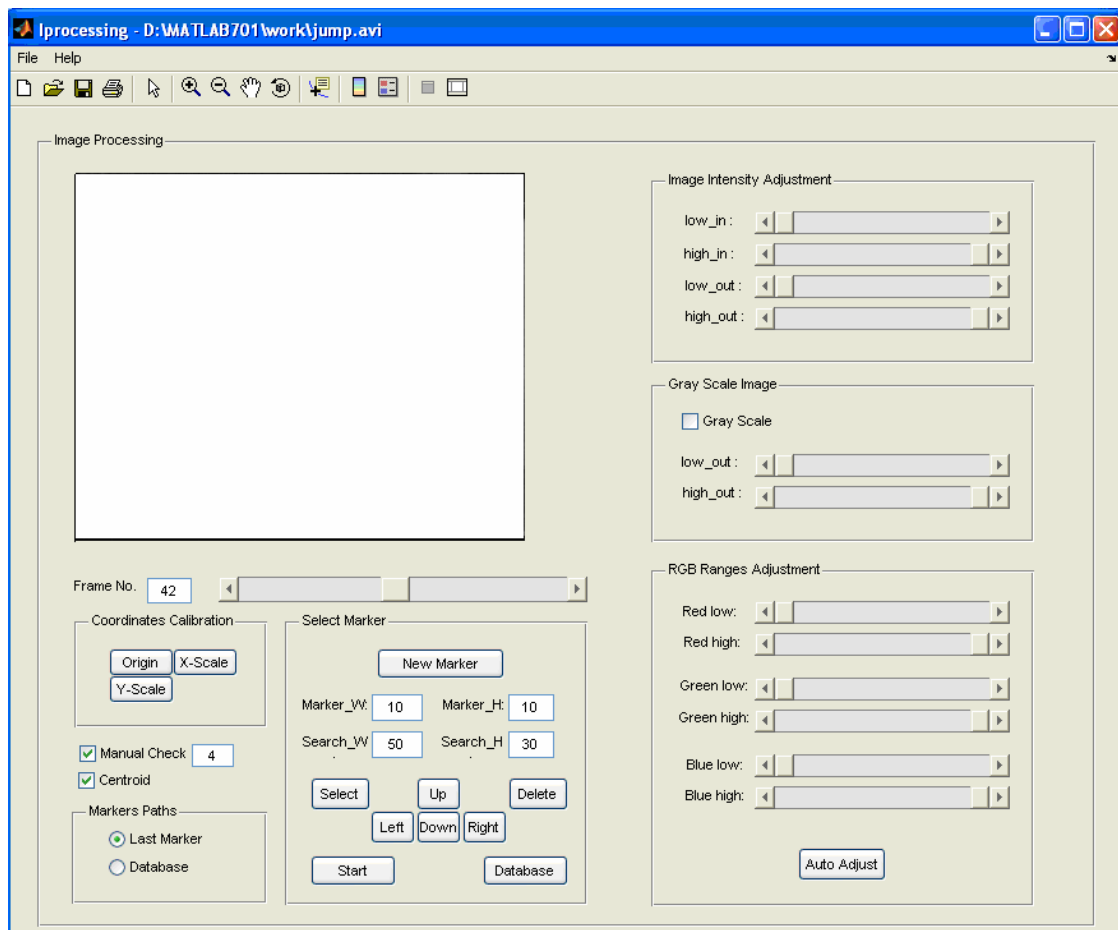
هدف برنامه پردازش تصویر، تشخیص موقعیت توپ و به دست آوردن مختصات دوبعدی آنها است. متداولترین روش برای پردازش تصویر استفاده از ماتریس RGB تک تک فریم‌ها می‌باشد که در این نرم‌افزار نیز از این روش استفاده شده است. روش کار بدین صورت است که کاربر موقعیت توپ را انتخاب می‌کند و نرم‌افزار با جستجوی توپ در تمام فریم‌های بعدی آن را پیدا کرده و مختصات دو بعدی آن را بر حسب پیکسل استخراج می‌کند. تکنیک به کار رفته برای پیدا کردن موقعیت توپ در هر فریم الگوریتم معروف به دست آوردن Correlation و مینیمم سازی آن بین تصویر توپ و فریم مورد جستجو است. که البته برای بالا بردن ضریب اطمینان جستجو و همچنین برای تسریع عملیات پردازش تصویر الگوریتم مذکور تعدیل و بهینه‌سازی شده است از جمله این تکنیکها می‌توان به موارد زیر اشاره کرد:

- ۱ - پیش‌بینی محل توپ در فریم جدید با استفاده از میانبایی نتایج فریم‌های قبلی
- ۲ - استفاده از یک ناحیه محدود (به جای کل فریم) به عنوان منطقه مورد جستجو برای تسریع عملیات جستجو

۳ - استفاده از انواع فیلترها برای حذف نویزهای تصویر

اسلایدرهای موجود در سمت راست شکل (۳-۱۳)، این امکان را به کاربر می‌دهند که با محدود کردن بازه نمایش رنگهای RGB تا حد ممکن تصاویر اضافی را از فریم‌ها برداشته و موقعیت توپ واضح‌تر و شفاف‌تر دیده شود.

این اصلاحات و تنظیمات زمانی مورد استفاده قرار می‌گیرد که نرم‌افزار نتواند به طور خودکار محل توپ را در تعدادی از فریم‌ها تشخیص دهد. در چنین مواقعی با انجام این تنظیمات دستی می‌توان به نرم‌افزار در پیدا کردن محل توپ کمک رساند.

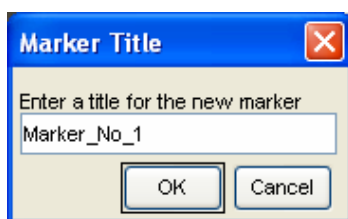


شکل (۳-۱۳) نمای کلی رابط نرم‌افزار پردازش تصویر

روش کار با این برنامه به ترتیب زیر است:

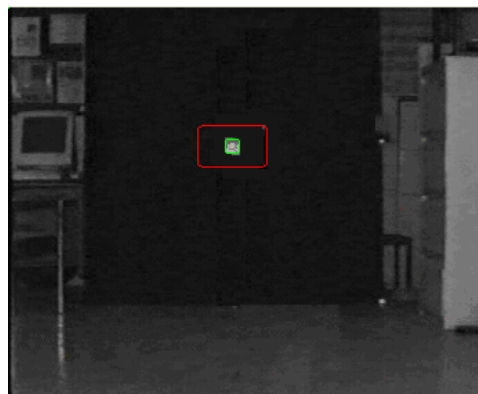
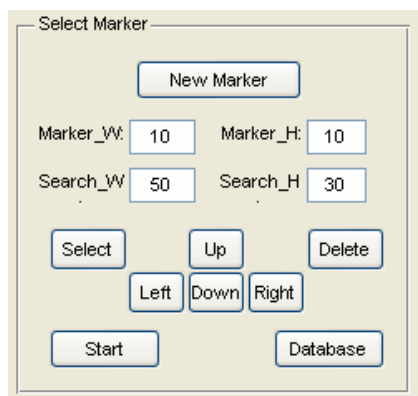
۱- با انتخاب گزینه Open از منوی فایل، می‌توان یک فایل ویدئویی با فرمت avi را باز و لود کرد. آخرین فریم آن مشاهده می‌گردد. توسط اسلایدر موجود می‌توان سایر فریم‌های تصویر را تک تک مشاهده کرد.

۲- با انتخاب دکمه New Marker، می‌توان توپ را به صورت دلخواه نامگذاری کرد (شکل (۳-۱۴)).



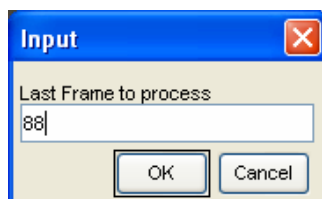
شکل (۳-۱۴) کادر محاوره‌ای ثبت نام توپ

۳- با زدن دکمه O می‌توان توپ مورد نظر را روی تصویر مورد نظر انتخاب کرد. در صورت نیاز به تغییر محل انتخاب توپ، با زدن دکمه X می‌توان عملیات را Cancel کرد. دکمه های U، D، L و R برای حرکت دادن باکس محاط توپ مورد نظر به بالا، پائین، چپ و راست تعبیه شده اند (شکل (۳-۱۵)، چپ). شایان توجه است که با انتخاب توپ دو مستطیل سبز و قرمز روی تصویر نشان داده می‌شود. مستطیل سبز باید محاط بر توپ باشد و مستطیل قرمز در واقع ناحیه‌ای است که نرم‌افزار در فریم بعدی برای جستجوی توپ استفاده می‌کند. اندازه این ناحیه تابعی از سرعت حرکت توپ مورد نظر می‌باشد. بدیهی است هر چه سرعت حرکت توپ بیشتر باشد این ناحیه نیز باید بزرگتر انتخاب شود. اندازه این دو مستطیل توسط اعدادی که در پانل انتخاب توپ از کاربر خواسته شده است تعیین می‌گردد.



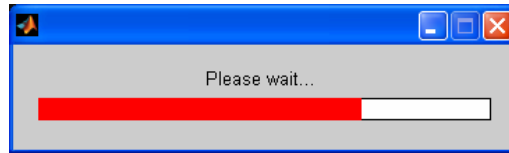
شکل (۱۵-۳) محل توپ (راست) و دکمه‌های تنظیم محل آن (چپ)

۴- پس از انتخاب دقیق محل توپ با زدن دکمه Start عملیات پردازش تصویر و یافتن موقعیت توپ در فریم‌های بعدی آغاز می‌شود. پیش از شروع عملیات، بازه فریم‌هایی که باید توپ در آنها جستجو شود از کاربر پرسیده می‌شود (شکل (۱۶-۳)). شروع بازه فریمی است که توپ در آن مشخص شده است و انتهای بازه به طور پیش فرض آخرین فریم فایل ویدئویی باز شده می‌باشد که می‌توان بطور دلخواه آن را تغییر داد.



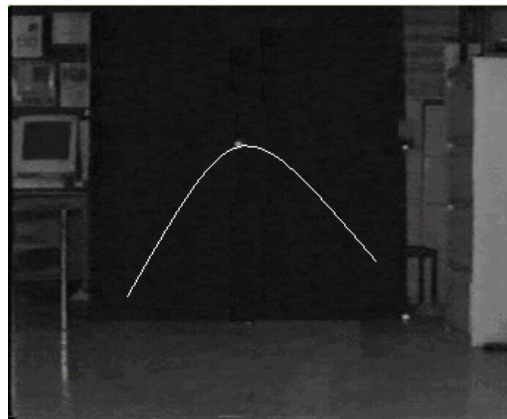
شکل (۱۶-۳) کادر محاوره‌ای مربوط به آخرین فریم بازه جستجوی توپ

با زدن OK عملیات پردازش تصویر شروع می‌شود و کادر محاوره‌ای شامل Progress Bar ظاهر می‌شود که میزان پیشرفت عملیات را نشان می‌دهد (شکل (۱۷-۳)).



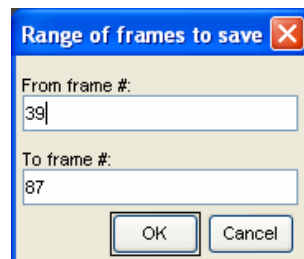
شکل (۱۷-۳) نوار پیشرفت عملیات پردازش تصویر

پس از اتمام عملیات پردازش تصویر مسیر حرکت توپ روی تصویر نشان داده می‌شود (شکل (۳-۳)). به کمک اسلایدر می‌توان توپ را روی مسیر رسم شده حرکت داد و از درستی نتایج اطمینان حاصل نمود.



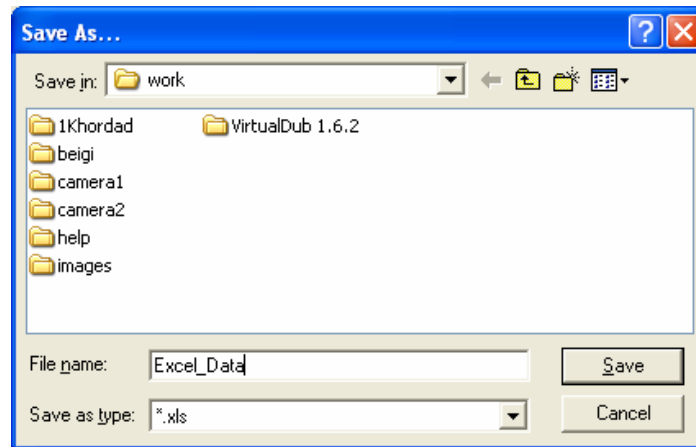
شکل (۱۸-۳) مسیر توپ حین حرکت

۵- با انتخاب دکمه Database اطلاعات x-y به دست آمده برای موقعیت توپ تحت نام انتخاب شده برای توپ، در حافظه ذخیره می‌شود. کادر محاوره‌ای مربوطه بازه فریم‌هایی که قرار است ذخیره شود را از کاربر درخواست می‌کند (شکل (۱۹-۳)).



شکل (۱۹-۳) کادر محاوره‌ای مربوط به بازه فریم‌هایی که قرار است ذخیره شود

۶- پس از اتمام عملیات و به دست آوردن نتایج همه موقعیت توپ می‌توان اطلاعات موجود در Database را به صورت یک فایل Excel ذخیره نمود. برای این کار باید گزینه Export excel ...file از منوی فایل انتخاب شود (شکل (۳-۲۰)).



شکل (۳-۲۰) ذخیره سازی اطلاعات به صورت فایل Excel

۸- برای باز کردن و مشاهده نتایج ذخیره شده در فایل‌های Excel بر روی فیلم می‌توان از گزینه Import Excel file استفاده کرد (شکل (۳-۲۱)). با انتخاب این گزینه، توسط یک کادر محاوره‌ای نام فایل اکسل از کاربر دریافت می‌شود و اطلاعات آن به Database موجود انتقال می‌یابد.

File Help	
Workspace	Ctrl+N
Open ...	Ctrl+O
Save a frame ...	Ctrl+F
Save a frame with paths ...	Ctrl+P
Import Excel File ...	Ctrl+I
Export Excel File ...	Ctrl+E
Exit	Ctrl+Q

شکل (۳-۲۱) گزینه‌های موجود در منوی فایل

۹- گزینه Save a Frame امکان ذخیره سازی فریم جاری را به صورت یک فایل jpg فراهم ساخته است.

۱۰- گزینه Save a Frame with paths امکان ذخیره کردن فریم جاری به همراه مسیر موقعیت

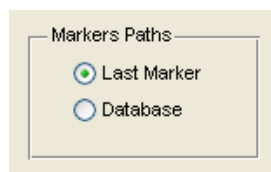
توپ را به صورت یک فایل jpg فراهم میسازد.

۱۱- گزینه Workspace فریم‌های Deinterlace شده در حافظه موقت را فراخوانی کرده و برای

پردازش تصویر آماده می‌کند.

۱۲- گزینه‌های موجود در پانل Balls Paths به ترتیب برای نمایش مسیر آخرین توپ و یا مسیر

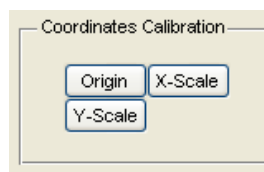
تمام موقعیت توپ‌های موجود در Database است (شکل (۳-۲۲)).



شکل (۳-۲۲) حالات مختلف نمایش مسیر موقعیت توپ

۱۳- به کمک پانل Coordinates Calibration می‌توان مختصات نقطه مبدا، جهات و اندازه

محورهای مختصات دوبعدی را انتخاب کرد (شکل (۳-۲۳)).

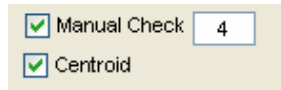


شکل (۳-۲۳) پانل کالیبراسیون دوبعدی

۱۴- فعال کردن گزینه Centroid سبب می‌شود که مرکز سطح توپ به عنوان نقطه معرف محل

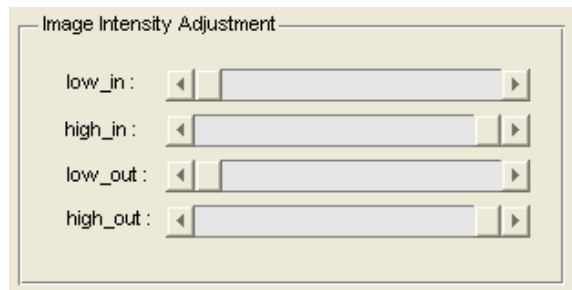
توپ انتخاب شود. گزینه Manual Check امکان و میزان مشارکت کاربر در تعیین محل موقعیت

توپ را تعیین می‌کند (شکل (۳-۲۴)).



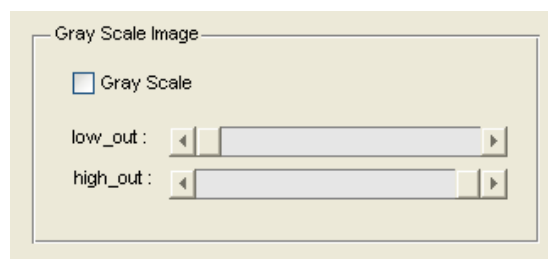
شکل (۳-۲۴) گزینه‌های Manual Check و Centroid

۱۳- اسلایدرهای موجود در پانل Image Intensity Adjustment برای تنظیمات کرانه پائین و بالای شدت نور تصویر ورودی و خروجی تعبیه شده است (شکل (۳-۲۵)).



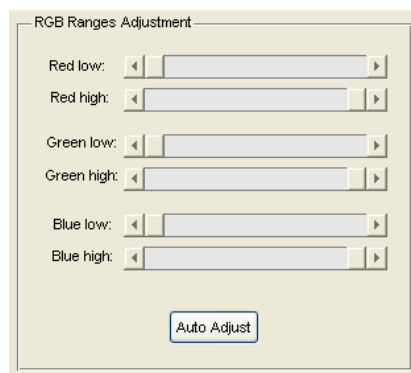
شکل (۳-۲۵) تنظیمات کرانه پائین و بالای شدت نور تصویر ورودی و خروجی

۱۴- اسلایدرهای موجود در پانل Gray Scale Image برای تنظیمات کرانه پائین و بالای تصویر تک رنگ خاکستری است. برای اعمال این تنظیمات لازم است که گزینه Gray Scale انتخاب شود (شکل (۳-۲۶)).



شکل (۳-۲۶) تنظیمات کرانه پائین و بالای تصویر تک رنگ خاکستری

۱۵- اسلایدرهای موجود در پانل RGB Image برای تنظیمات کرانه‌های پائین و بالای تصاویر RGB تعبیه شده است. در صورت انتخاب گزینه Gray Scale نمی‌توان این تنظیمات را اعمال کرد. گزینه Auto Adjust این تنظیمات را به طور خودکار انجام می‌دهد (شکل (۳-۲۷)).



شکل (۳-۲۷) تنظیمات کرانه‌های پائین و بالای تصاویر RGB

۳-۴- تئوری بازسازی مختصات سه بعدی

با داشتن حداقل دو نمای مختلف و به کارگیری اصول فتوگرامتری می توان مختصات فضایی نقاط هدف را تعیین کرد. که در این بخش به صورت اجمالی به اصول بازسازی مختصات سه بعدی پرداخته می شود. [۱۳]

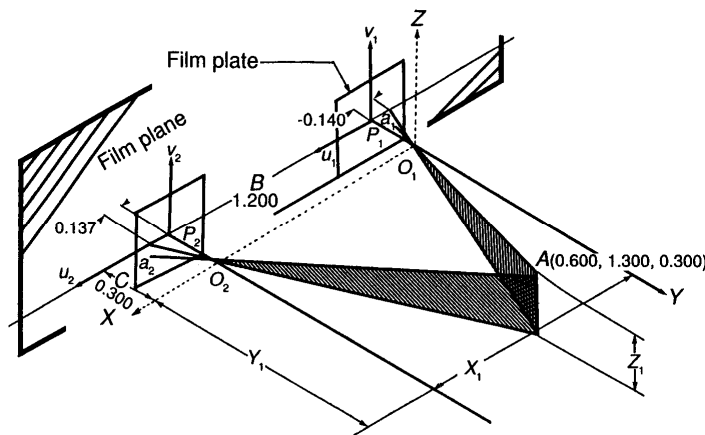
۳-۴-۱- تکنیکهای بازسازی

در صورت تعریف سیستم های مختصاتی مناسب می توان با استفاده از معادلات تبدیل سیستم مختصات نقطه هدف را تعیین کرد. این معادلات رابطه پارامترهای بیرونی نظیر موقعیت دوربین و پارامترهای درونی نظیر ویژگی های عدسی یا نوع دوربین را با موقعیت توپ یا نقاط هدف بیان می کنند. معادلات و روشهای حل معادله، برای دوربین ثابت یا متحرک مورد استفاده قرار می گیرند. البته اغلب اندازه گیری های سه بعدی حرکات با استفاده از دوربین های ثابت انجام می شوند. در عکسبرداری هوایی یا در سیستم های machine vision نظیر وسایل نقلیه خودکار، دوربین روی دستگاه نصب می شود و همراه با آن حرکت می کند و سپس جابجایی دوربین برای محاسبات مورد استفاده قرار می گیرد. فرآیند استاندارد اندازه گیری با استفاده از تصویربرداری، مستلزم اطلاع دقیقی از تمامی پارامترهای درونی و بیرونی دوربین است. هر خطا در یکی از پارامترها به کسب مقادیر نادرستی برای مختصات منجر می شود.

۳-۴-۱-۱- تکنیکهای بازسازی سنتی

اغلب تکنیکهای مجازی، از تکنیکهای کلاسیک نشأت گرفته اند. درک کافی از اصول اولیه به فهم بهتر فعالیتهای جاری منجر می شود. ترتیب قرارگیری مجموعه stereo-photogrammetric همراه

با فواصل نمونه در شکل (۳-۲۸) به نمایش درآمده است. فرضیه‌های مهم عبارتند از اینکه پارامترهای بیرونی دوربین معلوم باشند و صفحات فیلم در یک صفحه مشابه قرار گیرند. این امر به معنای آنست که محورهای اپتیکی آنها با یکدیگر موازی و عمود بر صفحه فیلم است. مبدأ محورهای u_i و v_i در نقطه میانی فیلم (principle point) (P_i) قرار گرفته است. محور اپتیکی هر دوربین از کانون آن عبور می‌کند و در امتداد نقطه میانی (principle point) دوربین قرار می‌گیرد. FCS به صورت قراردادی در کانون دوربین ۱ در O_1 قرار دارد و دوربین‌ها به اندازه $B=1/200\text{ m}$ از یکدیگر فاصله دارند. فاصله کانونی $C=0/300\text{ m}$ است.



شکل (۳-۲۸) ترتیب قرارگیری مجموعه stereo-photogrammetric در حالی که محورهای اپتیکی با یکدیگر موازیند.

فاصله دو دوربین از یکدیگر $1/200\text{ m}$ و فاصله کانونی $0/300\text{ m}$ است. شرایط هم‌راستایی و هم‌صفحه بودن برآورده شده است.

مختصات نقطه هدف A (بر حسب متر) در این مثال نسبت به نقطه O_1 از قبل معلوم بوده است. مقادیر آن از طریق محاسبه با استفاده از معادلات زیر قابل دستیابی است:

$$X_1 = (B \cdot u_1) / P \quad \text{رابطه (۳-۱)}$$

$$Y_1 = (B \cdot c) / P \quad \text{رابطه (۲-۳)}$$

$$Z_1 = (B \cdot v_1) / P \quad \text{رابطه (۳-۳)}$$

$P = u_1 - u_2$ است و برای تصحیح انطباق (parallax) در سیستم مختصات O۱ استفاده می‌شود.

مختصات هدف با توجه به این معادلات به صورت زیر است:

$$X_1 = (1.200 \cdot -0.140) / (-0.140 - 0.137) = 0.606m$$

$$Y_1 = (1.200 \cdot 0.300) / (-0.140 - 0.137) = 1.300m$$

$$Z_1 = (1.200 \cdot 0.690) / (-0.140 - 0.137) = 0.299m$$

با وجود سادگی این روابط ریاضی، لازم است در این تکنیک در مورد محل قرارگیری دوربین‌ها دقت شود. به همین منظور Ayoub و Ramsey (۱۹۷۰)، سیستم عکسبرداری خود و مشخصات آن از قبیل فاصله اصلی، نقطه اصلی و اعوجاج عدسی را کاملاً توصیف کردند.

معادلات (۱-۳) تا (۳-۳) را می‌توان به شکل عمومی‌تری بیان کرد:

$$\begin{bmatrix} u - u_p + \Delta u \\ v - v_p + \Delta v \\ -C \end{bmatrix} = \lambda [M] \begin{bmatrix} X - X_o \\ Y - Y_o \\ Z - Z_o \end{bmatrix} \quad \text{رابطه (۴-۳)}$$

در این معادلات u و v مختصات تصویر، u_p و v_p مختصات تصویر نقطه اصلی، Δu و Δv خطاهای u و v ، C فاصله کانونی دوربین، λ فاکتور معیار خطی، M یک ماتریس تبدیل ۳ در ۳ از تصویر به سیستم مختصات آزمایشگاه، X ، Y و Z مختصات آزمایشگاهی هدف، و X_o ، Y_o و Z_o مختصات کانون در سیستم X ، Y و Z است. از این معادله عمومی بسیاری از معادلات سه‌بعدی مجازی به دست می‌آید.

۳-۴-۱-۲- تکنیکهای بازسازی مجازی

امروزه تکنیکهای مجازی برای بازسازی مختصات سه بعدی توسعه یافته‌اند. بسیاری از آنها با بازچینی، بسط و ساده سازی معادله (۳-۴) به دست آمده‌اند. معادلات سابق به طور کامل به مجموعه‌ای از پارامترهای مجهول تبدیل شدند که تابع مختصات نقطه هدف و ثابتها هستند. ثابتها، روابط ریاضیاتی پیچیده‌ای هستند که یک یا چند پارامتر دوربین را در بر می‌گیرند. به منظور تعیین این پارامترها به صورت تجربی، یک ابزار کالیبراسیون متشکل از تعدادی نقطه با موقعیت دقیقاً اندازه‌گیری شده برای تعیین مجهولات در دو مرحله استفاده می‌شود. [۱۹]

در مرحله نخست از ابزار کالیبراسیون عکسبرداری می‌شود. مختصات نقطه هدف و این تصویر برای تعیین مقدار ثابتهای مجهول مورد استفاده قرار می‌گیرد. سپس نقاط هدف یا توپ فاقد مختصات معلوم، جایگزین ابزار کالیبراسیون می‌شوند و مختصات آنها از مختصات تصویر و ثابتهایی که قبلاً محاسبه شده‌اند به دست می‌آید. تعداد نقاط کالیبراسیون موردنیاز برای حل پارامترهای مجهول به تکنیک مجازی به کار رفته بستگی دارد.

تاکنون چندین تکنیک بازسازی سه‌بعدی برترانتشار یافته‌اند. در میان این روشها شاید روش تبدیل خطی مستقیم یا DLT (توسعه یافته توسط Marzan در سال ۱۹۷۵) بیش از سایرین مورد بررسی قرار گرفته و به صورت گسترده به کار رفته باشد.

روش DLT بر پایه تکنیک کلاسیک استوار است و به صورت زیر بیان می‌شود:

$$u + \Delta u = \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

$$v + \Delta v = \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1}$$

رابطه (۳-۵)

در این معادلات u و v مختصات تصویر و Δu و Δv تصحیح مختصات تصویر برای اعوجاج عدسی هستند. مختصات نقطه هدف X ، Y و Z ، و ثابتهای L_1 تا L_{11} پارامترهای DLT هستند

که علاوه بر پارامترهای درونی دوربین و فاکتورهای خطی اعوجاج عدسی، موقعیت و راستای دوربین را مشخص می‌سازند.

همانطور که پیش از این ذکر شد، در روش کلاسیک (شکل (۳-۲۸)) محورهای اپتیکی دوربین‌ها باید با یکدیگر موازی باشند و در صورت عدم احراز این شرط نمی‌توان از معادله (۳-۴) استفاده کرد و معادلات جدیدی برای تصحیح زاویه دوربین‌ها باید به کار رود. با استفاده از معادلات DLT لزومی ندارد دوربین‌ها موازی باشند اما باید دقت شود برای به حداقل رساندن خطاهای بازسازی، روابط بین زوایای همپوشانی و دوربین تعریف شده توسط Marzan (۱۹۷۵) رعایت شود.

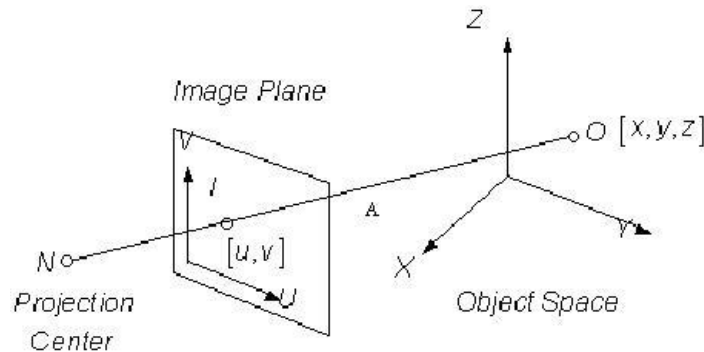
۳-۴-۱-۳- روش تبدیل خطی یا DLT

برای بازسازی سه‌بعدی مختصات و تبدیل مختصات دوبعدی تصاویر به مختصات سه‌بعدی فضای کاری، باید از روشهای کالیبراسیون استفاده شود.

کالیبراسیون دوربین همان به دست آوردن پارامترهای درونی شامل فاصله کانونی، مختصات نقطه اصلی در صفحه تصویر و نیز پارامترهای اعوجاج تصویر، و پارامترهای بیرونی شامل مختصات دوربین در فضای کاری و راستای محور اپتیکی است.

در روش DLT برای هر نقطه در فضای سه‌بعدی یک رابطه خطی بین مختصات دوبعدی تصویر روی فیلم و مختصات فضایی آن به دست می‌آید. معادلات این خطوط در واقع نگاشت یک نقطه خاص در فضای کاری بر فضای تصویر می‌باشند. بعد از نوشتن معادلات این نگاشتها و تبدیل مختصات، رابطه بین مختصات دوبعدی در تصویر و مختصات سه‌بعدی به دست می‌آید که شامل تعدادی پارامتر مجهول است. کالیبراسیون دوربین به معنای تعیین این پارامترهای مجهول از طریق وارد کردن داده‌های تعدادی از نقاط دارای مختصات معلوم (نقاط کنترلی) در معادلات DLT است.

برای مثال همانطور که در شکل (۳-۲۹) مشاهده می‌شود، O یک نقطه در فضای کاری است و I تصویر آن روی صفحه تصویر می‌باشد. در واقع I نتیجه نگاشت نقطه O روی صفحه تصویر است. اصولاً تمام بردارهای نگاشت از نقطه کاری (O) به نقاط تصویر (I) از نقطه N که اصطلاحاً نقطه گره ای (Projection Center) نامیده می‌شود، می‌گذرند (شرط هم‌راستایی).



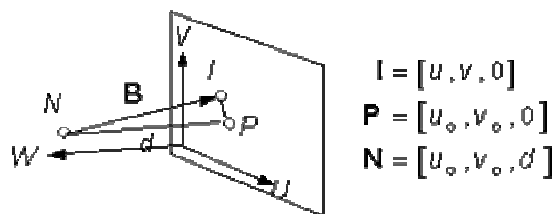
شکل (۳-۲۹) نگاشت بین فضای کاری و فضای تصویر

همانطور که مشخص است، X, Y و Z مختصات سه‌بعدی در فضای کاری و u و v مختصات دوبعدی تصویر هستند. اگر مختصات فضایی نقاط O و N را به ترتیب با $[x, y, z]$ و $[x_0, y_0, z_0]$ نشان دهیم، مختصات بردار A از O به N به صورت زیر خواهد بود:

$$\vec{A} = (x - x_0)\hat{i} + (y - y_0)\hat{j} + (z - z_0)\hat{k} \quad \text{رابطه (۳-۶)}$$

پس از آن برای دستیابی به مختصات سه‌بعدی تصویر، مطابق شکل (۳-۳۰)، خط w از نقطه N عمود بر صفحه تصویر رسم می‌شود و در نتیجه محور عمود بر u و v و نیز نقطه P یا نقطه اصلی با فاصله d (فاصله اصلی) از صفحه تصویر به دست می‌آید. بنابراین مختصات سه‌بعدی N و I در

سیستم مختصات تصویر به ترتیب $[u_0, v_0, d]$ و $[u, v, 0]$ خواهد بود.



شکل (۳-۳۰) نقاط اصلی و نقاط گره‌ای

در این حالت مختصات بردار B که از N به I کشیده می‌شود به شکل زیر خواهد بود:

$$\vec{B} = (u - u_0)\hat{e}_u + (v - v_0)\hat{e}_v + (-d)\hat{e}_w \quad \text{رابطه (۳-۷)}$$

که $\hat{e}_u$ و $\hat{e}_v$ و $\hat{e}_w$ به ترتیب بردارهای یکه محورهای u و v و w هستند. طبق شرط هم‌راستایی، بردار A در شکل (۳-۲۹) و بردار B در شکل (۳-۳۰) باید هم‌راستا باشند. اگر دو بردار در یک سیستم باشند شرط هم‌راستایی به صورت زیر برقرار خواهد بود:

$$\vec{B} = C.\vec{A} \quad \text{رابطه (۳-۸)}$$

در این معادله C یک مقدار ثابت است. بنابراین مختصات بردار A به صورت زیر به مختصات تصویر تبدیل شده و بردار A' به دست می‌آید.

$$\vec{A}' = R.\vec{A} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \vec{A} \quad \text{رابطه (۳-۹)}$$

در این معادله R ماتریس تبدیل مختصات فضای کاری به مختصات تصویر است. بنابراین طبق رابطه ۸ شرط هم‌راستایی به صورت زیر برقرار خواهد بود:

$$\vec{B} = C.\vec{A}' = C.R.\vec{A} = C \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} \quad \text{رابطه (۳-۱۰)}$$

با بسط معادله فوق، روابط (۳-۱۱) تا (۳-۱۳) بین مختصات دوبعدی و سه‌بعدی نقاط به دست می‌آید:

$u - u_0 = C.[r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)]$	رابطه (۱۱-۳)
$v - v_0 = C.[r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)]$	رابطه (۱۲-۳)
$-d = C.[r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)]$	رابطه (۱۳-۳)

در نتیجه می‌توان مقدار ثابت C را از معادله (۱۳-۳) به دست آورد:

$$C = \frac{-d}{[r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)]} \quad \text{رابطه (۱۴-۳)}$$

با جایگذاری معادله C در دو معادله (۱۰-۳) و (۱۱-۳)، رابطه بین مختصات دوبعدی (u, v) و سه‌بعدی (x, y, z) به این ترتیب خواهد بود:

$$\begin{aligned} u - u_0 &= -d \cdot \frac{r_{11}(x - x_0) + r_{12}(y - y_0) + r_{13}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)} \\ v - v_0 &= -d \cdot \frac{r_{21}(x - x_0) + r_{22}(y - y_0) + r_{23}(z - z_0)}{r_{31}(x - x_0) + r_{32}(y - y_0) + r_{33}(z - z_0)} \end{aligned} \quad \text{رابطه (۱۵-۳)}$$

با مرتب کردن روابط فوق، معادلات اصلی DLT به صورت زیر به دست خواهد آمد:

$$\begin{aligned} u &= \frac{L_1 x + L_2 y + L_3 z + L_4}{L_9 x + L_{10} y + L_{11} z + 1} \\ v &= \frac{L_5 x + L_6 y + L_7 z + L_8}{L_9 x + L_{10} y + L_{11} z + 1} \end{aligned} \quad \text{رابطه (۱۶-۳)}$$

$L1-L11$ پارامترهای DLT هستند که بر حسب پارامترهای موجود در معادله (۱۵-۳) در معادله

(۱۷-۳) به دست آمده‌اند.

