

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده : برق و رباتیک

گروه : کنترل

طراحی و پیاده‌سازی یک مدل جامع کنترل رباتیک با استفاده از شبکه‌های پتری

دانشجو : سهیل آراسته‌فر

اساتید راهنما

دکتر حیدر طوسی‌ان شاندیز

دکتر علی اکبر پویان

استاد مشاور

دکتر هومن یونسی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

ماه و سال انتشار : تیر ۱۳۸۸

دانشگاه صنعتی شاهرود

دانشکده : برق و رباتیک

گروه : کنترل

پایان نامه کارشناسی ارشد آقای / خانم سهیل آراسته فر
تحت عنوان: طراحی و پیاده‌سازی یک مدل جامع کنترل رباتیک با استفاده از شبکه‌های پتری

در تاریخ توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد
مورد ارزیابی و با درجه مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی : دکتر هومن یونسی		نام و نام خانوادگی : دکتر حیدر طوسی‌ان شاندیز
	نام و نام خانوادگی : -----		نام و نام خانوادگی : دکتر علی اکبر پویان

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی :		نام و نام خانوادگی : دکتر محمد حداد ظریف
			نام و نام خانوادگی : دکتر حسن پور
			نام و نام خانوادگی :
			نام و نام خانوادگی :

تقدیم به همسر فداکار و مهربانم.

نمی‌دانم آیا روزی می‌رسد که بتوانم آروزهای تو را به حقیقت تبدیل نمایم،
نمی‌دانم آیا می‌توانم زحمات و دلسوزی‌های تو را درک کنم،
نمی‌دانم آیا فداکاری بی‌دریغ تو ...

امیدوارم این هدیه کوچک را از من بپذیری
سهیل آراسته‌فر

پدر و مادر عزیزم،
دوستتان می‌دارم. نمی‌دانم چه بگویم ...

با تشکر فراوان از استاد بزرگوارم آقای دکتر حیدر طوسی‌ان شاندیز،
بزرگواری، صبر، تلاش و دلسوزی‌های جناب عالی را هرگز فراموش نخواهم کرد. امیدوارم به اندازه
روزهای باقیمانده از عمرم به شما خدمت کنم، هر چند که نمی‌شود گوشه‌ای از زحمات شما را جبران
نمود. راستی و درستی شما را دوست می‌دارم ...

با تشکر فراوان از استاد بزرگوارم آقای دکتر علی اکبر پویان،
دلسوزی، کوشش، زحمات، مراقبت‌ها و نگرانی‌های شما را هرگز فراموش نخواهم کرد. شخصیت،
منش و برخورد شما همیشه در ذهن من باقی است. مصاحبت با شما کلاس درس زندگی و چراغ راه
آینده من می‌باشد. نوع نگرش شما به انسان، جامعه و زندگی را دوست می‌دارم ...

با بهترین آرزوها
سهیل آراسته‌فر

تأیید می‌نمایم که مطالب مندرج در این پایان‌نامه نتیجه تحقیقات خودم می‌باشد و در صورت استفاده از نتایج دیگران مرجع آن را ذکر نموده‌ام.

کلیه حقوق مادی مترتب از نتایج مطالعات، آزمایشات و نوآوری ناشی از تحقیق موضوع این پایان‌نامه متعلق به دانشگاه صنعتی شاهرود می‌باشد.

ماه و سال: تیر ۱۳۸۸

چکیده فارسی

در این پایان نامه یک مدل جامع تطبیقی کنترل رباتیک بر اساس شبکه‌های پتری، جهت طراحی مسیر حرکت ربات معرفی شده است. در این راستا یک بیان جدید از طراحی مسیر در رباتیک و یک کلاس جدید در شبکه‌های پتری تعریف شده‌اند. بیان پیشنهادی - بیان ماسک- در طراحی به سرعت بیان مفصلی و به دقت بیان کارترین می‌باشد. بیان ماسک منجر به کاهش شدید تعداد معادلات سینماتیک معکوس شده است که حرکت بلافاصله (آنلاین) ربات را نتیجه می‌دهد.

در روش پیشنهادی، ماسک‌ها به صورت تطبیقی بر اساس یک معیار شباهت ساخته می‌شوند. معیار شباهت نزدیکی مسیر تولید شده به مسیر مطلوب را بر اساس واریانس خطا اندازه‌گیری می‌نماید. استفاده از روش مذکور منجر به حرکت هماهنگ در مفاصل ربات و سرعت بالای حرکت ربات می‌شود. مزیت دیگر روش این است که مستقل از رزولوشن مسیر می‌باشد. با کاهش شدید معادلات سینماتیک معکوس که بسیار زمان‌بر و حافظه‌بر هستند، تجهیزات ارزان‌تری به منظور انجام محاسبات مورد نیاز قرار می‌گیرد که مزیت دیگر این روش می‌باشد.

در کلاس پیشنهادی از شبکه پتری، تعریف کاملی از زمان صورت گرفته است. براساس این تعریف، گذار جدیدی پیشنهاد شده که یکی از کاربردهای آن حذف تضاد در شبکه می‌باشد. در این راستا، مدل بدون تضادی که منجر به حل مسائل زمانی معرفی شده، می‌شود تعریف خواهد شد.

فهرست مقالات اتخاذ شده

- [1] H. Toossian Shandiz, S. Arastehfar, "Two links robot controlling based on Time Petri net," *IEEE 6th International Symposium on Communication systems, Networks, and Digital Signal Processing*, pp. 609-612, Jul, 2008.
- [2] H. Toossian Shandiz, A. Pouyan, S. Arastehfar, "An adaptive control Petri net model for robot path tracking," in *Proc. ELSEVIER Journal of Control Engineering Practice*, Dec, 2008.
- [3] A. Pouyan, H. Toossian Shandiz, S. Arastehfar, and H. Younessi, "Eliminating Redundancy and Singularity in Robot Path Planning based on Masking," in *Proc. ELSEVIER Journal of Expert Systems with Applications*, Mar, 2009.

فهرست مطالب

فصل اول: مقدمه

۱	-----	۱-۱ - مقدمه‌ای بر رباتیک
۴	-----	۱-۱-۱ - ربات چیست؟
۷	-----	۱-۱-۲ - تاریخچه رباتیک
۹	-----	۱-۱-۳ - درجه آزادی ربات
۱۱	-----	۱-۱-۴ - مفاصل ربات
۱۲	-----	۱-۱-۵ - محدوده کاری و تکین بودن
۱۲	-----	۲-۱ - مقدمه‌ای بر شبکه‌پتری

فصل دوم: سینماتیک ربات

۱۵	-----	۱-۲ - مقدمه
۱۵	-----	۲-۲ - موقعیت و جهت
۱۶	-----	۲-۲-۱ - نقطه در فضا
۱۶	-----	۲-۲-۲ - بردار در فضا
۱۸	-----	۲-۲-۳ - بیان یک دستگاه در مبدا مختصات دستگاه مرجع
۱۹	-----	۲-۲-۴ - بیان یک دستگاه در فضا
۲۱	-----	۲-۲-۵ - بیان یک جسم صلب در فضا
۲۱	-----	۲-۳ - ماتریس‌های تبدیل همگن
۲۲	-----	۲-۴ - تبدیل
۲۷	-----	۲-۵ - معکوس ماتریس‌های تبدیل

۲۸	-----	۶-۲- سینماتیک مستقیم و معکوس ربات
----	-------	-----------------------------------

فصل سوم: طراحی مسیر (معرفی روش پیشنهادی)

۳۳	-----	۱-۳- مقدمه
۳۴	-----	۲-۳- مسیر و تراژکتوری
۳۵	-----	۳-۳- بیان متغیر مفصلی و بیان کارتزین
۳۸	-----	۴-۳- طراحی مسیر
۴۲	-----	۵-۳- آموزش ربات
۴۳	-----	۶-۳- بیان ماسک
۴۶	-----	۱-۶-۳- بیان ماسک مربعی
۴۹	-----	۲-۶-۳- بیان ماسک مربعی چسبنده
۵۰	-----	۳-۶-۳- بیان ماسک مستطیلی در ماسک مربعی
۵۲	-----	۴-۶-۳- بیان ماسک تطبیقی
۵۸	-----	۷-۳- بیان ماسک در مقایسه با بیان مفصلی و کارتزین

فصل چهارم: پیاده‌سازی روش (شبکه پتری)

۵۹	-----	۱-۴- مقدمه
۵۹	-----	۲-۴- سیستم‌های گسسته رخداد
۶۰	-----	۳-۴- مدل‌سازی سیستم‌های گسسته رخداد
۶۱	-----	۱-۳-۴- مدل‌های زمانی
۶۱	-----	۲-۳-۴- مدل‌های منطقی

۶۲	-----	۳-۳-۴- مقایسه مدل های زمانی و منطقی
۶۲	-----	۳-۳-۴-۱- بیان زمانی مساله
۶۳	-----	۳-۳-۴-۲- بیان منطقی مساله
۶۳	-----	۴-۴- بیان ریاضی مدل
۶۵	-----	۴-۵- بیان زبانی
۶۹	-----	۴-۶- کنترل سیستم های گسسته رخداد
۶۹	-----	۴-۶-۱- کنترل پذیری و نظارت سیستم های گسسته رخداد
۷۵	-----	۴-۶-۲- رویت پذیری
۷۵	-----	۴-۷- زمان در سیستم
۷۶	-----	۴-۸- شبکه های پتری
۷۸	-----	۴-۸-۱- فعال و آتش شدن گذار
۷۹	-----	۴-۸-۲- شبکه های پتری زمانی
۸۲	-----	۴-۸-۳- گذارهای زمانی دست کاری شده
۸۳	-----	۴-۹- پیاده سازی پتری
۸۴	-----	۴-۹-۱- پیاده سازی فرآیند ساخت ماسک
۸۸	-----	۴-۹-۲- پیاده سازی کنترل حرکت ربات

فصل پنجم: نتایج

۹۰	-----	۵-۱- مقدمه
۹۰	-----	۵-۲- شبیه سازی و نتایج تجربی

منابع

٩٥

منابع و ماخذ

فهرست اشکال

فصل اول		
۱۲	مفصل چرخشی	الف ۱
۱۲	مفصل کشویی	ب ۱
۱۳	عناصر شبکه پتری، به ترتیب از سمت چپ: حالت، گذار، کمان و توکن	۲
فصل دوم		
۱۶	نقطه در فضا	۱
۱۹	دستگاهی در دستگاه مرجع	۲
۲۰	دستگاه در دستگاه مرجع	۳
۲۱	یک جسم در فضا	۴
۲۳	جابه‌جایی خالص در فضا	۵
۲۵	چرخش خالص در فضا	۶
۲۹	دستگاه نهایی ربات در فضا	۷
۳۰	بیان دناویت- هارتنبرگ از ربات	۸
فصل سوم		
۳۴	دنباله‌ای از نقاط در فضا که تشکیل یک مسیر می‌دهند	الف ۱
۳۴	دنباله‌ای از نقاط در فضا که تشکیل یک مسیر می‌دهند	ب ۱
۳۶	مسیر غیر قابل پیش‌بینی	۲
۳۶	مسیر کارتزین	۳
۳۸	موارد نامطلوب در بیان کارتزین	الف ۴
۳۸	موارد نامطلوب در بیان کارتزین	ب ۴
۳۹	بازوی کارتزین دو لینکی	۵
۳۹	موقعیت A و B	۶
۴۰	جابجایی با سرعت یکسان	۷
۴۱	حرکت هماهنگ	۸
۴۱	حرکت بر روی خط راست	۹
۴۴	چهار ماسک ممکن	۱۰
۴۵	مسیر مطلوب	۱۱