می‌توان گفت منظور از کالیبراسیون دوربین به روش DLT، یافتن این ۱۱ پارامتر است. همانطور که پیش از این ذکر شد به دست آوردن این پارامترها از طریق اندازه‌گیری و در واقع به دست آوردن پارامترهای موجود در معادله (۱۵-۳) بسیار دشوار است و از دقت کافی برخوردار نیست.

بنابراین این پارامترها از طریق کالیبره کردن دوربین به دست می‌آیند. [۳،۲]

$$[d_u, d_v] = \left[\frac{d}{\lambda_u}, \frac{d}{\lambda_v} \right]$$

$$D = -(x_o r_{31} + y_o r_{32} + z_o r_{33})$$

$$L_1 = \frac{u_o r_{31} - d_u r_{11}}{D}$$

$$L_2 = \frac{u_o r_{32} - d_u r_{12}}{D}$$

$$L_3 = \frac{u_o r_{33} - d_u r_{13}}{D}$$

$$L_4 = \frac{(d_u r_{11} - u_o r_{31}) x_o + (d_u r_{12} - u_o r_{32}) y_o + (d_u r_{13} - u_o r_{33}) z_o}{D}$$

$$L_5 = \frac{v_o r_{31} - d_v r_{21}}{D}$$

$$L_6 = \frac{v_o r_{32} - d_v r_{22}}{D}$$

$$L_7 = \frac{v_o r_{33} - d_v r_{23}}{D}$$

$$L_8 = \frac{(d_v r_{21} - v_o r_{31}) x_o + (d_v r_{22} - v_o r_{32}) y_o + (d_v r_{23} - v_o r_{33}) z_o}{D}$$

$$L_9 = \frac{r_{31}}{D}$$

$$L_{10} = \frac{r_{32}}{D}$$

$$L_{11} = \frac{r_{33}}{D}$$

رابطه (۳-۱۷)

با مرتب کردن این معادلات بر حسب ۱۱ مجهول $L_1 - L_{11}$ رابطه ماتریسی زیر به دست می‌آید:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 x_1 & -u_1 y_1 & -u_1 z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1 x_1 & -v_1 y_1 & -v_1 z_1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} \quad \text{رابطه (۳-۱۸)}$$

معادله (۳-۱۶) معادله استاندارد سه‌بعدی DLT است، البته می‌توان خطاهای اپتیکی عدسی

دوربین را از این معادله کم کرد؛ به این صورت که:

$$u - \Delta u = \frac{L_1 x + L_2 y + L_3 z + L_4}{L_9 x + L_{10} y + L_{11} z + 1}$$

$$v - \Delta v = \frac{L_5 x + L_6 y + L_7 z + L_8}{L_9 x + L_{10} y + L_{11} z + 1}$$

رابطه (۱۹-۳)

در این معادله $[\Delta u, \Delta v]$ خطاهای اپتیکی هستند. خطاهای اپتیکی از روابط زیر به دست می‌آیند:

$$\Delta u = \xi(L_{12}r^2 + L_{13}r^4 + L_{14}r^6) + L_{15}(r^2 + 2\xi^2) + L_{16}\xi\eta$$

$$\Delta v = \eta(L_{12}r^2 + L_{13}r^4 + L_{14}r^6) + L_{16}(r^2 + 2\eta^2) + L_{15}\xi\eta$$

رابطه (۲۰-۳)

در این روابط:

$$[\xi, \eta] = [u - u_0, v - v_0]$$

$$r^2 = \xi^2 + \eta^2$$

رابطه (۲۱-۳)

سه پارامتر $L_{12} \dots L_{14}$ که در معادلات فوق به دلیل وجود تغییر شکل اپتیکی اضافه شده‌اند. در حالی که پارامترهای L_{15}, L_{16} بواسطه خارج از مرکز بودن این تغییرشکل در نظر گرفته شده‌اند. باید توجه داشت در روش DLT حداقل ۱۱ پارامتر باید وجود داشته باشد و برای دستیابی به جواب دقیق‌تر از تعداد پارامترهای بیشتری استفاده می‌شود. با توجه به رابطه (۱۶-۳) دو معادله و ۱۱ مجهول وجود دارد، در نتیجه حداقل ۱۱ معادله برای به دست آوردن این پارامترها ضروری است. پارامترهای معلوم در این معادله مختصات دوبعدی و سه‌بعدی مجموعه ای از نقاط هستند که به عنوان نقاط کنترلی به کار می‌روند و در واقع ۶ نقطه کنترلی (برای هر نقطه دو معادله) برای محاسبه پارامترها ضروری است. ماتریس حاصل از این شش معادله 11×11 خواهد بود که معکوس آن باید محاسبه شود. همانطور که مشخص است این ماتریس ذاتاً تعداد زیادی صفر دارد و معمولاً منفرد است. بنابراین از روش شبه معکوس استفاده می‌شود. شبه معکوس ماتریس A یک ماتریس B هم بعد AT است که در شرایط زیر صدق می‌کند:

$ABA = A$ $BAB = B$	
------------------------	--

این روش در واقع همان روش حداقل مربعات وزنی است که بهترین جواب را از راه کمینه کردن مربعات خطاها برای معادله پیدا می کند و هرچه تعداد معادلات بیشتر باشد دقت جواب بیشتر خواهد بود. در نتیجه برای افزایش دقت می توان از تعداد نقاط بیشتری استفاده کرد.

پس از به دست آوردن ۱۱ پارامتر ذکر شده، مختصات سه بعدی نقاط کاری از معادلات زیر به دست می آید:

$$\begin{bmatrix} L_1 - L_9 u & L_2 - L_{10} u & L_3 - L_{11} u \\ L_5 - L_9 v & L_6 - L_{10} v & L_7 - L_{11} v \\ L'_1 - L'_9 u' & L'_2 - L'_{10} u' & L'_3 - L'_{11} u' \\ L'_5 - L'_9 v' & L'_6 - L'_{10} v' & L'_7 - L'_{11} v' \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u - L_4 \\ v - L_8 \\ u' - L'_4 \\ v' - L'_8 \end{bmatrix} \quad \text{رابطه (۲۲-۳)}$$

همانطور که مشخص است، معادلات (۲۲-۳) از مرتب کردن روابط (۱۶-۳) بر حسب مجهولات z, y, x به دست می آید. برای به دست آوردن این سه مجهول، سه معادله و در نتیجه حداقل دو نما از فریم ضروری است. در این رابطه نیز مختصات دو بعدی دو نمای v, u و v', u' به ترتیب برای تصویر دوربین ۱ و دوربین ۲ برای حل معادله استفاده شده است. $L_1 - L_{11}$ و $L'_1 - L'_{11}$ نیز به ترتیب پارامترهای DLT برای دوربین های ۱ و ۲ هستند. در حالتی که روش استاندارد DLT به همراه پنج پارامتر اضافی و m دوربین ($m \geq 2$) مورد نظر باشد، رابطه فوق به شکل زیر خواهد بود:

$$\begin{bmatrix} \frac{v^{(1)} L_9^{(1)} - L_1^{(1)}}{R^{(1)}} & \frac{v^{(1)} L_{10}^{(1)} - L_2^{(1)}}{R^{(1)}} & \frac{v^{(1)} L_{11}^{(1)} - L_3^{(1)}}{R^{(1)}} \\ \frac{w^{(1)} L_9^{(1)} - L_5^{(1)}}{R^{(1)}} & \frac{w^{(1)} L_{10}^{(1)} - L_6^{(1)}}{R^{(1)}} & \frac{w^{(1)} L_{11}^{(1)} - L_7^{(1)}}{R^{(1)}} \\ \vdots & \vdots & \vdots \\ \frac{v^{(m)} L_9^{(m)} - L_1^{(m)}}{R^{(m)}} & \frac{v^{(m)} L_{10}^{(m)} - L_2^{(m)}}{R^{(m)}} & \frac{v^{(m)} L_{11}^{(m)} - L_3^{(m)}}{R^{(m)}} \\ \frac{w^{(m)} L_9^{(m)} - L_5^{(m)}}{R^{(m)}} & \frac{w^{(m)} L_{10}^{(m)} - L_6^{(m)}}{R^{(m)}} & \frac{w^{(m)} L_{11}^{(m)} - L_7^{(m)}}{R^{(m)}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{L_4^{(1)} - v^{(1)}}{R^{(1)}} \\ \frac{L_8^{(1)} - w^{(1)}}{R^{(1)}} \\ \vdots \\ \frac{L_4^{(m)} - v^{(m)}}{R^{(m)}} \\ \frac{L_8^{(m)} - w^{(m)}}{R^{(m)}} \end{bmatrix} \quad \text{رابطه (۲۳-۳)}$$

که در آن $[v, w] = [u - \Delta u, v - \Delta v]$ و $R^{(i)} = L^{(i)}_9 X + L^{(i)}_{10} Y + L^{(i)}_{11} Z + 1$ است. با استفاده

مجدد از روش حداقل مجموع مربعات می توان مختصات فضایی $[X, Y, Z]$ را به دست آورد.

۳-۴-۱-۴- موقعیت دوربین ها و مختصات نقطه اصلی

از معادله (۳-۱۷) می توان روابط زیر را به دست آورد:

$$\begin{aligned} L_1 X_0 + L_2 Y_0 + L_3 Z_0 &= -L_4 \\ L_5 X_0 + L_6 Y_0 + L_7 Z_0 &= -L_8 \\ L_9 X_0 + L_{10} Y_0 + L_{11} Z_0 &= -1 \end{aligned} \quad \rightarrow \begin{bmatrix} L_2 & L_3 \\ L_6 & L_7 \\ L_{10} & L_{11} \end{bmatrix} \cdot \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix} \quad \text{رابطه (۳-۲۴)}$$

با نوشتن معادلات فوق می توان مختصات فضایی مرکز پروژکشن را به دست آورد. همچنین

می توان به طور مشابه از معادلات (۳-۱۷) معادلات زیر را به دست آورد:

$$\begin{aligned} L_9^2 + L_{10}^2 + L_{11}^2 &= \frac{1}{D^2} [r_{31}^2 + r_{32}^2 + r_{33}^2] = \frac{1}{D^2} \\ (DL_1)(DL_9) + (DL_2)(DL_{10}) + (DL_3)(DL_{11}) &= u_0 [r_{31}^2 + r_{32}^2 + r_{33}^2] - d_u [r_{11} r_{31} + r_{12} r_{32} + r_{13} r_{33}] = u_0 \\ (DL_5)(DL_9) + (DL_6)(DL_{10}) + (DL_7)(DL_{11}) &= v_0 \end{aligned} \quad \text{رابطه (۳-۲۵)}$$

$$u_0 = D^2 (L_1 L_9 + L_2 L_{10} + L_3 L_{11}) = \frac{L_1 L_9 + L_2 L_{10} + L_3 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2}$$

$$v_0 = D^2 (L_5 L_9 + L_6 L_{10} + L_7 L_{11}) = \frac{L_5 L_9 + L_6 L_{10} + L_7 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2}$$

معادلات فوق بر اساس شرایط تعامد ماتریس تبدیل R برقرار است، یعنی:

$$\sum_{k=1}^3 r_{ik} r_{jk} = \delta_{ij} \quad \text{رابطه (۳-۲۶)}$$

$$\sum_{k=1}^3 r_{ki} r_{kj} = \delta_{ij}$$

می‌توان گفت حل معادله با روش حداقل مربعات به دلیل وجود خطاهای تجربی و اندازه‌گیری، شرایط تعامد را تضمین نمی‌کند. این موضوع یکی از مشکلات ذاتی روش DLT به شمار می‌رود. به همین دلیل در سال ۱۹۸۸ روش DLT بهبود یافته توسط هاتز ارائه شد.

۳-۴-۱-۵- روش DLT بهبود یافته

در روش DLT بهبود یافته که به اختصار MDLT نامیده می‌شود، معادله تعامدپذیری با معادلات استاندارد DLT ترکیب شده است و عملاً تعامدپذیری ماتریس تبدیل آن که در معادله (۳-۲۶) نشان داده شده است وارد معادلات می‌شود. این معادله تعامدپذیری بدین صورت به دست می‌آید که از معادله (۳-۲۵) که بر اساس شرط تعامد پذیری بدست آمده می‌توان نوشت [۱۳]:

$\frac{u_0}{D^2} = L_1 L_9 + L_2 L_{10} + L_3 L_{11}$	رابطه (۳-۲۷)
$\frac{v_0}{D^2} = L_5 L_9 + L_6 L_{10} + L_7 L_{11}$	رابطه (۳-۲۸)

از حاصلضرب معادلات (۳-۲۷) و (۳-۲۸)، و جایگذاری مقادیر u_0, v_0 و اعمال شرایط تعامدپذیری می‌توان نوشت:

$$(L_1 L_9 + L_2 L_{10} + L_3 L_{11}) \cdot (L_5 L_9 + L_6 L_{10} + L_7 L_{11}) = (L_1 L_5 + L_2 L_6 + L_3 L_7) \cdot (L_9^2 + L_{10}^2 + L_{11}^2) \quad \text{رابطه (۳-۲۹)}$$

به معادله (۳-۲۹) معادله تعامدپذیری گفته می‌شود. برای ترکیب این معادله و معادله استاندارد

DLT می‌توان L_1 را از معادله (۳-۲۹) بر حسب $L_2 \dots L_{11}$ نوشت:

$$L_1 = f(L_2 \dots L_{11}) \quad \text{رابطه (۳-۳۰)}$$

با ترکیب معادلات (۳-۳۰) و (۳-۱۷) و (۳-۱۹) می‌توان معادله (۳-۳۱) را به دست آورد:

$$u = f(L_2 \dots L_{11}) + \Delta u$$

$$v = g(L_2 \dots L_{11}) + \Delta v$$

رابطه (۳-۳۱)

معادلات فوق معادلات اصلی روش MDLT هستند. اگر Δu و Δv صفر فرض شوند، در آن صورت می‌توان نوشت:

$$\begin{aligned} u &= f(L_2 \dots L_{11}) \\ v &= g(L_2 \dots L_{11}) \end{aligned} \quad \text{رابطه (۳۲-۳)}$$

معادلات فوق دو معادله غیرخطی بر حسب $L_2 \dots L_{11}$ است که برای حل آن از تقریب تیلور برای خطی کردن و صرفنظر از مشتقات مرتبه بالاتر و روش تکرار استفاده می‌شود:

$$\begin{aligned} u &= f_0 + \frac{\partial f_0}{\partial L_i} \cdot d(L_i) \\ v &= g_0 + \frac{\partial g_0}{\partial L_i} \cdot d(L_i) \end{aligned} \quad \text{رابطه (۳۳-۳)}$$

که در آن f_0 و g_0 از رابطه زیر به دست می‌آیند:

$$\begin{aligned} f_0 &= f[L_i(\text{PreviousStep})] \\ g_0 &= g[L_i(\text{PreviousStep})] \end{aligned} \quad \text{رابطه (۳۴-۳)}$$

در معادلات (۳۳-۳) عبارات $\frac{\partial g_0}{\partial L_i} \cdot d(L_i), \frac{\partial f_0}{\partial L_i} \cdot d(L_i), f_0, g_0$ با استفاده از مراحل قبلی محاسبه

شده‌اند. مقادیر اولیه L_i نیز با روش استاندارد DLT به دست می‌آید. با استفاده از روش مجموع کمترین مربعات مقادیر $d(L_i)$ محاسبه می‌گردد. سپس با استفاده از رابطه زیر L_i در هر مرحله به دست می‌آید:

$$L_i(\text{PresentStep}) = d(L_i) + L_i(\text{PreviousStep}) \quad \text{(۳۵-۳)}$$

۳-۵- طراحی و کدنویسی نرم‌افزاری تبدیل مختصات سه‌بعدی

پس از پردازش تصویر و دریافت مختصات دوبعدی موقعیت توپ لازم است مختصات سه‌بعدی موقعیت توپ بازسازی شود. این کار را می‌توان با استفاده از روش تبدیل خطی مستقیم انجام داد.

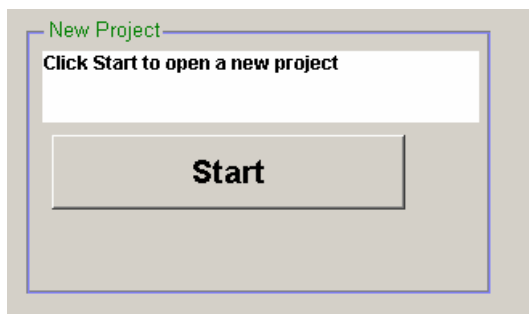
برای بررسی حرکت یک توپ و به دست آوردن مختصات سه‌بعدی بازسازی شده آن، تصویربرداری در دو مرحله با حداقل دو دوربین صورت می‌گیرد. در ابتدا تصاویری از چارچوب کالیبراسیون گرفته شده و سپس بدون تغییر دادن وضعیت دوربین‌ها، از اجرای حرکت مورد نظر در فضای چارچوب کالیبراسیون تصویربرداری صورت می‌گیرد. بنابراین در صورتی که تصویربرداری با دو دوربین انجام گیرد به کمک نرم‌افزار پردازش تصویر، چهار فایل حاوی مختصات پیکسلی حاصل از تصویر چارچوب و حرکت از دید هر دوربین به دست خواهد آمد که حاوی مختصات دوبعدی نقاط کنترلی چارچوب کالیبراسیون و موقعیت توپ در زمین می‌باشد. برای بازسازی سه‌بعدی، علاوه بر داشتن مختصات نقاط کنترلی مربوط به چارچوب کالیبراسیون که از تصویربرداری به دست آمده است، لازم است مختصات واقعی این نقاط از طریق اندازه‌گیری دقیق نیز ثبت شده باشد.

برای دستیابی به مختصات سه‌بعدی موقعیت توپ نرم‌افزاری با نام Transform در محیط MATLAB V7.0.4 طراحی و کدنویسی شد. همانطور که پیش از این نیز ذکر شد نرم‌افزار MATLAB به دلیل برخورداری از قابلیت‌های متعدد در محاسبات ماتریسی، وجود ابزارهای مختلف برای پردازش داده‌ها نظیر هموارسازی داده‌ها و همچنین امکان برقراری ارتباط با نرم‌افزار Excel، برای پیاده‌سازی این نرم‌افزار انتخاب شد. در واقع ورودی نرم‌افزار Transform مختصات موقعیت توپ و نقاط کنترلی چارچوب کالیبراسیون بر حسب پیکسل و مختصات واقعی نقاط کنترلی به صورت فایل‌های Excel است. پس از دریافت این فایل‌ها به صورت مجزا، با اعمال روش هموارسازی انتخاب شده توسط کاربر می‌توان یکی از روشهای بازسازی مختصات سه‌بعدی DLT یا MDLT را اجرا نموده و مختصات سه‌بعدی بازسازی شده موقعیت توپ را به صورت فایل Excel دریافت کرد.

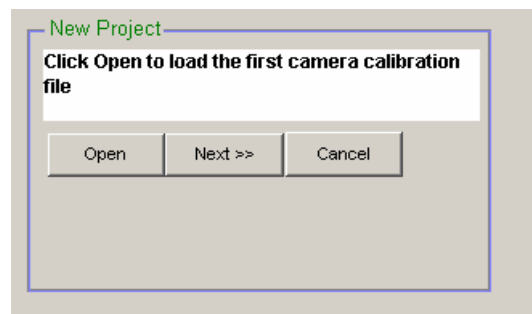
در ادامه توضیحات تکمیلی برای کار با نرم‌افزار Transform و قابلیت‌های برنامه برای دریافت خروجی‌های لازم ارائه می‌گردد. جزئیات کدنویسی این برنامه در پیوست ۵ آمده است.

۳-۵-۱- دریافت فایل‌های مختصات موقعیت توپ بر حسب پیکسل

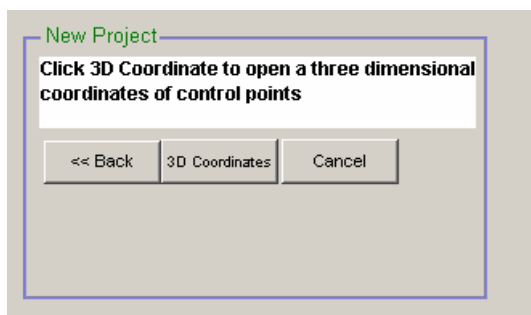
برای شروع یک بازسازی جدید پس از اجرای برنامه Transform، گزینه New از منوی File یا دکمه Start باید انتخاب گردد. در این مرحله ابتدا تعداد دوربین‌های مورد استفاده در تصویربرداری انتخاب می‌گردد، سپس به کمک توضیحات بالای پنجره می‌توان فایل‌های مناسب را در برنامه بارگذاری نمود. بعنوان مثال در صورتی که تصویربرداری به کمک دو دوربین صورت گرفته باشد، فایل‌های مختصات موقعیت توپ و چارچوب کالیبراسیون را باید در چهار مرحله انتخاب نمود. در هر مرحله با انتخاب دکمه Back امکان برگشت به مرحله قبلی میسر است. در مرحله آخر با انتخاب دکمه Real Coordinates، فایل مختصات واقعی نقاط کنترلی از کاربر درخواست می‌شود. پس از اتمام مراحل بارگذاری، به منظور تایید مراحل انجام شده باید دکمه Finish انتخاب شود (شکل (۳-۳۱)). با انتخاب دکمه Cancel در هر مرحله می‌توان از ادامه روند بارگذاری فایل‌ها صرفنظر نمود.



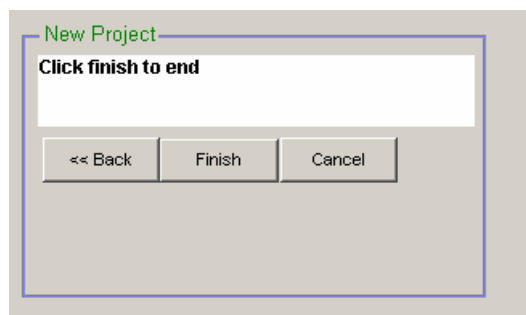
شروع بارگذاری فایل‌ها



بارگذاری یکی از فایل‌های پیکسلی چارچوب کالیبراسیون



بارگذاری فایل مختصات واقعی نقاط کنترلی

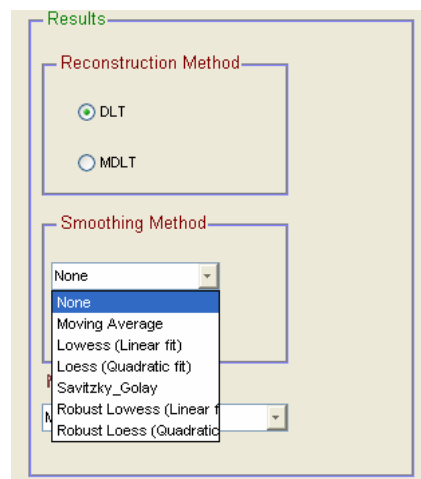
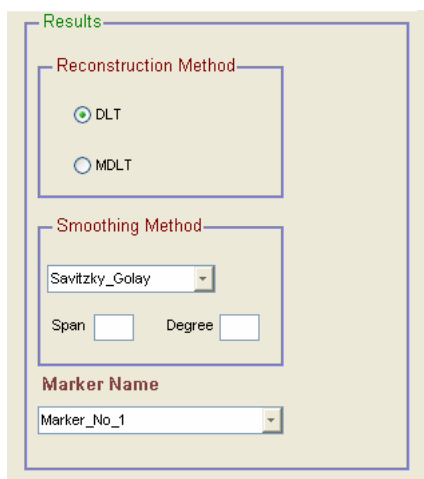


اتمام بارگذاری

شکل (۳-۳۱) توالی عملکرد برنامه Transform برای بارگذاری فایل‌های مختصات

۳-۵-۲- انتخاب روش هموارسازی و بازسازی

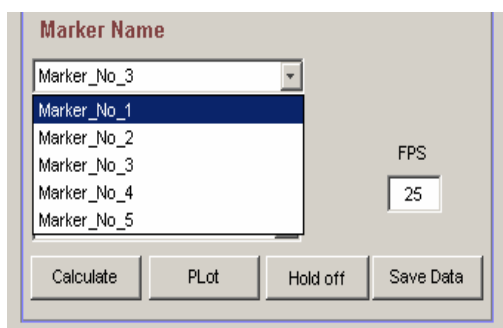
پس از بارگذاری فایل‌های مختصات، با انتخاب یکی از روشهای هموارسازی در منوی کرکره‌ای مربوطه می‌توان داده‌های پیکسلی را هموار نمود. برای بازسازی مختصات سه‌بعدی موقعیت توپ و به دست آوردن ضرایب کالیبراسیون دوربین‌ها یکی از روش‌های DLT یا MDLT از دکمه رادیویی مربوطه باید انتخاب گردد (شکل (۳-۳۲)).



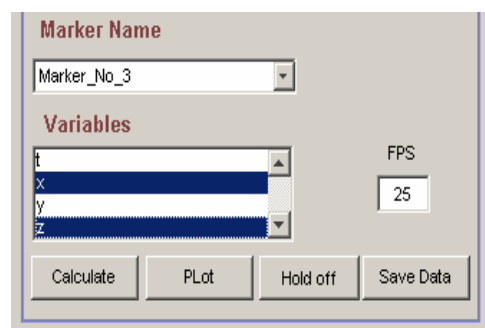
شکل (۳-۳۲) انتخاب روش بازسازی و هموارسازی

۳-۵-۳- دریافت خروجی‌ها

پس از انتخاب روش بازسازی و هموارسازی، با انتخاب دکمه Calculate توابع DLT و یا MDLT که در محیط MATLAB نوشته شده اند، اجرا می‌شوند و می‌توان ضرایب کالیبراسیون دوربین‌ها را مشاهده نمود. در قسمت FPS باید سرعت تصویربرداری بر حسب فریم بر ثانیه وارد گردد. این مقدار به صورت پیش فرض ۲۵ فریم بر ثانیه در نظر گرفته شده است. برای مشاهده مختصات بازسازی شده موقعیت توپ بر حسب هم و یا بر حسب زمان، ابتدا از منوی کرکره‌ای Markers name، توپ و سپس دو متغیر مورد نظر از لیست Variables انتخاب می‌گردند. با انتخاب دکمه Plot، نمودار دو متغیر انتخاب شده ترسیم می‌گردد. در صورت تمایل برای ترسیم بیش از یک نمودار برای یک توپ و یا دستیابی به دو متغیر خاص برای توپ، دکمه Hold on را می‌توان فعال نمود (شکل (۳-۳۳)). با انتخاب دکمه Save Data، مختصات سه‌بعدی بازسازی شده تمام موقعیت توپ در مسیر انتخاب شده توسط کاربر ذخیره خواهد شد.



انتخاب یک توپ خاص



انتخاب متغیرهای مورد نظر

شکل (۳-۳۳) انتخاب روش بازسازی و هموارسازی

با انتخاب گزینه Exit از منوی File می‌توان از برنامه Transform خارج شد.

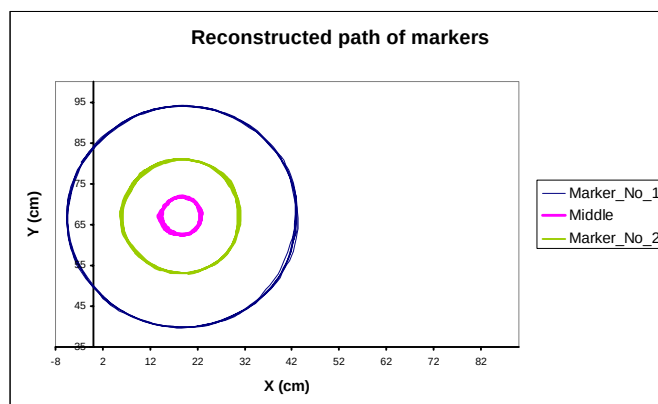
۳-۶- نمونه‌ای از اجرای نرم افزار تهیه شده

برای بررسی صحت نرم افزار تهیه شده آزمایشی تدارک دیده شد که در آن از ژيروسکوپ که دارای حرکت در صفحه افقی می‌باشد استفاده شد (شکل ۳-۳۴).



شکل (۳-۳۳) دستگاه ژيروسکوپ

بدین منظور سه مارکر غیر همراهی بر روی بخش دوار دستگاه نصب گردید. با استفاده از معادله حاکم بر حرکت دستگاه و اندازه گیری زمان ۱۰ دور چرخش ژيروسکوپ، سرعت زاویه ای دستگاه تعیین می‌شود. این آزمایش برای پنج سرعت زاویه ای مختلف تکرار گردید. پس از پردازش تصویر، مختصات سه بعدی مارکرها با استفاده از بخش 3D Reconstruction بازسازی شدند. شکل (۳-۳۵)، نمودار مسیر بازسازی شده حرکت مارکرها را نشان می‌دهد.



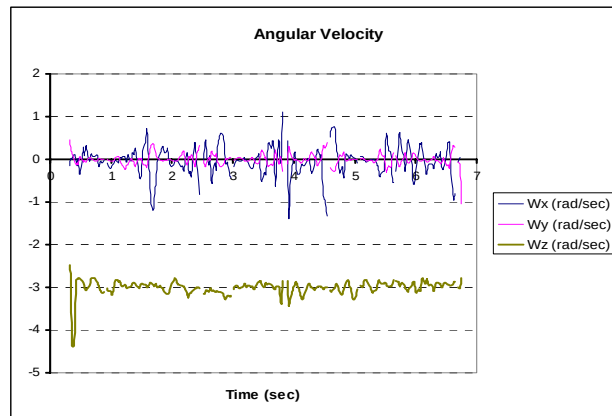
شکل (۳-۳۴) نمودار مسیر بازسازی شده مارکرها

در جدول زیر سرعت زاویه ای واقعی، سرعت زاویه ای محاسبه شده و درصد خطا برای پنج آزمایش آورده شده است.

جدول (۳-۱) مقایسه سرعت زاویه ای واقعی و محاسبه شده در آزمایش ژيروسکوپ

شماره آزمایش	۱	۲	۳	۴	۵
سرعت زاویه ای واقعی (rad/sec)	4.30	4.11	3.85	2.91	2.42
سرعت زاویه ای محاسبه شده (rad/sec)	4.35	4.09	3.81	2.98	2.42
خطا (%)	1.1	0.49	1.04	2.40	0.0

نمودار مربوط به سرعت‌های زاویه‌ای بر حسب زمان به ترتیب در شکل (۳-۳۶) آورده شده است.



شکل (۳-۳۵) تغییرات سرعت زاویه‌ای بر حسب زمان برای حرکت بخش دوار ژيروسکوپ (نتایج آزمایش ۴)

همانطور که در شکل (۳-۳۵) مشاهده می‌شود، مسیر بازسازی شده مارکرها به صورت دایره

متحدالمرکز بوده که با مسیر واقعی آنها تطابق خوبی دارد.

فصل ۴

تئوری کنترل پیش بین

کنترل پیش بین چنانکه از نامش بر می آید، یک روش کنترلی است که در آن پیش بینی رفتار آینده یک سیستم در فرمول بندی مسئله کنترل وارد می شود.

پیش بینی، مقادیر آینده یک متغیر را بر مبنای اطلاعات در دسترس سیستم به دست می دهد. هر چه اطلاعات نمایانگری از سیستم داشته باشیم، دقت پیش بینی بهتر خواهد بود. آنگاه می توان تخمین متغیرهای آینده سیستم را برای طراحی قانون کنترل به کاربرد تا عملکرد بهتر کنترلی، که معمولاً رسانیدن یا نگهداشتن خروجی در یک مقدار مطلوب است، بدست آید. این دسته از روشهای کنترلی که اطلاعات یا فرضیات مربوط به آینده سیستم را به هم پیوند می دهند، در حالت کلی کنترل پیش بین خوانده می شوند.

کنترل پیش بین معمولاً هنگامی به کار می رود که عملکرد بهتری نسبت به آنچه از کنترل غیر پیش بین بدست می آید، مورد نیاز باشد. این حالت شامل سیستمهایی می شود که رفتار آینده آنها می تواند کاملاً متفاوت از آنچه اکنون از آنها دیده می شود باشد، مانند سیستمهای با تاخیر زمانی، سیستمهای با مرتبه بالا، میرای ضعیف و غیر کمینه فاز .

کنترل پیش بین یا کنترل پیش بین مبتنی بر مدل (MPC or MBPC) که کنترل با افق پس رونده (Receding Horizon Control) یا کنترل با افق متحرک نیز نامیده می شود، به مفهوم امروزی از دهه هفتاد مطرح گردیده و از آن هنگام تاکنون به طرز قابل ملاحظه ای توسعه یافته است. واژه MPC به یک استراتژی به خصوص کنترل اختصاص ندارد بلکه گستره وسیعی از روشهای کنترلی را دربر می گیرد که در کمینه سازی تابع هدف برای بدست آوردن سیگنال کنترل از مدل صریح سیستم استفاده می کنند. کنترل پیش بین به دسته ای از الگوریتم های

کنترلی گفته می‌شود که دنباله ای از ورودی های کنترلی را با بهینه سازی رفتار آینده سیستم تعیین می‌کنند.

روش های MPC ، عموماً به کنترل کننده های خطی منجر می‌شوند. ایده هایی که با شدت و ضعف تقریباً در تمام روشهای کنترل پیش بین مورد استفاده قرار می‌گیرند عبارتند از:

- استفاده صریح از مدل برای پیش بینی خروجی فرآیند در بازه های زمانی (افق های) آینده.

- محاسبه دنباله کنترل که یک تابع هزینه را کمینه می‌کند.

- استراتژی پس روندگی (Receding Strategy): در این استراتژی در هر گام زمانی افق پیش

بینی به جلو انتقال می‌یابد. همچنین فقط اولین مقدار از دنباله سیگنال کنترل که در هر گام

محاسبه می‌شود، به سیستم اعمال می‌گردد.

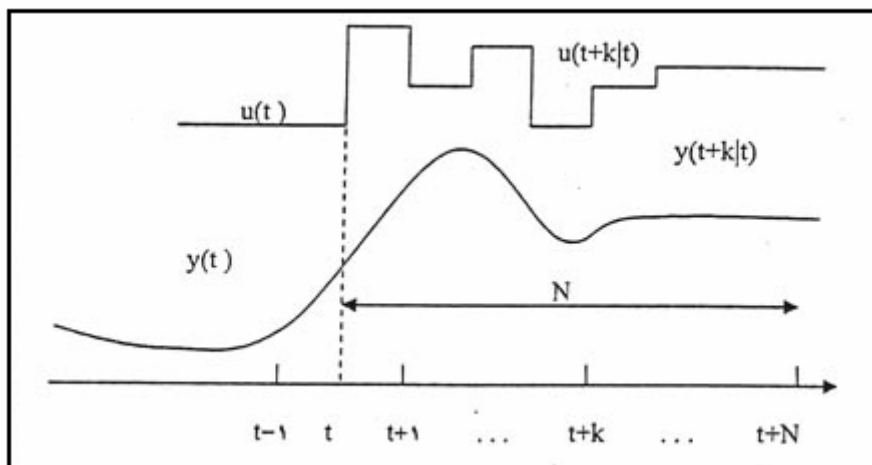
تفاوت عمده الگوریتمهای مختلف کنترل پیش بین، در مدل هایی است که برای نمایش فرآیند و

نویز به کار می‌گیرند و همچنین در توابع هزینه ای که کمینه می‌کنند.

۴-۲- استراتژی کلی کنترل پیش بین (MPC)

به طور کلی، همانطور که در شکل (۴-۱) نشان داده شده است، مراحل الگوریتم همه کنترل

کننده هایی که به خانواده MPC تعلق دارند، از این قرار است:



شکل (۴-۱) استراتژی MPC

۱- در هر گام، پیش بینی خروجی های آینده برای افق تعیین شده ای مانند N ، که افق پیش بینی نامیده می شود، با استفاده از مدل فرآیند بدست می آید. این خروجی های پیش بینی شده $(k = 1 \dots N, y(t+k | t))$ به مقادیر معلوم تا لحظه t (ورودی ها و خروجی های گذشته) و سیگنال های کنترل در گام های آتی $(k = 0 \dots N-1, u(t+k | t))$ که بعداً باید محاسبه شوند، بستگی دارد.

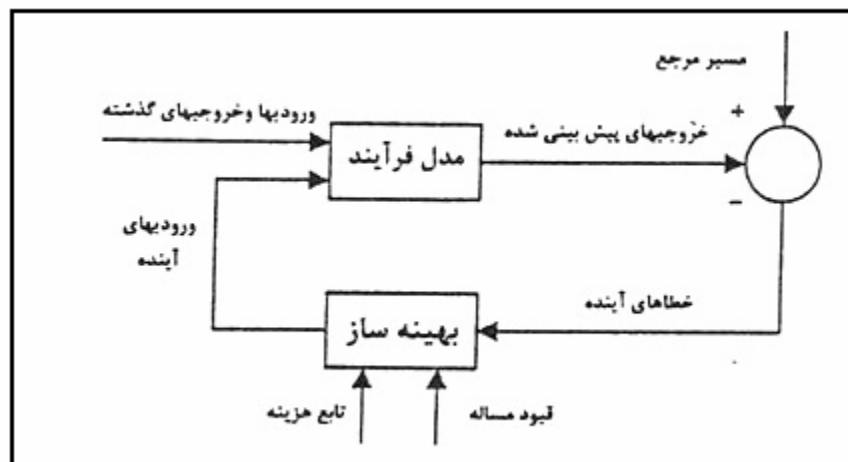
۲- مجموعه ای از سیگنال های کنترل، با بهینه سازی یک معیار معین به دست می آید. از آنجا که مطلوب است خروجی سیستم به یک پروفایل ورودی یا مسیر مرجعی مانند $\omega(t+k)$ نزدیک باشد، این معیار عموماً یک تابع درجه دوم از اختلاف خروجی پیش بینی شده و مسیر مرجع پیش بینی شده است. هزینه کنترل نیز در اغلب موارد در تابع معیار وارد می شود.

اگر تابع هزینه درجه دوم و مدل فرآیند خطی باشد و محدودیتی در نظر گرفته نشده باشد (مثل محدودیت روی سیگنال کنترل برای مدل کردن محدودیت های Actuator)، حل صریح مساله

امکان پذیر است و برای سیگنال کنترل فرمول بسته ای بدست می‌آید، در غیر اینصورت باید از روشهای بهینه سازی تکراری برای حل مساله طراحی استفاده کرد.

۳- سیگنال کنترل $u(t | t)$ به فرآیند اعمال می‌شود و باقی عناصر دنباله کنترل محاسبه شده کنار گذاشته می‌شوند. زیرا در گام بعدی نمونه برداری، $y(t+1)$ معلوم است و مرحله اول الگوریتم باید با این مقدار جدید تکرار شود و همه دنباله ها به روز گردند. بنابراین $u(t+1 | t+1)$ (که اصولاً به دلیل وجود اطلاعات جدید با $u(t+1 | t)$ متفاوت است) بر اساس مفهوم افق پس رونده، مجدداً محاسبه و به فرآیند اعمال می‌شود.

برای پیاده سازی این استراتژی، ساختار ساده نشان داده شده در شکل (۴-۲) به کار می‌رود.



شکل (۴-۲) ساختار کلی MPC

۴-۲-۱- مدل فرآیند

یک مدل برای پیش بینی خروجی های آینده فرآیند مورد استفاده قرار می‌گیرد. در این پیش بینی از مقادیر لحظات قبلی و زمان حال ورودی و خروجی ها و از دنباله سیگنال کنترل پیشنهاد شده از بهینه سازی، استفاده می‌گردد.

مدل فرآیند نقش مهمی در الگوریتم کنترل ایفا می‌کند. مدل انتخاب شده باید به خوبی دینامیک فرآیند را توصیف کند تا بتواند به طور دقیق خروجی های آینده را پیش بینی نماید. همچنین، مطلوب است که مدل حتی الامکان ساده باشد تا پیاده سازی الگوریتم پیش بینی، ساده و کم هزینه باشد.

مدل های مختلفی در فرمول بندی استراتژی های متفاوت MPC به کار می‌روند:

۱ - مدل پاسخ ضربه: یکی از مدل های مطلوب در صنعت مدل پاسخ ضربه بریده شده است. برای تعیین این مدل، خروجی فرآیند را در پاسخ به ورودی ضربه، اندازه گیری می‌کنند. این روش بعنوان یک سری وزنی یا مدل کانولوشن در روش MAC و به عنوان یک حالت خاص در GPC و EPSAC بکار می‌رود. رابطه خروجی با ورودی طبق معادله (۴-۱) است:

$$y(t) = \sum_{i=1}^{\infty} h_i u(t-i) \quad (۴-۱)$$

که در آن h_i ، نمونه های خروجی است هنگامی که فرآیند با ضربه واحد تحریک شده، با فرکانس خاصی از آن نمونه برداری شود. این جمع (سری) برای فرآیندهای پایدار بدون انتگرال گیر تنها با N مقدار مطابق رابطه زیر قطع خواهد شد:

$$y(t) = \sum_{i=1}^N h_i u(t-i) + e(t) = H(q^{-1})u(t)$$

$$H(q^{-1}) = h_1 q^{-1} + h_2 q^{-2} + \dots h_N q^{-N} \quad (۴-۲)$$

معادله پیش بینی بصورت زیر داده می‌شود.

$$\hat{y}(t+k/t) = \sum_{i=1}^N h_i u(t+k-i/t) = H(q^{-1})u(t+k/t) \quad (۴-۳)$$

مشکل عمده این روش آن است که پارامترهای زیادی برای مدل کردن پاسخ ضربه باید تعیین شود و ضمناً فقط برای فرآیندهای پایدار می‌تواند به کار رود و نیز N مقدار بزرگی است (در حدود ۴۰-۵۰)

این روش بطور وسیع در صنعت کاربرد دارد چون بسیار حسی است و بطور واضح اثر هر سیگنال کنترل را در یک خروجی معین منعکس می‌کند. اگر فرآیند چند متغیره باشد خروجیهای مختلف اثر ورودی را مطابق رابطه (۴-۴) منعکس می‌کنند.

$$y_j(t) = \sum_{k=1}^M \sum_{i=1}^N h_i^{kj} u^k(t-i) \quad (4-4)$$

یک مزیت بزرگ این روش این است که هیچ اطلاعات اولیه در مورد فرآیند نیاز نیست بنابراین شناسایی فرآیند ساده است و در همان حال دینامیکهای پیچیده مانند غیر مینیمم فاز بودن و یا تاخیر به آسانی قابل بررسی است.

۲- مدل پاسخ پله: ارتباط نزدیکی با مدل پاسخ ضربه دارد. در واقع با اعمال ورودی پله و اندازه گیری خروجی سیستم می‌توان این مدل را تعیین کرد. این مدل در روش DMC و روشهای مشابه استفاده می‌شود. برای سیستمهای پایدار پاسخ قطع شده بصورت (۴-۵) است.

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t-i) + e(t) = y_0 + G(q^{-1})(1-q^{-1})u(t) \quad (5-4)$$

که g_i نمونه های خروجی برای ورودی پله و $\Delta u(t) = u(t) - u(t-1)$ است. بنابراین معادله پیش بینی بصورت زیر خواهد بود.

$$\hat{y}(t+k/t) = y_0 + \sum_{i=1}^N g_i \Delta u(t+k-i/t) \quad (6-4)$$

مقدار y_0 را بدون اینکه به عمومیت مساله خلی وارد شود، می‌توان حذف کرد.

این روش مزایا و معایب روش قبلی را دارد و کنترل کننده از نوع یک است.

۳- مدل تابع تبدیل: مدل مورد استفاده دیگر، مدل تابع تبدیل است که بیشترین کاربرد را در مطالعات نظری مهندسی کنترل دارد. این مدل در روشهای EHAC، MUSMAR، MURHAC، مورد استفاده قرار می گیرد و از مفهوم تابع تبدیل با رابطه $C=B/A$ که بصورت (۷-۴) زیر تعریف می شود استفاده می کند.

$$\begin{aligned} A(q^{-1})y(t) &= B(q^{-1})u(t) \\ A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \\ B(q^{-1}) &= b_1q^{-1} + b_2q^{-2} + \dots + b_{n_b}q^{-n_b} \end{aligned} \quad (7-4)$$

معادله پیش بینی به صورت رابطه زیر است:

$$\hat{y}(t+k/t) = \frac{B(q^{-1})}{A(q^{-1})} u(t+k/t) \quad (8-4)$$

این روش برای سیستم های ناپایدار نیز بکار می رود و این مزیت را دارد که تنها نیاز به تعداد کمی پارامتر داشته، برای همه انواع فرآیندهای قابل اعمال است. همچنین اطلاعات اولیه ای در مورد فرآیند نیاز است، به خصوص مرتبه چند جمله ای های A، B باید مشخص باشد.

۴- مدل فضای حالت: مدل سازی به صورت معادلات حالت نیز در بعضی از فرمول بندیها مثل روش PFC مورد استفاده قرار می گیرد که مطابق رابطه زیر است:

$$\begin{cases} x(t) = Mx(t-1) + Nu(t-1) \\ y(t) = Qx(t) \end{cases} \quad (9-4)$$

x(t) بردار حالت، M، N، Q به ترتیب ماتریسهای سیستم، ورودی و خروجی هستند. پیش بینی برای این مدل طبق رابطه زیر است:

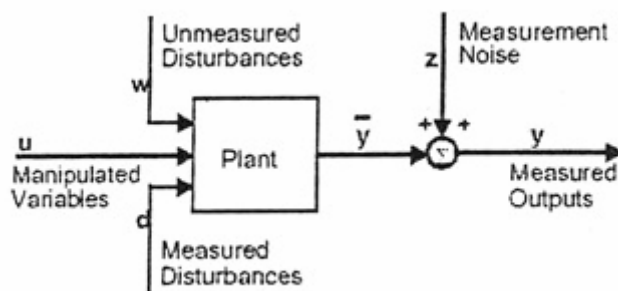
$$\hat{y}(t+k/t) = Q\hat{x}(t+k/t) = Q[M^k x(t) + \sum_{i=1}^k M^{i-1} Nu(t+k-i/t)] \quad (۱۰-۴)$$

مزیت این روش آن است که به راحتی می‌تواند برای فرآیندهای چند متغیره بکار رود. قانون کنترل از فیدبک ترکیب خطی بردار حالت بدست می‌آید، گر چه ممکن است حالت، تعبیر فیزیکی نداشته باشد. بنابراین اگر حالتها در دسترس نباشند بدلیل نیاز به استفاده از رویت گر ممکن است محاسبات پیچیده گردد.

علاوه بر روشهای فوق روشهای دیگری نیز برای مدلسازی فرآیند بکار می‌رود از جمله روشهای غیر خطی، فازی، عصبی و که بیشتر برای سیستم های غیر خطی به کار می‌روند و البته مساله بهینه سازی در آن صورت بسیار دشوار می‌شود.

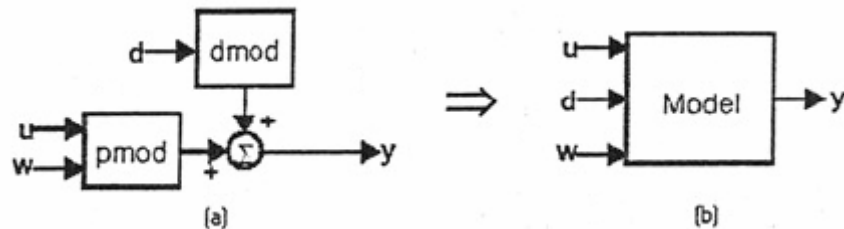
۲-۲-۴- مدل اغتشاش

برخی از روشهای MPC اغتشاشات قابل اندازه گیری را در مدل خود بکار برده و بوسیله کنترل پیشخور اثر آن را جبران می‌کنند. در شکل (۳-۴) یک سیستم واقعی که در آن نویز اندازه گیری و اغتشاش و نیز محل اثر آنها مشخص شده، نشان داده شده است.



شکل (۳-۴) سیستم واقعی با در نظر گرفتن نویز و اغتشاش

برای مدل‌سازی چنین سیستمی در شکل (۴-۴-۴) نویز و اغتشاشات غیر قابل اندازه‌گیری بصورت جمله d با دینامیک $dmod$ که در خروجی سیستم اثر می‌کند، همچنین اغتشاشات قابل اندازه‌گیری بصورت جمله w نشان داده شده است.



شکل (۴-۴) مدل سیستم با در نظر گرفتن نویز و اغتشاش

مدل اغتشاش می‌تواند به منظور بیان رفتاری که بوسیله مدل سیستم منعکس نمی‌شود (شامل اثر ورودیهای غیر قابل اندازه‌گیری، نویز و خطاهای مدل) باشد. بنابراین مدل می‌تواند به دو قسمت تقسیم شود: مدل فرآیند واقعی و مدل اغتشاش و هر دو برای پیش‌بینی لازم است. در شکل (۴-۴-۵) مدل سیستم بصورت یک بلوک دینامیکی که ورودیهای آن u , d , w و خروجی آن y است، نشان داده شده است.

انتخاب مدل برای نشان دادن اغتشاشات به همان اندازه مدل فرآیند مهم است. مدلی که فراوان استفاده می‌شود مدل CARIMA است که در آن اغتشاش به عنوان جمله جمع شونده با خروجی بصورت زیر معرفی می‌شود.

$$n(t) = \frac{C(z^{-1})}{D(z^{-1})} e(t) \quad (۴-۱۱)$$

در رابطه فوق چند جمله‌ای $D(z^{-1})$ شامل جمله $\Delta = 1 - z^{-1}$ است و $e(t)$ نویز سفید با میانگین صفر است. این مدل برای دو نوع اغتشاش مناسب است، تغییرات اتفاقی در زمانهای غیر قابل پیش

بینی (برای مثال تغییرات در کیفیت مواد) و حرکت براونی (در فرآیندهای تعادل انرژی) و بطور مستقیم در روشهای GPC، EPSAC، EHAC، UPC استفاده می‌شود. یادآوری می‌شود با در نظر گرفتن انتگرال گیر، کنترل بدون خطای ماندگار قابل دسترسی است.

۴-۲-۳- بهینه سازی و تابع هدف

بخش بهینه ساز جزء اساسی دیگری از استراتژی MPC است (شکل ۴-۲) که سیگنال کنترل را فراهم می‌سازد. اگر تابع هزینه درجه دوم باشد، سیگنال کنترلی به صورت تابعی از مقادیر قبلی و مسیر مرجع آینده قابل محاسبه است. در حضور محدودیت هایی که با نابرابری توصیف می‌شوند، حل مساله با روش های عددی که محاسبات بیشتری دارند، ممکن خواهد بود. حجم مسائل بهینه سازی به تعداد متغیرها و افق پیش بینی به کار رفته بستگی دارد و عموماً در حدی است که نیاز به محاسبات چندانی پیچیده ندارد. در حالت هایی که محدودیت وجود دارد یا شرایط خاص روی مقاوم بودن کنترل کننده قرار داده شده است، زمان لازم برای بهینه سازی به مراتب بیشتر از حالت عادی خواهد بود.

روشهای مختلف MPC توابع هدف متفاوتی را برای بدست آوردن قانون کنترل پیشنهاد می‌کنند. هدف اصلی همه آنها این است که خروجی آینده $y(t)$ در یک افق در نظر گرفته شده باید به سیگنال مرجع تعیین شده w را دنبال کند و در همان حال هزینه کنترلی لازم برای انجام این کار باید کمینه شود.

عبارت کلی برای چنین تابع هدفی به صورت زیر خواهد بود:

(۴-)

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j/t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \quad (12)$$

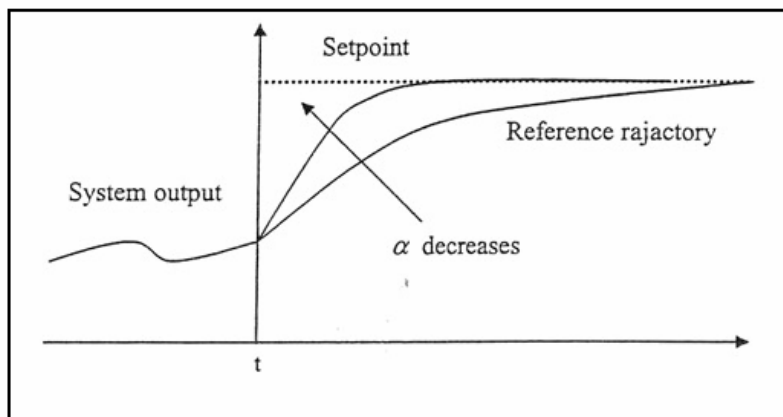
در بعضی روشها، قسمت دوم، که بیانگر هزینه کنترلی است، را به حساب نمی‌آوریم و در بعضی روشها نیز مقدار سیگنال کنترل (و نه نمو آن) ظاهر می‌شوند.

پارامترهای $N1$ و $N2$ به ترتیب افقهای مینیمم و ماکزیمم پیش بینی هدف و Nu افق کنترل است که لزوماً با ماکزیمم افق پیش بینی برابر نیستند، در مورد اینها در جای خود سخن خواهیم گفت. فقط اشاره می‌کنیم که این پارامترها بعنوان پارامترهای طراحی هستند که برای دستیابی به عملکرد دلخواه می‌توان آنها را تنظیم کرد.

یکی از مزایای کنترل پیش بین این است که اگر شکل آینده سیگنال مرجع از قبل مشخص باشد، سیستم می‌تواند عکس العمل مناسب را از قبل اعمال کند و لذا آثار تاخیر زمانی را در پاسخ فرآیند جبران سازد. شکل آینده سیگنال مرجع در بسیاری از کاربردها از قبل مشخص است. در تابع هدف (۴-۱۲) اکثریت روشها از سیگنال مرجعی استفاده می‌کنند که لزوماً منطبق بر سیگنال مرجع واقعی نیست (شکل (۴-۵)) و معمولاً یک تخمین هموار از مقادیر کنونی خروجی به سمت مرجع مطلوب است که با سیستم مرتبه اول زیر بیان می‌شود:

$$\omega(t)=y(t) \quad \omega(t+k)=\alpha\omega(t+k-1)+(1-\alpha)r(t+k) \quad k=1,...,N \quad (۴-۱۳)$$

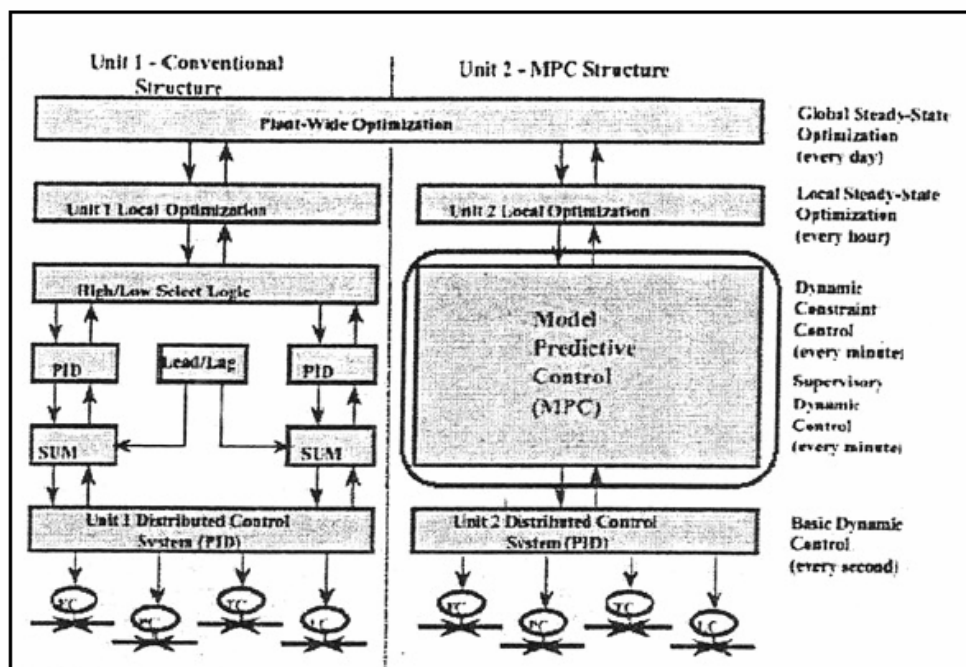
یک پارامتر بین صفر و یک است که یک مقدار قابل تنظیم به دست می‌دهد که بر پاسخ دینامیک سیستم اثر می‌گذارد.



شکل (۴-۵) تاثیر α بر ردیابی سیگنال

بهینه سازی در کنترل سلسله مراتبی : در سیستم های مدرن صنعتی، کنترل کننده پیش بین، بخشی از سیستم کنترلی سلسله مراتبی است. در شکل (۴-۶) یک ساختار متعارف کنترل در سمت چپ به همراه ساختار پیش بین در سمت راست نشان داده شده است .

در بالاترین سطح ساختار، یک بهینه ساز کلی، کمیت های بهینه حالت دائمی را برای هر قسمت سیستم تعیین می نماید. این کمیت ها به بهینه ساز محلی فرستاده می شوند. بهینه سازهای محلی از مدل مشروح تری از فرآیند استفاده می کنند و در فواصل زمانی کوتاه تری به کار گرفته می شوند.



شکل (۴-۶) ساختار کنترل سلسله مراتبی در یک سیستم پیشرفته صنعتی

این بهینه سازها به نوبه خود مقادیر مطلوب حالت دائمی را محاسبه می کنند و برای پیاده سازی به واحد کنترل دینامیک محدودیت ها می فرستند. این واحد موظف است که سیستم را از یک حالت دائمی به حالت مطلوب دیگر برساند و در عین خارج شدن از محدودیت های معین، کمینه باشد. در ساختار متعارف، پیاده سازی با ترکیب الگوریتم هایی چون PID و Lead/Lag (بلوک L/L شکل (۴-۶)) و بعضی عملیات منطقی صورت می گیرد. در استراتژی MPC، همه اجزای این واحد با یک کنترل کننده پیش بین جایگزین می شود.

۴-۳- مزایای کنترل پیش بین MPC

در مقایسه با سایر روشهای کنترلی، MPC چندین مزیت قابل توجه دارد:

- برای افرادی که دانش کمتری در کنترل دارند، MPC بسیار جذاب است، چون مفاهیم، مبتنی بر درک حسی هستند و تنظیم پارامترهای کنترل کننده پیچیدگیهای خاص روشهای دیگر را ندارد.

- MPC می‌تواند برای کنترل دسته وسیعی از فرآیندهای به کار گرفته شود. سیستم‌هایی با دینامیک ساده یا پیچیده مثل سیستم‌های غیر مینیمم فاز، ناپایدار و سیستم‌های با تاخیر طولانی با این روش قابل کنترل می‌باشند.

- تعمیم این روش به حالت چند متغیره به سادگی قابل انجام است.

- MPC ذاتاً توانایی جبران‌سازی تاخیر زمانی را دارد.

- برای اغتشاش‌های قابل اندازه‌گیری، MPC با ایجاد سیگنال پیشخور توانایی جبران‌سازی را داراست.

- چون قانون کنترل خطی است، پیاده‌سازی کنترل کننده‌های MPC نسبتاً ساده است.

- لحاظ کردن محدودیت‌های مساله کنترل به طور نسبتاً ساده‌ای قابل انجام است.

- در مواردی که خروجی مرجع در لحظات آینده، از قبل معلوم است (مانند رباتیک)، MPC بسیار موفق عمل می‌کند.

- به طور کلی درجه آزادی در این روش زیاد است.

- MPC برخلاف بسیاری از روشهای کنترلی از صنعت آغاز شده است.

- بر اساس اصول اولیه، امکان توسعه این روش در آینده فراهم است.

۴-۴- مشکلات کنترل پیش بین MPC

همانگونه که منطقی به نظر می‌رسد MPC مشکلات خاص خود را نیز داراست. به عنوان نمونه، گر چه قانون کنترلی حاصل از الگوریتم MPC به سادگی قابل پیاده سازی است و محاسبات نسبتاً کمی دارد، لیکن استخراج قانون کنترلی پیچیده تر از کنترل کننده های کلاسیک PID است. اگر دینامیک فرآیند تغییر نکند، محاسبات مربوط به تعیین سیگنال کنترل به صورت خارج خط انجام می‌شود. اما در کنترل تطبیقی سیستم های تغییر پذیر با زمان و نامعلوم، کلیه محاسبات باید در هر گام نمونه برداری انجام گیرد و با لحاظ کردن محدودیت های سیستم و کنترل کننده، حجم محاسبات بیشتر هم خواهد شد. اگر چه با کامپیوترهای پیشرفته امروزی، حجم محاسبات اصولاً مشکل اساسی محسوب نمی‌گردد، اما اغلب کامپیوترهایی که در کنترل فرآیندهای صنعتی به کار می‌روند، چندان پیشرفته نیستند و علاوه بر محاسبات کنترلی فرآیند، وظایف دیگری نظیر ارتباط با کاربر، انتقال داده ها، ثبت اطلاعات و ... را نیز بر عهده دارند.

شاید بتوان گفت که بزرگترین مشکل برای به کار بردن الگوریتم های MPC در فرآیندهای صنعتی، لزوم در دست بودن مدل مناسبی از فرآیند است. الگوریتم طراحی مبتنی بر اطلاع پیشین از مدل است و روشن است که مزایای حاصل شده، از دقیق نبودن مدل و مطابق با واقعیت نبودن آن تاثیر می‌پذیرد.

برجسته ترین محدودیت های MPC در صنعت را می‌توان به شرح زیر برشمرد:

- مدل های پاسخ ضربه و پاسخ پله دارای پارامترهای زیادی هستند و کاربرد الگوریتم کنترل را به فرآیندهای پایدار محدود می‌سازند.

- حل مسائل بهینه سازی دینامیک، زیر بهینه هستند.

- عموماً فرض می‌شود که اغتشاش موجود در خروجی ثابت است.

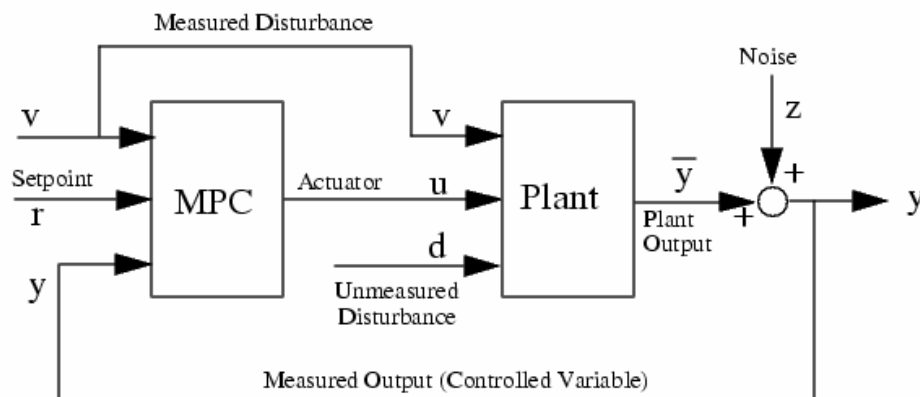
- برای حصول پایداری نامی، تنظیم پارامترها ضروری است.

- به عدم قطعیت در مدل به طور مناسب توجه نشده است.

علیرغم فقدان اثبات های ریاضی دقیق برای پایداری و مقاوم بودن الگوریتم ها، در عمل نشان داده شده است که MPC استراتژی مناسبی برای کنترل فرآیندهای صنعتی است. اخیراً کاربردهای موفق فراوانی از کنترل پیش بین نه تنها در فرآیندهای صنعتی بلکه در سیستم های گوناگونی از بازوهای روبات گرفته تا بیهوشی بالینی گزارش شده است.

۴-۵- پیاده سازی کنترل پیش بین با مطلب

جعبه ابزار MPC مطلب شامل مجموعه توابعی برای تحلیل و طراحی سیستم های کنترل پیش بین می باشد در محیط سیمولینک اینکار توسط بلوک MPC انجام پذیر می باشد شکل (۴-۷) زیر این بلوک و ورودیها و خروجیهای آن را نشان می دهد. [14,15]



شکل (۴-۷) بلوک و ورودیها و خروجیهای کنترلر MPC (مرجع [14])

همانطور که مشاهده می شود این بلوک دارای سه ورودی و یک خروجی (سیگنال کنترلی) می باشد شکل زیر نمونه ای از کاربرد کنترلر MPC را برای یک پلنت نشان می دهد هدف مساله

میل دادن مقادیر خروجی پلنت، y ، بسمت مقدار مرجع، r ، می باشد که این کار با تنظیم سیگنال کنترلی u انجام می شود پلنت نشان داده شده در شکل ممکن است علاوه بر ورودی u دارای ورودیهای اغتشاش نیز باشد که v مبین اغتشاش قابل اندازه گیری و d اغتشاش غیر قابل اندازه گیری می باشد کنترلر MPC برای تولید سیگنالهای کنترلی دقیقتر، از مقادیر اغتشاش قابل اندازه گیری نیز استفاده می کند. سومین ورودی کنترلر MPC سیگنال خروجی پلنت، y ، می باشد با داشتن این مقادیر کنترلر MPC سیگنالهای u را طوری تولید می کند، که سیگنالهای خروجی y به مقادیر تعیین شده در سیگنال مرجع r میل کند میزان دقت این ردگیری به مشخصات پلنت بستگی دارد. که در ساختاری تحت عنوان MPC Controller تعریف می شود و جزء پارامترهای ورودی بلوک دیاگرام MPC می باشد. و بکمک دستور زیر تعریف می شود

$$\text{Mpc_controller} = \text{mpc}(\text{sys}, T_s, P, m)$$

که sys بردار فضای حالت پلنت می باشد. T_s زمان نمونه برداری P معرف وسعت پیش بینی (prediction horizon) و m معرف وسعت کنترل (control horizon) می باشد. [14,15]

فصل ۵

پیاده سازی کنترلر پیش بین

برای ربات CamCrane

در این فصل به ارائه شیوه‌ای نوین در طراحی سیستم‌های دینامیکی و پیاده سازی کنترلر بر روی آن بکمک ارتباط دو نرم افزار شبیه ساز دینامیکی و نرم افزار طراح کنترلر می‌پردازیم و با استفاده از آن به پیاده سازی کنترلر پیش بین برای ربات CamCrane می‌پردازیم.

شیوه های سنتی پیاده سازی کنترلر بر روی سیستم‌های دینامیکی مستلزم بدست آوردن معادلات ریاضی توصیف کننده مدل دینامیکی تحت کنترل بود. که فقط برای مدل‌های دینامیکی ساده از دقت کافی برخوردار بود و برای مدل‌های پیچیده مستلزم ساده سازیها و فرضیات خطی سازی بود که متعاقباً از دقت عملکرد مدل دینامیکی می‌کاست. در روش پیشنهادی این پایان نامه، مدل دینامیکی CamCrane بدون هیچگونه ساده سازی (بعبارتی دقیقاً همان مدل طراحی شده برای ساخت) برای طراحی کنترلر مورد استفاده قرار می‌گیرد. که اینکار از طریق طراحی مدل دینامیکی در محیط نرم افزار تحلیل دینامیکی MSC Visual Nastran® و طراحی کنترلر در محیط نرم افزار Simulink و لینک نمودن این دو نرم افزار صورت می‌گیرد. در این روش سیگنالهای کنترلی از نرم افزار نسترن به نرم افزار سیمولینک ارسال می‌شود و عملگرها و موتورهای سیستم دینامیکی مطابق سیگنالهای دریافتی عمل می‌کنند کار نرم افزار نسترن محاسبه وضعیت جدید مدل دینامیکی در پاسخ به سیگنالهای ورودی است و خروجیها که شامل تمام پارامترهای دلخواه (system's configuration) از مدل دینامیکی می‌باشد به نرم افزار سیمولینک بصورت فیدبک ارسال می‌شود.

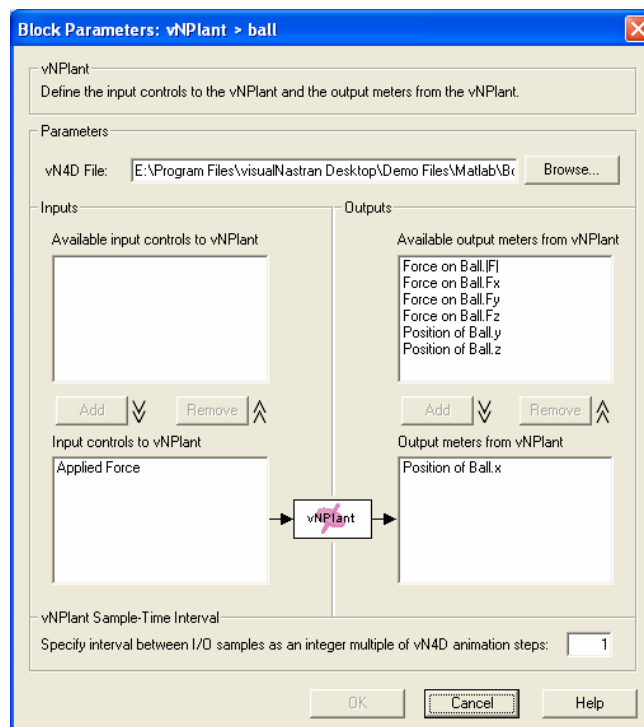
در این فصل ابتدا به طراحی کنترلر MPC برای ربات CamCrane در محیط سیمولینک می‌پردازیم و در ادامه نحوه ایجاد ارتباط بین دو نرم افزار مدل ساز دینامیکی و طراح کنترلر توضیح داده می‌شود و در نهایت نتایج شبیه سازیها و بحث بر روی نتایج ارائه خواهد شود.