۴۶	طراحی مستطیلی	۱۲
۴۷	طراحی مربعی ۱	الف ۱۳
۴۷	طراحی مربعی ۲	ب ۱۳
۴۸	ماسک مربعی	۱۴
۴۹	حالت اشتباه ۱	الف ۱۵
۴۹	حالت اشتباه ۲	ب ۱۵
۵۰	ماسک مربعی چسبنده	۱۶
۵۱	ماسک مستطیلی چسبنده	۱۷
۵۳	۳ قطعه مجزا	الف ۱۸
۵۳	۱ قطعه مجزا	ب ۱۸
۵۳	حالت اشتباه ۱	الف ۱۹
۵۴	حالت اشتباه ۲	ب ۱۹
۵۵	ماسک تطبیقی	۲۰
۵۵	نقطه ممکن بعدی	۲۱
۵۶	خطا در ماسک	۲۲

فصل چهارم

۶۰	تراژکتوری	۱
۶۲	خط تولید نمونه	۲
۶۳	تراژکتوری خط تولید نمونه	۳
۶۶	گراف ماشین نمونه	۴
۶۷	گراف دو ماشین همزمان	۵
۶۸	ضرب دو ماشین نمونه	۶
۷۱	حلقه کنترلی	۷
۷۱	حلقه نظارتی	۸
۷۲	بازی موش و گربه	۹
۷۲	گراف بازی موش و گربه	۱۰
۷۳	مدل بازی موش و گربه	۱۱
۷۴	مدل کنترلی بازی موش و گربه	۱۲
۷۴	گراف کنترل بازی	۱۳
۷۴	حلقه نظارتی بازی	۱۴
۷۷	حالت	الف ۱۵
۷۷	گذار	ب ۱۵
۷۷	حالت دارای توکن	ج ۱۵
۷۷	کمان وزن دار	الف ۱۶

۷۷	کمان موازی هموزن	ب ۱۶
۸۰	گذار اتمیک قبل از آتش	الف ۱۷
۸۰	گذار اتمیک بعد از آتش	ب ۱۷
۸۱	گذار سه‌فاز قبل از آتش	الف ۱۸
۸۱	گذار سه‌فاز بعد از آتش	ب ۱۸
۸۲	گذار زمانی دستکاری شده	۱۹
۸۳	کلاس جدید معرفی شده	۲۰
۸۵	چهار ماسک ممکن	۲۱
۸۶	پیاده‌سازی پتری روش تطبیقی ساخت ماسک	۲۲
۸۷	گذارهای متناظر با مسیر	۲۳
۸۸	پیاده‌سازی پتری کنترل موتورها	۲۴

فصل پنجم

۹۰	مسیر مطلوب	الف ۱
۹۰	کوتاه‌ترین مسیر	ب ۱
۹۰	مسیر پیموده شده	ج ۱
۹۱	مسیر مطلوب	الف ۲
۹۱	مسیر پیموده شده	ب ۲
۹۱	مسیر پیموده شده	۳
۹۲	مسیر مطلوب ۲۲۹ پیکسل	الف ۴
۹۲	واریانس = $0/15$ بزرگترین واریانس محاسبه شده = $0/11$	ب ۴
۹۲	واریانس = $0/078$ بزرگترین واریانس محاسبه شده = $0/0757$	ج ۴
۹۲	واریانس = $0/055$ بزرگترین واریانس محاسبه شده = $0/053$	د ۴
۹۲	واریانس = $0/0046$ بزرگترین واریانس محاسبه شده = $0/0045$	ه ۴
۹۳	مسیر مطلوب ۹۶۶ پیکسل	الف ۵
۹۳	واریانس = $0/08$ بزرگترین واریانس محاسبه شده = $0/0728$	ب ۵
۹۳	واریانس = $0/005$ بزرگترین واریانس محاسبه شده = $0/0046$	ج ۵
۹۴	مسیر مطلوب ۱۱۵۳ پیکسل	الف ۶
۹۴	واریانس = $0/05$ بزرگترین واریانس محاسبه شده = $0/03$	ب ۶
۹۴	واریانس = $0/005$ بزرگترین واریانس محاسبه شده = $0/0044$	ج ۶

فهرست جداول

	فصل سوم	
۴۲	۱	متغیرهای مفصلی
	فصل چهارم	
۸۷	۱	حالات در پیاده‌سازی ماسک تطبیقی
۸۹	۲	حالات در پیاده‌سازی حلقه کنترلی موتور

فهرست معادلات

فصل دوم

۱۶	-----	۱
۱۷	-----	۲
۱۷	-----	۳
۱۷	-----	۴
۱۷	-----	۵
۱۷	-----	۶
۱۷	-----	۷
۱۸	-----	۸
۱۸	-----	۹
۱۹	-----	۱۰
۱۹	-----	۱۱
۲۰	-----	۱۲
۲۰	-----	۱۳
۲۰	-----	۱۴
۲۱	-----	۱۵
۲۱	-----	۱۶
۲۲	-----	۱۷
۲۲	-----	۱۸
۲۳	-----	۱۹
۲۴	-----	۲۰
۲۴	-----	۲۱
۲۴	-----	۲۲
۲۵	-----	۲۳
۲۶	-----	۲۴
۲۶	-----	۲۵
۲۶	-----	۲۶
۲۷	-----	۲۷
۲۷	-----	۲۸
۲۸	-----	۲۹

۲۸	-----	۳۰
۳۲	-----	۳۱
۳۲	-----	۳۲

فصل سوم

۵۷	-----	۱
----	-------	---

فصل چهارم

۶۴	-----	۱
۶۴	-----	۲
۶۵	-----	۳
۶۵	-----	۴
۶۶	-----	۵
۶۶	-----	۶
۶۸	-----	۷
۷۰	-----	۸
۷۰	-----	۹
۷۲	-----	۱۰
۷۵	-----	۱۱
۷۵	-----	۱۲
۸۵	-----	۱۳



دانشکده برق و رباتیک

پایان نامه کارشناسی ارشد

طراحی و پیاده‌سازی یک مدل جامع کنترل رباتیک با استفاده از شبکه‌های پتری

سهیل آراسته‌فر

اساتید راهنما :

دکتر حیدر طوسی‌ان شاندیز

دکتر علی اکبر پویان

ماه و سال انتشار : تیر ۱۳۸۸



Shahrood University of Technology

Electrical Engineering Faculty

**Designing and Simulating a Robust Petri-Net Based Robotic
Control Model**

Soheil Arastehfar

Supervisors:

Dr. Heydar Toossian Shandiz

Dr. Ali Akbar Pouyan

Date: Jun 2009

فصل اول

۱-۱- مقدمه‌ای بر رباتیک

انسان همیشه به دنبال ساخت وسایل گوناگون به منظور فراهم آوردن راحتی و آسایش است. ربات‌ها از زمانی که انسان شروع به ساخت وسایل گوناگون نمود، همیشه در ذهن انسان بودند. تاکنون ربات‌های زیادی به منظور انجام حرکات و اعمال و رفتار انسان ساخته شده‌اند و در این پایان‌نامه ما به طور خاص به ربات‌های صنعتی^۱ که به منظور انجام حرکاتی مانند دست انسان^۲ ساخته می‌شوند می‌پردازیم. بر این اساس، ابتدا سعی می‌کنیم مطالبی که خواننده را با موضوع آشنا می‌کند بیان می‌کنیم. به این منظور، سینماتیک ربات^۳، برای آنالیز موقعیت ربات را در فصل ۲ بیان می‌کنیم. بعد از آن، بیان مفصلی^۴ و کارتیزین^۵ جهت بیان جابجایی ربات و همچنین بیان جدیدی با استفاده از ماسک‌ها را در فصل ۳ بنیانگذاری می‌نماییم. سپس در فصل چهارم، سیستم‌های گسسته رخداد^۶ را مورد بحث قرار می‌دهیم و در این راستا، حرکت ربات را از دید یک سیستم گسسته رخداد مورد بررسی قرار می‌دهیم. در فصل ۴، شبکه‌های پتری^۷ را به عنوان یک ابزار جهت بیان گرافیکی و نظارت یک سیستم گسسته رخداد معرفی می‌کنیم. سپس، بیان ماسک برای حرکت ربات و کنترل موقعیت ربات را براساس شبکه‌های پتری پیاده‌سازی می‌کنیم. به این منظور، کلاسی جدید از شبکه‌های پتری را در فصل ۴ معرفی خواهیم نمود. در این پایان‌نامه، با ابداع دو موضوع جدید در بحث رباتیک و شبکه‌های پتری، بیان قدرتمند از طراحی مسیر^۸ و کنترل موقعیت ربات را پایه‌ریزی و پیاده‌سازی خواهیم کرد. در آخر، در فصل ۵، شبیه‌سازی و

۱. Industrial Robots

۲. Arm

۳. Robot Kinematics

۴. Joint Description

۵. Cartesian Description

۶. Discrete Event Systems

۷. Petri Nets

۸. Path Planning

نتایج تجربی روش پیشنهادی نشان داده خواهد شد.

ربات‌ها با استفاده گسترده در خطوط تولید کارخانجات، ابزار قدرتمندی در صنعت امروز به شمار می‌آیند. ربات‌ها قادرند عملیات بسیاری را انجام دهند و وظایفشان را با دقت و تکرارپذیری^۱ بالا انجام دهند که این مهم در صنعت بسیار مطلوب می‌باشد. همچنین ربات‌ها به امکانات رفاهی، استراحت و امنیت کاری که نیروهای انسانی به آن نیازمندند، نیازی ندارند. در نتیجه، استفاده از ربات‌ها در صنعت امروز بسیار گسترده است و به سرعت نیز در حال پیشرفت می‌باشد [۱]. کمپانی‌های زیادی در سطح جهان از جمله Adept Robotics, Staubli Robotics, Kuka Robotics, Fanuc Robotics به ساخت ربات‌های صنعتی می‌پردازند. اگرچه در حال حاضر هزاران ربات در صنعت در حال انجام عملیات گوناگون می‌باشند، با این وجود آن‌ها نمی‌توانند کاملاً جایگزین نیروی انسانی شوند. با این حال، ربات‌ها هر جا که مفید باشند مورد استفاده قرار می‌گیرند و استفاده از آن‌ها به سرعت در حال پیشرفت است. برنامه‌ریزی و توصیف سطح بالای^۲ ربات‌های صنعتی و ربات‌های متحرک^۳ در کاربردهای پیچیده رباتیکی امروزی بسیار گسترده شده است [۲]-[۶]. در این پایان‌نامه، ربات یک معماری هیبرید ناهمگن^۴ است. این معماری قادر است اعمال active و proactive را در قالب یک ساختار بگنجانند. خصوصاً که، ما ربات‌ها را در دو لایه متمایز تصمیم‌گیری^۵ و عملیاتی^۶، معماری (ساختاربندی) می‌نماییم. لایه تصمیم‌گیر یک توصیف سطح بالا از محیط را برای انتخاب عملیات مناسب برقرار می‌کند. در حالیکه، لایه عملیاتی یک توصیف سطح پایین را ارائه می‌دهد که جهت ارزیابی شرایط و عملیاتی کردن، اعمال رفتاری پایه‌ای به کار می‌رود. معماری هیبرید می‌تواند بر این اساس که اطلاعات چگونه بیان می‌شوند، به کلاس‌های متعددی تفکیک

۱. Repeatability

۲. High Level Specification and Programming

۳. Mobile Robot

۴. Heterogeneous Hybrid Architecture

۵. Deliberative Layer

۶. Operational Layer

شود. این ساختارها می‌توانند همگن^۱ باشند، اگر اطلاعات در هر دو سطح تصمیم‌گیری و عملیاتی به یک صورت بیان شوند. در غیر اینصورت این ساختارها ناهمگن^۲ بوده و اطلاعات در این دو سطح به دو صورت متمایز بیان می‌شوند [۲]-[۹]. هدف ما در این پایان‌نامه معرفی یک روش توصیفی جدید جهت توصیف رفتارهای حرکتی بازوها (ربات‌های صنعتی) و پیاده‌سازی این روش روی ربات‌های صنعتی می‌باشد. روش پیشنهادی ما بر اساس شبکه‌های پتری [۱۰] که یک توصیف سطح بالای گرافیکی و یک ابزار مدل‌سازی برای سیستم‌های دینامیک گسسته رخداد می‌باشد توصیف و پیاده‌سازی شده است. شبکه‌های پتری به طور گسترده‌ای در مدل کردن سیستم‌های دینامیک^۳ مورد استفاده قرار گرفته‌اند [۳]، [۵]، [۱۱]-[۱۸]. این شبکه‌ها از میان ابزارهای دیگر کاندیدای مناسبی جهت آنالیز کیفی^۴ عملکرد وظایف ربات‌ها^۵ می‌باشند.