۵-۲- ایجاد ارتباط بین مطلب و نسترن

نرم افزار MSC Visual Nastran® توسعه یافته نرم افزار MSC Working Model® است که توسط شرکت MSC تولید شده است. از قابلیت‌های عمده این نرم افزار می‌توان به شبیه سازی دینامیکی و تحلیل‌های استاتیکی و المان محدود آن اشاره کرد که در این پروژه نیز از این قابلیت‌ها برای طراحی ربات CamCrane استفاده شده است یکی از ویژگی‌های اساسی این نرم افزار قابلیت لینک شدن آن نرم افزار کنترلی سیمولینک می‌باشد که به این طریق می‌توان از کنترلرهای طراحی شده در محیط سیمولینک برای کنترل سیستم‌های دینامیکی پیاده سازی شده در محیط نسترن استفاده کرد. این ارتباط از طریق بلوک vNPlantBlock میسر شده است شکل (۵-۱) پارامترهای ورودی و خروجی این بلوک را نشان می‌دهد. [17,18]

روش کار با این بلوک بدینصورت است که ابتدا در فیلد مربوط به اسم فایل vN4D File مدل سیستم دینامیکی طراحی شده در محیط نسترن مشخص می‌شود با اینکار لیستی از پارامترهای ورودی و خروجی مدل در قسمت مربوط به Inputs و Outputs نشان داده می‌شود این پارامترها در واقع پارامترهایی هستند که در محیط نرم افزار نسترن توسط کاربر در مرحله بعد از مدلسازی مشخص می‌شود پارامترهای ورودی بلوک شامل موقعیت زاویه ای، سرعت، شتاب یا گشتاور موتورها و عملگرها می‌باشد و پارامترهای خروجی بلوک شامل تمام مشخصات (موقعیت، سرعت خطی یا زاویه ای، شتاب خطی یا زاویه ای، مومنتم و ...) تک تک قطعات موجود در مدل دینامیکی می‌باشد. همه پارامترهای مورد نیاز اعم از پارامترهای ورودی و خروجی باید در مرحله پس از مدلسازی توسط کاربر تعریف شود تا در محیط نرم افزار سیمولینک و توسط بلوک vNPlantBlock قابل دسترسی باشد. همانطور که در شکل فوق مشاهده می‌شود لازم است که

پارامترهای ورودی و خروجی بلوک vNPlantBlock مجدداً در قسمت مربوط به Input controls به vNPlant و Output meters from vNPlant مشخص شود. با مشخص کردن این پارامترها عملیات تعریف بلوک vNPlantBlock خاتمه می‌یابد و می‌توان از آن در محیط نرم افزار سیمولینک استفاده کرد. لازم به ذکر است که با اجرای بلوک vNPlantBlock در محیط سیمولینک در واقع مدل دینامیکی طراحی شده در محیط نرم افزار نسترن اجرا می‌شود و این بلوک وظیفه انتقال داده ها بین این دو نرم افزار را انجام می‌دهد.



شکل (۵-۱) پارامترهای ورودی و خروجی بلوک vNPlantBlock

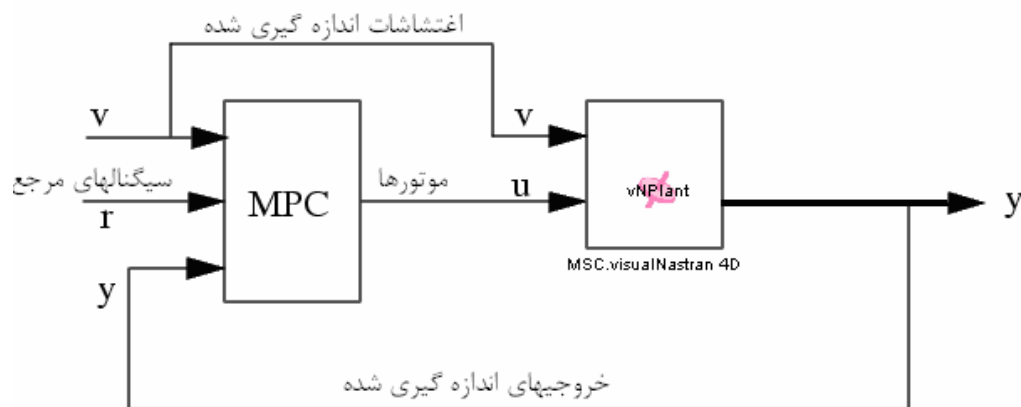
در این پروژه نیز از مدل دینامیکی CamCrane پیاده سازی شده در محیط نسترن برای طراحی کنترلر و انجام شبیه سازی ها استفاده شده است. پارامترهای ورودی بلوک vNPlantBlock بشرح زیر می‌باشد:

- ۱ - سرعت زاویه ای موتور DC محرک واگن روی ریل
 - ۲ - سرعت زاویه ای موتور DC محرک اتصال لولایی قائم بازوی ربات
 - ۳ - سرعت زاویه ای موتور DC محرک اتصال لولایی افقی بازوی ربات
 - ۴ - سرعت زاویه ای موتور DC محرک بازوی تلسکوپی
 - ۵ - سرعت زاویه ای موتور DC محرک اتصال لولایی افقی دوربین
 - ۶ - سرعت زاویه ای موتور DC محرک اتصال لولایی قائم دوربین
- همانطور که مشاهده می‌شود همه پارامترهای ورودی بلوک vNPlantBlock سرعت زاویه ای موتورهای محرک قسمت‌های مختلف ربات می‌باشد بدیهی است که پارامترهای خروجی نیز در واقع فید بک هایی است که از سنسورهای نصب شده در قسمت‌های مختلف ربات تولید می‌شود. این پارامترها بشرح زیر می‌باشد.

- ۱ - موقعیت خطی واگن بر روی ریل
 - ۲ - موقعیت زاویه ای اتصال لولایی قائم بازوی ربات
 - ۳ - موقعیت زاویه ای اتصال لولایی افقی بازوی ربات
 - ۴ - موقعیت شعاعی بازوی تلسکوپی
 - ۵ - موقعیت زاویه ای اتصال لولایی افقی دوربین
 - ۶ - موقعیت زاویه ای اتصال لولایی قائم دوربین
- با این تعاریف بلوک vNPlantBlock برای مدل ربات CamCrane دارای شش ورودی و شش خروجی خواهد بود و می‌توان از آن برای پیاده سازی کنترلر MPC استفاده نمود.

۳-۵- مدلسازی کنترلر MPC برای ربات CamCrane در محیط سیمولینک

جعبه ابزار MPC مطلب شامل مجموعه توابعی برای تحلیل و طراحی سیستم های کنترل پیش بین می باشد. در محیط سیمولینک اینکار توسط بلوک MPC انجام پذیر می باشد شکل (۲-۵) ورودیها و خروجیهای سیستم کنترلی را نشان می دهد.



شکل (۲-۵) ورودیها و خروجیهای سیستم کنترل پیش بین

همانطور که مشاهده می شود این بلوک دارای سه ورودی و یک خروجی می باشد هدف مساله میل دادن مقادیر خروجی پلنت، y ، بسمت مقدار مرجع، r ، می باشد که این کار با تنظیم سیگنال کنترلی u انجام می شود پلنت نشان داده شده در شکل ممکن است علاوه بر ورودی u دارای ورودیهای اغتشاش نیز باشد که v مبین اغتشاش قابل اندازه گیری می باشد. سومین ورودی کنترلر MPC سیگنال خروجی پلنت، y ، می باشد با داشتن این مقادیر کنترلر MPC سیگنالهای u را طوری تولید می کند، که سیگنالهای خروجی y به مقادیر تعیین شده در سیگنال مرجع r میل کند میزان دقت این ردگیری به مشخصات پلنت بستگی دارد. که در ساختاری تحت عنوان MPC Controller تعریف می شود و جزء پارامترهای ورودی بلوک دیاگرام MPC می باشد. مهمترین

مساله ای که در طراحی کنترلر مواجه هستیم تابع تبدیل مبین پلنت می باشد همانطور که گفته شد مدل دینامیکی پلنت تهیه شده برای CamCrane بسیار پیچیده می باشد و عملاً نمی توان تابع تبدیل آن را بدست آورد برای حل این مشکل ایده System Identification مورد بررسی قرار گرفت در این ایده از جعبه ابزار System Identification نرم افزار مطلب برای شناسایی سیستم و بدست آوردن تابع تبدیل معرف پلنت استفاده می شود به این صورت که مدل دینامیکی تهیه شده از پلنت برای طیف وسیعی از سیگنالهای ورودی سیموله شده و نتایج خروجی ثبت می شود سپس با استفاده از جعبه ابزار System Identification تابع تبدیلی که این نگاشت را بین ورودی و خروجی انجام می دهد بدست می آید که در واقع تخمینی از تابع تبدیل پلنت خواهد بود.

مدلهای نسبتاً دقیق مستلزم تهیه بانک اطلاعات با چندین هزار سیمولشین با ورود های متنوع می باشد چون برای مدل دینامیکی CamCrane یک مدل نسبتاً پیچیده محسوب می شود این کار بسیار هزینه بر است به همین خاطر مدل دینامیکی فقط برای چندین حالت مختلف سیمولشین شد و تابع تبدیل تقریبی پلنت از روی آن بدست آمد. در ادامه به ارائه معادلات دینامیکی ربات به روش لاگرانژ می پردازیم. ابتدا انرژی جنبشی و پتانسیل قسمتهای مختلف سیستم را به دست می آوریم:

$$k_1 = 1/2 m_1 \dot{x}_1^2 \quad \text{رابطه (۱-۵)}$$

$$u_1 = 0 \quad \text{رابطه (۲-۵)}$$

$$k_2 = 1/2 m_2 (\dot{x}_1 + d_1 \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - d_1 \sin \theta_1 \cos \theta_2 \dot{\theta}_2)^2 + 1/2 m_2 (d_1 \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - d_1 \sin \theta_1 \sin \theta_2 \dot{\theta}_2)^2 + 1/2 m_2 (d_1 \cos \theta_2 \dot{\theta}_2)^2 + 1/2 I_{xx2} \dot{\theta}_1^2 + 1/2 I_{zz2} \dot{\theta}_2^2 \quad \text{رابطه (۳-۵)}$$

$$u_2 = m_2 d_1 g \sin \theta_1 \quad \text{رابطه (۴-۵)}$$

$$k_3 = 1/2m_3(\dot{x}_1 + x_2 \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - x_2 \sin \theta_1 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \sin \theta_1)^2 + \\ 1/2m_3(x_2 \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - x_2 \sin \theta_1 \sin \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \cos \theta_1)^2 + 1/2m_2(x_2 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \sin \theta_2)^2 + \\ 1/2I_{xx3}\dot{\theta}_1^2 + 1/2I_{zz3}\dot{\theta}_2^2$$

رابطه (۵-۵)

$$u_3 = m_3 x_2 g \sin \theta_1 \quad \text{رابطه (۶-۵)}$$

$$k_4 = 1/2m_4(\dot{x}_1 + (d_2 + x_2) \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - (d_2 + x_2) \sin \theta_1 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \sin \theta_1)^2 + \\ 1/2m_4((d_2 + x_2) \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - (d_2 + x_2) \sin \theta_1 \sin \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \cos \theta_1)^2 + \\ 1/2m_4((d_2 + x_2) \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \sin \theta_2)^2 + 1/2I_{xx4}(\dot{\theta}_1 + \dot{\theta}_3 \cos \theta_2)^2 + 1/2I_{zz4}(\dot{\theta}_2 + \dot{\theta}_4 \cos \theta_3)^2$$

رابطه (۷-۵)

$$u_4 = m_4(d_2 + x_2)g \sin \theta_1 \quad \text{رابطه (۸-۵)}$$

مجموع انرژیهای پتانسیل و جنبشی بصورت زیر می شود.

$$u = u_1 + u_2 + u_3 + u_4 = m_2 d_1 g \sin \theta_1 + m_3 x_2 g \sin \theta_1 + m_4(d_2 + x_2)g \sin \theta_1 \quad \text{رابطه (۹-۵)}$$

$$k = k_1 + k_2 + k_3 + k_4 = 1/2m_1\dot{x}_1^2 + 1/2m_2(\dot{x}_1 + d_1 \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - d_1 \sin \theta_1 \cos \theta_2 \dot{\theta}_2)^2 + \\ 1/2m_2(d_1 \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - d_1 \sin \theta_1 \sin \theta_2 \dot{\theta}_2)^2 + 1/2m_2(d_1 \cos \theta_2 \dot{\theta}_2)^2 + \\ 1/2I_{xx2}\dot{\theta}_1^2 + 1/2I_{zz2}\dot{\theta}_2^2 + 1/2m_3(\dot{x}_1 + x_2 \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - x_2 \sin \theta_1 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \sin \theta_1)^2 + \\ 1/2m_3(x_2 \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - x_2 \sin \theta_1 \sin \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \cos \theta_1)^2 + \\ 1/2m_2(x_2 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \sin \theta_2)^2 + 1/2I_{xx3}\dot{\theta}_1^2 + 1/2I_{zz3}\dot{\theta}_2^2 + 1/2m_4(\dot{x}_1 + (d_2 + x_2) \cos \theta_1 \cos \theta_2 \dot{\theta}_1 - \\ (d_2 + x_2) \sin \theta_1 \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \sin \theta_1)^2 + 1/2m_4((d_2 + x_2) \sin \theta_1 \cos \theta_2 \dot{\theta}_1 - \\ (d_2 + x_2) \sin \theta_1 \sin \theta_2 \dot{\theta}_2 + \dot{x}_2 \cos \theta_2 \cos \theta_1)^2 + 1/2m_4((d_2 + x_2) \cos \theta_2 \dot{\theta}_2 + \dot{x}_2 \sin \theta_2)^2 + \\ 1/2I_{xx4}(\dot{\theta}_1 + \dot{\theta}_3 \cos \theta_2)^2 + 1/2I_{zz4}(\dot{\theta}_2 + \dot{\theta}_4 \cos \theta_3)^2$$

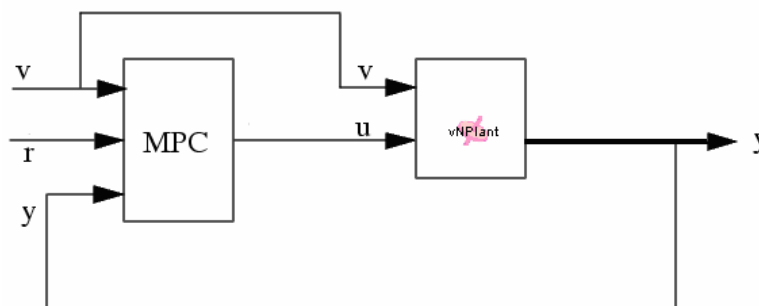
رابطه (۱۰-۵)

با جایگزاری در معادله لاگرانژ معادله دینامیکی ربات CamCrane بدست می آید (عملیات مربوط به مشتق گیری و جایگزاری در معادله لاگرانژ و همچنین فرم نهایی معادلات دینامیکی در پیوست ۶ آمده است).

$$\frac{d}{dt} \frac{\partial k}{\partial \dot{\theta}} - \frac{\partial k}{\partial \theta} + \frac{\partial u}{\partial \theta} = \tau \quad \text{رابطه (۵-۱۱)}$$

وردیهای سیستم $(u_1, u_2, u_3, u_4, u_5, u_6)$ پارامترهای I, V, ω_r پلنت می باشد و خروجیهای سیستم $(y_1, y_2, y_3, y_4, y_5, y_6)$ سیگنالهای کنترلی می باشند که در اینجا X_s, K_s, K_r در نظر گرفته شده اند که ضرایب بهره مربوط به موتورها می باشد.

شکل (۵-۳) مدل طراحی شده در سیمولینک برای کنترل ربات را نشان می دهد در این مدل بلوک مربوط به پلنت در واقع رابط دو نرم افزار نستر و سیمولینک است که در بخش قبل بطور کامل توضیح داده شد. هدف کنترلر MPC هدایت دوربین به موقعیت فضایی و زاویه ای مشخص شده در قسمت سیگنالهای مرجع می باشد نتایج مربوط به نحوه کنترل و سرعت آن در رسیدن به نقطه مطلوب در انتهای فصل آمده است.



شکل (۵-۳) مدل طراحی شده در سیمولینک برای کنترل ربات

لازم به ذکر است که سیگنالهای مرجع مورد نیاز برای کنترلر MPC توسط نرم افزار سه بعدی ساز تولید می شود. بدینصورت که پس از مشخص شدن موقعیت فضایی توپ در ورزشگاه، این نرم افزار

موقعیت فضایی و زاویه ای مناسب برای دوربین را پیشنهاد می کند و بعنوان پارامترهای مرجع برای کنترلر، به آن ارسال می کند.

۵-۴- شبیه سازی و بحث بر روی نتایج

هدف از شبیه سازی اعمال یکسری سیگنالهای مرجع از نرم افزار سه بعدی ساز به کنترلر و مشاهده عملکرد پاسخ ربات به این سیگنالها می باشد. بعنوان مثال فرض می شود که در لحظه $t = 0$ ربات دارای شرایط اولیه ای بصورت زیر می باشد.

$$x_1 = -1m, \dot{x}_1 = 0m/s, \ddot{x}_1 = 0m/s^2$$

$$\theta_1 = .50rad, \dot{\theta}_1 = 1rad/s, \ddot{\theta}_1 = 1rad/s^2$$

$$\theta_2 = 0.35rad, \dot{\theta}_2 = 0.25rad/s, \ddot{\theta}_2 = 0.5rad/s^2$$

$$x_2 = 1.35m, \dot{x}_2 = 0.45m/s, \ddot{x}_2 = 0.3m/s^2$$

$$\theta_3 = 1rad, \dot{\theta}_3 = 3rad/s, \ddot{\theta}_3 = 1rad/s^2$$

$$\theta_4 = 0rad, \dot{\theta}_4 = 3rad/s, \ddot{\theta}_4 = 1rad/s^2$$

که x موقعیت خطی واگن بر روی ریل و θ موقعیت زاویه ای اتصال لولایی قائم بازوی ربات می باشد حال فرض می کنیم در همان لحظه $t = 0$ سیگنال مرجعی با مشخصات زیر برای کنترلر ارسال می شود.

$$x_1 = 1m, \dot{x}_1 = 0m/s, \ddot{x}_1 = 0m/s^2$$

$$\theta_1 = 0rad, \dot{\theta}_1 = 1rad/s, \ddot{\theta}_1 = 1rad/s^2$$

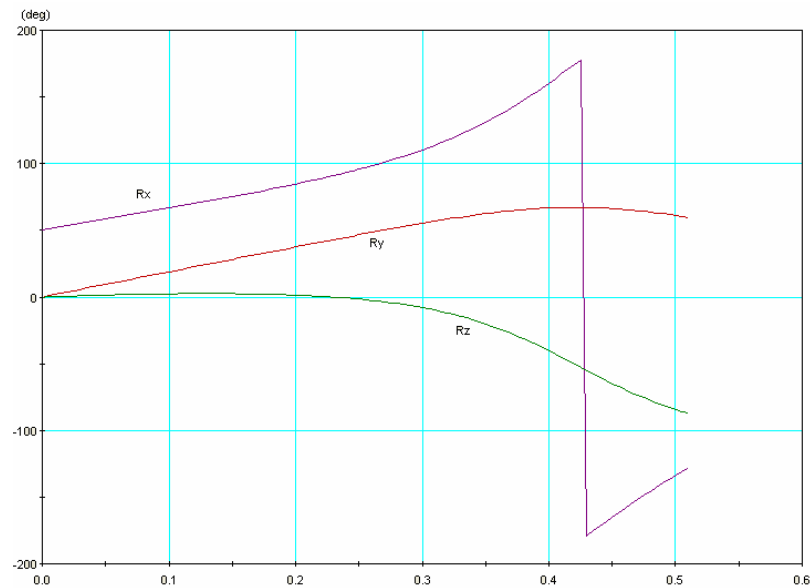
$$\theta_2 = 2.75rad, \dot{\theta}_2 = 0.5rad/s, \ddot{\theta}_2 = 0.5rad/s^2$$

$$x_2 = 1.35m, \dot{x}_2 = 0.45m/s, \ddot{x}_2 = 0.3m/s^2$$

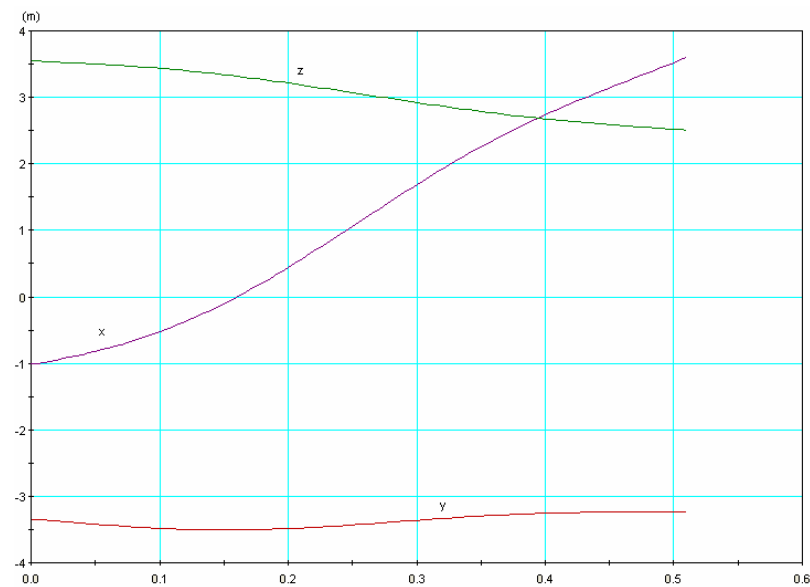
$$\theta_3 = -1.5rad, \dot{\theta}_3 = 3rad/s, \ddot{\theta}_3 = 1rad/s^2$$

$$\theta_4 = -1.5rad, \dot{\theta}_4 = 3.5rad/s, \ddot{\theta}_4 = 1rad/s^2$$

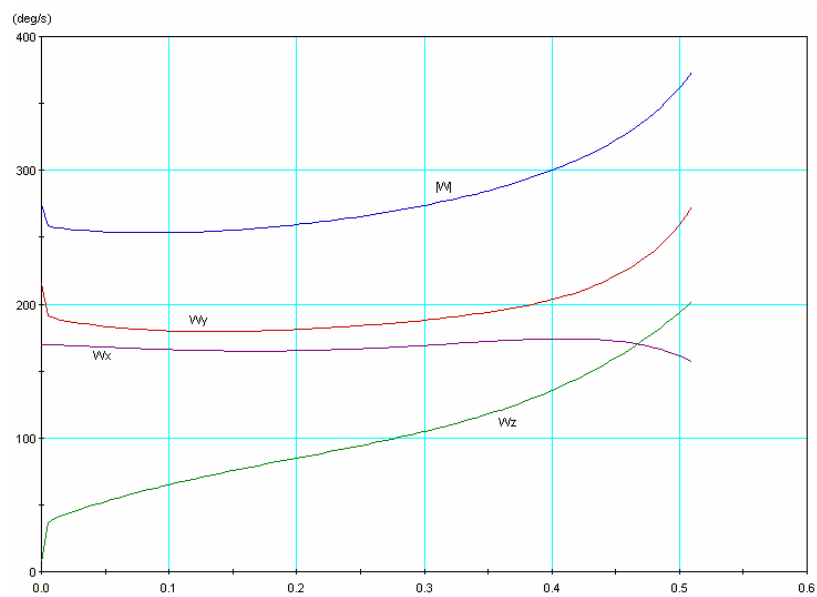
برای این سیگنال نتایج شبیه سازی شامل نمودارهای موقعیت، سرعت و شتاب خطی واگن روی ریل نسبت به زمان در شکل‌های (۴-۵) تا (۹-۵) و نمودارهای موقعیت، سرعت و شتاب خطی و زاویه‌ای دوربین نسبت به زمان در شکل‌های (۹-۵) تا (۱۲-۵) آمده است.



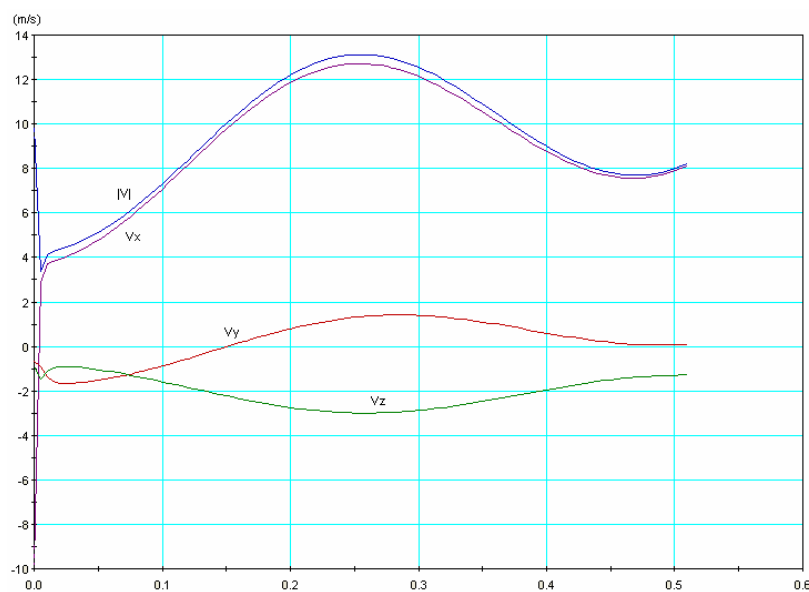
شکل (۴-۵) نمودار موقعیت زاویه‌ای دوربین



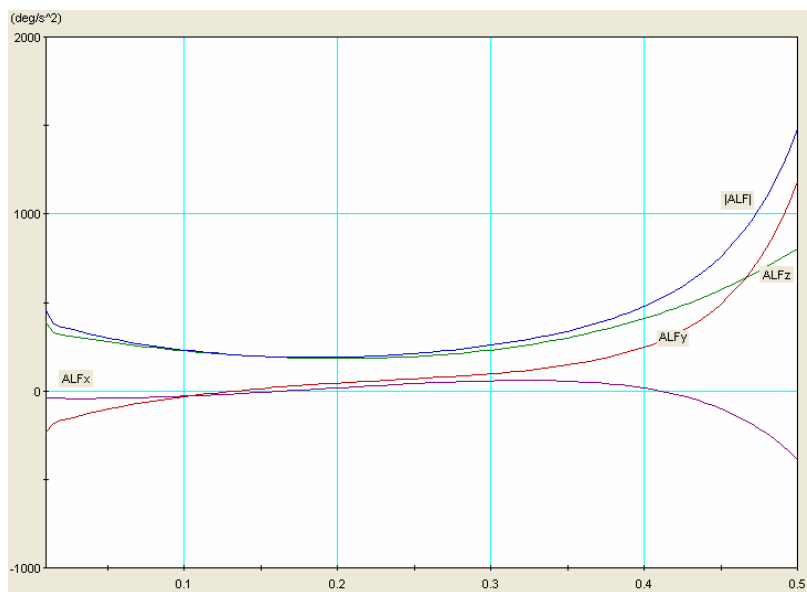
شکل (۵-۵) نمودار موقعیت خطی دوربین



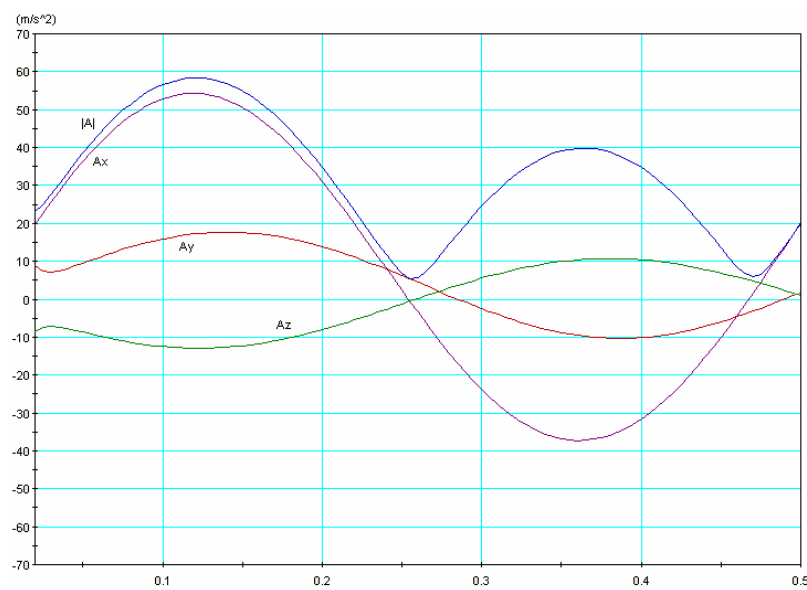
شکل (۵-۶) نمودار سرعت زاویه ای دوربین



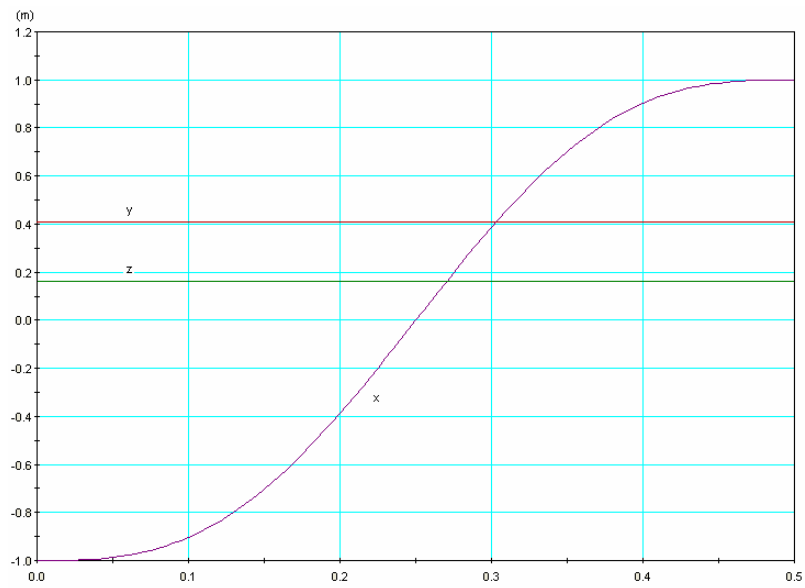
شکل (۵-۷) نمودار سرعت خطی دوربین



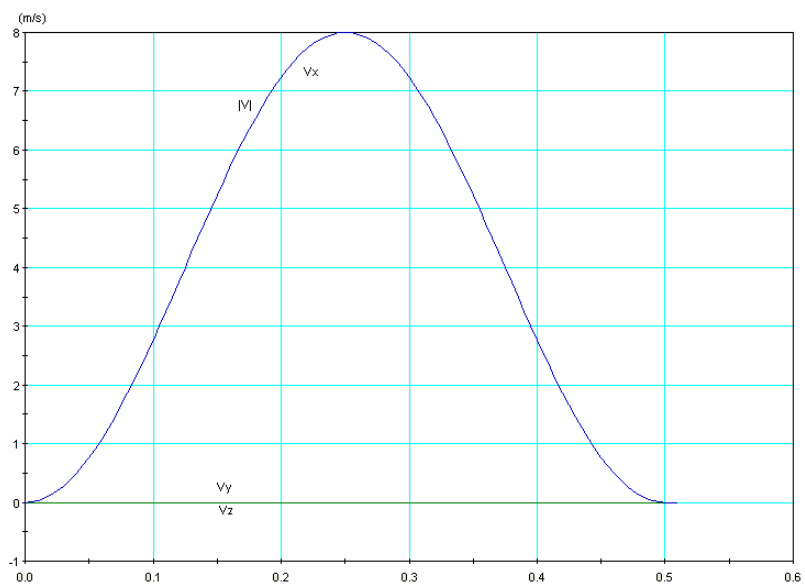
شکل (۵-۸) نمودار شتاب زاویه ای دوربین



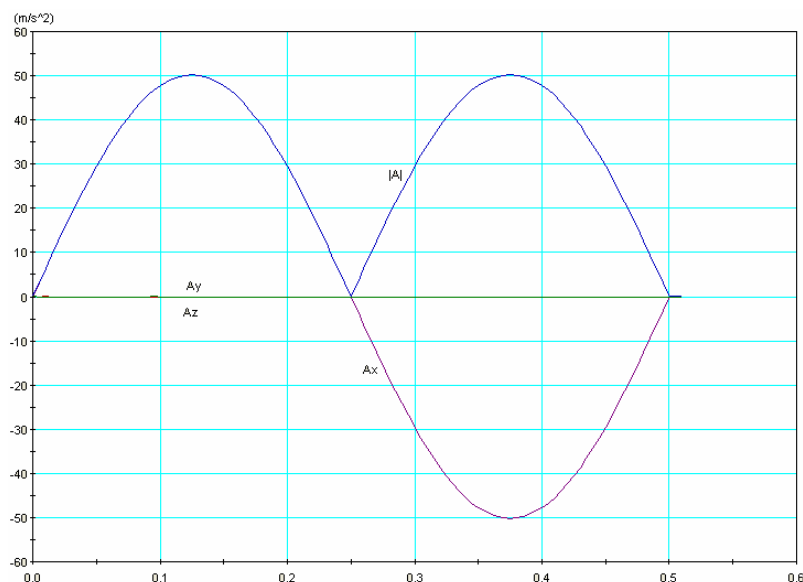
شکل (۵-۹) نمودار شتاب خطی دوربین



شکل (۵-۱۰) نمودار موقعیت خطی واگن



شکل (۵-۱۱) نمودار سرعت خطی واگن



شکل (۵-۱۲) نمودار شتاب خطی واگن

جزئیات بیشتر در مورد شبیه سازیها و نتایج آنها بدلیل حجم بودن نتایج آنها در CD ضمیمه پایان نامه موجود می باشد همچنین فایل انیمیشن نتایج که نحوه حرکت ربات و جهت گیری دوربین به سمت پوزیشن تعریف شده را نشان می دهد در CD ضمیمه موجود می باشد.

برای اجرای مدل برای هر حالت دلخواه از سیگنالهای مرجع کفایت مقادیر سیگنال مرجع که بصورت برداری با شش مقدار ثابت است در بلوک vNPlantBlock وارد شود با اجرای مدل در محیط سیمولینک، نرم افزار نسترن نیز خود بخود اجرا شده و مدل ربات را لود خواهد کرد و حرکات قسمتهای مختلف مکانیزم ربات را بصورت انیمیشن نشان خواهد داد. همچنین نمودارهای مربوط به سیگنالهای فیدبک از شش سنسور نصب شده بر روی مدل به همراه سیگنالهای کنترلی تولید شده و توسط کنترلر MPC ترسیم می شود.

لازم به ذکر است که اجرای مدل مستلزم نصب نرم افزارهای مطلب ورژن 7.0.4 و MSC VisualNastran® می باشد.

فصل ۶

طراحی سخت افزارهای

کنترلی

سیستم پردازش تصویر ارائه شده در فصل سوم به همراه نرم افزار کنترلی پیاده سازی شده در فصل پنجم به محاسبات بسیار زیادی نیاز دارد که از عهده سخت افزارهای معمولی خارج است. کامپیوترهای امروزی سرعت محاسباتی بسیار بالایی دارند ولی سیستم عاملهایی مثل ویندوز یا لینوکس از کاربرد مورد نیاز ما فاصله زیادی دارد و جوابگوی نیاز ما نمی باشد. برای پیاده سازی سخت افزارهای کنترلی بر روی کامپیوترهای شخصی نرم افزار مطلب سیستم عاملی بسیار کوچک و فوق العاده سریع را تدارک دیده که علاوه بر فراهم ساختن سرعت محاسباتی بسیار بالا امکان تبادل اطلاعات با سخت افزارها را نیز آسان نموده است. این کرنل بر روی سخت افزاری تحت عنوان Xpc-Target نصب و اجرا می شود.

در این فصل به طراحی سخت افزارهای کنترلی پرداخته می شود و بطور کلی به دو بخش اصلی تقسیم شده است. در بخش نخست مقدماتی در رابطه با مباحث نرم افزاری درگیر با تولید کدهای Real time (شامل ایجاد مدل Real time، مراحل پروسه تولید کد، معرفی RTW و ساختار آن و محیطهای سخت افزاری و نرم افزاری RTW) ارائه می شود سپس مباحث مربوط به سخت افزار xPC target (شامل معرفی xPC، ابزارها و امکانات محیط سخت افزاری و نرم افزاری RTW و نحوه کنترل آن، استفاده از شتاب دهنده برای بهینه سازی سرعت اجرایی مدل و نحوه استفاده از xPC target برای تولید کد قابل اجرا بر روی سیستم) توضیح داده شده است.

۶-۲- ایجاد مدل Real time در محیط RTW برای سیستم کنترلی

نرم افزار RTW برای ایجاد مدل‌هایی جهت تست Real-Time روی انواع سخت‌افزارهای xPC-Target مورد استفاده قرار می‌گیرد. در این بخش نحوه تبدیل مدل‌های ایجاد شده در سیمولینک به یک مدل Real Time توضیح داده می‌شود.

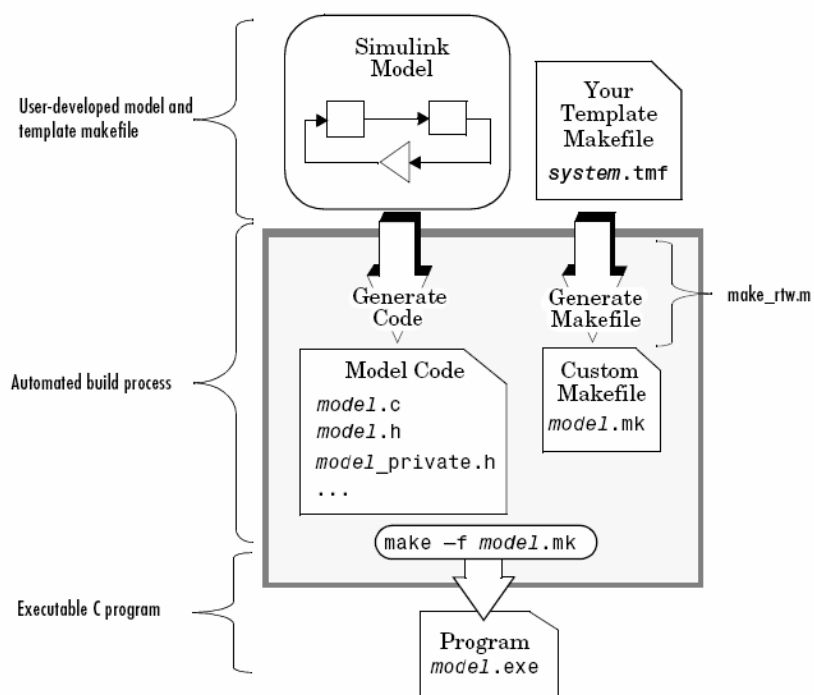
۶-۲-۱- معرفی RTW

RTW مجموعه امکاناتی در مطلب و سیمولینک است که امکان مدلسازی Real Time سیستم‌های دینامیکی را روی انواع سخت‌افزارها را فراهم می‌سازد این Toolbox امکانات زیر را دربر دارد:

- ۱ - تولید اتوماتیک کد برای انواع Target Platform
- ۲ - مسیری سریع و مستقیم از طراحی سیستم به پیاده‌سازی آن
- ۳ - ایجاد ارتباطی قوی و محکم بین مطلب و سیمولینک
- ۴ - محیط گرافیکی ساده برای استفاده کاربر
- ۵ - تولید مستقیم کد C از مدل‌های سیمولینک
- ۶ - امکان ارتباط Real Time بین سیمولینک و مدل در حال اجرا بر روی Target
- ۷ - Rapid Simulation: کمک شتابدهنده سیمولینک می‌توان بطور متوسط سرعت سیمولیشن را بین ۵ تا ۲۰ برابر افزایش داد. در حالت نرمال سیمولینک در مد مفسر کار می‌کند که کلا سرعت زیادی ندارد. در حالیکه کدهای تولید شده توسط شتابدهنده سیمولینک برای پردازش سریع بنحو احسن بهینه شده‌اند.

۶-۲-۲- ایجاد یک مدل Real Time

روش اتوماتیک تولید کد توسط RTW قادر به تولید برنامه‌های Real Time روی انواع محیط‌های host می‌باشد. در این روش از امکانات make برای کامپایل و لینک نمودن سورس کدها استفاده می‌شود. یک برنامه سطح بالای M-file پروسه برنامه‌سازی RTW را کنترل و هدایت می‌کند شکل (۶-۱) این فرآیند را بطور کامل نشان می‌دهد.



شکل (۶-۱) فرآیند تولید کد توسط RTW

۶-۲-۳- مراحل پروسه تولید کد

پروسه تولید کد طی دو مرحله انجام می‌شود که برای تکمیل پروسه و ایجاد یک برنامه قابل اجرا به دو مرحله دیگر نیز احتیاج دارد (شکل (۶-۲)).

این چهار مرحله پروسه تولید کد با زدن کلید Build در کادر محاوره‌ای RTW بطور اتوماتیک پشت سرهم انجام می‌شود این چهار مرحله عبارتند از:

۱ - تحلیل بلوک دیاگرام و کامپایل آن به model.rtw توسط RTW.

۲ - فراخوانی TLC برای تبدیل model.rtw به کدهای C.

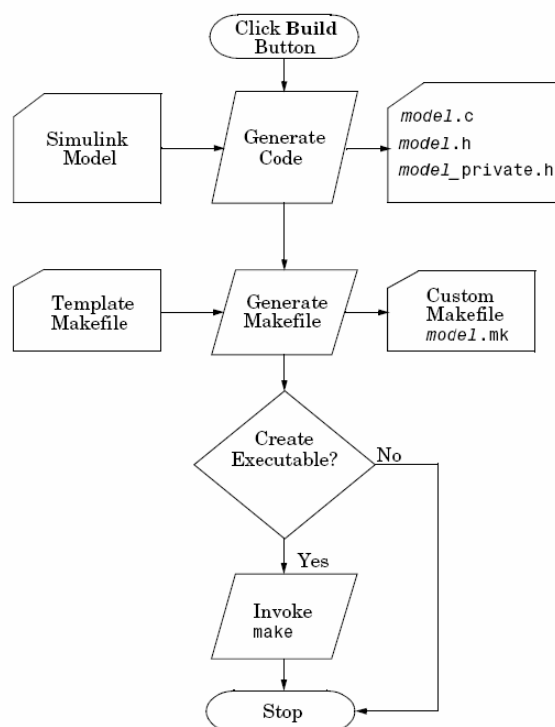
۳ - تولید فایل make file از روی مشخصات سیستم Target.

۴ - کامپایل و لینک نمودن obj فایلها و Library برای تولید کد قابل اجرا.

چنانچه که قرار باشد مدل روی یک میکروکنترلر یا برد DSP پیاده‌سازی شود فقط کافیست

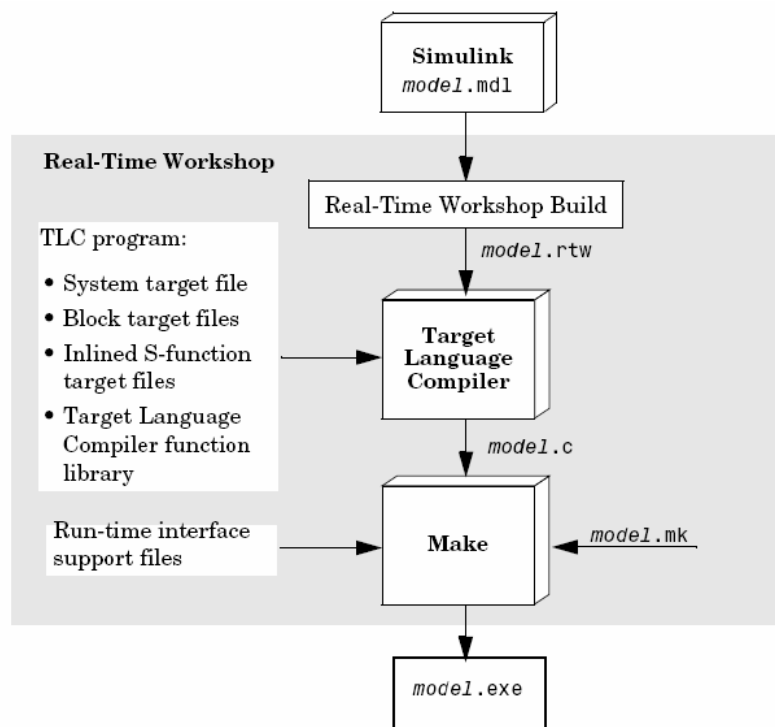
فایلهای سورس کد ایجاد شود. سپس این فایلها روی چیپست Download شود. شکل (۶-۳)

فرآیند اتوماتیک تولید کد را نشان می‌دهد.



شکل (۶-۲) مراحل کنترل پروسه تولید اتوماتیک کد

در صورتی که گزینه HTML Report فعال باشد مجموعه‌ای از فایل‌های HTML برای گزارش مراحل پروسه Build ایجاد می‌شود.



شکل (۳-۶) پروسه تولید کد توسط RTW

تعاریف Target & Host:

Target: عبارت از یک محیط سخت‌افزاری با یک سیستم عامل می باشد که قرار است کدهای ایجاد شده بر روی آن اجرا شود. فرآیند تولید کد Target توسط System Target file و make file و دستور make کنترل می‌شود. برای انتخاب Target دلخواه کافیسست از بین ایتهمهای موجود موارد دلخواه انتخاب شود.

Host: سیستمی است که مطلب، سیمولینک و RTW بر روی آن اجرا می‌شود. توسط پروسه تولید کد می‌توان فایل‌های قابل اجرا بر روی Target را تولید کرد.

۶-۲-۴- تولید S-function برای RTW

در این بخش به نحوه تولید S-function‌هایی که با RTW سازگاری دارند پرداخته می‌شود. با یکسری مفاهیم اساسی شروع کرده و به نحوه تولید یک S-function کاملاً بهینه می‌پردازیم.

کاربرد S-function

S-function قادر به حل طیف وسیعی از انواع مسائل و مشکلاتی است که هنگام کار با سیمولینک یا RTW با آنها مواجه می‌شویم این مسائل عبارتند از:

- پیاده‌سازی مجموعه‌ای از الگوریتم‌ها با سیمولینک یا RTW.

- نیاز به کدهای بهینه برای دستیابی به سرعت اجرایی بالا.

- ایجاد ارتباط بین کدهای دست‌نویس C با RTW.

S-function این امکان را به کاربر می‌دهد که الگوریتم‌های مورد نظر خود را با حداکثر قابلیت‌های ممکنه در سیمولینک پیاده کند البته به همان میزان و پیچیدگی، به حافظه و CPU سیستم نیاز خواهد داشت.

سه نوع S-function وجود دارد که هر کدام دارای ویژگی‌ها و کاربردهای خاصی است:

Noninlined S-function

Wrapper S-function

Fullyinlined S-function

حال به توضیح این سه نوع S-function می‌پردازیم:

- Noninlined S-function: Noninlined S-function تابعی از نوع C-mex است که قابل

استفاده برای سیمولینک و RTW می‌باشد. این نوع S-function توسط S-function API و به

زبان C پیاده‌سازی می‌شوند سپس سیمولینک و RTW بر حسب نیازهای خود روتینهای مورد نظر را از S-function احضار می‌کند. در این نوع S-function نیاز به حافظه و CPU زیاد است.

– S-function Wrapper: این نوع S-function مناسب برای ایجاد ارتباط بین کدهای دست‌نویس C با RTW می‌باشد. در این شیوه کدهای C مورد نظر در یک ماژول جداگانه نوشته می‌شود و در S-function فقط روتینهای تعریف شده در ماژول فرخوانی می‌شود. به خاطر همین موضوع که در S-function هیچ کدی در رابطه با الگوریتم نوشته نشده است S-function Wrapper گفته می‌شود.

– Fullyinlined S-function: این نوع S-function برای سیمولینک و RTW حکم بلوکهای Built-in را دارد به همین خاطر این نوع S-function نیاز به دو بار پیاده‌سازی دارند یکبار برای سیمولینک (C-mexS-function) و یکبار برای RTW (TLC). میزان پیچیدگی فایل TLC به میزان پیچیدگی الگوریتم پیاده‌سازی شده و میزان کارآیی مورد نیاز از S-function بستگی دارد.

۶-۲-۵- معرفی RTWin

RTW راه‌حلی است برای مدلسازی و تست Real Time مدل‌های دینامیکی بر روی یک PC. در واقع یک PC هم بعنوان Host و هم Target شناخته می‌شود.

در این محیط سخت‌افزاری بکمک مطلب و سیمولینک و Stateflow مدل سیمولینک ساخته می‌شود. مدل حاصله را می‌توان بوسیله سیمولینک در مد نرمال سیمولیشن کرد. سپس کد اجرایی مدل را توسط RTW تولید کرد و در نهایت فایل اجرایی حاصله را می‌توان با استفاده از External mode سیمولینک بصورت Real Time سیمولیشن کرد.

تعامل بین External mode سیمولینک RTWin باعث ایجاد یک محیط گرافیکی رابط کاربر شده است که امکانات زیر را فراهم می‌کند:

۱) Signal Visualization: تمامی بلوکهای Scope سیمولینک قابل استفاده خواهند بود.

۲) قابلیت تغییر پارامترها: توسط جعبه محاوره بلوکها می توان پارامترهای بلوکها را تغییر داد.

۳) کنترل Real Time.

۶-۲-۶ ساختار RTWin

۱- RTWin Kernel: RTWin از یک کرنل کوچک برای اطمینان از کارکرد Real Time استفاده

می کند این کرنل روی حلقه صفرم CPU اجرا می شود و از کلاک داخلی PC استفاده می کند.

جزئیات کرنل:

Interrupt: کرنل تمام وقفه ها را قبل از اینکه ویندوز دریافت کند گرفته و بررسی می کند.

Scheduler: لیست جدول ساعت.

ارتباط با سخت افزار I/O: رابط سخت افزاری I/O کرنل بکمک بلوکهای I/O امکان ارتباط بردهای

I/O را فراهم کرده است اگر برد سخت افزاری مورد نظر بدرستی نصب شده باشد درایور امکان

استفاده Real Time را فراهم خواهد ساخت.

Analog Input, Analog output, Digital input, Digital output, Counter, Encoder input.

درایورهای مورد نظر را برای گرفتن اطلاعات از input و ارسال اطلاعات به output فراخوانی

می کنند. این درایورها نیز روی حلقه صفرم CPU (CPU Ring zero) اجرا می شوند.

۲ - Real Time Application: بصورت Real Time روی PC اجرا می شود.

۳ - Signal Acquisition and Analysis: می توان سیگنالها را از ورودی درحین اجرای Real

Time Application گرفت این کار توسط ابزارهای زیر صورت می گیرد:

- Signal Tracing: ابزاری برای دریافت و مشاهده سیگنالها بصورت Real Time است.

- Signal Logging: ابزاری برای دریافت سیگنال بصورت Real Time است بطوریکه پس از اتمام مرحله دریافت سیگنال می توان اطلاعات را پردازش نمود.

۴- Parameter Tuning: می توان بصورت Real Time در حین اجرای برنامه پارامترهای بلوکها را تغییر داد و بلافاصله نتیجه را مشاهده کرد.

Simulink External mode: External mode سیمولینک نیازمند یک رابط برای ارسال پارامترهای سیمولینک به سیستم تحت کنترل و دریافت اطلاعات و مقادیر پارامترهای جدید بلوکها از آنها می باشد. در بعضی از RTW ها مثل TORNADO/VME Target، از پروتکل TCP/IP برای این ارتباطات استفاده می شود. تحت RTWin کامپیوتر host بعنوان Target نیز استفاده می شود. بنابراین فقط یک درایور مجازی برای تبادل پارامترها بین مطلب و سیمولینک توسط حافظه نیاز است .

Signal Acquisition: شما می توانید در حین اجرای Real Time سیگنالهای ورودی را دریافت (Capture) و نمایش دهید. سیگنالها از Real Time Application دریافت می شود و سیمولینک توسط بلوکهای Scope نمایش می دهد.

Parameter Tuning: Simulink External mode این قابلیت را فراهم کرده که می توان پارامترهای بلوکهای سیمولینک را در حین اجرا تغییر داد.

Simulink External mode: پارامترهای تغییر یافته را به Real Time Application ارسال می کند و این کار در حالیکه سیستم بصورت Real Time اجرا می شود انجام می پذیرد.

Data buffers & Transferring data: در هر بازه زمانی از Real Time Application، سیمولینک اطلاعات ورودی به سیستم را در قسمتی از حافظه ذخیره می کند تا اینکه بافر مورد نظر پر شود.

به محض اینکه بافر پر شد سیمولینک اطلاعات ورودی را بحالت معلق در می‌آورد تا اطلاعات بافر شده را ارسال کند و در عین این حال Real Time Application نیز بکار خود ادامه می‌دهد.

۶-۲-۷- معرفی TLC

TLC یکی از ملحقات RTW است که امکان استفاده از کدهای C را برای بلوکهای سیمولینک فراهم می‌سازد. این امر بطورکلی با inline نمودن S-function باعث کارایی بلوکها از لحاظ سرعت می‌شود.

فرآیند TLC

TLC تنها برای یک هدف طراحی شده است و آن تبدیل فایل توصیف مدل Model.rtw به کد Target. و به عبارتی دیگر TLC بعنوان یک جزء از Real Time فایل فرم میانی (Intermediate RTW form) یعنی Model.rtw را به کد C تبدیل می‌کند بطوریکه در کدهای تولید شده توسط TLC، توانائیها و محدودیتهای پردازنده Target نیز در نظر گرفته شده است.

البته TLC بیشتر بصورت یک پردازشگر متن عمل می‌کند به این صورت که کدهای همه بلوکهای سیمولینک موجود می‌باشد و TLC فقط با استفاده از فایلهای Target و فایل Model.rtw کدهای ANSI C تولید می‌کند اگر کمی با HTML، Perl و مطلب آشنایی داشته باشید، در خواهید یافت که TLC کلکسیونی از ایده‌های بکار رفته در آنها می‌باشد اینکه از یک گرامر (Syntax) از نوع mark-up استفاده می‌کند ایده‌ای برگرفته شده از HTML است. توانایی انعطاف پذیری کدهای نوشته شده را از Perl به ارث برده، و توانایی پردازش انواع دیتاها (همچون مطلب) را دارا می‌باشد. کدهای تولید شده توسط TLC کدهای بسیار بهینه شده هستند که بطور کامل Comment شده‌اند. TLC قابلیت کارکرد باهمه نوع مدلهای سیمولینک چه خطی، غیر خطی، پیوسته،

گسسته و هیبرید را دارد. همه بلوکهای سیمولینک بطور اتوماتیک به کدهای متناظرشان تبدیل می‌شوند به غیر از آنهایی که از توابع مطلب بصورت M-file استفاده کرده‌اند.

TLC بکمک اطلاعات موجود در Block Target file از فایل Model.rtw کدهای C تولید می‌کند. اطلاعات موجود در (Block Target file) بطور کلی مشخصات سخت افزاری سیستم Target را تعیین می‌کند تا کدهای تولید شده توسط TLC هماهنگی کاملی با Target داشته باشد.

۶-۲-۸- مفاهیم RTWin

RTWin راه‌حلی برای مدل‌سازی و تست سیستمهای دینامیکی بر روی یک PC است در واقع محیطی را فراهم ساخته که در آن از یک PC هم بعنوان host استفاده می‌شود و هم بعنوان Target.

در این محیط که روی یک PC یا Laptop قابل اجرا است بکمک مطلب، سیمولینک و Stateflow مدل مورد نظر ساخته می‌شود. این مدل توسط سیمولینک و در مدرنمال قابل سیمولیشن است ولی می‌توان بکمک RTW، Stateflow Coder و C Compiler فایل اجرایی از مدل تهیه نمود و آنرا بصورت Real Time سیمولیشن کرد.

۶-۲-۹- محیط سخت‌افزاری RTWin

سخت افزارهای مورد نیاز عبارتند از یک PC و بردهای I/O

۱ - PC

کامپیوتر مورد استفاده می‌تواند Desktop، Laptop، یا Notebook با Microsoft Windows 98، Windows NT، Windows 2000، Millennium یا XP باشد.

RTWin از بردهای I/O استاندارد سازگار با PC پشتیبانی می‌کند. هنگامی که مدل بصورت Real Time در حال اجرا است RTWin Target اطلاعات را از ورودیهای استاندارد گرفته و بعنوان بلوک Input از آن استفاده می‌کند بلافاصله آنها را پردازش کرده و به پورت برد I/O ارسال می‌کند. بردهای RTWin Target: I/O طیف وسیعی از بردهای I/O را پشتیبانی می‌کند از جمله این بردها، بردهای I/O از نوع ISA، PCI، PCMCIA هستند اینها شامل بردهای (A/D) Analog to digital، (D/A) Digital to Analog، Digital input، Digital output، Encoder input و ... هستند که بالغ بر دویست نوع برد می‌شود.

I/O driver block library: کتابخانه درایورهای بلوکهای I/O شامل درایورهای استاندارد برای بردهای I/O می‌باشد. که بالغ بر دویست نوع درایور I/O را پشتیبانی می‌کند و بعنوان رابط جهان بیرون Real Time Application عمل می‌کنند.

روش کار با این بلوک درایورها کاملاً مشابه سایر بلوکها می‌باشد.

از جمله این درایورها می‌توان به موارد زیر اشاره کرد:

۱ - سنسورها و عملگرهایی که به بردهای I/O وصل می‌شوند.

۲ - بردهای I/O که برای تبدیل ولتاژ به مقادیر عددی و یا مقادیر عددی به ولتاژ بکار می‌روند.

۳ - بردهایی که مقادیر عددی را از ورودی می‌خوانند یا مقادیر عددی را به خروجی ارسال می‌کنند.

البته درایورهای RTW محدود به موارد فوق نیست برای از بین بردن هر گونه محدودیت در این زمینه RTW این امکان را فراهم ساخته که کاربر درایور مورد نظر را نوشته و به کتابخانه درایورهای RTW اضافه کند.

۶-۲-۱۰- محیط نرم افزاری RTWin

امکانات نرم افزاری RTWin محیطی را برای طراحی، ساخت و تست مدلها را هم در مد Normal و هم Real Time فراهم می‌سازد.

(۱) Non Real Time Simulation: با استفاده از سیمولینک می‌توان مدل‌های مورد نظر را طراحی کرد و در مد Normal سیمولینک آنرا بصورت Non Real Time سیمولیشن کرد. در حالت Non Real Time سیمولینک از یک Time Vector برای سیمولیشن استفاده می‌کند. در این روش به محض اینکه مقادیر خروجی مدل برای یک Time Step محاسبه شد سیمولینک Time Step بعدی را شروع می‌کند. و از آنجا که این Time Vector هیچ ارتباطی با Clock سخت‌افزار ندارد به همین خاطر خروجی مدل بصورت Non Real Time خواهد بود.

(۲) Real Time Execution: برای یک اجرای Real Time بایستی Real Time Application ایجاد شده و از مد External سیمولینک استفاده کرد. Real Time Application: RTW، RTWin و C Compiler یک فایل اجرایی درست می‌کنند که می‌تواند بر روی کرنل بصورت Real Time اجرا شود.

Real Time Application از پارامترهای اولیه موجود در بلوکهای سیمولینک در هنگام تولید کد استفاده می‌کند. توجه به این نکته حائز اهمیت است که اگر مدل سیمولینک بصورت Continuouse Time طراحی شده باشد باید از الگوریتم انتگرالگیر Fixed-step استفاده کرد.

محیط نرم افزاری RTWin از منابع Real Time سیستم برای اجرای Real Time Application استفاده می‌کند.

براساس Time Sample Rate انتخاب شده RTWin Target از یکسری وقفه‌ها برای اجرای Real Time Application استفاده می‌کند. اینکار هم باعث اجرای Real time می‌شود و هم اینکه Real

Application Time با سرعت مناسبی به اجرای خود ادامه می‌دهد. هر بار که وقفه‌ای ارسال می‌شود تمامی خروجی‌های مدل محاسبه می‌شود.

۶-۳- معرفی xPC Target و تولید کد قابل اجرا بر روی آن

xPC Target جزئی از خانواده نرم‌افزارهای سیمولیشن سیستم‌های Real Time می‌باشد بعضی از این نرم‌افزارها برای مدل‌سازی Real Time ضروری هستند و بعضی دیگر فقط کاربردهای خاصی دارند. در ادامه به معرفی xPC Target ابزارها و امکانات آن می‌پردازیم و نحوه کنترل xPC Target توسط کاربر توضیح داده می‌شود.

۶-۳-۱- معرفی xPC Target

xPC Target راه‌حلی برای مدل‌سازی و سیمولیشن Real Time سیستم‌های دینامیکی روی PC های استاندارد است xPC Target از یک کامپیوتر که جدا از host PC است برای اجرای Real Time Application استفاده می‌کند.

در محیط host PC بکمک سیمولینک و مطلب و Stateflow می‌توان مدلهایی از سیستم‌های دینامیکی را طراحی کرده و بصورت Non Real Time آنها را سیمولیشن کرد xPC Target این امکان را فراهم ساخته که کاربر با اضافه نمودن بلوکهای I/O و با استفاده از یک host PC حاوی Stateflow و RTW و کامپایلرهای C/C++ بتواند فایل‌های اجرایی بسازد سپس فایل‌های اجرایی را از host PC به Target منتقل کند. کامپیوتر Target که از کرنل Real Time xPC Target استفاده می‌کند با اجرای برنامه منتقل شده روی آن می‌تواند سیستم را بطور Real Time کنترل کند.

سخت‌افزارهای مورد نیاز:

یک PC برای host و یک PC برای Target که PC مورد نیاز برای Target می‌تواند یک PC از نوع Desktop، Laptop، Industrial PC، PC/104، PC/104+ و یا Compact PCI باشد.

نرم‌افزارهای مورد نیاز:

xPC Target برای اجرا نیازمند نرم‌افزارهای زیر می‌باشد.

Microsoft Visual C/C++ (Version 5.0 , 6.0 or 7.0)

Watcom C/C++ (10.6 or 11.0)

مطلب و سیمولینک و RTW

۶-۳-۲- ابزارهای مورد نیاز برای xPC Target

xPC Target ورژن PC که بر روی host PC با سیستم عامل ویندوز نصب می‌شود و برای اجرا به محصولات نرم‌افزاری زیر نیازمند است.

۱ - Matlab: توسط Command مطلب می‌توان با xPC Target و Target Application ارتباط برقرار کرد. و در حالیکه سیستم بصورت Real Time اجرا می‌شود. کنترل کامل بر آن داشت این کار توسط xPC Target functions صورت می‌گیرد.

این توابع اعمال زیر را میسر می‌سازد.

الف) کنترل Real Time مثل Start، Stop و Download کردن Target Application.

ب) دریافت سیگنال و تحلیل آنها بصورت Real Time.

ج) قابلیت تغییر پارامترهای بلوکها در حالیکه Target Application بصورت Real Time اجرا می‌شود.

۲ - Simulink: بکمک این نرم‌افزار می‌توان سیستم دینامیکی و کنترلرهای مناسب برای آن سیستم طراحی کرد. xPC Target هماهنگی کاملی با انواع بلوکهای discrete-time و continuous-time دارد. البته در یک سیستم continuous-time برای تولید کد بوسیله RTW حتما باید از الگوریتم انتگرالگیری fixed-step استفاده کرد.

توجه به این نکته بسیار حائز اهمیت است که فقط S-functionهایی که به زبان C نوشته شده‌اند توسط RTW ساپورت می‌شوند در حالیکه S-functionهای M-file این قابلیت را ندارند. در xPC Target می‌توان ورودی و خروجی سیستم فیزیکی را با بلوکهای I/O جایگزین کرد و اطلاعات را از یک سیستم فیزیکی واقعی دریافت کرد. کتابخانه I/O نرم‌افزار xPC Target بیش از ۴۰۰ نوع درایور را ساپورت می‌کند. این درایورها دائما در حال افزایش هستند و همه به زبان C Code S-function نوشته شده‌اند.

۳ - RTW: بلوکهای سیمولینک و چارتهای Stateflow را به کد تبدیل می‌کند. که توسط xPC Target و بکمک دستور تولید کد RTW (Build) می‌توان Target Application را به روی کامپیوتر Target انتقال داد. از امکانات دیگر RTW پشتیبانی از سیستمهای Multirate است که امکان بهینه کردن کدها از لحاظ سرعت و کارایی را فراهم می‌سازد.

۴ - کامپایلر C: بکمک کامپایلر C و RTW فایل Target Application ایجاد می‌شود که این فایل بر روی سیستم Target حاوی کرنل xPC Target قابل اجرا است. xPC Target کامپایلرهای زیر را ساپورت می‌کند:

Watcom C/C++ (10.6 or 11.0)

Microsoft Visual C/C++ (Version 5.0 , 6.0 or 7.0)

۵ - XPC Embedded: امکان ارتباط Real Time با Target Application را فراهم می‌سازد.

البته xPC Target یک محصول مجزا محسوب می‌شود که نیازمند License جداگانه است. از قابلیت‌های xPC Target می‌توان به موارد زیر اشاره کرد.

الف) امکان بوت کردن Target PC از درایوهایی غیر از فلاپی درایو مثل هارد دیسک و یا flash memory.

ب) امکان ایجاد فایل‌های اجرایی برای Target PC بطوریکه این فایل‌ها کاملاً مستقل از host PC عمل می‌کنند.

۶ - امکان ایجاد برنامه‌های با رابط گرافیکی قابل اجرا بر روی host PC برای کنترل، تغییر پارامترها و دریافت اطلاعات از Target Application: برای ایجاد این برنامه‌ها می‌توان از توابع API خود xPC Target استفاده کرده و در محیطی مثل Visual Basic کدهای مورد نظر را نوشت.

۶-۳-۳- امکانات xPC Target

محیط xPC Target با فراهم ساختن ابزارهای زیر عملیات مدلسازی و تست Real Time سیستم‌های دینامیکی را ممکن ساخته است:

۱- Real Time Kernel

۲- Real Time Application

۳- Signal Acquisition

۴- Parameter Tuning

۱ - Real Time Kernel: xPC Target به هیچ سیستم عاملی (Windows, Linux, Dos) نیاز

ندارد و از یک کرنل بسیار بهینه بنام xPC Target Kernel بعنوان سیستم عامل استفاده می‌کند.

کرنل مورد نظر را خود xPC Target با ایجاد یک دیسک بوت کننده در اختیار کاربر قرار می‌دهد. البته xPC Target Embedded نیازمند سیستم عامل Dos است.

Target Boot Disk: این دیسک نیاز به نصب هر گونه نرم‌افزاری را برطرف می‌کند و امکان دستیابی به هارد دیسک Target PC را فراهم می‌کند. به این ترتیب می‌توان پس از اتمام عملیات تست Real Time بر روی xPC Target از Target بعنوان یک PC استفاده کرد. در این روش هیچ نرم‌افزاری بطور دائم بر روی PC نصب نخواهد شد.

Target PC Bios: تنها نرم‌افزار مورد نیاز برای اجرای کرنل xPC Target است. به همین خاطر xPC Target برای کنترل بهینه سیستم با اجازه کاربر تغییراتی را در Bios اعمال می‌کند. به محض لود شدن Bios، بایوس بدنال دیسک بوت می‌گردد این دیسک شامل دو بخش ۱۶ بیتی و ۳۲ بیتی است. ابتدا قسمت ۱۶ بیتی توسط CPU در مد Real اجرا می‌شود و باعث می‌شود که اولاً CPU به مد Protected منتقل شود و ثانیاً Descriptor table ایجاد شود در نهایت با اجرای بخش ۳۲ بیتی، xPC Target Kernel اجرا می‌شود. پس از اجرای Kernel اطلاعاتی راجع به host-Target نشان داده می‌شود.

سپس کرنل سعی می‌کند فایل Target Application را از کامپیوتر host بر روی Target منتقل کند. به محض اتمام عملیات لود و آماده شدن Target Application می‌توان توسط توابع xPC Target و سایر امکانات با Target Application ارتباط برقرار نموده و آنرا کنترل کرد.

۲ - Real Time Application: RTW, Stateflow Coder و کامپایلر مدلهای سیمولینک را به Real Time Application تبدیل می‌کنند Target Application حاصله می‌تواند بصورت Real Time روی PC استاندارد بدون نیاز به هر گونه سیستم عاملی اجرا شود Target Application دارای مشخصات زیر است:

Memory Model: Target Application بصورت یک برنامه ویندوز NT کامپایل شده که از مدل flat memory استفاده می‌کند. این برنامه سپس به یک Image مناسب برای xPC Target تبدیل می‌شود که بدون نیاز به هر گونه سیستم عاملی قابل اجرا است.

Task Execution Time: Target Application قادر به اجرای taskهایی با سرعت 100kHz (۱۰ میکرو ثانیه) است

۳ - Signal Acquisition: کرنل xPC Target سیگنالهای ورودی را از Target Application را در RAM ذخیره می‌کند و برای کاربر امکان آنالیز و مشاهده آنها را فراهم می‌کند. علاوه بر آن xPC Target امکاناتی مثل Signal Monitoring, Signal logging و Signal tracing را نیز فراهم ساخته است.

۴ - Parameter Tuning: بسیاری از بلوکهای سیمولینک دارای پارامترهای قابل تنظیم هستند که می‌توان مقادیر آنها را قبل و یا در حین اجرای Target Application بصورت Real تنظیم کرد. برای این تنظیمات xPC Target دو روش ارائه می‌کند:

الف) Interactive: قابلیت تنظیم فعال را در حین اجرای Real Time فراهم می‌کند تغییرات ایجاد شده بر روی پارامترها بلافاصله در خروجی سیستم منعکس می‌شود.

ب) Script Procedures: مجموعه دستورالعملهای تنظیم پارامترها را می‌توان بصورت یک Script نوشته و توسط کامپیوتر host آنها اجرا کرد.

۶-۳-۴ محیط سخت‌افزاری xPC Target

محیط سخت‌افزاری تشکیل شده است از کامپیوتر host، بردهای I/O کامپیوتر Target و ارتباط سریال یا شبکه‌ای بین host و Target.

۱- host PC: هر گونه PC با سیستم عامل ویندوز برای اینکار جوابگو می باشد این PC بایستی دارای یک فلاپی درایو و یک پورت سریال برای ارتباط کارت آداپتور اترنت باشد.

۲- Target PC: برای کامپیوتر Target می توان از هر گونه PC با پردازنده Intel 386/486/Pentium یا AMD k5/k6/Athlon استفاده کرد.

این کامپیوتر نیز باید دارای یک فلاپی درایو و یک پورت سریال برای ارتباط آداپتور کارت اترنت باشد.

یک کامپیوتر Target می تواند یکی از موارد زیر باشد:

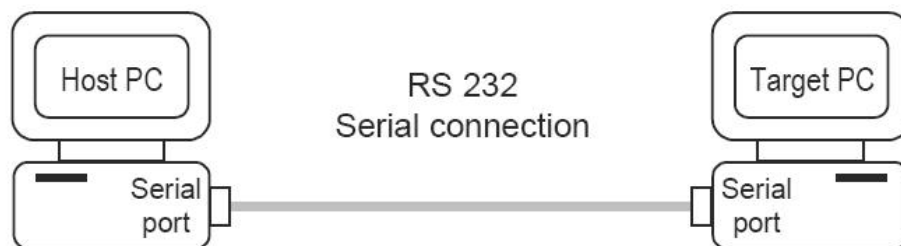
۱- Desktop PC

۲- Industrial PC

هنگام استفاده از Industrial PC می توان سخت افزارهای Compact PCI, PC/104+, PC/104 و یا Signal Board Computer(SBC) را انتخاب کرد.