همانطور که قبلاً توضیح داده شد، ربات‌ها سیستم‌هایی جهت به انجام رسانیدن یک وظیفه یا یک عملیات هستند. به این منظور، ربات می‌بایست یک مسیر از پیش تعیین‌شده‌ای را به منظور انجام عملیات از پیش تعیین‌شده بپیماید. تاکنون، روش‌های زیادی جهت پیمایش یک مسیر بیان شده‌اند، که هر کدام دارای مزایا و معایبی هستند. پیمایش مسیر در صنعت رباتیک به صورت گسترده‌ای مورد توجه پژوهشگران قرار گرفته است [۱۹]-[۳۲]. در این پایان‌نامه، ما یک مدل تطبیقی^۶ بر اساس شبکه‌های پتری زمانی^۷ [۱۶] را جهت پیمایش مسیر توسط ربات، ارائه خواهیم کرد. نتایج تجربی بدست آمده نشان داده‌اند، روش پیشنهادی دارای مزایای بسیاری نسبت به روش‌های معرفی شده قبلی توسط دیگر

۱. Homogeneous

۲. Heterogeneous

۳. Dynamic Systems

۴. Qualitative Performance Analysis

۵. Robot Task

۶. Adaptive

۷. Time Petri Nets

پژوهشگران می‌باشد. یکی از اهداف روش‌های پیمایش مسیر، پیمایش سریع مسیر می‌باشد. در این راستا، روش پیشنهادی ما به اندازه روش‌های پر سرعتی که تاکنون معرفی شده‌اند سریع می‌باشد. مزیت دیگر و در واقع یکی از مهمترین مزایای روش پیشنهادی ما که در روش‌های دیگر دیده نمی‌شود، این است که پیمایش مسیر در این روش مستقل از رزولوشن مسیر می‌باشد و این روش می‌تواند مسیر با هر رزولوشن مطلوبی را به درستی پیماید. بر اساس روش پیشنهادی، ربات می‌تواند مسیر با هر شکل دلخواه را به صورت زنده طی نماید. این مهم با کاهش فوق‌العاده محاسبات سینماتیکی معکوس^۱ که بینهایت زمان بر^۲ و حافظه بر^۳ می‌باشند محقق می‌شود [۳۳]-[۳۹]. این روش همچنین می‌تواند در مباحث کنترلی رباتیک از جمله طراحی تراژکتوری^۴، کنترل سرعت و دینامیک ربات‌ها به کار برده شود. یکی از مهمترین مزایای این روش این است که محاسبات اضافی سینماتیکی و نقاط تکین^۵ [۱]، [۲۹]، [۳۴]، [۳۵] بر اساس این روش خود به خود کنار گذاشته می‌شوند و روش هیچ‌گاه، هیچ محاسبه‌ای را برای یک نقطه تکین انجام نمی‌دهد. روش پیشنهادی به گونه‌ای طرح‌ریزی شده است که می‌تواند بر اساس پارامترهای خطا به منظور استفاده‌های گوناگون تنظیم شود. ما در طراحی خود، از واریانس خطا جهت تنظیم دقت روش پیشنهادی استفاده کرده‌ایم که معیار^۶ تطبیق در روش کنترل تطبیقی^۷ ما می‌باشد.

۱-۱-۱- ربات چیست؟

یک بازوی صنعتی^۸ بسیار مشابه یک جرثقیل می‌باشد. یک جرثقیل از لینک‌هایی^۹ که مانند یک

۱. Inverse Kinematics

۲. Time Consume

۳. Space Consume

۴. Trajectory Planning

۵. Singularity

۶. Criterion

۷. Adaptive Control

۸. Industrial Robot

۹. Link

زنجیر پشت سر هم توسط مفاصل^۱ به هم متصل شده‌اند تشکیل شده است که هر مفصل توسط یک محرک^۲، حرکت کرده می‌شود. یک بازو نیز، مانند یک جرثقیل از لینک‌هایی که توسط مفاصل پشت سر هم به هم متصل شده‌اند، تشکیل شده است. همچنین هم بازو و هم جرثقیل می‌تواند بار مشخص شده‌ای را حمل کنند، و هر کدام توسط یک کنترلر مرکزی که محرک‌ها را کنترل می‌کند، کنترل می‌شوند. جرثقیل توسط شخصی که آن را به کار می‌برد، کنترل می‌شود، در حالیکه یک بازو به وسیله کامپیوتر یا ابزار مشابه آن کنترل می‌شود. این تفاوت یک تفاوت اساسی میان وسایلی که بازو نامیده می‌شوند، می‌باشد. این تفاوت مرز میان یک بازوی ساده و یک ربات را مشخص می‌نماید.

مطلوب است که ربات‌ها به گونه‌ای طراحی شوند که قادر باشند چندین وظیفه و عمل را تنها با یک تغییر ساده در برنامه انجام دهند. حرکات یک ربات توسط یک کنترلر تحت نظارت یک کامپیوتر کنترل می‌شود که این کامپیوتر می‌تواند در حال اجرای برنامه‌های متعددی از کنترل ربات باشد. بنابراین تغییر در قسمتی از برنامه می‌تواند مابقی قسمت‌های برنامه را تحت‌الشعاع قرار داده و کاربر را به تغییر در قسمت دیگر برنامه وادار نماید.

اکنون به بیان چند تعریف ارائه شده از ربات می‌پردازیم:

۱. ربات‌ها ماشین‌آلاتی هستند که برنامه‌ریزی می‌شوند تا اجسام را جابجا کنند یا با استفاده از

ابزارهایی خاص عملیات خاصی را انجام دهند در نتیجه می‌توان ربات را به صورت زیر تعریف

نمود:

"ماشین کنترل‌پذیر اتوماتیک که قابلیت برنامه‌ریزی برای انجام کارهای گوناگون با قابلیت

تکرارپذیری و خستگی‌ناپذیری را دارد."

۲. انجمن ربات امریکا (RIA)^۱ تعریفی از ربات به صورت زیر ارائه می‌دهد:

"ربات، بازوی چندمنظوره قابل برنامه‌ریزی است که برای جابجایی مواد، قطعات، ابزار یا دستگاه-های مخصوص طراحی شده است و به وسیله حرکتهای برنامه‌ریزی شده، کارهای مختلف را به اجرا درمی‌آورد."

۳. تعریف نه‌چندان خوب دیگری از ربات به صورت زیر آمده است:

"ربات‌ها تک بازو، کور، بدون فهم، کر و لال، بدون احساس و درک نسبت به کاری که انجام می‌دهند، هستند."

اما برخلاف خیلی از ماشین‌ها، ربات‌ها به خاطر شباهتشان به انسان به راحتی برای انجام دادن یک کار قابل تصور هستند. کشورهای مختلف استانداردهای متفاوتی نسبت به وسیله‌ای که آن را ربات می‌نامند، ارائه کرده‌اند. به عنوان مثال JIRA^۲ در ژاپن کلاس‌های متعددی از ربات را به صورت زیر بیان می‌کند:

- کلاس ۱: Manual-Handling Devices: یک وسیله با چندین درجه آزادی که توسط یک کاربر حرکت داده می‌شود.
- کلاس ۲: Fixed-Sequence Robot: یک وسیله که یک وظیفه یا عمل را بر طبق یک برنامه از پیش تعیین‌شده تغییرناپذیر انجام می‌دهد.
- کلاس ۳: Variable-Sequence Robot: از لحاظ تعریف مانند کلاس ۲ می‌باشد، اما با این تفاوت که تغییر در آن آسان می‌باشد.
- کلاس ۴: Playback Robot: یک کاربر وظایف مطلوب را به صورت دستی با حرکت دادن ربات

۱. Robotic Institute of America

۲. Japanese Industrial Robot Association

- انجام می‌دهد. در حین جابجایی‌ها ربات حرکات را به منظور اجرای دوباره ذخیره می‌کند.
- بنابراین این کلاس از ربات‌ها به صورت دستی آموزش می‌بینند تا یک عمل را انجام دهند.
- کلاس ۵: Numerical Control Robot: در این کلاس ربات بر طبق برنامه‌ای که به آن داده شده است، جابجایی‌های مطلوب را انجام می‌دهد تا این که به صورت دستی آموزش دیده باشد.
 - کلاس ۶: Intelligent Robot: یک ربات که مجهز به ابزاری برای شناخت محیط است و توانایی انجام یک وظیفه یا عمل را با وجود تغییر در شرایط محیطی دارد.
- در ژاپن هر وسیله‌ای که در یکی از این کلاس‌ها قرار بگیرد، ربات نامیده می‌شود در حالیکه در امریکا، انجمن ربات امریکا (RIA) تنها کلاس‌های ۳ تا ۶ را به عنوان ربات معرفی می‌نماید. و این در حالی است که در فرانسه، AFR^۱ ربات‌ها را به صورت زیر کلاس‌بندی می‌نماید:
- Type A: ابزارهای دستی^۲ که توسط کاربر کنترل می‌شوند.
 - Type B: ابزارهای اتوماتیک با سیکل کاری از پیش تعیین شده.
 - Type C: ابزارهایی که قابل برنامه‌ریزی هستند و به صورت سرو^۳ به منظور ایجاد یک مسیر پیوسته یا نقطه به نقطه کنترل می‌شوند.
 - Type D: مشابه Type C می‌باشند، با این تفاوت که دارای توانایی کسب اطلاعات از محیط می‌باشند.

۲-۱-۱- تاریخچه رباتیک

در این قسمت تنها به ذکر خلاصه‌ای از تاریخ علم رباتیک می‌پردازیم و رویدادهایی که منجر به

۱. Association Française de Robotique

۲. Manual Devices

۳. Servo

تغییرات اساسی در روند عمل رباتیک شده‌اند را بیان می‌کنیم:

- ۱۹۲۲ کارل کاپک^۱ نمایش‌نامه‌نویس اهل چک در این سال نمایش‌نامه‌ای با نام Rossum's Universal Robots نوشت و در این نمایش‌نامه از کلمه "Rabota" استفاده کرد که به معنی کارگر می‌باشد [۴۰].
- ۱۹۲۶ جرج دول^۲ وسیله‌ای ساخت که می‌توانست آموزش ببیند و عملیات آموزش دیده را تکرار کند. همچنین در این سال اکرت^۳ و ماچلی^۴ کامپیوتر ENIAC را در دانشگاه پنسیلوانیا^۵ ساختند.
- ۱۹۵۲ اولین ماشین NC در MIT ساخته شد.
- ۱۹۵۴ جرج دول اولین ربات برنامه‌پذیر را ارائه داد.
- ۱۹۵۵ دناویت^۶ و هارتنبرگ^۷ ماتریس تبدیل همگن^۸ را ارائه دادند [۴۱].
- ۱۹۶۱ اختراع جرج دول به نام "Programmed Article Transfer" در امریکا ثبت شد که پایه‌ای بر ربات‌های Unimate بود.
- ۱۹۶۲ اولین ربات صنعتی به نمایش گذاشته شد و GM اولین رباتش را در تولید نصب کرد.
- ۱۹۶۷ Unimate، ربات MakII را معرفی کرد. اولین ربات به ژاپن برای نقاشی با استفاده از اسپری صادر شد.