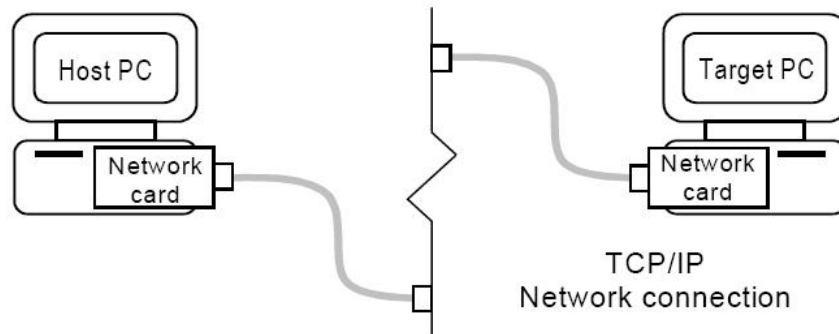
Host-Target Connection: xPC Target دو نوع ارتباط بین host و Target را ساپورت می کند.

۱- سریال: کامپیوترهای host و Target مستقیماً توسط پورت RS232 و با کابل سریال به هم وصل می شوند. سرعت انتقال داده در این شیوه بین 1200 bounds تا 115200 bounds می باشد.



شکل (۴-۶) ارتباط سریال RS 232

۲- شبکه: کامپیوترهای host و Target توسط شبکه به هم وصل می‌شوند. شبکه می‌تواند LAN یا Internet و یا کابل اترنت باشد. هر دو کامپیوتر توسط آداپتور کارتهای اترنت به شبکه وصل می‌شوند و طبق پروتکل TCP/IP با هم ارتباط برقرار می‌کنند. سرعت انتقال داده بین 10 تا 100 مگابیت بر ثانیه خواهد بود.



شکل (۵-۶) شبکه کامپیوترهای host و Target طبق پروتکل TCP/IP

I/O Driver: xPC Target طیف وسیعی از انواع بردهای I/O را پشتیبانی می‌کند از جمله ISA، PC/104، PC/104+ و Compact PC. این درایورها توسط بلوکهای در سیمولینک تعریف شده است که کاربر براحتی می‌تواند از آنها استفاده کند.

کتابخانه بردهای I/O: این کتابخانه شامل بلوکهای سیمولینک برای xPC Target است. I/O Support: کتابخانه I/O بالغ بر ۱۵۰ نوع درایور را ساپورت می‌کند. این بردها قابل نصب بر روی انواع busهای PC و PC/104 stock یا Industrial PC Chassis می‌باشد. xPC Target انواع بردهای I/O زیر را پشتیبانی می‌کند.

Analog input(A/D)

Analog output(D/A)

که برای ارتباط با انواع سنسورها و عملگرها با Target Application مناسب می‌باشد.

Digital input & output: برای ارتباط با سوئیچنگها، on/off devices و پورت پارالل مناسب می‌باشد.

RS232 support: از پورتهای COM1 و COM2 برای ارتباط سریال با بردهای خارجی استفاده می‌کند.

CAN Support: این رابط امکان ارتباط از طریق شبکه CAN بین Target Application و ابزارهایی مثل Remote Sensor و Actuator را فراهم می‌سازد.

سایر بردهایی که xPC Target پشتیبانی می‌کند عبارتند از: Counter Timer, UPD, GPIB, Watchdog, Incremental encoder, Shared memory.

۶-۳-۵- محیط نرم‌افزاری xPC Target

محیط نرم‌افزاری ابزاری است برای طراحی، ساخت و تست Target Application بصورت Non Real Time و Real Time این محیط همچنین امکان ارتباط host با Target را نیز فراهم می‌سازد. این محیط نرم‌افزاری امکانات زیر را فراهم ساخته است:

الف) ارتباط بین host و Target

برای تبادل اطلاعات بین host PC و Target PC می‌توان از ارتباط سریال RS232 یا ارتباط شبکه‌ای استفاده کرد. این اطلاعات شامل موارد زیر است:

۱- Target Application: برای انتقال Target Application از کامپیوتر host به کامپیوتر Target.

۲- Control: برای کنترل Target Application از جمله تغییر Start، Stop، تغییر Sample

Time و یا گرفتن اطلاعاتی در مورد کارایی Target Application و CPU بکار می‌رود.

ب) Rapid Prototyping Process

فرآیند مدل‌سازی سریع شامل طراحی و ساخت مدل روی host PC و اجرا و تست Target Application روی Target PC است.

فرآیند مدل‌سازی سریع شامل مراحل زیر است:

- ۱ - ساخت مدل سیمولینک و Stateflow.
- ۲ - سیمولشین مدل بصورت Non Real Time.
- ۳ - ساخت فایل اجرایی Target Application.
- ۴ - اجرای فایل Target Application بصورت Real Time روی xPC Target.
- ۵ - دریافت سیگنال از Target Application و مشاهده آن توسط xPC Target Scope.
- ۶ - تنظیم پارامترهای بلوکها توسط xPC Target Remote Control یا سیمولینک.

۶-۳-۶- نحوه کنترل xPC Target

محیط xPC Target یک محیط توسعه یافته برای ارتباط با کامپیوتر Target می‌باشد که ابزارها و امکانات متعدد نرم‌افزاری و سخت‌افزاری را برای ارتباط با کامپیوتر Target فراهم ساخته است.

الف) xPC Target Remote Control Tool

xPC Target از یک محیط گرافیکی برای ارتباط با Target Application استفاده می‌کند. این محیط گرافیکی با استفاده از توابع xPC Target و توابع گرافیکی مطلب ساخته شده است. این برنامه توسط دستور xpcrcctool قابل اجرا است.

xPC Target Real Time دارای امکانات زیر است:

۱ - محیط: با استفاده از xPC Target Setup می‌توان مشخصات محیط نرم‌افزار شامل نحوه ارتباط بین کامپیوترهای host و Target، نوع و محل کامپایلر C و مشخصات PC Target را تغییر داد.

۲ - کنترل: برای ساخت، انتقال و اجرای Target Application بکار می‌رود. همچنین تغییر زمان شروع Sample Time ها بدون نیاز به تولید مجدد کد از امکانات این تابع می‌باشد.

۳ - Signal Acquisition: بکمک xPC Target Simulink Viewer می‌توان بطور اکتیو اسکوپهای host یا Target به سیستم اضافه کرد.

۴ - تنظیم پارامترها: با استفاده از xPC Target Simulink Viewer می‌توان بطور اکتیو پارامترهای TargetApplication را تغییر داد.

(ب) Matlab command Line Interface:

روش دیگر برای ارتباط با xPC Target استفاده از محیط Command مطلب است. توابع xPC Target را می‌توان هم در محیط Command مطلب و یا بصورت یک M-file اجرا کرد. xPC Target شامل بیش از ۹۰ تابع مطلب است که برای کنترل Target Application از روی host PC بکار می‌رود.

محیط گرافیکی رابط کاربر مکمل بسیاری از امکانات xPC Target است. مشابه محیط مطلب نیز توابعی برای Environment، Control، Signal Acquisition و Parameter Tuning فراهم کرده است.

(ج) Simulink External Mode

Simulink External Mode برای ارتباط بین بلوک دیاگرام سیمولینک و Target Application می‌باشد. بلوک دیاگرام ارتباط گرافیکی با Target Application که بصورت Real Time اجرا

می‌شود را فراهم می‌سازد. بطوریکه با تغییر پارامترهای بلوکها بلافاصله پارامترهای Target Application نیز تغییر می‌کند.

سایر ابزارهای ارتباطی با Target Application عبارت است از:

۱ - Simulink with xPC Target Scope Blocks: یکسری بلوکهای اسکوپ را برای مشاهده و

پیگیری سیگنالها به xPC Target اضافه می‌کند

۲ - Target PC Command Line Interface: توابع xPC Target را بر روی Target

Application اجرا می‌کند

۳ - Web Browser Interface: بکمک Microsoft Internet Explorer یا Netscape Navigator

به xPC Target وصل می‌شود و امکان کنترل xPC Target را از روی هر کامپیوتری امکانپذیر

می‌سازد. همچنین اگر Target و host توسط کابل RS232 به هم وصل شده باشد باز می‌توان از

Web Browser برای کنترل استفاده کرد.

۶-۳-۷- استفاده از شتابدهنده سیمولینک برای افزایش سرعت اجرایی مدل

شتابدهنده سیمولینک باعث اجرای سریع مدل‌های سیمولینک می‌شود. برای اینکار شتابدهنده

سیمولینک و Real Time Workshop و C Compiler برای تولید مدل‌های اجرایی سریع استفاده

می‌شود.

شتابدهنده از کدهای C تولید شده توسط کامپایلر برای سیمولیشن مدل دینامیکی استفاده

می‌کند، که ذاتا دارای سرعت اجرایی بالایی هستند. در حالیکه در حالت Normal کدهای مدل

توسط مفسر سیمولینک اجرا می‌شود که اغلب سرعت اجرایی پائینی دارند.

شتابدهنده کدهای C تولید کرده و توابع mex مطلب با احضار کامپایلر عملیات Link را انجام می‌دهد. از طرف دیگر در روش Accelerated، سیمولینک دیگر نیاز به انجام محاسبات اضافی مخصوص مفسر سیمولینک که در حالت Normal صورت می‌گیرد، ندارد.

مراحل انجام شده طی فرایند Acceleration بصورت زیر می‌باشد:

۱ - تولید کدهای C و کامپایل آن.

۲ - ذخیره کدهای در شاخه جاری.

۳ - ذخیره فایل‌های mex در شاخه جاری.

۴ - اجرای مدل کامپایل شده.

پس از تولید فایل‌های اجرایی توسط شتابدهنده در اجراهای بعدی دیگر نیازی به کامپایل مجدد و تولید فایل‌های اجرایی نیست مگر اینکه تغییراتی در مدل سیستم داده شود که در اینصورت سیمولینک مجدداً اقدام به تولید فایل اجرایی جدید خواهد کرد.

۶-۳-۸- استفاده از RTW برای تولید کد Real Time

گزینه Build Model از RTW منوی Tools از مدل سیمولینک کدهای Real Time می‌سازد. پروسه تولید کد طی دو مرحله انجام می‌شود که برای تکمیل پروسه و ایجاد یک برنامه قابل اجرا به دو مرحله دیگر نیز احتیاج دارد.

این چهار مرحله پروسه تولید کد با زدن کلید Build بطور اتوماتیک پشت سرهم انجام می‌شود. این چهار مرحله عبارتند از:

۱ - تحلیل بلوک دیاگرام و کامپایل آن به model.rtw توسط RTW.

۲ - فراخوانی TLC برای تبدیل model.rtw به کدهای C.

۳ - تولید فایل make file از روی مشخصات سیستم Target.

۴ - کامپایل و لینک نمودن obj فایلها و Library برای تولید کد قابل اجرا.

۶-۳-۹- استفاده از xPC Target برای تولید کد قابل اجرا بر روی سیستم

آخرین مرحله از فرآیند برنامه‌سازی Machine Vision استفاده از xPC Target برای مدل‌سازی و سیمولیشن Real Time سیستمهای دینامیکی روی PCهای استاندارد است. در محیط host PC بکمک سیمولینک و مطلب و Stateflow می‌توان مدلهایی از سیستمهای دینامیکی را طراحی کرده و بصورت Non Real Time آنها را سیمولیشن کرد. xPC Target این امکان را فراهم ساخته که کاربر با اضافه نمودن بلوکهای I/O و با استفاده از یک host PC حاوی Stateflow و RTW و کامپایلرهای C/C++ بتواند فایل‌های اجرایی بسازد سپس فایل‌های اجرایی را از host PC به Target منتقل کند. کامپیوتر Target که از کرنل xPC Target Real Time استفاده می‌کند با اجرای برنامه منتقل شده روی آن می‌تواند سیستم را بطور Real Time کنترل کند.

فصل ۷

نتیجه گیری و

پیشنهادهات

۷-۱- نتیجه گیری

در این پایان نامه طرح جدیدی تحت عنوان ربات CamCrane برای فیلمبرداری از اجسام متحرک سریع ارائه شد و مشخصات آن شامل طراحی مکانیزم ربات، سیستم پردازش تصویر و سخت افزار و نرم افزار کنترلی بطور کامل توضیح داده شد. موارد کاربرد این نوع ربات و مزایا و برتریهای آن نسبت به گونه های قبلی (که اغلب صرفاً یک مکانیزم بودند و توسط اپراتور کنترل می شد.) در فصل اول توضیح داده شد.

در فصل دوم به طراحی مکانیزم ربات پرداخته شد و پس از تعریف اهداف و خواسته های مورد نیاز از مکانیزم طرحی بهینه ارائه و طراحی شد و جزئیات مربوط به انتخاب قطعات مخصوصاً موتورها، عملگرها، گیربکس ها، میل راهنما، یاتاقانها و سایر قطعات بررسی شد. تحلیلهای دینامیکی و استاتیکی مکانیزم طراحی شده نیز در انتهای این فصل ارائه شد.

در فصل سوم به یکی از مشکلات اساسی در طراحی ربات CamCrane پرداخته شد و آن بدست آوردن موقعیت سه بعدی توپ در ورزشگاه بصورت Real-time می باشد. طی دو دهه گذشته تحقیقات وسیعی در زمینه Object Detection بکمک پردازش تصویر انجام گرفته است و الگوریتمهای متنوعی با قابلیت های گوناگون ارائه شده است ولی با این حال پیچیدگیهای پیاده سازیهای این الگوریتمها و مخصوصاً حجم عملیات بسیار بالای آنها دو مشکل اساسی نرم افزارهای پردازش تصویر محسوب می شود. در این فصل الگوریتمی هوشمند برای تشخیص توپ ارائه شد و با بهینه سازیهایی که بر روی آن انجام شده سعی گردید که حجم عملیات تقلیل یابد و برای اجرای Real-time روی کامپیوترهای شخصی سازگار گردد. در ادامه نحوه پیاده سازی نرم افزار سه بعدی ساز که از جدیدترین شیوه سه بعدی سازی استفاده می کند توضیح داده شد.

یکی دیگر از نتایج این پایان نامه ارائه روشی برای مدلسازی دینامیکی سیستمهای پیچیده و کنترل آنها بود که در فصل پنجم توضیح داده شد. نحوه لینک کردن نرم افزار شبیه ساز دینامیکی و نرم افزار طراح کنترلر ارائه شد و مدل طراحی شده برای CamCrane با این شیوه کنترل گردید.

نتایج فصل پنجم نشان می دهد که کنترلر MPC که در حال حاضر جزو پرکاربردترین کنترلرهای صنعتی محسوب می شود، برای کنترل ربات CamCrane نیز بسیار مناسب می باشد نتایج شبیه سازیها نشان می دهد که این کنترلر تمام اهداف کنترلی مورد نیاز را برآورده می کند. به همین ترتیب نکته قابل توجه در فصل ششم ارائه شیوه جدید پیاده سازی کنترلرها برروی سخت افزار xPC بود که علی رغم هزینه بسیار پائین نسبت به نمونه های متداول امروزی، از سرعت پردازش بالاتری نیز برخوردار است و پیاده سازی و کار با آن نیز بسیار راحت تر می باشد.

۲-۷- پیشنهادات

با توجه به رشد روزافزون صنعت، بطور قطع در آینده ای نزدیک رباتهای CamCrane به عرصه ظهور خواهند رسید و بتدریج جای شیوه های صنعتی فیلمبرداری را تصرف خواهند کرد و تصاویری با کیفیت بسیار بالاتر تهیه خواهند نمود.

در این پایان نامه سعی شده طرح کلی و جامع برای ربات CamCrane ارائه شود، و برای تکمیل شدن بحث، تحقیق برروی مباحث زیر پیشنهاد می شود.

۱ - توسعه میدان کاری مکانیزم برای پوشش دادن نواحی وسیعتر

۲ - نرم افزار تهیه شده در این پروژه قادر به پردازش ۲۵ فریم بر ثانیه می باشد. برای فیلمبرداری از اجسام بسیار سریع نیاز به سرعت پردازشی ۱۰۰ و حتی ۲۰۰ فریم بر ثانیه احساس می شود.

۳- برای بهبود عملکرد کنترلر MPC پیشنهاد می‌شود از طریق جعبه ابزار System

Identification نرم افزار مطلب مدل ریاضی دقیقتری از ربات CamCrane بدست آید.

۴- موقعیت فضایی و نحوه جهت گیری دوربین نسبت به مسیر توپ یک امر سیلقه ای می‌باشد

که تابع نظرات شخصی کارگردان می‌باشد تهیه نرم افزاری که بتواند سیگنالهای مرجع مورد نیاز

کنترلر را بصورت حرفه ای و با استفاده از پارامترهای معرف نظرات شخصی کارگردان تولید کند

می‌تواند کیفیت فیلمبرداری را دو چندان کند.

مقاله مستخرج از پایان نامه

طراحی مکانیکی ربات فیلم برداری از اجسام متحرک

حسن ظهور، احمد کلاهی

دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف

چکیده

در این مقاله رباتی تحت عنوان CamCrane برای فیلمبرداری از اجسام متحرک سریع در دامنه‌ای وسیع معرفی و طراحی گردیده است. طراحی کامل ربات CamCrane، که شامل سه قسمت اصلی طراحی مکانیزم، طراحی سیستم پردازش تصویر و پیاده سازی کنترلر می‌باشد، توضیح داده شده است. در طراحی مکانیزم، مکانیزمهای فعلی بهینه شده است. طراحی سیستم پردازش تصویر شامل انتخاب قطعات سخت افزاری و پیاده سازی نرم افزار برای کاربرد ربات CamCrane تطبیق داده شده است. در شبیه سازی نرم افزار کنترلی برای اطمینان از عملکرد آن از ایده لینک دو نرم افزار مدل ساز دینامیکی و طراح کنترلر استفاده شده و در طراحی سخت افزاری برای پیاده سازی کنترلر نیز از سخت افزارهای نوین تحت عنوان xPC استفاده گردیده است. و در نهایت نتایج مدلسازیها و شبیه سازیهای انجام شده ارائه گردیده است.

کلمات کلیدی: CamCrane، xPC، مکانیزم، پردازش تصویر، مدلسازی، شبیه سازی، سیمولینک

۱) مقدمه

Camera Crane مکانیزمی است که به فیلمبردار امکان مانور و حرکت دوربین را در محدوده‌ای وسیع فراهم می‌کند و عموماً در صنعت فیلم سازی و تصویر برداری از صحنه های ورزشی کاربرد دارد شکل(۱) نمونه‌ای از این مکانیزم را نشان می‌دهد که برای فیلم برداری از مسابقه فوتبال مورد استفاده قرار می‌گیرد. طول بازوی این مکانیزم در حدود ۱۵ متر می‌باشد و دوربین در انتهای بازو قرار دارد این مکانیزم پایه ثابت بوده و بر روی خودرو نصب می‌شود.



شکل(۱) نمونه‌ای از مکانیزم Camera Crane

بطور کلی طرح ربات CamCrane طراحی جدید می‌باشد که در راستای تکامل مکانیزمهای Camera Crane ارائه می‌شود ولی با توجه به پیچیدگیهای طرح هیچ کدام به مرحله ساخت کامل نرسیده است هدف ما در این پایان‌نامه ارائه طراحی کامل برای ربات فیلم بردار از اجسام متحرک می‌باشد علاوه بر این نتایج شبیه سازیهای کامپیوتری نیز برای اطمینان از عملکرد صحیح طرح ارائه شده است و قسمت پردازش تصویر و نرم‌افزار کنترلی بطور کامل طراحی و پیاده سازی شده است و برای حالتی خاص مورد تست قرار گرفته است و قسمتهای سخت‌افزاری پروژه شامل طراحی مکانیزم و سخت‌افزار کنترلی فقط در حد طراحی و ارائه شبیه سازیها ارائه شده است.

۲) طراحی مکانیزم ربات

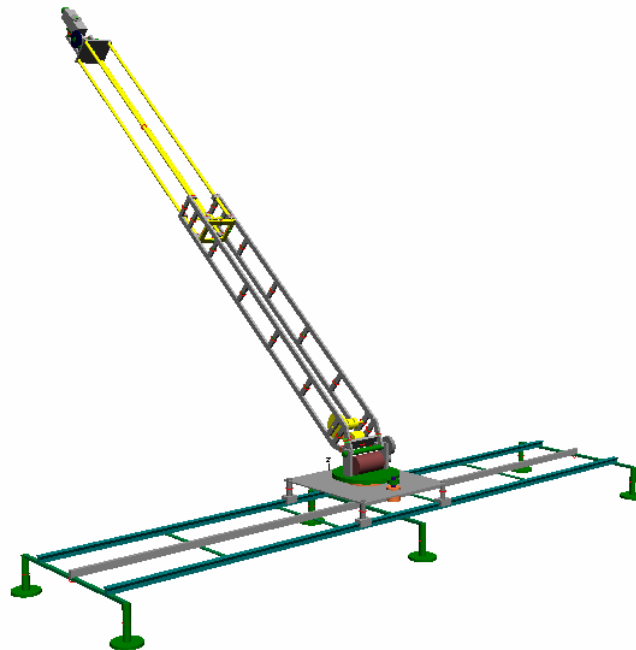
مکانیزم انتخاب شده برای CamCrane يك مکانیزم شش درجه آزادي است که می‌تواند دوربین فیلمبرداری را در تمام پوزیشن‌ها (فضای کاری این ربات ناحیه‌ای به وسعت $5 \times 10 \times 15$ متر می‌باشد) با هر موقعیت زاویه‌ای و جهت‌گیری دلخواه قرار دهد. پس از بررسی انواع مکانیزم‌های موجود مکانیزمی با طرح کلی زیر مورد تأیید قرار گرفت.

۱) کل مجموعه مکانیزم ربات بر روی واگنی نصب می‌شود که این واگن بر روی ریلی بطول ۵ متر قرار دارد و امکان حرکات انتقالی را فراهم می‌سازد.

۲) بازوی ربات مکانیزمی تلسکوپی می‌باشد که طول کلی آن ۵ متر می‌باشد که سه متر آن بازوی اصلی و ۲ متر نیز بازوی تلسکوپی می‌باشد و بر اساس دستورات کنترلی دوربین می‌تواند در فاصله شعاعی ۳ تا ۵ متر پوزیشن‌دهی شود.

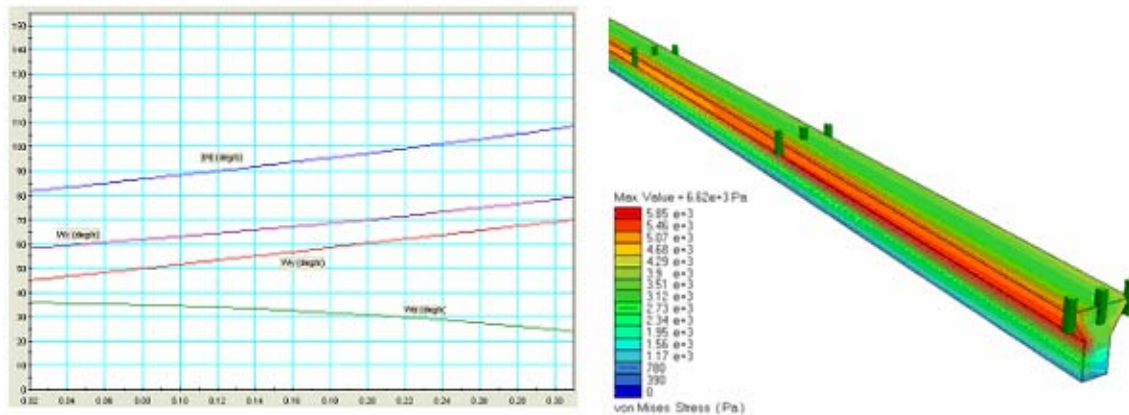
۳) دو اتصال لولایی عمود بر هم، دو درجه آزادي دورانی برای بازوی ربات را فراهم می‌کند.

۴) مشابهاً دو اتصال لولایی عمود بر هم دو درجه آزادي دورانی برای دوربین نصب شده در انتهای بازو را فراهم می‌کند. شکل (۲) مکانیزم پیشنهادی برای ربات CamCrane را نشان می‌دهد.



شکل (۲-۱۲) مدل سه بعدی طرح اولیه مکانیزم ربات CamCrane

مدلسازی سه بعدی انجام شده با یکسری تغییرات به محیط نرم افزار MSC Visual Nastran® منتقل شده است، تا تحلیلهای استاتیکی و دینامیکی بر روی آن انجام گردد. با تحلیلهای استاتیکی مقادیر تنش و کرنش در قسمتهای مختلف سازه را بدست آمد. و در تحلیلهای دینامیکی سرعت و شتاب T_k قطعات سازه به همراه نیروهای وارده به آنها را تعیین شده است. شکل (۳) نمونه‌ای از این تحلیلهای را نشان می‌دهد.



شکل (۳) نتایج تحلیل استاتیکی تنشهای وان میز و وارده بر ریل- نمودار سرعت زاویه‌ای بارو

۳) پردازش تصویر و بدست آوردن مختصات سه بعدی

هدف از پردازش تصویر بدست آوردن مختصات سه بعدی تولید اطلاعات فیدبک از مختصات سه بعدی توپ برای سیستم کنترلی می‌باشد از میان انواع روشهای موجود برای بدست آوردن مختصات سه بعدی توپ در زمین فوتبال، روش پردازش تصویر مناسب شناخته شد برای این منظور دوربینهای متعددی در محلهای مناسبی از ورزشگاه نصب می‌شود که هر کدام می‌تواند محدوده مشخصی از زمین بازی را در دید خود داشته باشد.

برای ثبت و اندازه‌گیری حرکت توپ با استفاده از تصویربرداری ویدیویی، به سخت افزارهای متعددی نیاز است این

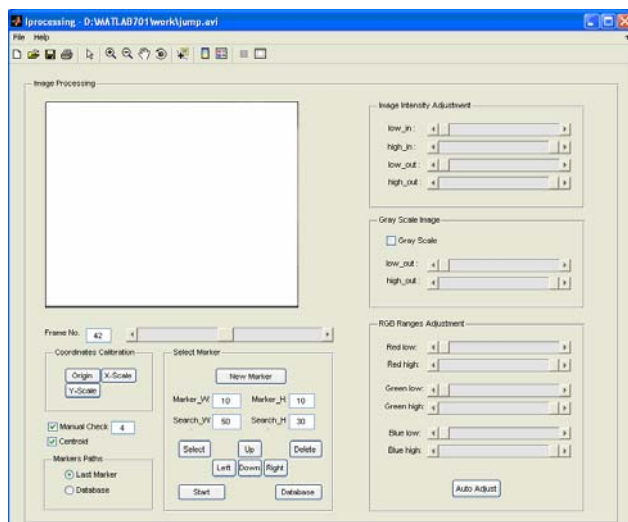
سخت افزارهای به شرح زیر می‌باشند.

- دوربینهای تصویربرداری و کارت واسطه
- کامپیوترهای پردازش تصویر
- چارچوب کالیبراسیون
- همزمان ساز دوربین ها (سنکرونایزر)

سیستم نرم‌افزاری پردازش تصویر شامل نرم‌افزارهای جداگانه‌ای برای دریافت تصویر از دوربین، استخراج فریم‌های فشرده شده از تصاویر، سنکرون کردن دوربین‌ها از طریق شناسایی فریم مبنا در هر یک، پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ از دید هر دوربین است. به طور کلی بخشهای زیر برای این نرم‌افزار در نظر گرفته شده است:

- دریافت تصویر از دوربین
- استخراج فریم‌های فشرده شده از تصاویر
- پردازش تصویر و به دست آوردن مختصات دو بعدی موقعیت توپ
- سنکرون کردن دوربین‌ها از طریق شناسایی فریم مبنا در هر یک
- ترکیب اطلاعات دو دوربین و به دست آوردن مختصات سه بعدی موقعیت توپ

کلیه قسمت‌ها در محیط نرم‌افزار MATLAB® پیاده‌سازی و اجرا می‌شوند. علت انتخاب محیط نرم‌افزاری MATLAB برای پیاده‌سازی نرم‌افزار پردازش تصویر، به قابلیت‌های متعدد آن در محاسبات ماتریسی برمی‌گردد که امکان انجام محاسبات حجیم پردازش تصویر با سرعت بالا را فراهم می‌کند. وجود ابزارهایی برای کار با تصاویر ویدئویی و پردازش تصویر و امکان برقراری ارتباط با دوربین فیلمبرداری برای دریافت تصاویر از دوربین، همچنین وجود کتابخانه‌ای عظیم از انواع توابع مورد نیاز برای برنامه‌نویسی سبب شده است که از این نرم‌افزار به عنوان پلت فرمی برای پیاده‌سازی سیستم نرم‌افزاری پردازش استفاده شود. شکل زیر اینترفیس نرم افزار پردازش تصویر را نشان می‌دهد.



شکل (۴): نمای کلی رابط نرم‌افزار پردازش تصویر

۴) پیاده سازی کنترلر پیش بین برای ربات CamCrane

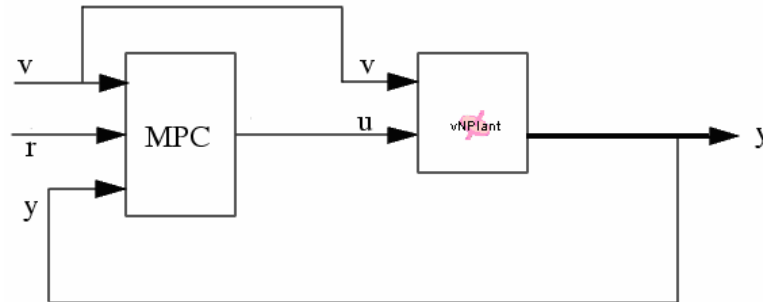
در مقاله مدل دینامیکی CamCrane بدون هیچگونه ساده سازی (بعبارتی دقیقاً همان مدل طراحی شده برای ساخت) برای طراحی کنترلر مورد استفاده قرار می‌گیرد. که اینکار از طریق طراحی مدل دینامیکی در محیط نرم افزار تحلیل دینامیکی MSC Visual Nastran® و طراحی کنترلر در محیط نرم افزار Simulink و لینک نمودن این دو نرم افزار صورت می‌گیرد. در این روش سیگنالهای کنترلی از نرم افزار نسترن به نرم افزار سیمولینک ارسال می‌شود و عملگرها و موتورهای سیستم دینامیکی مطابق سیگنالهای دریافتی عمل می‌کنند کار نرم افزار نسترن محاسبه وضعیت جدید مدل دینامیکی در پاسخ به سیگنالهای ورودی است و خروجیها که شامل تمام پارامترهای دلخواه (system's configuration) از مدل دینامیکی می‌باشد به نرم افزار سیمولینک بصورت فیدبک ارسال می‌شود.

پارامترهای ورودی مدل دینامیکی بشرح زیر انتخاب شده است:

- ۱ – سرعت زاویه ای موتور DC محرک واگن روی ریل
 - ۲ – سرعت زاویه ای موتور DC محرک اتصال لولایی قائم بازوی ربات
 - ۳ – سرعت زاویه ای موتور DC محرک اتصال لولایی افقی بازوی ربات
 - ۴ – سرعت زاویه ای موتور DC محرک بازوی تلسکوپی
 - ۵ – سرعت زاویه ای موتور DC محرک اتصال لولایی افقی دوربین
 - ۶ – سرعت زاویه ای موتور DC محرک اتصال لولایی قائم دوربین
- همانطور که مشاهده می‌شود پارامترهای ورودی سرعت زاویه‌ای موتورهای محرک قسمتهای مختلف ربات می‌باشد بدیهي است که پارامترهای خروجی نیز در واقع فید بک‌هایی است که از سنسورهای نصب شده در قسمتهای مختلف ربات تولید می‌شود. این پارامترها بشرح زیر می‌باشد.

- ۱ – موقعیت خطی واگن بر روی ریل
- ۲ – موقعیت زاویه ای اتصال لولایی قائم بازوی ربات
- ۳ – موقعیت زاویه ای اتصال لولایی افقی بازوی ربات
- ۴ – موقعیت شعاعی بازوی تلسکوپی
- ۵ – موقعیت زاویه ای اتصال لولایی افقی دوربین
- ۶ – موقعیت زاویه ای اتصال لولایی قائم دوربین

با در اختیار داشتن مدل دینامیکی می‌توان به پیاده سازی کنترلر بر روی مدل پرداخت. شکل (۵) مدل طراحی شده در سیمولینک برای کنترل ربات را نشان می‌دهد در این مدل بلوک مربوط به پلنت در واقع رابط دو نرم افزار نسترن و سیمولینک است. هدف کنترلر MPC هدایت دوربین به موقعیت فضایی و زاویه ای مشخص شده در قسمت سیگنالهای مرجع می‌باشد نتایج مربوط به نحوه کنترل و سرعت آن در رسیدن به نقطه مطلوب در انتهای مقاله آمده است.



شکل (۵) مدل طراحی شده در سیمولینک برای کنترل ربات

برای کنترل ربات از کنترلرهای پیش بین (Model Predictive Control) یا MPC استفاده شده است. کنترلر پیش بین چنانکه از نامش بر می‌آید، یک روش کنترلی است که در آن پیش بینی رفتار آینده یک سیستم در فرمول بندی مسئله کنترل وارد می‌شود.

پیش بینی، مقادیر آینده یک متغیر را بر مبنای اطلاعات در دسترس سیستم به دست می‌دهد. هر چه اطلاعات نمایانگرتری از سیستم داشته باشیم، دقت پیش بینی بهتر خواهد بود. آنگاه می‌توان تخمین متغیرهای آینده سیستم را برای طراحی قانون کنترل به کاربرد تا عملکرد بهتر کنترلی، که معمولاً رسانیدن یا نگهداشتن خروجی در یک مقدار مطلوب است، بدست آید.

لازم به ذکر است که سیگنالهای مرجع مورد نیاز برای کنترلر MPC توسط نرم افزار سه بعدی ساز تولید می‌شود. بدینصورت که پس از مشخص شدن موقعیت فضایی توپ در ورزشگاه، این نرم افزار موقعیت فضایی و زاویه ای مناسب برای دوربین را پیشنهاد می‌کند و بعنوان پارامترهای مرجع برای کنترلر، به آن ارسال می‌کند.

۶) طراحی سخت افزارهای کنترلی

سیستم پردازش تصویر به همراه نرم افزار کنترلی پیاده سازی شده به محاسبات بسیار زیادی نیاز دارد که از عهده سخت افزارهای معمولی خارج است. کامپیوترهای امروزی سرعت محاسباتی مناسبی دارند ولی سیستم عاملهایی مثل ویندوز یا لینوکس از کاربرد مورد نیاز ما فاصله زیادی دارد و جوابگوی نیاز ما نمی باشد. برای پیاده سازی سخت افزارهای کنترلی بر روی کامپیوترهای شخصی، نرم افزار مطلب سیستم عاملی بسیار کوچک و فوق العاده سریع را تدارک دیده که علاوه بر فراهم ساختن سرعت محاسباتی بسیار بالا امکان تبادل اطلاعات با سخت افزارها را نیز آسان نموده است. این کرنل بر روی سخت افزاری تحت عنوان Xpc-Target نصب و اجرا می شود.

۷) شبیه سازی و بحث بر روی نتایج

هدف از شبیه سازی اعمال یکسری سیگنالهای مرجع از نرم افزار سه بعدی ساز به کنترلر و مشاهده عملکرد پاسخ ربات به این سیگنالها می باشد. بعنوان مثال فرض می شود که در لحظه $t = 0$ ربات دارای شرایط اولیه ای بصورت زیر می باشد.

$$x_1 = 2.35m, \dot{x}_1 = 0.35m/s, \ddot{x}_1 = 0.2m/s^2$$

$$\theta_1 = 2.35rad, \dot{\theta}_1 = 2.35rad/s, \ddot{\theta}_1 = 0.2rad/s^2$$

$$\theta_2 = 0.35rad, \dot{\theta}_2 = 0.65rad/s, \ddot{\theta}_2 = 0.3rad/s^2$$

$$x_2 = 1.35m, \dot{x}_2 = 0.45m/s, \ddot{x}_2 = 0.3m/s^2$$

$$\theta_3 = 1.35rad, \dot{\theta}_3 = 0.25rad/s, \ddot{\theta}_3 = 0.6rad/s^2$$

$$\theta_4 = 2.25rad, \dot{\theta}_4 = 0.45rad/s, \ddot{\theta}_4 = 0.6rad/s^2$$

که موقعیت خطی واگن بر روی ریل و موقعیت زاویه ای اتصال لولایی فائم بازوی ربات می باشد حال فرض می کنیم در همان لحظه $t = 0$ سیگنال مرجعی با مشخصات زیر برای کنترلر ارسال می شود.

$$x_1 = 2.35m, \dot{x}_1 = 0.35m/s, \ddot{x}_1 = 0.2m/s^2$$

$$\theta_1 = 2.35rad, \dot{\theta}_1 = 2.35rad/s, \ddot{\theta}_1 = 0.2rad/s^2$$

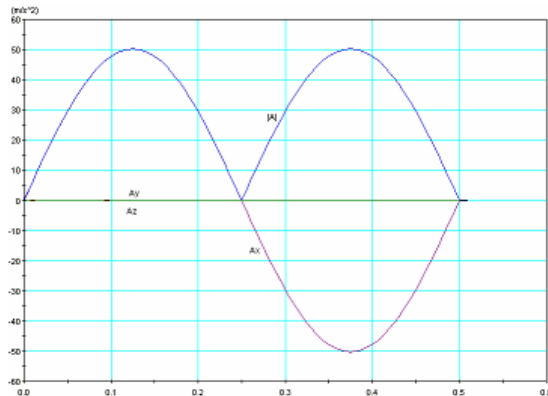
$$\theta_2 = 0.35rad, \dot{\theta}_2 = 0.65rad/s, \ddot{\theta}_2 = 0.3rad/s^2$$

$$x_2 = 1.35m, \dot{x}_2 = 0.45m/s, \ddot{x}_2 = 0.3m/s^2$$

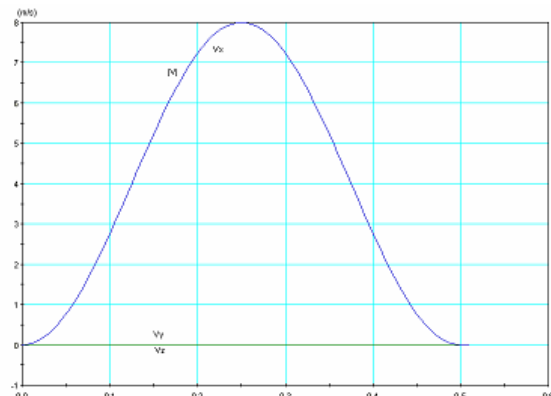
$$\theta_3 = 1.35 \text{ rad}, \dot{\theta}_3 = 0.25 \text{ rad/s}, \ddot{\theta}_3 = 0.6 \text{ rad/s}^2$$

$$\theta_4 = 2.25 \text{ rad}, \dot{\theta}_4 = 0.45 \text{ rad/s}, \ddot{\theta}_4 = 0.6 \text{ rad/s}^2$$

برای این سیگنال نتایج شبیه سازی شامل نمودارهای شتاب خطی و سرعت واگن روی ریل نسبت به زمان در شکل‌های (۶) و (۷) آمده است.



شکل (۷) نمودار شتاب خطی واگن



شکل (۶) نمودار سرعت خطی واگن

نتیجه‌گیری

در این مقاله طرح جدیدی تحت عنوان ربات CamCrane برای فیلمبرداری از اجسام متحرک سریع ارائه شد و مشخصات آن شامل طراحی مکانیزم ربات، سیستم پردازش تصویر و سخت افزار و نرم افزار کنترلی بطور کامل توضیح داده شد. موارد کاربرد این نوع ربات و مزایا و برتریهای آن نسبت به گونه های قبلی (که اغلب صرفاً یک مکانیزم بودند و توسط اپراتور کنترل می‌شد.) توضیح داده شد.

در ادامه الگوریتمی هوشمند برای تشخیص توپ ارائه شد، و با بهینه سازیهایی که بر روی آن انجام شد، سعی گردید که حجم عملیات تقلیل یابد و برای اجرای Real-time روی کامپیوترهای شخصی سازگار گردد. سپس نحوه پیاده سازی نرم افزار سه بعدی ساز که از جدیدترین شیوه سه بعدی سازی استفاده می‌کند توضیح داده شد.

یکی دیگر از نتایج این مقاله ارائه روشی برای مدلسازی دینامیکی سیستمهای پیچیده و کنترل آنها بود. نحوه لینک کردن نرم افزار شبیه ساز دینامیکی و نرم افزار طراح کنترلر ارائه شد و مدل طراحی شده برای CamCrane با این شیوه کنترل گردید.

1. A. Yamada, Y. Shirai and J. Miura, 'Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games', ICPR, 2002
2. TRACKING FOOTBALL PLAYERS WITH MULTIPLE CAMERAS , ICIP, 2004 Ming Xu James Orwell Graeme Jones
3. J. Segen, S. (Gopal) Pingali, A camera-based system for tracking people in real time, Proceedings of 13th International Conference on Pattern Recognition (ICPR'96), pages 63-67, Vienna
4. R. Tsai, 'An efficient and accurate camera calibration technique for 3D MachineVision', (١٩٨٤), ٣٢٣-٣٢٤. Proc. CVPR, pp
5. Real-Time Object Tracking for Soccer-Robots without Color Information, 2004 Andr e Treptow, Andreas Masselli and Andreas Zell
6. J. Amat, A. Monferrer, J. Batlle, X. Cufi, 1999, a low-cost underwater vehicle, Elsevier Science Ltd.
7. S. Negahdaripour, A.Khamene, 2000, Motion-based Compression Underwater Video Imagery for the Operation of Unmanned Submersible Vehicles, Elsevier Science Ltd.
8. Meng Joo Er, Chang Boon Low, Khuan Holm Nah, Moo Heng Lim, Shee Yong Ng, 2002, Real-time implementation of a dynamic fuzzy neural networks controller for a SCARA, Elsevier Science Ltd.
9. Meng Joo Er, Moo Heng Lim, Hui Song Lim, 2001, Real-time hybrid adaptive fuzzy control of a SCARA robot, Elsevier Science Ltd.
10. C. Larose, S. Guerette, F. Guay, A. Nolet, T. Yamamoto, H. Enomoto, Y. Kono, Y. Hasegawa, H.Taoka, 2003, Afully digital real-time power system simulator based on PC-cluster, Elsevier Science Ltd.
11. F.R. Driscoll, R.G. Lueck, M. Nahon, 1998, The motion of a deep-sea remotely operated vehicle system, Elsevier Science Ltd.
12. F. Pezeshkpour, VFM, Release 0.1 December 1998, Mathworks.com
analysis of Blanchi (1995), Three-dimensional 1- Paul Allard, Ian A.F.Stokes, Jean-Pierre kinetics human movement, Human
- 13, Kwon3D Motion Analyzing Web, 13- Kwon Y. H., Theoretical foundation,
- 14, <http://www.Mathworks.com>
15. MATLAB 6.5, PDF Documents, Mathworks, 2002

16. MSDN, Microsoft corporation, 2002
17. MSC Working Model 4D, Help Documentations, 2000
18. MSC Visual Nastran, Help Documentations, 2002

۱۹-فرهمنډ ف .، نریماني ر .، آبادپورا آ . و همکاران، طراحی و توسعه سیستم اندازه گیری و تحلیل سینماتیک و دینامیک حرکات ورزشی، گزارش تحقیقات بین دانشگاهی، دانشگاه صنعتی شریف، ۱۳۸۱

Mechanical Design of a Robot with Real-time Motion Tracking Camera

AHMAD COLAHI DAMIRCHIDARACI

*Mechanical Engineering Department
Sharif University of Technology*

Abstract

This thesis introduces new kind of robot named *CamCrane* to camera high speed moving objects in large domains. This robot is an enhancement of Camera Cranes mechanisms, which have wide applications in cinema industries. Having no control systems and even autonomous systems made these mechanisms to be unable in tracking of high speed objects. So in this thesis an improved version of Camera Cranes mechanism, CamCrane Robot, for tracking of high speed objects are proposed.

This dissertation deals with designing, analysis, modeling and control of CamCrane robot. It is challenged in three main fields: design of robot's Mechanism, design of Image Processing system and implementation of Control System. In design of mechanism our try mainly is concentrated on optimization of Camera Cranes mechanism, increasing the response time of mechanism, reducing of its weight and designing of actuators to handle the high speed motions are discussed. The Image Processing system is used to detect the desired moving object that must be traced. Selecting of required hardware and programming the image processing software are included in Image processing system's design. Finally implementation of robot's controller and design of its hardware are presented in details.

In the simulation of control software the new method of making links between Dynamic Modeling software and Control System Design software are implemented. And in design of control system's hardware the new type of hardware named xPC have been used. For reliable design numerous kinds of modeling and simulations are presented through the dissertation.

Keywords: CamCrane, xPC, Mechanism, Image Processing, Modeling, Simulation, Simulink

1. A. Yamada, Y. Shirai and J. Miura, 'Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games', ICPR, 2002.
2. TRACKING FOOTBALL PLAYERS WITH MULTIPLE CAMERAS , ICIP, 2004 Ming Xu James Orwell Graeme Jones
3. J. Segen, S. (Gopal) Pingali, A camera-based system for tracking people in real time, Proceedings of 13th International Conference on Pattern Recognition (ICPR'96), pages 63-67, Vienna.
4. R. Tsai, 'An efficient and accurate camera calibration technique for 3D MachineVision', Proc. CVPR, pp.(1986) ,344-353.
5. Real-Time Object Tracking for Soccer-Robots without Color Information, 2004 Andr e Treptow, Andreas Masselli and Andreas Zell
6. J. Amat, A. Monferrer, J. Batlle, X. Cufi, 1999, a low-cost underwater vehicle, Elsevier Science Ltd.
7. S. Negahdaripour, A.Khamene, 2000, Motion-based Compression Underwater Video Imagery for the Operation of Unmanned Submersible Vehicles, Elsevier Science Ltd.
8. Meng Joo Er, Chang Boon Low, Khuan Holm Nah, Moo Heng Lim, Shee Yong Ng, 2002, Real-time implementation of a dynamic fuzzy neural networks controller for a SCARA, Elsevier Science Ltd.

9. Meng Joo Er, Moo Heng Lim, Hui Song Lim, 2001, Real-time hybrid adaptive fuzzy control of a SCARA robot, Elsevier Science Ltd.
10. C. Larose, S. Guerette, F. Guay, A. Nolet, T. Yamamoto, H. Enomoto, Y. Kono, Y. Hasegawa, H. Taoka, 2003, A fully digital real-time power system simulator based on PC-cluster, Elsevier Science Ltd.
11. F.R. Driscoll, R.G. Lueck, M. Nahon, 1998, The motion of a deep-sea remotely operated vehicle system, Elsevier Science Ltd.
12. F. Pezeshkpour, VFM, Release 0.1 December 1998, Mathworks.com
- 13- Paul Allard, Ian A.F. Stokes, Jean-Pierre Blanchi (1995), Three-dimensional analysis of human movement, Human kinetics
- 14- Kwon Y. H., Theoretical foundation, Kwon3D Motion Analyzing Web, www.Kwon3d.com
15. <http://www.Mathworks.com>,
16. MATLAB 6.5, PDF Documents, Mathworks, 2002
17. MSDN, Microsoft corporation, 2002
18. MSC Working Model 4D, Help Documentations, 2000
19. MSC Visual Nastran, Help Documentations, 2002

۲۰-فرهمند ف.، نریمانی ر.، آبادپورآ. و همکاران، طراحی و توسعه سیستم اندازه گیری و تحلیل سینماتیک و دینامیک حرکات ورزشی، گزارش تحقیقات بین دانشگاهی، دانشگاه صنعتی شریف،

۱۳۸۱

۲۱-کوروش زندی، "کنترل پیشبین چند ماشینه توربین برق بادی از نوع تغذیه دابل" پایان

نامه کارشناسی ارشد دانشگاه زنجان، ۱۳۸۴

پیوست

پیوست ۱: سیمولینک و نحوه مدلسازی سیستمهای دینامیکی

در این بخش به معرفی سیمولینک، نحوه مدل سازی سیستمهای دینامیکی و سیمولیشن آنها توسط سیمولینک پرداخته می شود.

معرفی سیمولینک

سیمولینک نرم افزاری است که قابلیت های مدل سازی، سیمولیشن و آنالیز سیستمهای متغیر با زمان را فراهم می سازد به این سیستمها اصطلاحا سیستمهای دینامیکی گفته می شود. این نرم افزار عموما برای شبیه سازی سیستمهایی مثل مدارهای الکتریکی، مکانیکی و هیدرولیکی بکار می رود. سیمولیشن یک سیستم دینامیکی توسط سیمولینک پروسه ای دو مرحله ای می باشد:

مرحله اول: طراحی مدل گرافیکی سیستم توسط امکانات محیط نرم افزار

مرحله دوم: آنالیز سیستم در بازه زمانی مورد نظر توسط سیمولینک

نحوه مدل سازی سیستمهای دینامیکی

سیمولینک کتابخانه ای از انواع بلوکهای استاندارد تدارک دیده که کاربر می تواند بکمک آنها و محیط طراحی سیمولینک سیستم دینامیکی مورد نظر را طراحی کند. در زیر کلمات کلیدی در مدل سازی سیمولینک تعریف شده است.

بلوک دیاگرام: بلوک دیاگرام نمایشی گرافیکی از مدل سیستم دینامیکی است. هر سیستم دینامیکی تشکیل شده است از چندین بلوک که توسط یکسری خطوط بهم وصل شده اند.

بلوک: هر بلوک بیانگر یک المان سیستم دینامیکی است که با توجه به ورودیها و توابع حالت تعریف شده در هر زمان سیگنال خاصی را تولید و ارسال می‌کند.

حالت: بلوکها می‌توانند دارای بردار حالت باشند. بردار حالت متغیری است که خروجی بلوک را با توجه به مقدار بردار ورودی و مقدار بلوک در زمان قبلی تعیین می‌کند. بلوکی که دارای بردار حالت است، همواره نتایج بردار حالت در مرحله قبلی را ذخیره می‌کند تا در محاسبات زمان حال خود از آنها استفاده کند. به همین خاطر این بلوکها دارای نوعی حافظه برای ذخیره نتایج مرحله قبلی خود هستند. بعنوان مثال بلوک انتگرالگیر سیمولینک که از بردار حالت استفاده می‌کند برای انتگرالگیری از ورودی از لحظه شروع سیمولیشن تا زمان حال استفاده می‌شود. واضح است که مقدار

انتگرال در هر لحظه به نتیجه انتگرالهای زمان قبلی بستگی دارد البته بلوکهایی مثل بلوک Gain وجود دارد که فاقد بردار حالت هستند. خروجی این بلوک حاصلضرب مقدار ورودی در یک مقدار ثابت است.

تابع سیستم (System Function): هر بلوک سیمولینک دارای یک مجموعه توابع سیستم است که روابط متغیر با زمان بین ورودیها، بردار حالت و خروجی بلوک را مشخص می‌کند این توابع عبارتند از:

۱- تابع خروجی (Output Function): تابعی است که خروجی را بکمک مقادیر ورودی بردار حالت و زمان مشخص می‌کند.

۲- Update Function: تابعی است که برای محاسبه بردار حالت گسسته در استپ بعدی مورد استفاده قرار می‌گیرد. ورودیهای این تابع بردار حالت زمان حال و مقادیر ورودی بلوک می‌باشد.

۳- Derivative Function: تابعی است که رابطه بین مشتقات تابع حالت سیستم پیوسته و مقادیر فعلی تابع حالت و بردار ورودی را تعریف می‌کند.

بطور سمبلیک توابع سیستم بصورت زیر تعریف می‌شود:

$$y = f_o(t, x, u) \text{ Output Function}$$

$$X_{d_{k+1}} = f_u(t, x, u) \text{ Update Function}$$

$$X'_c = f_d(t, x, u) \text{ Derivative Function}$$

که در آن :

$$X = \begin{bmatrix} X_c \\ X_{d_k} \end{bmatrix}$$

که t زمان، X بردار حالت بلوک، u بردار ورودی بلوک، y بردار خروجی بلوک، X_d مشتق بردار حالت گسسته و X'_c مشتق بردار حالت پیوسته می‌باشد.

هنگام سیمولیشن، سیمولینک در هر استپ زمانی با احضار این توابع مقادیر بردار حالت و بردار خروجی را حساب می‌کند.

پارامترهای بلوک: بسیاری از مشخصات بلوکها پارامتری هستند. تعریف پارامتریک این مشخصات امکان تغییر مقادیر این مشخصات را در حین آنالیز مدل فراهم می‌سازد که در دستیابی به یک مدل بهینه بسیار مفید می‌باشد.

بلوکهایی که دارای این قابلیت باشند بلوکهای tunable گفته می‌شود بلوکهایی که فاقد این خاصیت باشند فقط می‌توان قبل از شروع آنالیز مقادیر پارامتریک آنها را تعیین کرد.

بلوکهای Discrete & Continues: بلوکهای سیمولینک به دو نوع پیوسته و گسسته تقسیم‌بندی می‌شوند. در بلوکهای پیوسته مقادیر خروجی بلوک با تغییر مقدار ورودی تغییر می‌کند اما در بلوکهای گسسته زمان پاسخگویی به ورودی فقط در لحظات خاصی امکان‌پذیر است که به Block Sample Time معروف است. در این نوع بلوکها مقادیر خروجی بلوک بین هر دو استپ همواره مقدار ثابت استپ قبلی است. هر بلوک گسسته دارای پارامتری تحت عنوان زمان نمونه‌برداری

است که در آن زمانها عملیات نمونه‌برداری و بهنگام‌سازی نتایج بلوک را انجام می‌دهد. نمونه‌ای از اینگونه بلوکها، بلوک Discrete Pulse Generator می‌باشد.

البته بعضی از بلوکها، مثل بلوک Gain، بسته به اینکه توسط بلوکهای پیوسته یا گسسته راه‌اندازی شده باشند، می‌توانند متناظرا پیوسته یا گسسته عمل کنند.

Solver: مدل‌های سیمولینک همواره مشتقات زمانی از بردارهای حالت را فراهم می‌سازد به همین خاطر سیمولینک هنگام سیمولیشن باید مقادیر پیوسته‌ای از بردار حالت را در هر لحظه محاسبه کند که اینکار را توسط انتگرالگیری از مشتقات موجود انجام می‌دهد.

انواع روشهای انتگرالگیری موجود است که هر کدام برای کاربردهای ویژه‌ای تدارک دیده شده است. سیمولینک از یک Solver تحت عنوان ODE (Ordinary Differential Equation) استفاده می‌کند که روشی بسیار موثر، پایدار و دقیق در محاسبات انتگرالگیری عددی است. البته خود این Solver زیرمجموعه‌های متنوعی دارد که سیمولینک به کاربر اجازه انتخاب آنها را می‌دهد.

نحوه سیمولیشن سیستمهای دینامیکی

منظور از سیمولیشن سیستمهای دینامیکی محاسبه مقادیر توابع حالت و بردار خروجی بلوکها در بازه‌های زمانی مشخص می‌باشد اینکار با زدن کلید Start از منوی Simulation شروع می‌شود.

شروع سیمولیشن یک مدل دینامیکی طی دو فاز صورت می‌گیرد

الف) فاز مقدار دهی اولیه مدل (Model Initialization): در این فاز بطور خلاصه عملیات زیر انجام می‌شود:

۱- محاسبه عبارات پارامتریک بلوکها و مقادیر عددی آنها

۲- تعیین نوع دیتاهای بلوکها: در این مرحله ابتدا نوع دیتاهای ورودی و خروجی بلوکها تعیین می‌شود سپس سیمولینک چک می‌کند که آیا ارتباط سیگنالهای بین بلوکها با توجه به نوع دیتاها امکانپذیر است یا نه.

۳- محاسبه میزان حافظه مورد نیاز برای بردارهای تعریف شده، بردارهای حالت، پارامترها و متغیرهای زمان اجرای مدل

۴- بهینه‌سازی بلوکها.

۵ - جایگزینی بلوکهای Subsystem: در این مرحله مدل از الگوی هیراچی خارج می‌شود و بلوکهای Subsystem با مدل‌های متناظر تعریف شده برای آنها جایگزین می‌شود.

۶ - عملیات سرت بلوکها به ترتیب اجرایی.

۷ - تعیین زمانهای نمونه‌برداری تک‌تک بلوکها.

۸ - تخصیص حافظه مورد نیاز برای متغیرها و بردارها.

ب) فاز اجرایی مدل: در این فاز سیمولیشن وارد فاز اجرایی می‌شود. سیمولینک مقادیر توابع حالت و بردارهای خروجی تک‌تک بلوکها را در زمانهای مشخص شده محاسبه می‌کند به این زمانها اصطلاحاً Time Step گفته می‌شود. زمانهای بین هر دو Time Step، Step Size نامیده می‌شود. مقادیر این دو عدد تابعی از نوع Solver انتخابی برای سیمولیشن و نوع سیستم دینامیکی از لحاظ داشتن ناپیوستگیها (Zero Crossing) می‌باشد.

پرونده‌های صورت گرفته در هر Time Step بشرح زیر می‌باشد:

۱ - بهنگام نمودن مقادیر خروجی بلوکها: سیمولینک با احضار تابع Output Function و ارسال مقادیری مثل زمان، بردار حالت و بردار ورودی مقدار خروجی را محاسبه می‌کند و چنانچه زمان موجود مضرب صحیحی از زمان نمونه‌برداری بلوک باشد آنرا به پورت خروجی بلوک ارسال می‌کند.

۲ - بهنگام نمودن تابع حالت مدل بلوک: سیمولینک با احضار تابع حالت اقدام به بهنگام‌سازی تابع حالت مدل بلوک می‌کند سیمولینک با انتگرالگیری از مشتقات گسسته زمانی تابع حالت، تابع حالت پیوسته‌ای را محاسبه می‌کند.

همچنین سیمولینک با احضار تابع مشتقگیر اقدام به محاسبه مشتقات زمانی تابع حالت می‌کند.

۳ - بررسی ناپیوستگیهای توابع حالت مدل‌های بلوک‌دیاگرام: سیمولینک بکمک تکنیکی به نام Zero-Crossing Detection ناپیوستگیهای موجود در تابع حالت را کشف می‌کند.

۴ - محاسبه مقدار t برای Time Step بعدی

سیمولینک این چهار مرحله را تکرار می‌کند تا زمان مشخص شده برای سیمولیشن تمام شود.

تعیین ترتیب بهنگام‌سازی بلوکها: هنگام سیمولیشن، سیمولینک در هر Time Step اقدام به بهنگام‌سازی توابع حالت و مقادیر خروجی بلوکها می‌کند مسئله قابل توجه در این پروسه ترتیب بهنگام‌سازی بلوکها در یک سیستم دینامیکی می‌باشد. در یک سیستم دینامیکی مقادیر ورودی هر بلوک وابسته به مقدار خروجی بلوک دیگر است و در یک سیستم بهم پیوسته متشکل از بلوکهای متعدد مشخص سازی ترتیب بهنگام‌سازی نتایج اهمیت بسزائی دارد. چرا که در غیر اینصورت مدل دینامیکی بدرستی سیمولیشن نخواهد شد. برای حل این مساله سیمولینک در فاز Model Initialization اقدام به تهیه ترتیبی مناسب برای بهنگام‌سازی بلوکها می‌کند، و برای اینکار از روش Direct Feedthrough استفاده می‌کند.

- روش Direct Feedthrough: به منظور تهیه لیستی از ترتیب بهنگام‌سازی بلوکها، سیمولینک پورتهای ورودی بلوکها را بر حسب رابطه خروجی آنها به ورودی، دسته‌بندی می‌کند. پورت ورودی که مقدار زمان حال آن در تعیین مقدار خروجی بلوک دیگری در همین زمان نقش داشته باشد Direct Feedthrough نامیده می‌شود. بعنوان مثال بلوکهای مثل Gain و Sum دارای Direct Feedthrough Port هستند. در حالیکه بلوکهای مثل Integrator و Constant اینگونه نیستند.

-قوانین ترتیب بلوکها: سیمولینک از قوانین پایه‌ای برای سرت بلوکها استفاده می‌کند. نتیجه اعمال این قوانین تهیه لیستی از ترتیب بهنگام‌سازی بلوکها است که ردیفهای اول این لیست بلوکهایی که بدون Port Feedthrough هستند قرار دارند سپس بلوکهای Feedthrough Port قرار می‌گیرند که ترتیب آنها بگونه‌ای است که نتایج ورودی درستی برای بلوکهای بعدی ایجاد می‌کنند.

Solver: سیمولینک برای سیمولیشن سیستمهای دینامیکی اقدام به محاسبه توابع حالت و بردارهای خروجی در Time Step های مشخص می‌کند. به پروسه محاسبه توابع حالت سیستم حل مدل گفته می‌شود. هیچ روش کارآمدی برای حل همه انواع مدلها وجود ندارد. به همین خاطر سیمولینک انواع Solver ها را برای سیمولیشن طیف وسیعی از سیستمهای دینامیکی تدارک دیده است. گزینه Simulation Parameters اجازه انتخاب انواع Solver های موجود در سیمولینک را فراهم می‌کند.

روشهای حل Fixed-Step و Variable Step:

از یک دیدگاه کلی Solver های سیمولینک به دو دسته Fixed-Step و Variable Step تقسیم‌بندی می‌شوند.

Fixed-Step Solver: در این روش سیمولینک در یکسری بازه‌های زمانی ثابت، بین زمان شروع و خاتمه سیمولیشن محاسبات خود را انجام می‌دهد. بازه‌های زمانی را می‌توان هم بصورت دستی و هم توسط یکسری روشهایی که توسط خود سیمولینک فراهم شده انتخاب کرد. بطور کلی هر چه این بازه‌های زمانی کوچکتر باشند نتایج دقیقتری از مدل دینامیکی حاصل خواهد شد.

Variable Step Solver: بازه‌های زمانی سیمولیشن توسط سیمولینک بصورت متغیر در نظر گرفته می‌شود. جایی که سیستم دینامیکی دچار تغییرات سریع است این بازه‌ها کوتاه در نظر

گرفته می‌شود و جایی که سیستم به حالت پایدار رسیده این بازه‌های زمانی تا حد امکان بزرگ در نظر گرفته می‌شود. بطوریکه در نهایت دقت مورد نظر در حل مدل حاصل شود.

هنگام سیمولیشن یک سیستم دینامیکی سیمولینک ناپیوستگی‌های متغیرهای حالت را به روش Zero Crossing Detection دائماً چک می‌کند. اگر سیمولینک ناپیوستگی را در متغیر حالت تشخیص دهد مقادیر Time Step را در این بازه زمانی به دو قسمت قبل از ناپیوستگی و بعد از ناپیوستگی تقسیم می‌کند.

از آنجا که ناپیوستگی‌های متغیر حالت اغلب عامل تغییرات ناگهانی در یک سیستم دینامیکی هستند به همین خاطر سیمولیشن ناحیه ناپیوستگی بسیار حائز اهمیت است. بعنوان مثال در سیمولیشن برخورد یک توپ با دیوار اگر لحظه برخورد بین دو Time Step رخ دهد سیمولیشن دچار اشتباه خواهد شد و بنظر خواهد رسید که توپ از دیوار عبور کرده است.

برای جلوگیری از این مساله واقع شدن یک Time Step بر روی ناپیوستگیها بسیار مهم است. سیمولاتوری که همه کارها را مطلقاً بر عهده Solver قرار دهد مطمئناً در اینگونه موارد دچار مشکل خواهد شد. بعنوان مثال در حالت Fixed Step کوچک نمودن بازه‌های زمانی در کل باعث بهتر شدن نتایج مدل می‌شود اما هیچ تضمینی برای رخداد ناپیوستگیها بر روی این Time Stepها وجود ندارد.

یک روش برای حل این مشکل استفاده از Variable Time Step Solver می‌باشد. در این روش Time Stepهای بهینه برای سیمولیشن انتخاب می‌شود بطوریکه اگر تغییرات مدل دینامیکی شدید باشد Time Stepها کوچک در نظر گرفته می‌شود و برعکس. از لحاظ تئوری این روش برای ناپیوستگیها نیز مؤثر می‌باشد، چرا که در ناپیوستگیها عموماً تغییرات مدل دینامیکی شدید است به همین خاطر گامهای زمانی در نظر گرفته شده برای سیمولیشن کوتاه و کوتاهتر می‌شود، اما مشکلی که هست افزایش حجم محاسبات سیمولیشن می‌باشد.

روش Zero Crossing Detection: سیمولینک از روشی بنام Zero Crossing Detection برای حل این مسئله استفاده می‌کند با این تکنیک هر بلوک می‌تواند مجموعه‌ای از Crossing-Zeroها را تشخیص داده و آنها را تحت عنوان توابع Crossing-Zero ثبت کند. این توابع به اینصورت عمل می‌کنند که هرگاه ناپیوستگی در سیستم رخ می‌دهد تغییر علامت می‌دهند به این ترتیب در پایان محاسبات هر Time Step سیمولینک این توابع را احضار کرده و نتایج آنها را چک می‌کند اگر نتایج حاکی از تغییر علامت این توابع باشد، سیمولینک به وجود ناپیوستگی در این بازه زمانی پی می‌برد. برای مشخص ساختن موقعیت دقیقتر ناپیوستگی سیمولینک از میانمایی مقادیر نتایج لحظه فعلی و لحظه قبلی استفاده می‌کند. به این ترتیب سیمولینک می‌تواند نقاط ناپیوستگی را با دقت خوبی تشخیص داده و سیمولیشن دقیقتری از مدل ارائه دهد.

معرفی S-function

S-function یک زبان کامپیوتری برای توصیف بلوکها در سیمولینک می‌باشد S-function را می‌توان به زبانهای Matlab، C، C++، Ada یا فرترن نوشت کدهای C، C++، Ada و فرترن توسط امکانات Matlab به mex فایل کامپایل می‌شوند و مثل سایر فایلها هنگام نیاز بطور دینامیکی به مطلب لینک می‌شوند. فراخوانی S-function الگوی خاصی دارد که امکان ارتباط آنرا با Solver سیمولینک فراهم می‌سازد. این ارتباط بسیار شبیه نحوه فراخوانی بلوکها توسط سیمولینک می‌باشد. S-function این امکان را می‌دهد که توسط یکسری قوانین تعریف شده، بلوکهای مورد نظر در سیمولینک ایجاد شود. از قابلیت‌های دیگر S-function ارتباط آن Real Time Workshop است.

پیوست ۲: مشخصات دوربین سرعت بالای Mikrotron مدل CAMMC1302

Mikrotron GmbH

Landshuter Str. 20-22

D-85716 Unterschleissheim

Germany

Tel.: +49 (0) 89-72 63 42-00

Fax: +49 (0) 89-72 63 42-99

info@mikrotron.de

www.mikrotron.de

- MC1302 monochrome
- MC1303 RGB colour with BAYER Filter
- Up to 100 fps @ 1280 (H) x 1024 (V) resolution
- 8 or 10 Bit
- "Base" Camera Link® output with 132 MB/s max. data rate
- Random selection of window size and position
- Asynchronous global shutter
- Horizontal and vertical pixelbinning
- Low power consumption
- Small, compact size

Flexible resolution and speed

The MC1302/03 of Mikrotron is a high-speed megaPixel CMOS camera with a resolution of 1280 (H) x 1024 (V) pixel and up to 100 fps. Contrary to high resolution CCD sensors, modern CMOS sensors offer high resolution and extremely high data rates. Because window size, position (ROI), and clock frequency is randomly programmable, resolution and frame rate of MC1302/03 can be easily adapted to any specific requirements.

Easy configuration

MC1302/03 configuration is done via the Camera Link® interface. Eight sets of camera configuration schemes can be stored in non-volatile memory of the cameras microcontroller. Even the FPGA configuration file can be downloaded for very special customer needs e.g. image preprocessing in MC1302/03 hardware.

Window size and position (ROI) on the sensor plus clock speed are programmable. With a pixel clock of 33 MHz on the “Base” Camera Link® connector and a window size of 100 x 100 pixel a frame rate of more than 5.000 fps can be achieved. Maximum video data rate at the 2 x 8 -Bit “Base” Camera Link® connector is 132 MB/s.

"Freeze Frame" full frame shutter

The MC1302/03 has a "Freeze Frame" shutter. This shutter "freezes" and stores the complete full frame at the end of the exposure time while exposing the next image. This results in extremely sharp, high-quality images, even if the object moves very fast. MC1302/03 can expose images synchronously or asynchronously.

In synchronous (free run) mode exposure time equals frame time or, for shorter exposure times, the electronic shutter can be activated with a minimal shutter time of 4µs. In asynchronous mode an image is exposed by an external trigger.

Camera Link® compatible data output

Using 2x10-Bit data output the full dynamic range of the sensor is available on the output. If less bandwidth and more speed is desired, any 8-Bits out of the 10-Bits of the sensor can be selected and output as 2x8-Bit within the “Base” Camera Link® output format.

Horizontal and vertical Pixelbinning

With high frame rates the exposure time is short. For more sensitivity the level of two adjacent or pixel in one row or one column or both can be summed.

Camera configuration tool

The MC1302/03 is delivered with a configuration program for Windows® operating systems. The clearly defined control surface allows to work with it intuitively. All parameters of the camera, e.g. image size and position, speed, shutter, exposure time may be set directly or via sliders. The resulting commands are transmitted by the serial interface of Camera Link®. The selected settings may be stored in the camera

or in a separate file. Also integrated is a monitor to show the transmitted commands and answers from the camera.