۱. Karel Capek

۲. George Devol

۳. Eckert

۴. Mauchly

۵. University of Pennsylvania

۶. Denavit

۷. Hartenberg

۸. Homogeneous Transformation Matrices

- ۱۹۶۸ در استنفورد SRI^۱ یک ربات هوشمند به نام Shakey ساخته شد.
- ۱۹۷۲ IBM عملیات ساخت یک ربات کارتیزین^۲ برای استفاده در خط تولیدش را شروع کرد و بالاخره IBM 7565 را به منظور فروش توسعه داد.
- ۱۹۷۳ سینسیناتی میلاکرن^۳ ربات T3 را معرفی کرد که در صنعت بسیار مورد علاقه قرار گرفت.
- ۱۹۷۸ اولین ربات PUMA توسط Unimation به GM عرضه شد.
- ۱۹۸۲ GM و Fanuc ژاپن یک قرارداد برای ساخت ربات‌های GMFanuc امضا کردند: صنایع وستینگ هاوس^۴ Unimation را خریداری کرد و بعداً آن را به Staubli در سوئیس فروخت.
- ۱۹۸۳ رباتیک به یک موضوع محبوب در صنعت و دانشگاه تبدیل شد و افراد بسیاری در کشورهای مختلف به یادگیری علم رباتیک پرداختند.
- ۱۹۹۰ سینسیناتی میلاکرن به ABB در سوئیس واگذار شد. کارخانجات کوچک ربات‌سازی از بازار خارج شدند و تنها تعداد محدودی از کارخانجات بزرگ ربات‌سازی در بازار باقی ماندند.

۳-۱-۱- درجه آزادی^۵ ربات

یک جسم صلب^۶ و یک نقطه بر روی آن را در نظر بگیرید. برای اینکه جسم در یک موقعیت دلخواه در فضا قرار بگیرد، کافی است موقعیت نقطه در نظر گرفته شده تعریف شود. موقعیت یک نقطه در فضا با سه پارامتر بیان می‌شود. در این حالت اگرچه جسم در مکانی دلخواه در فضا قرار می‌گیرد، اما

۱. Stanford Research Institute

۲. Cartesian Robot

۳. Cincinnati Milacron

۴. Westinghouse

۵. Degrees of Freedom (DOF)

۶. Rigid Body

جسم در هر حالت چرخشی می‌تواند در مکان مورد نظر قرار گیرد. در واقع بینهایت حالت وجود دارد که جسم می‌تواند در یک نقطه قرار داده شود. در نتیجه، به منظور توصیف یک جسم در فضا، می‌بایست هم موقعیت مکانی نقطه در نظر گرفته شده بر روی جسم و هم موقعیت چرخشی جسم نسبت به این نقطه مشخص شود. به بیان دیگر ۶ پارامتر (۳ پارامتر جهت توصیف موقعیت نقطه در نظر گرفته شده بر روی جسم و ۳ پارامتر جهت توصیف چرخش جسم نسبت به این نقطه) نیاز است تا یک جسم در فضا به صورت کامل توصیف شود. بنابراین، می‌بایست ۶ درجه آزادی وجود داشته باشد تا بتوان یک جسم را به صورت کامل در فضا قرار داد، در غیر اینصورت توانایی ربات در قرار دادن جسم در فضا کاسته می‌شود.

هر مفصل بسته به نوع، تعدادی درجه آزادی به ربات می‌دهد. مفاصل می‌توانند کشویی، چرخشی، گوی کاسه‌ای و یا ... باشند که هر کدام دارای درجه آزادی مشخصی هستند. یک دستگاه کارترین سه بعدی را تصور کنید، حرکت در راستای هر یک از محورها نشانه یک درجه آزادی است و اگر ما اجازه حرکت در هر سه راستا را داشته باشیم در این صورت می‌توانیم در داخل یک مکعب در هر جابجایی دلخواهی قرار بگیریم. حال فرض کنید، اجازه نداشته باشیم که در راستای محور Z حرکت کنیم. در این صورت ما تنها می‌توانیم در صفحه‌ای که محور Z برای ما تعیین می‌کند جابجا شویم. در این جا ما یک درجه آزادی از حرکت را از دست داده‌ایم. جابجایی در راستای هر یک از محورها یک درجه آزادی به یک ربات می‌دهد و رباتی که در راستای هر سه محور اجازه حرکت دارد، دارای سه درجه آزادی است. سه درجه دیگر آزادی، چرخش حول هر یک از محورها است. که بیانگر آزادی در قرارگرفتن در جهتی خاص می‌باشد. این ۶ درجه آزادی ما را قادر می‌سازد تا در هر جابجایی و جهت دلخواهی در فضا قرار بگیریم. در واقع این ۶ درجه ربات را قادر می‌سازد تا فعالیتی مانند دست انسان (از کتف تا انتهای مچ) انجام دهد. ربات‌ها عموماً دارای ۵ یا ۶ درجه آزادی هستند که ۳ تا از آن‌ها جهت قرار دادن نقطه نهایی بازو در مکان مناسب استفاده می‌شوند و ۲ یا ۳ تای دیگر مربوط به موقعیت چرخشی نقطه نهایی

بازو می‌باشند. ۶ درجه آزادی برای یک بازو کافی است تا بازو بتواند جسم را در هر موقعیت مکانی و چرخشی در فضا قرار دهد.

درجه آزادی یک مکانیزم توسط رابطه کوتزباخ^۱ بدست می‌آید و با تعمیم این رابطه به صورت زیر می‌توان درجه آزادی یک ربات را محاسبه نمود:

$$m = 6(n - 1) - 5j_1 - 4j_2 - 3j_3 - 2j_4 - j_5 \quad (۱ - ۱)$$

حرف m درجه آزادی، n تعداد لینک‌های ربات و j_1, j_2, j_3, j_4, j_5 تعداد مفاصلی هستند که به ترتیب دارای ۱، ۲، ۳، ۴ و ۵ درجه آزادی می‌باشند.

۴-۱-۱- مفاصل ربات

اگرچه انواع زیادی مفصل وجود دارد از جمله مفاصل خطی، چرخشی، گوی کاسه‌ای و ... با این حال استفاده از مفاصل کشویی^۲ و چرخشی^۳ در رباتیک مرسوم است. استفاده از مفاصل دیگر همچون مفصل گوی کاسه‌ای که درجه آزادی بیشتر از یک را برای ربات تامین می‌کنند، به دلیل کنترل پیچیده و سخت در رباتیک مرسوم نمی‌باشند. هر مفصل چرخشی و کشویی یک درجه آزادی به ربات اضافه می‌کند، که می‌توانند توسط محرکه‌های هیدرولیکی^۴، بادی^۵ و یا الکتریکی^۶ حرکت داده شوند. مفاصل چرخشی و کشویی به ترتیب در شکل‌های ۱-۱ الف و ۱-۱ ب نشان داده شده‌اند.

۱. Kutzbach

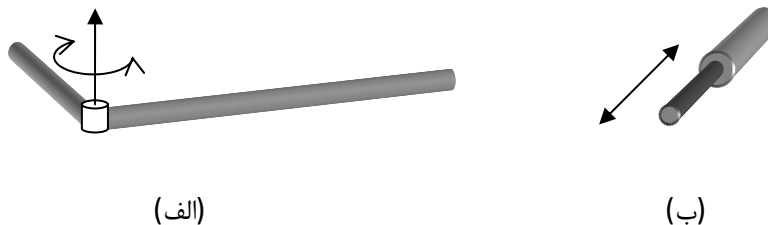
۲. Prismatic (P)

۳. Revolute (R)

۴. Hydraulic Actuators

۵. Pneumatic Actuators

۶. Electric Actuators



شکل ۱-۱ مفاصل ربات (الف) مفصل چرخشی (ب) مفصل کشویی.

۵-۱-۱- محدود کاری^۱ و تکین بودن

محدوده کاری و نقاط تکین محدودیت‌های حرکتی ربات را نمایش می‌دهند که وابسته به شکل و پیکربندی ربات^۲ می‌باشند. محدوده کاری مجموعه‌ای از نقاط را در فضا نمایش می‌دهد که ربات می‌تواند به آن‌ها دسترسی کامل داشته باشد [۴۲]. دسترسی کامل به این معنی است که، ربات می‌تواند به این موقعیت‌ها تحت هر دورانی دست یابد. نقاط تکین نقاطی هستند که یا ربات نمی‌تواند به آن‌ها دسترسی داشته باشد و یا دسترسی ربات به آن‌ها یک دسترسی کامل نمی‌باشد. برای اطلاعات بیشتر به [۱] رجوع کنید.

۳-۱- مقدمه‌ای بر شبکه پتری

شبکه‌های پتری [۱۰] ابزار قدرتمندی در مدل‌سازی سیستم‌ها می‌باشد. در واقع این شبکه‌ها می‌توانند سیستم‌های دینامیک را از نگاه سیستم‌های گسسته‌رخداد مدل کنند [۳]، [۵]، [۱۱] - [۱۸]. در صنعت نقش نظارتی این شبکه استفاده گسترده دارد و به دلیل گرافیکی بودن این ابزار بسیار کاربر پسند

۱. Workspace

۲. Robot Configuration

می‌باشد.

در شبکه‌های پتری چهار عنصر اصلی وجود دارد که گذار^۱، حالت^۲، کمان^۳ و توکن^۴ نامیده می‌شوند. شکل ظاهری این چهار عنصر در شکل ۱-۲ نمایش داده شده است.



شکل ۱-۲ عناصر شبکه پتری، به ترتیب از سمت چپ: حالت، گذار، کمان و توکن

عملکرد این چهار عنصر توسط مثال زیر به راحتی بیان می‌شود. به این منظور خط تولید یک کارخانه را در نظر بگیرید. حالت‌ها مانند یک بافر^۵ در این خط تولید می‌باشند که قطعات در آن قرار می‌گیرد و گذارها مانند ماشین‌آلاتی هستند که عملیاتی را بر روی قطعات انجام می‌دهند. کمان‌ها مانند غلتک‌ها و لیفتراک‌ها می‌باشند که قطعات را به مکان‌های موردنظر جابجا می‌کنند و قطعات همان توکن‌ها هستند.

عملیات در این شبکه‌ها به این صورت است که توکن‌ها وارد شبکه شده و گذارها بنابر پیکربندی شبکه آتش^۶ می‌شوند. در اثر آتش شدن گذارها، توکن‌ها نابود شده و ساخته می‌شوند. گذارها زمانی می‌توانند آتش شوند که تمامی شرایط ورودی آن‌ها برآورده شوند یعنی گذار فعال^۷ شده است. آتش شدن گذار توسط کاربر تعریف می‌شود.

۱. Transition

۲. Place

۳. Arc

۴. Token

۵. Buffer

۶. Fire

۷. Enable

۸. Carl Adam Petri

مفهوم شبکه پتری اولین بار در پایان‌نامه کارل آدام پتری^۱ [۴۳] در سال ۱۹۶۲ در دانشکده ریاضیات و فیزیک دانشگاه دارمتاد^۱ در آلمان غربی مطرح شد [۴۳]-[۴۴]. سپس این پایان‌نامه مورد توجه پژوهشگران در سرتاسر جهان قرار گرفت و توسعه داده شد که در منابع [۴۵]-[۴۸] از سال ۱۹۷۰ تا ۱۹۷۵ آورده شده‌اند. در این سال‌ها یک گروه در دانشگاه MIT بر روی نظریه پتری پژوهش‌های بسیاری انجام دادند و مقالات و پایان‌نامه‌های بسیاری را ارائه دادند و در جولای سال ۱۹۷۵ یک کنفرانس از شبکه‌های پتری و موضوعات وابسته در این دانشگاه برگزار شد [۴۹]. تعدادی از مقالات نوشته شده در بحث شبکه‌های پتری در [۵۰] گردآوری شده‌اند که این مقالات تا قبل از سال ۱۹۸۰ ارائه شده‌اند. مقالات دیگری تا سال ۱۹۸۴ که در آلمان و سایر کشورهای اروپایی نوشته شده‌اند در پیوست کتاب [۵۱] آورده شده‌اند. منابع [۵۲]-[۵۴] مباحث مقدمه‌ای خوبی را برای آشنایی بیشتر با شبکه‌های پتری ارائه داده‌اند. موضوع شبکه پتری تا به حال مورد علاقه پژوهشگران در سرتاسر جهان قرار گرفته و مقالات بسیاری نوشته شده است.

فصل دوم

۱-۲- مقدمه

در این فصل به بیان مختصری از سینماتیک ربات‌ها می‌پردازیم. ابتدا توضیح می‌دهیم نقاط در فضا چگونه بیان می‌شوند و سپس به بیان جهت می‌پردازیم. اکنون آماده‌ایم تا یک جسم را در فضا بیان کنیم. این مطالب پایه‌ای به ما کمک می‌کنند تا بتوانیم نقاط دلخواه وابسته به ربات و یا لینک‌های یک ربات را در فضا بیان کنیم. به این منظور و با استفاده از این مطالب به بیان سینماتیک مستقیم^۱ و معکوس ربات می‌پردازیم. با استفاده از سینماتیک مستقیم می‌توانیم موقعیت و جهت هر نقطه از ربات را با استفاده از موقعیت مفاصل^۲ آن بیان کنیم و همچنین با استفاده از سینماتیک معکوس قادر خواهیم بود تا جابجایی مورد نیاز در موقعیت مفاصل را جهت قرار گرفتن نقطه‌ای از ربات در موقعیت و جهت دلخواه را محاسبه کنیم.

در تمامی این حالات ما از ماتریس‌ها جهت بیان موقعیت‌ها و جهت‌ها استفاده می‌کنیم. ماتریس-ها ما را قادر می‌سازند تا به راحتی حجم زیادی از اطلاعات را محاسبه، نمایش و توضیح دهیم. و در آخر هم به توضیح روش دناویت- هارتنبرگ^۳ به منظور پیاده‌سازی محاسبات سینماتیک مستقیم و معکوس ربات می‌پردازیم. مطالبی که در این فصل بیان می‌شود تنها جهت آشنایی خواننده با موضوع می‌باشد، تا خواننده مطالب بیان شده در فصل آینده را به راحتی درک نماید.

۲-۲- موقعیت و جهت

در این قسمت به بیان موقعیت و جهت اجسام در فضا می‌پردازیم به این منظور ابتدا به مفاهیم موقعیت و جهت می‌پردازیم و به طور جداگانه نشان می‌دهیم که یک موقعیت دلخواه و یا جهت دلخواه

۱. Forward Kinematics

۲. Joint variable

۳. Denavit-Hartenberg

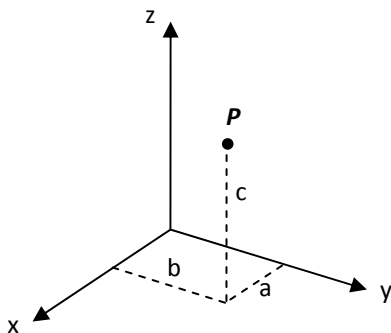
چگونه به صورت ماتریس بیان می‌شوند. با توجه به اینکه موقعیت، جهت، دستگاه، جابجایی و چرخش را به آسانی می‌توان با استفاده از ماتریس‌ها نمایش داد، از ابتدا به بیان ماتریسی می‌پردازیم.

۱-۲-۲- نقطه در فضا

بر اساس یک دستگاه مختصات کارتزین^۱ پایه^۲، تمامی نقاط در فضا با سه متغیر x, y, z بیان می‌شوند. نقاط با استفاده از بردارهای یکه و متغیرهای فضایی ذکر شده به صورت زیر بیان می‌شوند.

$$P = a\vec{i} + b\vec{j} + c\vec{k} \quad (۱-۲)$$

متغیرهای a, b, c جابجایی‌ها به ترتیب در راستای محور x, y, z در دستگاه مختصات مرجع می‌باشند، نقطه P و جابجایی‌های a, b, c در شکل ۱-۲ نمایش داده شده‌اند.



شکل ۱-۲ نقطه در فضا

۲-۲-۲- بردار در فضا

یک بردار در دستگاه مختصات کارتزین با استفاده از متغیرهای کارتزین نقاط ابتدا و انتهایش به صورت کامل بیان می‌شود. به این منظور بردار V را با نقاط ابتدائی و انتهایی A و B فرض کنید. نقاط A

۱. Cartesian coordinate

۲. Reference Frame

و B در فضا به صورت زید به طور کامل بیان می‌شوند.

$$\begin{cases} A = a\vec{i} + b\vec{j} + c\vec{k} \\ B = d\vec{i} + e\vec{j} + f\vec{k} \end{cases} \quad (۲ - ۲)$$

بردار V از اختلاف متغیرهای نظیر به نظیر A و B به دست می‌آید. و به صورت زیر نمایش داده می‌شود.

$$\vec{V} = x\vec{i} + y\vec{j} + z\vec{k} \quad (۳ - ۲)$$

که x, y, z در معادله ۲-۴ محاسبه شده‌اند.

$$\begin{cases} x = a - d \\ y = b - e \\ z = c - f \end{cases} \quad (۴ - ۲)$$

همچنین یک نقطه در فضا را می‌توان به صورت برداری بیان نمود که ابتدایش مبدا مختصات و انتهایش

نقطه مورد نظر می‌باشد معادله ۲-۵ این حقیقت را بیان می‌کند.

$$\begin{cases} P = x\vec{i} + y\vec{j} + z\vec{k} \\ \vec{V} = x\vec{i} + y\vec{j} + z\vec{k} \end{cases} \quad (۵ - ۲)$$

یک بردار در فضا را می‌توان با استفاده از ابعادش در فضا به صورت ماتریسی زیر نمایش داد.

$$\vec{V} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (۶ - ۲)$$

هر بردار در فضا را می‌توان علاوه بر فرم بالا به صورت جهت واحد و اندازه نیز نمایش داد. در این حالت

بردار موردنظر اندازه‌ای چون k و جهت واحدی چون v دارد و به صورت معادله ۲-۷ داده می‌شود.

$$\vec{V} = k \cdot \vec{v} \quad (۷ - ۲)$$

همچنین می‌توان هر دو بردار موازی را با استفاده از یک ضریب نمایش داد که در این حالت ضرایب مثبت

نمایشگر دو بردار هم‌جهت و ضرایب منفی نشان دهنده دو بردار خلاف جهت می‌باشند. طبق این واقعیت

می‌توان هر بردار را به صورت یک جهت و یک اندازه نمایش داد در نتیجه می‌توان از چهار متغیر برای نمایش یک بردار استفاده کرد بر این اساس بردار V هم جهت با خودش و دارای ضرایب ۱ می‌باشد که به صورت زیر نمایش داده می‌شود.

$$\vec{V} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (۸ - ۲)$$

همچنین بردار V را می‌توانیم به صورت زیر نمایش دهیم جایی که $x = \frac{a}{w}$, $y = \frac{b}{w}$, $z = \frac{c}{w}$ می‌باشد.

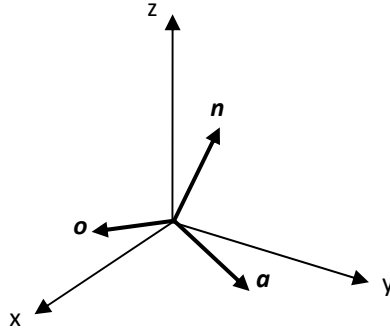
$$\vec{V} = \begin{bmatrix} a \\ b \\ c \\ w \end{bmatrix} \quad (۹ - ۲)$$

ضریب w می‌تواند اندازه یک بردار را به صورت دلخواه افزایش یا کاهش دهد بدون اینکه در جهت آن تغییری ایجاد نماید. در این حالت اگر w از یک کمتر باشد، اندازه بردار کاهش می‌یابد و در صورتی که بزرگتر از یک باشد اندازه بردار افزایش می‌یابد. در صورتی که w مساوی یک باشد، تغییری در اندازه بردار نمی‌دهد و w مساوی صفر، بردار را به یک جهت خالص تبدیل می‌نماید.

۳-۲-۲- بیان یک دستگاه در مبدا مختصات دستگاه مرجع

سه بردار $\vec{n}, \vec{o}, \vec{a}$ که دو به دو بر هم عمود می‌باشند یک دستگاه مختصات (F) را بیان نمایند فرض کنید که مبدا این دستگاه یعنی نقطه‌ای که سه بردار همدیگر را قطع کرده‌اند در مبدا مختصات دستگاه مرجع قرار گرفته است و دستگاه F حول نقطه مبدا دستگاه مرجع چرخانده شده است. در این حالت برای نمایش دستگاه F کافی است سه بردار $\vec{n}, \vec{o}, \vec{a}$ را در دستگاه مرجع بیان کنیم. در این حالت این سه بردار به صورت زیر داده می‌شوند.

$$[\vec{n}, \vec{o}, \vec{a}] = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (10-2)$$



شکل ۲-۲ دستگاهی در دستگاه مرجع

برای نمایش دستگاه F کافی است این سه بردار را به ترتیب در یک ماتریس 3×3 قرار دهیم در نتیجه دستگاه F در دستگاه مرجع به صورت زیر داده می‌شود.

$$F = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (11-2)$$

در نتیجه در صورتی که دستگاه F هیچ چرخشی نسبت به دستگاه مرجع نداشته باشد دستگاه F توسط یک ماتریس واحد 3×3 بیان می‌شود که به ترتیب بردارهای $\vec{n}, \vec{o}, \vec{a}$ به بردارهای $\vec{x}, \vec{y}, \vec{z}$ منطبق می‌باشند.

۲-۲-۴- بیان یک دستگاه در فضا

دستگاه F را بدون هیچ چرخشی نسبت به دستگاه مرجع در فضا در نظر بگیرید. دستگاه F تنها دارای جابجایی نسبت به دستگاه مرجع می‌باشد که این جابجایی با بردار V نمایش داده شده است. در این حالت برای بیان دستگاه F می‌توان از یک ماتریس جهت و یک بردار جابجایی V استفاده نمود که دستگاه مورد نظر به صورت زیر بیان می‌شود.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (۱۲-۲)$$

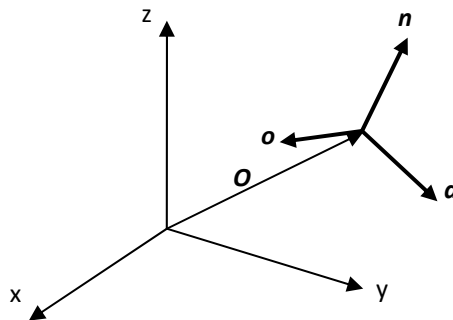
اکنون ما می‌خواهیم دستگاه F را تنها با یک ماتریس نمایش دهیم. به این منظور بردارهای جهت و موقعیت را در یک ماتریس مربعی با ضرایب اندازه به صورت زیر نمایش می‌دهیم.

$$F = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۱۳-۲)$$

در نتیجه دستگاه F که در مبدا مختصات دستگاه مرجع قرار دارد و با استفاده از رابطه ۲-۱۱ داده می‌شود را می‌توان به صورت زیر نمایش داد که هم‌ارز رابطه ۲-۱۱ می‌باشد.

$$F = \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۱۴-۲)$$

حال می‌توانیم با ترکیب این دو رابطه یک دستگاه F با هر جابجایی و هر چرخشی دلخواه در فضا نسبت به دستگاه مرجع را به صورت زیر نمایش دهیم.