Technical Data	
MC1302 Sensor	linear response, monochrome
MC1303 Sensor	RGB colour with BAYER filter and UV-IR filter
Number of pixel	1280(H) x 1024 (V)
Pixel size	12 x 12 µm
Optical active area	15.36 (H) x 12.29 (V) mm 19.67 mm diagonal
Fill factor	40%
Spectral bandwidth	400 ... 800 nm
Illumination @ ADC Vref = 1V	1600 LSB/lux-sec at 550 nm
Internal dynamic	59 dB
Frame rates	100 fps @ 1.280 x 1.024 Pixel 1.155 fps @ 360 x 360 Pixel 5.099 fps @ 100 x 100 Pixel
Video output	"Base" Camera Link®, 2 x 8-Bit or 2 x 10-Bit
Pixel clock	20 - 85 MHz
Synchronisation	Internal or external pixel clock
Asynchronous shutter	Internal timer, 1024 steps, 4 µs to 32 ms or by pulse width of trigger signal
Gain	Digital x 1, 2, 4
Camera configuration	"Base" Camera Link®
Power supply	8 ... 24 V DC
Power consumption max.	4.5W
Thermal resistance typ.	0.17°/W
Case temperature	+5...50°C
Shock & vibration	70g, 7grms
Lens	C-mount, F-mount with adapter
Dimensions (W x H x D)	63 x 63 x 47 mm
Weight	app. 300 g

Connectors			
6-pin. Hirose power connector			
Pin	Signal	Pin	Signal
1	VCC	4	DGND*
2	VCC	5	GND
3	STRB	6	GND

*DGND...digital GND for STRB signal



"Base" Camera Link® Connector	
All signals according to "Base" Camera Link® specification	
Signal	Description
CC1	EXP
CC4	External Clock



پیوست ۳: مشخصات دوربین JVC مدل GR-DVL9800U

Integrated Digital Still Camera

680,000 Pixel 1/3" Progressive Scan CCD with Progressive Color Filter. High Band Processor for expanded 520 lines of horizontal resolution and a full 480 lines of vertical resolution

200,000 pixel - 3.5" High Resolution - Anti-glare, 270-degree rotating LCD color monitor

RM-V718U Multi-function Remote Control

F1.8 Bright lens

Digital CyberCam Video Camera

Electronic viewfinder with color 0.55" CRT

3 Mode Retractable Built-in Flash: Auto, Manual, Red Eye

Removeable Flash Memory, 4 MB MMC Card, Illuminated Panel with mode dial, Auto Standby for easy operation

10X Optical Hyper Zoom

200X Digital Hyper Zoom with Spline Interpolation. 10X Digital Pinpoint Zoom during playback

Digital Image Stabilizer

Hyper High-Speed Recording and Pro-Slow Playback

16x9 Squeeze Mode

PCM Digital Stereo Audio

Snap Shot Modes: Full screen,Frame, Pin-up,Negative, Multi-Picture, 4-Frame / 9-Frame

Program AE with 8 Digital effects and 17 scene transitions

Iris Lock

Self Timer

Wide Mode and Progressive Super Wide mode .7X zoom

Auto Macro

Manual focus, Exposure, and White Balance modes

Black Fader

Digital Wipes and Fades

Variable-speed Shutter (1/250, 1/100, 1/60 sec.)

High Density Image Recording, Mini DV NTSC (SD specifications)

BN-V607U 700 mAh Lithium-ion, High Capacity, rechargeable battery pack

Multi-voltage AC Power Adapter/battery charger

i.Link Digital Input/Output (IEEE 1394 compliant) DV in/out

PC Terminal and RS232 connection cable and software for Still Image Transfer & manipulation(PC/AT compatible)

J-Terminal

S-Video Output

S-Video input

CDROM Software:JVC MultiMedia Navigator, JLIP Video Capture, JLIP Video Producer, Picture Navigator, Presto! Mr. Photo, Presto! PhotoAlbum, Presto! ImageFolio
4 MB Multimedia Card, Shoulder strap, AV Output Cable, S-Video Cable, Editing Cable, JLIP cable, cleaning cloth

Audio Dubbing

Random Assemble Editing with digital effect/scene transition selections

1 year parts, 90 days labor warranty

1.54 lbs including battery & tape

Width 2 2/3"

Height 3 3/4"

Depth 5 7/15"

Power 4.4 Watts using viewfinder

پیوست ۴: جزئیات کدنویسی نرم افزار دریافت و پردازش تصویر

الف) جزئیات کدنویسی نرم افزار دریافت تصویر از دوربین

همانطور که پیش از این ذکر شد نرم افزار Capturing در محیط برنامه نویسی MATLAB و با

استفاده از GUI و جعبه ابزارهای زیر تهیه شده است:

Image Acquisition Toolbox : این جعبه ابزار شامل مجموعه توابعی برای دریافت تصاویر از

دوربین ها می باشد.

Image Processing Toolbox: این جعبه ابزار شامل مجموعه توابعی برای پردازش تصویر

می باشد.

توابع بکار رفته در این نرم افزار به شرح زیر است:

`:function Capturing_OpeningFcn(hObject, eventdata, handles, varargin)`

به محض اجرای نرم افزار این تابع فراخوانی می شود و یک شی از ورودی تصویر ایجاد کرده و آن را

نمایش می دهد، در واقع عملیات مربوط به پیش نمایش تصاویر گرفته شده از دوربین در این

فانکشن انجام می شود.

`:function Close_Exit_Callback(hObject, eventdata, handles)`

هنگام خارج شدن از برنامه اجرا می شود و اجرای برنامه را متوقف می کند.

`:function Deinterlacing_Callback(hObject, eventdata, handles)`

فریم های اخذ شده از دوربین را به نرم افزار Deinterlacing ارسال می کند.

:function Show_NO_Callback(hObject, eventdata, handles)

تعداد فریم مورد نمایش را از کاربر می‌گیرد.

:function openfile_Callback(hObject, eventdata, handles)

با فعال کردن دکمه Open file این فانکشن اجرا می‌شود. ابتدا یک کادر محاوره‌ای باز می‌شود و از کاربر نام فایل را پرسیده و آن را به حافظه لود می‌کند.

:function Save_AS_Callback(hObject, eventdata, handles)

با فعال کردن دکمه Save As این فانکشن اجرا می‌شود و با نمایش دادن کادر محاوره‌ای مربوط به ذخیره فایل، نام فایل را از کاربر گرفته و بازه فریم‌هایی را که توسط کاربر مشخص شده است به صورت یک فایل avi ذخیره می‌کند.

:function slider1_Callback(hObject, eventdata, handles)

با تغییر دادن اسلایدر مربوط به نمایش فریم‌ها این فانکشن اجرا می‌شود و علاوه بر نمایش فریم مربوطه شماره فریم را نیز درج می‌کند.

:function Start_capture_Callback(hObject, eventdata, handles)

به تعداد فریم‌های مشخص شده توسط کاربر از ورودی تصویر، فریم اخذ کرده و آن را در حافظه موقت نگهداری می‌کند.

(ب) جزئیات کدنویسی بخش Deinterlacing:

نرم افزار Deinterlacing در محیط برنامه نویسی MATLAB و با استفاده از GUI و جعبه ابزار های زیر تهیه شده است:

Curve Fitting Toolbox: این جعبه ابزار شامل مجموعه توابعی برای میانجایی و تصحیح داده ها است.

Image Processing Toolbox: این جعبه ابزار شامل مجموعه توابعی برای پردازش تصویر می باشد.

توابع به کار رفته در این نرم افزار به این شرح هستند:

`:function Deinterlacing_OpeningFcn(hObject, eventdata, handles, varargin)`

این فانکشن به محض اجرای نرم افزار فراخوانی می شود و مقداردهی اولیه تعدادی از متغیرها در آن انجام می گیرد.

`:function By_Mouse_Callback(hObject, eventdata, handles)`

با انتخاب دکمه By mouse این فانکشن اجرا می شود و به کاربر این امکان را می دهد که به کمک ماوس ناحیه ای را از تصویر که لازم است پردازش شود، انتخاب کند. خروجی این فانکشن مختصات ناحیه انتخاب شده توسط کاربر است.

`:function Contents_menu_Callback(hObject, eventdata, handles)`

این فانکشن زمانی اجرا می شود که کاربر از منوی Help گزینه Contents را انتخاب کند با اجرای این فانکشن مطالب مربوط به Help نرم افزار که به صورت فایل Html است نشان داده می شود.

`:function Exit_menu_Callback(hObject, eventdata, handles)`

به هنگام خروج از Help نرم افزار این فانکشن اجرا می شود.

```
:function Frame_NO_Callback(hObject, eventdata, handles)
```

این فانکشن زمانی فراخوانی می شود که کاربر شماره فریم را وارد کرده و کلید Enter را بزند. با اجرای این فانکشن فریم مورد نظر نشان داده می شود.

```
:function import_workspace_Callback(hObject, eventdata, handles)
```

این فانکشن با انتخاب گزینه Workspace از منوی فایل اجرا شده و اطلاعات مربوط به پروسه Capturing را به حافظه فراخوانی می کند.

```
:function Open_menu_Callback(hObject, eventdata, handles)
```

با انتخاب گزینه Open از منوی فایل این فانکشن اجرا شده و کادر محاوره ای برای انتخاب فایل نشان داده می شود. با انتخاب نام فایل، اطلاعات مربوط به آن فایل در حافظه بارگذاری شده و آخرین فریم فایل مربوطه نشان داده می شود.

```
:function popupmenu1_Callback(hObject, eventdata, handles)
```

این فانکشن مربوط به منوی کرکره ای می باشد و از کاربر نوع تصویر را می پرسد.

```
:function Save_menu_Callback(hObject, eventdata, handles)
```

تصاویر Deinterlace شده توسط این فانکشن روی هارد دیسک ذخیره می شود.

```
:function show_selection_Callback(hObject, eventdata, handles)
```

با کلیک کردن روی دکمه By Position این فانکشن اجرا می‌شود و ناحیه‌ای از تصویر که مختصات آنها توسط کاربر داده شده است توسط مستطیل قرمز رنگ احاطه می‌شود.

```
function slider1_Callback(hObject, eventdata, handles)
```

با اعمال هرگونه تغییر روی اسلایدر مربوط به فریم‌ها، این فانکشن اجرا می‌شود و فریم مربوطه را نشان می‌دهد.

```
function Start_Deinterlacing_Callback(hObject, eventdata, handles)
```

با زدن دکمه Start Deinterlacing این فانکشن فراخوانی می‌شود و الگوریتم پیاده‌سازی شده برای Deinterlacing بر روی فریم‌های مشخص شده توسط کاربر اجرا می‌شود. با اجرای این فانکشن ابتدا بازه فریم‌های مورد نظر از کاربر پرسیده می‌شود، سپس عملیات Deinterlacing بر روی ناحیه‌ای از فریم که قبلاً توسط کاربر انتخاب شده است انجام می‌شود.

الگوریتم پیاده‌سازی شده بدین صورت است که ابتدا سطرهای زوج و فرد هر فریم جدا سازی می‌شود و از هر فریم دو تصویر ساخته می‌شود که تصویر اول از سطرهای زوج فریم اولیه و تصویر دوم از سطرهای فرد فریم اولیه ساخته شده است. سطرهای خالی دو تصویر فاصله در مرحله بعد توسط میانبایی درجه سوم پر می‌شود و در نهایت تصاویر ساخته شده در ماتریس مربوط به فریم‌های Deinterlace شده جایدهی می‌شود.

ج) جزئیات کدنویسی برنامه پردازش تصویر و به دست آوردن مختصات دوبعدی مارکرها
نرم‌افزار Improcessing در محیط برنامه نویسی MATLAB و با استفاده از GUI و جعبه ابزارهای زیر تهیه شده است:

Curve Fitting Toolbox: این جعبه ابزار شامل مجموعه توابعی برای میانمایی و تصحیح داده‌ها است.

Image Processing Toolbox: این جعبه ابزار شامل مجموعه توابعی برای پردازش تصویر می‌باشد.

Excel Link Toolbox: این جعبه ابزار شامل مجموعه توابعی برای ارتباط با نرم‌افزار Excel است. توابع به کار رفته در این نرم‌افزار به این شرح هستند:

`:function All_markers_Callback(hObject, eventdata, handles)`

با فراخوانی این تابع اطلاعات مربوط به مسیر مارکرهای موجود در Database نشان داده می‌شود.

`:function Auto_adjust_Callback(hObject, eventdata, handles)`

با انتخاب کلید Auto Adjust این فانکشن اجرا می‌شود و تنظیمات خودکار اسلایدها را انجام می‌دهد.

`:function blue_high_Callback(hObject, eventdata, handles)`

`function blue_low_Callback(hObject, eventdata, handles)`

`function green_high_Callback(hObject, eventdata, handles)`

`function gray_low_Callback(hObject, eventdata, handles)`

`function red_high_Callback(hObject, eventdata, handles)`

`function red_low_Callback(hObject, eventdata, handles)`

شش فانکشن فوق مربوط به تنظیمات اسلایدهای RGB هستند با اعمال تغییرات روی هر یک از این اسلایدها، فانکشن مربوطه اجرا شده و با توجه به بازه رنگهای تعیین شده توسط اسلایدرها، فریم‌ها نشان داده می‌شود. به عنوان مثال اگر مقادیر اسلایدرها به صورت زیر باشد:

Red low = 100 Red high = 200

Blue low = 0 Blue high = 0

Green low = 0 Green high = 0

فقط پیکسلهایی نشان داده می‌شوند که مقدار رنگ قرمز آنها بین ۱۰۰ تا ۲۰۰ است و از سایر پیکسلها که بازه رنگ قرمز آنها خارج از این محدود است صرفنظر می‌شود. در ضمن از پیکسلهای آبی و سبز نیز صرفنظر شده و نشان داده نخواهند شد.

```
function Database_Callback(hObject, eventdata, handles)
```

با انتخاب دکمه Database این فانکشن اجرا شده و از کاربر بازه فریم‌هایی را می‌پرسد که اطلاعات مربوط به مارکرهای آنها باید ذخیره شود. سپس اطلاعات در حافظه موقت و در Database مربوطه ذخیره می‌شود.

```
function exit_menu_Callback(hObject, eventdata, handles)
```

با اجرای این فانکشن عملیات مربوط به توقف اجرای برنامه و خارج شدن از حافظه انجام می‌گیرد.

```
function Frame_NO_Callback(hObject, eventdata, handles)
```

این فانکشن با وارد کردن شماره فریم درخواستی برای نمایش و کلیک دکمه Enter اجرا می‌شود و فریم مربوط را نمایش می‌دهد.

```
function gray_high_Callback(hObject, eventdata, handles)
```

```
function gray_low_CreateFcn(hObject, eventdata, handles)
```

این دو فانکشن مربوط به اسلایدرهای تصویر خاکستری هستند و بازه رنگ خاکستری قابل نمایش در تصویر را مشخص می‌کنند به عنوان مثال با انتخاب gray low = 50 و gray high = 100 همه پیکسلهایی که مقدار عددی رنگ خاکستری آنها بین ۵۰ تا ۱۰۰ باشد نشان داده می‌شود.

```
function Gray_scale_Callback(hObject, eventdata, handles)
```

با انتخاب چک باکس نمایش تصویر خاکستری این فانکشن اجرا شده فریم‌ها را به صورت تک رنگ نشان می‌دهد.

```
function Help_contents_Callback(hObject, eventdata, handles)
```

این فانکشن زمانی فراخوانی می‌شود که گزینه Contents از منوی Help انتخاب شود در این صورت فایل Html مربوط به Help برنامه نشان داده می‌شود.

```
function imageupdate(hObject,handles)
```

این فانکشن توسط سایر فانکشن‌ها فراخوانی می‌شود و هدف از آن به روزرسانی فریم نمایش داده شده است به عنوان مثال وقتی اسلاید مربوط به تغییر فریم تغییر پیدا می‌کند فانکشن مربوط به اسلاید از این فانکشن برای نمایش فریم جدید استفاده می‌کند.

```
function IProcessing_OpeningFcn(hObject, eventdata, handles, varargin)
```

این فانکشن به محض اجرای نرم‌افزار فراخوانی شده و مقداردهی اولیه تغییرهای مورد استفاده در نرم‌افزار را انجام می‌دهد.

```
function last_marker_Callback(hObject, eventdata, handles)
```

فراخوانی این فانکشن سبب می‌شود که تنها مسیر مربوط به آخرین مارکر روی فریم نشان داده شود.

```
function load_workspace_Callback(hObject, eventdata, handles)
```

با انتخاب گزینه Workspace از منوی فایل این فانکشن اجرا می‌شود و فریم‌های موجود در حافظه موقت را لود می‌کند.

```
function Marker_name_Callback(hObject, eventdata, handles)
```

با زدن دکمه New maker این فانکشن اجرا می‌شود و نام مارکر جدید را از کاربر پرسیده و Database را برای ذخیره کردن مقادیر مربوط به مختصات مارکر آماده می‌کند.

```
function move_down_Callback(hObject, eventdata, handles)
```

```
function move_up_Callback(hObject, eventdata, handles)
```

```
function move_left_Callback(hObject, eventdata, handles)
```

```
function move_right_Callback(hObject, eventdata, handles)
```

این چهار فانکشن مربوط به حرکت دادن موقعیت مستطیل‌های معرف مارکر هستند. با زدن هر کدام از کلیدهای Right, Left, Up, Down به ترتیب یکی از این فانکشنها اجرا شده و موقعیت این مستطیل‌ها را به پایین، بالا، چپ یا راست انتقال می‌دهد.

```
function Open_excel_Callback(hObject, eventdata, handles)
```

این فانکشن با انتخاب گزینه Import Excel از منوی فایل اجرا شده و اطلاعات موجود در فایل اکسل را لود می‌کند و در Database قرار می‌دهد.

```
function popupmenu2_Callback(hObject, eventdata, handles)
```

با انتخاب منوی کرکره‌ای این فانکشن اجرا می‌شود و نوع الگوریتم مورد استفاده برای پردازش تصویر را مشخص می‌کند.

```
function Save_as_menu_Callback(hObject, eventdata, handles)
```

با انتخاب گزینه Export Excel از منوی فایل این فانکشن اجرا می‌شود با نمایش کادر محاوره‌ای مربوط به ذخیره سازی فایل، اسم فایل اکسل را از کاربر گرفته و اطلاعات موجود در Database

شامل نام مارکرها شماره فریم‌ها، زمان و مختصات دوبعدی هر یک از مارکرها را در فایل اکسل ذخیره سازی می‌کند.

```
function slider1_Callback(hObject, eventdata, handles)
```

این فانکشن با کلیک کردن روی اسلایدر نمایش تک تک فریم‌ها اجرا می‌شود و فریم مربوطه را نشان می‌دهد.

```
function slider2_Callback(hObject, eventdata, handles)
```

```
function slider3_Callback(hObject, eventdata, handles)
```

```
function slider4_Callback(hObject, eventdata, handles)
```

```
function slider5_Callback(hObject, eventdata, handles)
```

این چهار فانکشن مربوط به تنظیمات شدت نور تصویر ورودی و خروجی هستند که با تغییر هر کدام از اسلایدرهای مربوط فانکشن متناظر با آن اجرا شده و پس از انجام تغییرات اعمال شده روی تصویر، آن را نمایش می‌دهند.

```
function Start_Process_Callback(hObject, eventdata, handles)
```

این فانکشن با کلیک بر روی دکمه Start فعال می‌شود و در واقع بدنه اصلی برنامه پردازش تصویر است. با اجرای این فانکشن ابتدا شماره آخرین فریمی که پردازش خواهد شد از کاربر پرسیده می‌شود. به طور پیش فرض نرم‌افزار پردازش را روی تمام فریم‌هایی که بین فریم جاری و آخرین فریم لود شده اند انجام می‌دهد. الگوریتم مورد استفاده در این فانکشن به منظور به دست آوردن Correlation و مینیمم سازی آن بین تصویر مارکر و فریم مورد جستجو است که این کار توسط فانکشن Fndc انجام می‌شود. خروجی فانکشن Fndc موقعیت دوبعدی محل مارکر و فریم جدید است که پس از تعیین موقعیت مطلق آنها به Database اضافه می‌شود.

در این فانکشن عملیات دیگری همچون پیش‌بینی محل مارکر در فریم از طریق میانبایی و نیز الگوریتمی مربوط به تشخیص احتمال خطای فانکشن findc گنجانده شده است.

پیوست ۵: جزئیات کدنویسی نرم افزار بازسازی مختصات سه بعدی

جزئیات کدنویسی نرم افزار بازسازی مختصات سه بعدی به شرح زیر است:

(الف) دریافت فایل های مختصات پیکسلی مارکرها

`:function Transform_OpeningFcn(hObject, eventdata, handles, varargin)`

به محض اجرای نرم افزار، این تابع فراخوانی می شود و یک شی از نمای کلی نرم افزار ایجاد کرده و آن را نمایش می دهد، در واقع عملیات مربوط به بازسازی مختصات سه بعدی مارکرها در این تابع انجام می شود.

`:function Start_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Start این تابع اجرا می شود. بارگذاری فایل ها با اجرای این تابع آغاز می شود.

`:function Open_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Open این تابع اجرا می شود. ابتدا یک کادر محاوره ای باز شده و از کاربر مسیر و نام اولین فایل پیکسلی مارکرها پرسیده و آن را در حافظه بارگذاری می نماید.

`:function Next_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه های Next این تابع اجرا می شود. یک کادر محاوره ای باز شده و از کاربر مسیر و نام فایل پیکسلی مارکرها پرسیده می شود.

`:function Import3d_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Real Coordinates این تابع اجرا می شود. در کادر محاوره ای باز شده مسیر و نام فایل مختصات واقعی نقاط کنترلی را می توان وارد نمود.

:function Back_pushbutton_Callback(hObject, eventdata, handles)

با فعال کردن دکمه Back این تابع اجرا می‌شود. با اجرای این تابع امکان برگشت به مرحله قبلی امکان پذیر است.

:function Finish_pushbutton_Callback(hObject, eventdata, handles)

با فعال کردن دکمه Finish این تابع اجرا شده و مراحل بارگذاری قبلی تایید می‌گردد.

:function Cancel_pushbutton_Callback(hObject, eventdata, handles)

با فعال کردن دکمه Cancel این تابع اجرا شده و در هر مرحله می‌توان از ادامه روند بارگذاری فایل ها صرفنظر نمود.

ب) انتخاب روش بازسازی و هموارسازی

بخش انتخاب روش هموارسازی در محیط برنامه نویسی Matlab و با استفاده از جعبه ابزار Curve Fitting Toolbox تهیه شده است. این جعبه ابزار شامل مجموعه توابعی برای هموارسازی داده‌ها است. به عنوان مثال، در صورت انتخاب روش Moving average در هموارسازی داده ها، دستور زیر فراخوانده می‌شود.

smooth(C1_constant(:,i),C1_constant(:,i+1),span,'moving')

:function uipanel_SelectionChangeFcn(hObject, eventdata, handles)

با انتخاب یکی از روش های DLT یا MDLT توسط دکمه های رادیویی، یکی از توابع DLT یا MDLT انتخاب می‌شود تا در مراحل بعدی این توابع اجرا گردند.

:function

[xr,yr,zr]=DLT(f1,f2,ca1,ca2,A3d_coor,markernum1,markernum2,Frame_no)
با اجرای این تابع مختصات پیکسلی مارکرها، با استفاده از روش DLT که در فصل ۴ به آن اشاره شد، بازسازی می‌شود. در این تابع f1 و f2 مختصات پیکسلی نقاط کنترلی چارچوب کالیبراسیون، ca1 و ca2 مختصات پیکسلی مارکرها، A3d_coor مختصات واقعی نقاط کنترلی، markernum1 تعداد نقاط کنترلی، markernum2 تعداد مارکرها، مورد بررسی و Frame_no تعداد فریم‌های تصویربرداری شده، می‌باشد. برای به دست آوردن ضرایب کالیبراسیون دوربین‌ها و استفاده از روش شبه معکوس از دستور Pinv() استفاده شد.

:function

[xr,yr,zr]=MDLT(f1,f2,ca1,ca2,A3d_coor,markernum1,markernum2,Frame_no)
با اجرای این تابع مختصات پیکسلی مارکرها، با استفاده از روش MDLT که در فصل ۴ به آن اشاره شد، بازسازی می‌شود. در این تابع f1 و f2 مختصات پیکسلی نقاط کنترلی چارچوب کالیبراسیون، ca1 و ca2 مختصات پیکسلی مارکرها، A3d_coor مختصات واقعی نقاط کنترلی، markernum1 تعداد نقاط کنترلی، markernum2 تعداد مارکرها، مورد بررسی و Frame_no تعداد فریم‌های تصویربرداری شده، هستند. جزئیات مربوط به نحوه محاسبه ضرایب کالیبراسیون دوربین‌ها با استفاده از روش تکرار و حداقل مربعات، در فصل ۴ آمده است.

ج) دریافت خروجی‌ها

:function Markername_popupmenu_Callback(hObject, eventdata, handles)

با انتخاب نام مارکر موردنظر از منوی کشویی Marker name این تابع اجرا می‌شود.

:function listbox1_Callback(hObject, eventdata, handles)

در این منوی فهرست، کاربر دو متغیر از متغیر های x ، y ، z و t را برای رسم نمودار انتخاب می نماید. در صورت عدم انتخاب دو متغیر، پیغام خطایی دال بر عدم انتخاب دو متغیر در هنگام فعال نمودن دکمه Plot ظاهر خواهد شد.

`:function FPS_edit_Callback(hObject, eventdata, handles)`

با وارد نمودن سرعت تصویربرداری بر حسب فریم بر ثانیه این تابع اجرا شده و با توجه به تعداد فریم های تصویربرداری شده متغیر t را ایجاد می نماید.

`:function Calculate_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Calculate این تابع با توجه به نوع روش بازسازی و هموارسازی، توابع DLT یا MDLT را اجرا نموده و ضرایب کالیبراسیون دوربین ها را در خروجی نمایش می دهد.

`:function PLOT_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Plot دو متغیر انتخاب شده در منوی فهرست برای یک مارکر خاص بر حسب هم ترسیم می گردد.

`:function Hold_togglebutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Hold off این تابع اجرا شده و دکمه به حالت Hold on تغییر وضعیت می دهد.

`:function SaveData_pushbutton_Callback(hObject, eventdata, handles)`

با فعال کردن دکمه Save data این تابع اجرا می‌شود. یک کادر محاوره‌ای باز شده و از کاربر مسیر و نام فایل مختصات بازسازی شده تمامی مارکرها جهت ذخیره سازی، پرسیده می‌شود.

پیوست ۶: معادلات دینامیکی ربات CamCrane به روش لاگرانژ

در این معادلات k انرژی جنبشی و u انرژی پتانسیل می باشد و اندیسهای ۱ تا ۴ معرف چهار

بخش مختلف ربات (واگن، بازوی تلسکوپی، بازوی تلسکوپی متحرک و دوربین) می باشد.

$$k_1 = 1/2 * m_1 * \dot{x}_1^2$$

$$k_2 = 1/2 * m_2 * (\dot{x}_1 + d_1 * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - d_1 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2)^2 + 1/2 * m_2 * (d_1 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - d_1 * \sin(\theta_1) * \sin(\theta_2) * \dot{\theta}_2)^2 + 1/2 * I_{xx2} * \dot{\theta}_1^2 + 1/2 * I_{xx2} * \dot{\theta}_2^2$$

$$k_3 = 1/2 * m_3 * (\dot{x}_1 + x_2 * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - x_2 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1))^2 + 1/2 * m_3 * (x_2 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - x_2 * \sin(\theta_1) * \sin(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \cos(\theta_1))^2 + 1/2 * m_3 * (x_2 * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \sin(\theta_2))^2 + 1/2 * I_{xx3} * \dot{\theta}_1^2 + 1/2 * I_{xx3} * \dot{\theta}_2^2$$

$$k_4 = 1/2 * m_4 * (\dot{x}_1 + (d_2 + x_2) * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - (d_2 + x_2) * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1))^2 + 1/2 * m_4 * ((d_2 + x_2) * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - (d_2 + x_2) * \sin(\theta_1) * \sin(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \cos(\theta_1))^2 + 1/2 * m_4 * ((d_2 + x_2) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \sin(\theta_2))^2 + 1/2 * I_{xx4} * (\dot{\theta}_1 + \dot{\theta}_3 * \cos(\theta_2))^2 + 1/2 * I_{xx4} * (\dot{\theta}_2^2 + \dot{\theta}_4 * \cos(\theta_3))^2$$

$$k = k_1 + k_2 + k_3 + k_4 \quad u = m_2 * d_1 * g * \sin(\theta_1) + m_3 * x_2 * g * \sin(\theta_1) + m_4 * (d_2 + x_2) * g * \sin(\theta_1)$$

$$\text{diff}(k, \dot{x}_1) =$$

$$m_1 * \dot{x}_1 + m_2 * (\dot{x}_1 + d_1 * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - d_1 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2) + m_3 * (\dot{x}_1 + x_2 * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - x_2 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1)) + m_4 * (\dot{x}_1 + (d_2 + x_2) * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - (d_2 + x_2) * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1))$$

$$\text{diff}(k, \dot{x}_2) =$$

$$m_3 * (\dot{x}_1 + x_2 * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - x_2 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1)) * \cos(\theta_2) * \sin(\theta_1) + m_3 * (x_2 * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - x_2 * \sin(\theta_1) * \sin(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \cos(\theta_1)) * \cos(\theta_2) * \cos(\theta_1) + m_3 * (x_2 * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \sin(\theta_2)) * \sin(\theta_2) + m_4 * (\dot{x}_1 + (d_2 + x_2) * \cos(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - (d_2 + x_2) * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \sin(\theta_1)) * \cos(\theta_2) * \sin(\theta_1) + m_4 * ((d_2 + x_2) * \sin(\theta_1) * \cos(\theta_2) * \dot{\theta}_1 - (d_2 + x_2) * \sin(\theta_1) * \sin(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \cos(\theta_2) * \cos(\theta_1)) * \cos(\theta_2) * \cos(\theta_1) + m_4 * ((d_2 + x_2) * \cos(\theta_2) * \dot{\theta}_2 + \dot{x}_2 * \sin(\theta_2)) * \sin(\theta_2)$$

$$\text{diff}(k, \dot{\theta}_1) =$$

$$\begin{aligned}
& m2*(dx1+d1*cos(tet1)*cos(tet2)*dtet1- \\
& d1*sin(tet1)*cos(tet2)*dtet2)*d1*cos(tet1)*cos(tet2)+m2*(d1*sin(tet1)*cos(tet2)*dtet1- \\
& d1*sin(tet1)*sin(tet2)*dtet2)*d1*sin(tet1)*cos(tet2)+Ixx2*dtet1+m3*(dx1+x2*cos(tet1)*cos(tet2)*dtet1- \\
& x2*sin(tet1)*cos(tet2)*dtet2+dx2*cos(tet2)*sin(tet1))*x2*cos(tet1)*cos(tet2)+m3*(x2*sin(tet1)*cos(tet2)* \\
& dtet1- \\
& x2*sin(tet1)*sin(tet2)*dtet2+dx2*cos(tet2)*cos(tet1))*x2*sin(tet1)*cos(tet2)+Ixx3*dtet1+m4*(dx1+(d2+x \\
& 2)*cos(tet1)*cos(tet2)*dtet1- \\
& (d2+x2)*sin(tet1)*cos(tet2)*dtet2+dx2*cos(tet2)*sin(tet1))*(d2+x2)*cos(tet1)*cos(tet2)+m4*((d2+x2)*sin \\
& (tet1)*cos(tet2)*dtet1- \\
& (d2+x2)*sin(tet1)*sin(tet2)*dtet2+dx2*cos(tet2)*cos(tet1))*(d2+x2)*sin(tet1)*cos(tet2)+Ixx4*(dtet1+dtet3 \\
& *cos(tet2))
\end{aligned}$$

$$\begin{aligned}
& diff(k,dtet2) = \\
& -m2*(dx1+d1*cos(tet1)*cos(tet2)*dtet1-d1*sin(tet1)*cos(tet2)*dtet2)*d1*sin(tet1)*cos(tet2)- \\
& m2*(d1*sin(tet1)*cos(tet2)*dtet1-d1*sin(tet1)*sin(tet2)*dtet2)*d1*sin(tet1)*sin(tet2)+Ixx2*dtet2- \\
& m3*(dx1+x2*cos(tet1)*cos(tet2)*dtet1- \\
& x2*sin(tet1)*cos(tet2)*dtet2+dx2*cos(tet2)*sin(tet1))*x2*sin(tet1)*cos(tet2)- \\
& m3*(x2*sin(tet1)*cos(tet2)*dtet1- \\
& x2*sin(tet1)*sin(tet2)*dtet2+dx2*cos(tet2)*cos(tet1))*x2*sin(tet1)*sin(tet2)+m3*(x2*cos(tet2)*dtet2+dx2 \\
& *sin(tet2))*x2*cos(tet2)+Ixx3*dtet2-m4*(dx1+(d2+x2)*cos(tet1)*cos(tet2)*dtet1- \\
& (d2+x2)*sin(tet1)*cos(tet2)*dtet2+dx2*cos(tet2)*sin(tet1))*(d2+x2)*sin(tet1)*cos(tet2)- \\
& m4*((d2+x2)*sin(tet1)*cos(tet2)*dtet1- \\
& (d2+x2)*sin(tet1)*sin(tet2)*dtet2+dx2*cos(tet2)*cos(tet1))*(d2+x2)*sin(tet1)*sin(tet2)+m4*((d2+x2)*cos \\
& (tet2)*dtet2+dx2*sin(tet2))*(d2+x2)*cos(tet2)+2*Ixx4*(dtet2^2+dtet4*cos(tet3))*dtet2
\end{aligned}$$

$$diff(k,dtet3) = Ixx4*(dtet1+dtet3*cos(tet2))*cos(tet2)$$

$$diff(k,dtet4) = Ixx4*(dtet2^2+dtet4*cos(tet3))*cos(tet3)$$

$$diff(k,x1) = 0$$

$$\begin{aligned}
& diff(k,x2) = \\
& m3*(dx1+x2*cos(tet1)*cos(tet2)*dtet1- \\
& x2*sin(tet1)*cos(tet2)*dtet2+dx2*cos(tet2)*sin(tet1))*(cos(tet1)*cos(tet2)*dtet1- \\
& sin(tet1)*cos(tet2)*dtet2)+m3*(x2*sin(tet1)*cos(tet2)*dtet1- \\
& x2*sin(tet1)*sin(tet2)*dtet2+dx2*cos(tet2)*cos(tet1))*(sin(tet1)*cos(tet2)*dtet1- \\
& sin(tet1)*sin(tet2)*dtet2)+m3*(x2*cos(tet2)*dtet2+dx2*sin(tet2))*cos(tet2)*dtet2+m4*(dx1+(d2+x2)*cos(
\end{aligned}$$

$$dx2*\sin(tet2)*\cos(tet1))+m4*((d2+x2)*\cos(tet2)*dtet2+dx2*\sin(tet2))*(-\\(d2+x2)*\sin(tet2)*dtet2+dx2*\cos(tet2))-Ixx4*(dtet1+dtet3*\cos(tet2))*dtet3*\sin(tet2)$$

$$\text{diff}(k,tet3) = -Ixx4*(dtet2^2+dtet4*\cos(tet3))*dtet4*\sin(tet3)$$

$$\text{diff}(k,tet4) = 0$$

$$\text{diff}(u,x1) = 0$$

$$\text{diff}(u,x2) = m3*g*\sin(tet1)+m4*g*\sin(tet1)$$

$$\text{diff}(u,tet1) = m2*d1*g*\cos(tet1)+m3*x2*g*\cos(tet1)+m4*(d2+x2)*g*\cos(tet1)$$

$$\text{diff}(u,tet2) = 0$$

$$\text{diff}(u,tet3) = 0$$

$$\text{diff}(u,tet4) = 0$$



Sharif University of Technology
Mechanical Engineering Department

Design of a Robot with Real-time Motion Tracking Camera

Ahmad Colahi Damirchidaraci

Supervisor: Prof. H. Zohour

November 2005