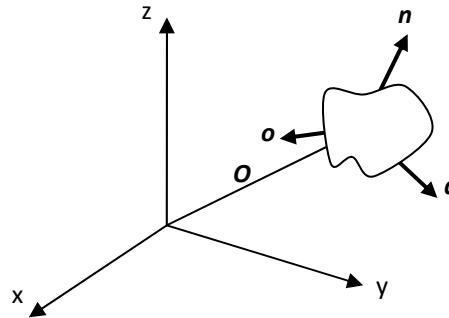


شکل ۲-۳ دستگاه در دستگاه مرجع

$$F = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۲-۱۵)$$

۲-۲-۵- بیان یک جسم صلب در فضا

برای بیان یک جسم صلب در فضا، کافی است یک دستگاه مختصات مربعی همچون F را به آن بچسبانیم و دستگاه F را در دستگاه مختصات مرجع بیان کنیم. در این حالت کلیه جابجایی‌ها و چرخشهای جسم با استفاده از دستگاه F قابل بیان می‌باشد یعنی، یک جسم در فضا با یک ماتریس بیان می‌شود که این ماتریس دستگاه چسبیده به جسم را در دستگاه مرجع بیان می‌کند. شکل ۲-۴ یک جسم و دستگاه چسبیده به آن را در فضا نمایش می‌دهد. بیان ماتریس جسم در معادله ۲-۱۶ آورده شده است.



شکل ۲-۴ یک جسم در فضا

$$F_{object} = \begin{bmatrix} n_x & o_x & a_x & O_x \\ n_y & o_y & a_y & O_y \\ n_z & o_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۲-۱۶)$$

۲-۳- ماتریس‌های تبدیل همگن

مطلوب است که ماتریس‌هایی را که برای بیان اجسام در فضا استفاده می‌کنیم به صورت مربعی بیان شوند. مزیت این نوع بیان در این است که معکوس ماتریس‌های مربعی نسبت به ماتریس‌های مستطیلی به راحتی محاسبه می‌شود و همچنین نتیجه ضرب دو ماتریس مربعی به صورت مربعی و با همان ابعاد باقی می‌ماند. در نتیجه در صورتی که بخواهیم چرخش و جابجایی یک جسم در فضا را نمایش دهیم می‌بایست ماتریس چرخش و بردار جابجایی مبدا دستگاه چسبیده به جسم را در یک ماتریس نمایش دهیم در این حالت با حفظ ضرایب اندازه تک‌تک بردارها به یک ماتریس 4×4 می‌رسیم که در معادله ۲-۱۷ آورده شده است و یک ماتریس همگن نامیده می‌شود.

$$A = \begin{bmatrix} n_x & o_x & a_x & O_x \\ n_y & o_y & a_y & O_y \\ n_z & o_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-17)$$

۲-۴- تبدیل

یک جابجایی در فضا یک تبدیل نامیده می‌شود. یک جسم یا یک دستگاه در فضا می‌تواند نسبت به دستگاه مختصات مرجع سه نوع تبدیل انجام دهد.

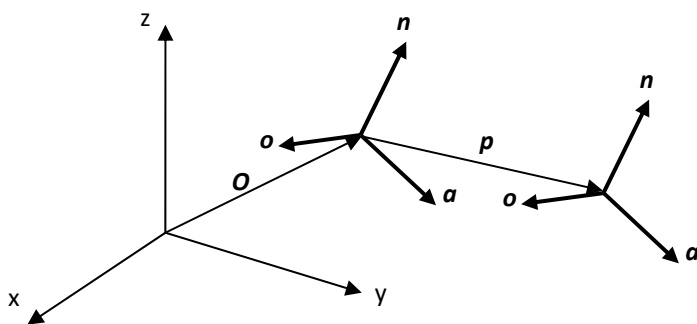
۱- یک جابجایی خطی خالص: در این حالت جسم یا دستگاه بدون هیچ‌گونه چرخشی نسبت به دستگاه مرجع، تنها در دستگاه جابجا می‌شود تا به یک موقعیت جدید نقل مکان نماید. در این حالت بردارهای نمایش‌دهنده جهت بدون تغییر مانده و جسم یا دستگاه در جهت قبلی باقی می‌ماند و تنها بردار جابجایی تغییر می‌نماید. چنین تبدیلی با ماتریس T نمایش داده می‌شود که در معادله زیر آورده شده است.

$$T = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-18)$$

همانطور که در این ماتریس دیده می‌شود، هیچ چرخشی بیان نشده و تنها جابجایی در راستای بردار P مشهود می‌باشد. موقعیت جدید جسم یا دستگاه موردنظر توسط رابطه زیر بدست می‌آید که نشان‌دهنده-ی پیش ضرب ماتریس T در ماتریس بیان‌کننده‌ی جسم یا دستگاه است. همانطور که ماتریس نتیجه نشان می‌دهد هیچ نوع تغییری در ماتریس جهت به وجود نیامده است و تنها در بردار جابجایی شاهد تغییر می‌باشیم که تغییر به صورت مجموع بردار جابجایی و بردار موقعیت مبدا دستگاه جسم یا مبدا دستگاه داده می‌شود. این واقعیت در شکل ۲-۵ نیز نمایش داده شده است.

$$A_{new} = T \times A_{old} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & O_x \\ n_y & o_y & a_y & O_y \\ n_z & o_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & o_x & a_x & O_x + p_x \\ n_y & o_y & a_y & O_y + p_y \\ n_z & o_z & a_z & O_z + p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (۲-۱۹)$$



شکل ۲-۵ جابه‌جایی خالص در فضا

۲- یک جسم یا دستگاه در فضا می‌تواند تنها یک چرخش خالص نسبت به دستگاه مرجع انجام دهد. در این حالت دستگاه یا جسم طول مبدا مختصاتش چرخشی انجام داده است. واضح است که چرخش حول مبدا دستگاه متصل به جسم منجر به هیچ‌گونه جابجایی مبدا دستگاه متصل نسبت به دستگاه مرجع

نمی‌شود. چنین تبدیل یا ماتریسی آورده شده در معادله ۲-۲۰ داده می‌شود. همانطور که در ماتریس R دیده می‌شود، بردار جابجایی خطی ماتریس دارای مقادیر صفر در راستای محورهای دستگاه مرجع می‌باشد که، نشان‌دهنده باقی ماندن دستگاه متصل به جسم در مختصات قدیم خود می‌باشد. در ماتریس R تنها جهت‌هایی نشان داده شده‌اند که دستگاه متصل به جسم را از جهت قدیم به یک جهت جدید منتقل می‌نماید.

$$R = \begin{bmatrix} nr_x & or_x & ar_x & 0 \\ nr_y & or_y & ar_y & 0 \\ nr_z & or_z & ar_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20 - 2)$$

جهت جدید جسم با استفاده از پیش ضرب ماتریس جهت ماتریس R در ماتریس جهت دستگاه چسبیده به جسم به دست می‌آید که در رابطه ۲-۲۱ آورده شده است.

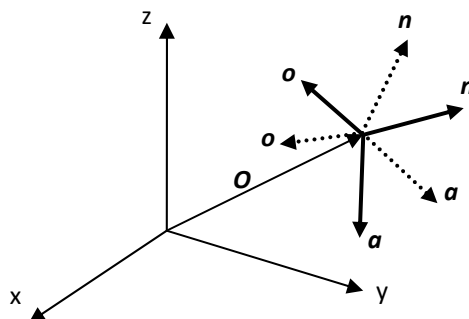
$$Direction = \begin{bmatrix} nr_x & or_x & ar_x \\ nr_y & or_y & ar_y \\ nr_z & or_z & ar_z \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (21 - 2)$$

همچنین با حفظ ابعاد ماتریس می‌توان دستگاه جدید را به صورت زیر نیز بیان کرد که تنها منجر به تغییر در بردارها جهت می‌شود و بردار جابجایی را بدون تغییر باقی می‌گذارد در این حالت نیز ماتریس دستگاه چسبیده به جسم از پیش ضرب ماتریس R در ماتریس دستگاه چسبیده به جسم به دست می‌آید.

$$A_{new} = R \times A_{old} = \begin{bmatrix} nr_x & or_x & ar_x & 0 \\ nr_y & or_y & ar_y & 0 \\ nr_z & or_z & ar_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & O_x \\ n_y & o_y & a_y & O_y \\ n_z & o_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \dot{n}_x & \dot{o}_x & \dot{a}_x & O_x \\ \dot{n}_y & \dot{o}_y & \dot{a}_y & O_y \\ \dot{n}_z & \dot{o}_z & \dot{a}_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22 - 2)$$

شکل ۲-۶ نشان‌دهنده یک چرخش خالص نسبت به دستگاه مرجع می‌باشد.



شکل ۲-۶ چرخش خالص در فضا

۳- یک جسم یا دستگاه می‌تواند در یک تبدیل ترکیبی از جابجایی خطی و چرخشی را انجام دهد و در یک موقعیت و چرخش متفاوت نسبت به دستگاه مرجع قرار گیرد. در این صورت ترتیب تبدیل بسیار مهم بوده و می‌بایست ترتیب ضرب ماتریس‌های تبدیل به درستی انجام شود. در صورتی این ترتیب تغییر کند، نتیجه تبدیل متفاوت و اشتباه می‌باشد. ترتیب تبدیلات جهت یک تبدیل ترکیبی به صورت زیر داده می‌شود:

۱- چرخش به اندازه θ_x درجه حول محور x

۲- جابجایی در طول بردار $[x, y, z]$

۳- چرخش به اندازه θ_y درجه حول محور y

که x, y, z محورهای دستگاه مختصات مرجع می‌باشند. طی این ترتیب از تبدیلات، تبدیل نهایی به صورت زیر داده می‌شود:

$$RT = R_y \times T \times R_x \quad (23 - 2)$$

بنابراین نتیجه نهایی به صورت زیر می‌باشد.

$$A_{New} = RT \times A_{Old} \quad (24 - 2)$$

همانطور که مشاهده می‌کنیم ترتیب ضرب‌ها در جهت عکس ترتیب ضرب‌ها در جهت عکس ترتیب تبدیلات می‌باشد رابطه ۲-۲۴ گویای این مطلب است که این دو هیچ تناقضی با هم نداشته و رابطه ۲-۲۵ بر اساس ترتیب ذکر شده، نوشته شده است.

$$\begin{cases} A_1 = R_x \times A_{Old} \\ A_2 = T \times A_1 \\ A_{New} = R_y \times A_2 \end{cases} \quad (25 - 2)$$

المان‌های ماتریس چرخش به دلیل قیدهای دستگاه مختصات کارتیزین ساده می‌شوند که این خصوصیات در زیر آورده شده‌اند.

- سه بردار دستگاه مختصات (n, o, a) دو به دو بر هم عمود می‌باشند،
- اندازه بردارهای a, o, n واحد می‌باشد.

که این قیدها به خصوصیات زیر منجر می‌شوند.

- (I) $\vec{n} \cdot (\text{the dot product}) \vec{o} = 0$
- (II) $\vec{n} \cdot \vec{a} = 0$
- (III) $\vec{a} \cdot \vec{o} = 0$
- (IV) $|\vec{n}| = 1$
- (V) $|\vec{o}| = 1$
- (VI) $|\vec{a}| = 1$

در نتیجه معادله به صورت زیر نوشته می‌شود.

$$A_{new} = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x & 0 \\ 0 & \sin \theta_x & \cos \theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26 - 2)$$

با ساده‌سازی این معادله بر اساس خصوصیات ذکر شده به معادله زیر می‌رسیم.

$$A_{new} = \begin{bmatrix} \cos \theta_y & \sin \theta_y \sin \theta_x & \sin \theta_y \cos \theta_x & x \cos \theta_y + z \sin \theta_y \\ 0 & \cos \theta_x & -\sin \theta_x & y \\ -\sin \theta_y & \cos \theta_y \sin \theta_x & \cos \theta_y \cos \theta_x & -x \sin \theta_y + z \cos \theta_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27-2)$$

۵-۲- معکوس ماتریس‌های تبدیل

در این حالت موقعیت جدید جسم یا دستگاه و تبدیلاتی که انجام شده تا جسم یا دستگاه از یک موقعیت قدیم در این موقعیت قرار بگیرند، معلوم می‌باشد. مطلوب است تا بتوانیم بر اساس این معلومات موقعیت قدیم جسم یا دستگاه را محاسبه کنیم. در نتیجه، ابتدا می‌بایست عکس تبدیلات انجام شده را محاسبه نماییم سپس، این تبدیلات محاسبه شده می‌بایست در جهت عکس معادله ۲-۲۶ در هم ضرب شوند. در آخر، با پیش ضرب نتیجه بدست آمده در معادله ۲-۲۸ موقعیت قدیم جسم یا دستگاه به دست می‌آید.

$$\begin{aligned} (R_x^{-1} \times T^{-1} \times R_y^{-1})A_{new} &= (R_x^{-1} \times T^{-1} \times R_y^{-1}) \times R_y \times T \times R_x \times A_{old} \\ A_{old} &= (R_x^{-1} \times T^{-1} \times R_y^{-1})A_{new} \end{aligned} \quad (28-2)$$

جهت بدست آوردن معکوس یک ماتریس مربعی می‌بایست مراحل زیر انجام شود.

- دترمینان ماتریس محاسبه شود.
- ترانزپوز ماتریس محاسبه شود.
- کهادهای ماتریس ترانزپوز محاسبه شود.
- ماتریس بدست آمده بر دترمینان تقسیم شود.

اگرچه ماتریس‌های تبدیل پیچیده می‌باشند اما برای بدست آوردن معکوس آن‌ها نیازی به مراحل بالا نیست و به راحتی بدون انجام محاسبات پیچیده می‌توان معکوس ماتریس‌های تبدیل را محاسبه نمود.

دترمینان تمامی ماتریس‌های تبدیل برابر ۱ می‌باشد. همچنین کهاد ماتریس ترانهاده در مرحله سوم به راحتی از ترانهاده کردن ماتریس جهت و عملیات کوچکی که در زیر آمده است محاسبه می‌شوند.

$$\begin{cases} O_x \rightarrow -\vec{O}_x \cdot \vec{n} \\ O_y \rightarrow -\vec{O}_y \cdot \vec{o} \\ O_z \rightarrow -\vec{O}_z \cdot \vec{a} \end{cases} \quad (29 - 2)$$

در حالت کلی معکوس ماتریس A به صورت زیر داده می‌شود:

$$A = \begin{bmatrix} n_x & o_x & a_x & O_x \\ n_y & o_y & a_y & O_y \\ n_z & o_z & a_z & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30 - 2)$$

$$A^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\vec{O}_x \cdot \vec{n} \\ o_x & o_y & o_z & -\vec{O}_y \cdot \vec{o} \\ a_x & a_y & a_z & -\vec{O}_z \cdot \vec{a} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

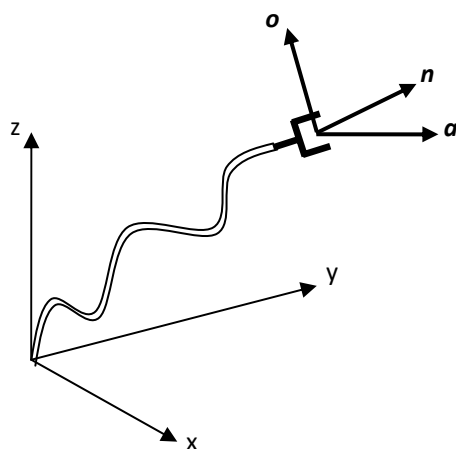
۲-۶- سینماتیک مستقیم و معکوس ربات

یک بازو که پیکربندی آن کاملاً معلوم و مشخص می‌باشد را در نظر بگیرید. به این معنی که ابعاد کلیه لینک‌ها و موقعیت تمامی مفاصل‌ها معلوم می‌باشد. محاسبه موقعیت فضایی و چرخشی نقطه نهایی ربات بر اساس ابعاد لینک‌ها و موقعیت مفاصل، سینماتیک مستقیم ربات نامیده می‌شود. به عبارت دیگر، با استفاده از معادلات سینماتیک مستقیم، موقعیت و جهت نقاط روی ربات در هر لحظه در صورتی که موقعیت مفاصل مشخص باشد قابل محاسبه می‌باشد. محاسبه مقدار جابجایی در موقعیت مفاصل ربات، جهت دستیابی نقطه نهایی ربات به موقعیت و جهت خاص، سینماتیک معکوس ربات نامیده می‌شود.

در واقع ابزاری که مورد استفاده ما قرار خواهد گرفت، سینماتیک معکوس ربات می‌باشد. به این دلیل که ما علاقه‌مند هستیم تا ربات در هر لحظه در موقعیت و جهت خاصی که مطلوب است باید قرار بگیرد. در این حالت کنترلر ربات نیازمند موقعیت مفاصل مطلوب در هر لحظه می‌باشد تا بتواند ربات را

در مسیر دلخواه به موقعیت و جهت نهایی برساند. که این محاسبات با استفاده از معادلات سینماتیک معکوس انجام می‌شود.

همانطور که گفته شد، به منظور بیان موقعیت و جهت یک جسم صلب در فضا، یک دستگاه مختصات را به جسم می‌چسبانیم و دستگاه را در دستگاه مرجع بیان می‌کنیم. شکل ۲-۷ نقطه نهایی ربات و دستگاه چسبیده به آن را به همراه دستگاه مختصات مرجع نمایش می‌دهد.

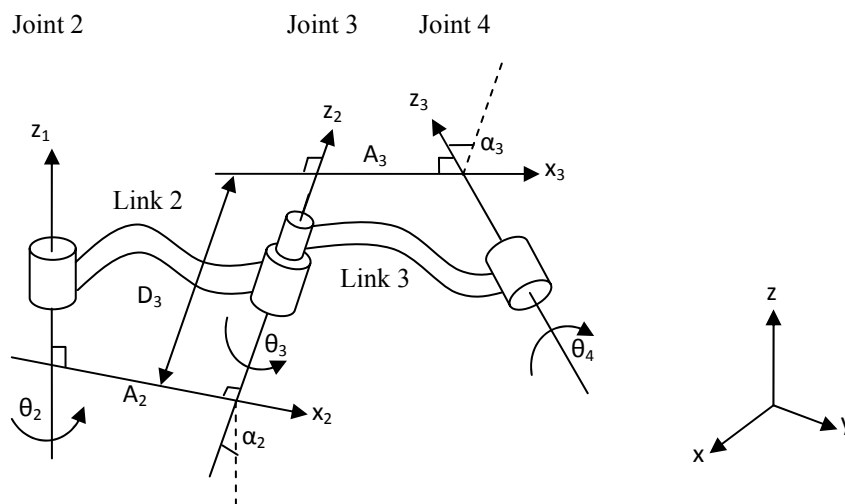


شکل ۲-۷ دستگاه نهایی ربات در فضا

حال به بیان روش دناویت و هارتنبرگ در محاسبه معادلات سینماتیک ربات می‌پردازیم. در سال ۱۹۹۵، دناویت و هارتنبرگ در ASME Journal Of Applied Mechanics مقاله‌ای به چاپ رسانید که در آن روشی را برای بدست آوردن محاسبات سینماتیکی بیان نمود. روش دناویت و هارتنبرگ بسیار ساده و قابل پیاده‌سازی بر روی ربات با هر شکل و پیکربندی بود. این روش هم اکنون به یک روش استاندارد در محاسبه سینماتیک ربات تبدیل شده است. به منظور بیان روش دناویت و هارتنبرگ، یک بازو را که از چند مفصل و لینک که پشت سر هم متصل شده‌اند را در نظر بگیرید. مفاصل می‌توانند کشویی یا چرخشی باشند و جهت بیان انواع دیگر مفاصل از ترکیب این مفصل‌ها استفاده می‌کنیم.

لینک‌ها می‌توانند با هر شکل و هر طولی باشد، که این طول شامل طول صفر هم می‌باشد. بنابراین ربات مورد بررسی ابتدا می‌بایست تحت این شرایط بیان شود. به این منظور، ابتدا به هر مفصل یک دستگاه مختصات کارتزین به صورتی که ذکر خواهد شد می‌چسبانیم. سپس روشی را برای بدست آوردن ماتریس تبدیل از یک مفصل به مفصل بعد یا قبل از آن بنیان می‌نهیم.

شکل ۲-۸، سه مفصل و دو لینک از یک ربات را نمایش می‌دهد. پیکربندی نمایش داده شده در این شکل کلی بوده و به راحتی می‌تواند مفاصل در رباتهای واقعی را بیان کند. این مفاصل می‌توانند خطی، چرخشی و یا ترکیبی از آنها باشند.



شکل ۲-۸ بیان دناویت-هارتنبگ از ربات

در روش دناویت هارتنبگ محور y در دستگاههای مختصات نشان داده نمی‌شود و هیچ‌گونه چرخشی نسبت به این محور انجام نمی‌شود. روش محورگذاری یا قرار دادن دستگاههای محلی در مفاصل به صورت زیر آورده می‌شود. این روش در شکل ۲-۸ نمایش داده شده است.

- همه مفاصل توسط محور Z بیان می‌شوند. اگر مفصل کشویی باشد محور Z دستگاه مختصات محلی در طول حرکت مفصل قرار می‌گیرد. همچنین اگر مفصل چرخشی باشد محور Z بر اساس قاعده دست راست در محور چرخش مفصل قرار می‌گیرد.
 - محورهای Z متحرک بوده و خطوط عمود بر دو محور متوالی عمود نامیده می‌شود. محور X در جهت این عمود قرار داده می‌شود. همانطور که در شکل ۲-۸ نمایش داده شده است.
 - اگر دو محور Z متوالی، موازی هم باشند. محور X به گونه‌ای انتخاب می‌شود که، مبدا دستگاه مختصات محلی قبلی‌اش را قطع نماید.
 - در صورتی که دو محور Z متوالی، متقاطع باشند محور X به صورت عمود بر صفحه محورهای Z انتخاب شده و جهت آن از ضرب خارجی دو محور Z بدست می‌آید.
- همانطور که در شکل ۲-۸ نشان داده شده است، θ چرخش حول محور Z ، A طول عمود بر دو محور Z ، D طول روی محور Z مابین دو محور X متوالی و α زاویه میان دو محور Z متوالی حول محور X را بیان می‌کنند. عموماً، θ و D متغیرهای مفصلی نامیده می‌شوند.
- حال فرض کنیم در دستگاه محلی Z_n قرار داریم و می‌خواهیم طی یک تبدیل به دستگاه محلی Z_{n+1} منتقل شویم، در این صورت تبدیلات زیر می‌بایست پشت سر هم و به ترتیب انجام شوند.
- ۱- چرخش حول محور Z_n به اندازه θ_{n+1} ، تا X_n و X_{n+1} با هم موازی شوند.
 - ۲- جابجایی خطی در طول محور Z_n به اندازه D_{n+1} تا X_n و X_{n+1} بر هم منطبق شوند.
 - ۳- جابجایی خطی در طول محور X_n به اندازه A_{n+1} تا مبدا دستگاه مختصات X_n و X_{n+1} بر هم منطبق شوند.
 - ۴- چرخش محور Z حول محور Z به اندازه α_{n+1} تا Z_n و Z_{n+1} بر هم منطبق شوند.
- در نتیجه تبدیل مطلوب به صورت زیر داده می‌شود:

$$\begin{aligned}
A_n^{n+1} &= DH_{n+1} = \\
&\begin{bmatrix} \cos \theta_{n+1} & -\sin \theta_{n+1} & 0 & 0 \\ \sin \theta_{n+1} & \cos \theta_{n+1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\
&\begin{bmatrix} 1 & 0 & 0 & A_{n+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_{n+1} & -\sin \alpha_{n+1} & 0 \\ 0 & \sin \alpha_{n+1} & \cos \alpha_{n+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (31 - 2) \\
&= \begin{bmatrix} \cos \theta_{n+1} & -\sin \theta_{n+1} \cos \alpha_{n+1} & \sin \theta_{n+1} \sin \alpha_{n+1} & A_{n+1} \cos \theta_{n+1} \\ \sin \theta_{n+1} & \cos \theta_{n+1} \cos \alpha_{n+1} & -\cos \theta_{n+1} \sin \alpha_{n+1} & A_{n+1} \sin \theta_{n+1} \\ 0 & \sin \alpha_{n+1} & \cos \alpha_{n+1} & D_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

که n شماره مفصل می باشد آنگاه سینماتیک مستقیم ربات به صورت زیر داده می شود.

$$A_0^n = A_0^1 \times A_1^2 \times \dots \times A_{n-1}^n = DH_1 \times DH_2 \times \dots \times DH_n \quad (32 - 2)$$

که تعداد مفصل ربات به اندازه n+1 می باشد.

همانطور که قبلاً بیان شد، حل معادلات سینماتیک معکوس ربات مطلوب ما می باشد در این

حالت A_0^n در دسترس بوده و می بایست متغیرهای مفصلی بدست می آیند. در نتیجه می بایست ۱۲ معادله

و ۱۲ مجهول حل شوند تا متغیرها بدست بیایند.

فصل سوم

۱-۳- مقدمه

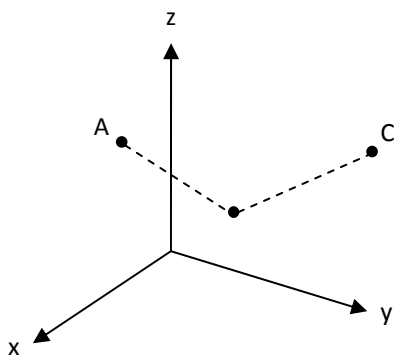
در فصل قبل، مروری بر سینماتیک ربات را انجام دادیم و آموختیم که چگونه با استفاده از معادلات سینماتیک مستقیم موقعیت و جهت نقطه نهایی ربات را بر اساس متغیرهای مفصلی معلوم بدست آوریم و همچنین آموختیم که چگونه متغیرهای مفصلی را به منظور قرار گرفتن نقطه نهایی ربات در موقعیت و جهت مطلوب را با استفاده از معادلات سینماتیک معکوس بدست آوریم. به منظور اینکه ربات از یک نقطه به نقطه دیگر به صورت کنترل شده حرکت کند، مسیر یا یک تراژکتوری حرکت ربات می‌بایست طراحی شود. در این حالت مطلوب است که ربات به منظور دستیابی به نقطه مطلوب از مسیری خاص عبور کند.

در این فصل دو روش عمومی در طراحی مسیر را بیان می‌کنیم که، بیان متغیر مفصلی و بیان کارتزین نامیده می‌شوند. همچنین روشی جدید را در این فصل به منظور طراحی مسیر ارائه خواهیم داد که در برابر این دو روش ذکر شده بسیار قدرتمند می‌باشد. این روش بر اساس روش‌های تطبیقی به منظور حرکت ربات در یک مسیر مناسب بنا نهاده شده است. در این فصل معیاری به نام معیار شباهت را به منظور پیاده‌سازی کنترل تطبیقی بیان خواهیم نمود.

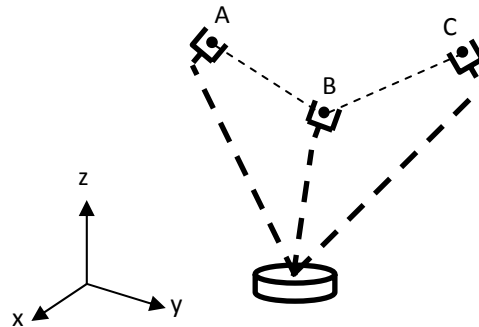
روش‌های موجود دنباله‌ای از مسیرها را تولید می‌نمایند که ربات به صورت کنترل شده آن را می‌پیماید، در صورتی که در روش معرفی شده ربات در طول تکه مسیرهایی که طراحی می‌شوند نیز به صورت کنترل شده حرکت می‌نماید. به دلیل این نوع کنترل در روش‌های قدیمی، دقت در این روش‌ها بسیار حساس نسبت به قطعات مسیر طراحی شده می‌باشد. در صورتی که در روشی که معرفی می‌شود، دقت قابل تنظیم بوده و یک مسیر مناسب بر اساس دقت مطلوب طراحی می‌شود. به این دلیل، ربات می‌تواند هر مسیری را بر اساس دقت مطلوب طی نماید.

۲-۳- مسیر و تراژکتوری

بدون در نظر گرفتن پیکربندی ربات، می‌توان یک مسیر را به صورت دنباله محدودی از نقاط بدون زمان در فضا که به ترتیب خاصی به دنبال هم قرار گرفته‌اند و شامل نقطه شروع و پایان هستند، تعریف نمود. به عبارت دیگر، مسیر دنباله‌ای از پیکربندی‌های ربات با ترتیب مشخص می‌باشد. به عنوان مثال، با توجه به شکل ۱-۳ الف، ربات می‌بایست به ترتیب به نقاط C, B, A دست یابد، این نقاط یک مسیر را بیان می‌کنند. یعنی همانطور که در شکل ۱-۳ ب نشان داده شده است دنباله‌ای از پیکربندی ربات که به ترتیب موقعیت‌های C, B, A را نشان می‌دهد، بیان‌کننده یک مسیر می‌باشد.



(الف)



(ب)

شکل ۱-۳ دنباله‌ای از نقاط در فضا که تشکیل یک مسیر می‌دهند.

هر چند در طراحی مسیر نقاطی که می‌بایست توسط ربات لمس شوند دارای زمان‌بندی نمی‌باشند، اما در یک تراژکتوری نقاط زمان‌دار می‌باشند و ربات باید به هر نقطه در زمان مشخصی دست یابد. بدون در نظر گرفتن یک برنامه زمان‌بندی برای نقاط ربات مسیر بین دو نقطه را همیشه به صورت یکسان می‌پیماید، در صورتی که با در نظر گرفتن یک برنامه زمان‌بندی برای نقاط بر اساس زمان‌بندی‌های متفاوت برای مجموعه نقاط یکسان مسیرهای پیموده شده متفاوت می‌باشند.

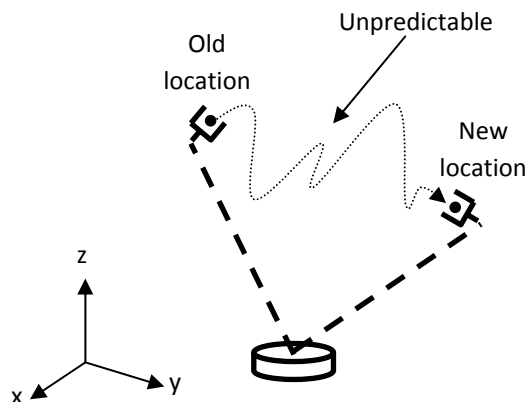
یکی از مزایای روش ما این است که ربات مجموعه نقاط یکسان را حتی در زمان بندی‌های متفاوت به صورت یکسان طی نماید. در این صورت تمامی مسیر طی شده توسط ربات قابل پیش‌بینی می‌باشد که مزیت بسیار مهمی است. مسیرهای طراحی شده در صورتی متفاوت می‌باشند که دقت مطلوب کاربر تغییر نماید، اما در این حالت نیز مسیر و تمام تراژکتوری‌های ممکن برای دقت مورد نظر مشابه می‌باشند. همچنین مسیر طی شده بین دو نقطه متوالی کاملاً مشخص می‌باشد در صورتی که در روش‌های دیگر این مسیر به راحتی قابل پیش‌بینی نمی‌باشد. در این فصل ما روش جدید را جهت طراحی مسیر تعریف خواهیم کرد.

۳-۳- بیان متغیر مفصلی و بیان کارتزین

رباتی را در نظر بگیرید که می‌بایست جهت رسیدن به نقطه‌ای مورد نظر در فضا هدایت شود. با استفاده از معادلات سینماتیک معکوس، متغیرهای مفصلی ربات برای نقطه مورد نظر محاسبه می‌شوند تا اینکه ربات بتواند با برآوردن تغییرات مورد نیاز در متغیرهای مفصلی به نقطه مورد نظر دست یابد. به این منظور، مفصل‌های ربات را از موقعیت مفصلی کنونی به اندازه تغییرات مورد نیاز حرکت می‌دهیم که در این حالت ربات در یک مسیر غیر قابل پیش‌بینی از موقعیت کنونی به موقعیت مطلوب حرکت می‌کند، همانطور که در شکل ۳-۲ نمایش داده شده است. بیان حرکت ربات با استفاده از متغیرهای مفصلی آن بیان مفصلی نامیده می‌شود. در این حالت بدون در نظر گرفتن موقعیت لحظه به لحظه ربات، مفاصل تنها جابجایی‌های خود را انجام می‌دهند تا به موقعیت مفصلی مطلوب دست یابند. غیر قابل پیش‌بینی بودن مسیر از آن جهت است که محرک‌های مفاصل می‌توانند با شتاب و سرعت متفاوتی در طول جابجایی خود حرکت کنند.

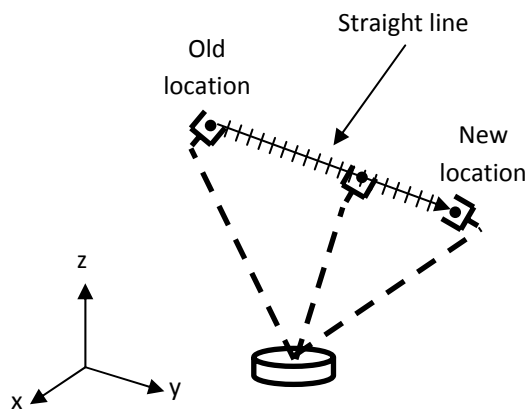
حال فرض کنید که مطلوب است تا ربات در طول یک مسیر از پیش تعریف شده از موقعیت قدیم خود به یک موقعیت جدید هدایت شود. در این حالت می‌توان نقاط محدودی را بر روی مسیر

انتخاب و ربات را در طول این نقاط هدایت کرد. به این منظور می‌بایست معادلات سینماتیک معکوس



شکل ۳-۲

برای این نقاط انتخاب شده، محاسبه شوند. همانطور که انتظار می‌رود هر چه تعداد این نقاط اضافه شود، ربات با دقت بیشتری در طول مسیر حرکت داده می‌شود اما این امر مستلزم این است که تعداد معادلات سینماتیک معکوس بیشتری حل شود. در صورتی که این معادلات بسیار زمان‌بر و حافظه‌بر می‌باشند. شکل ۳-۳ رباتی را نمایش می‌دهد که می‌بایست جهت دستیابی به موقعیت جدید در طول یک خط

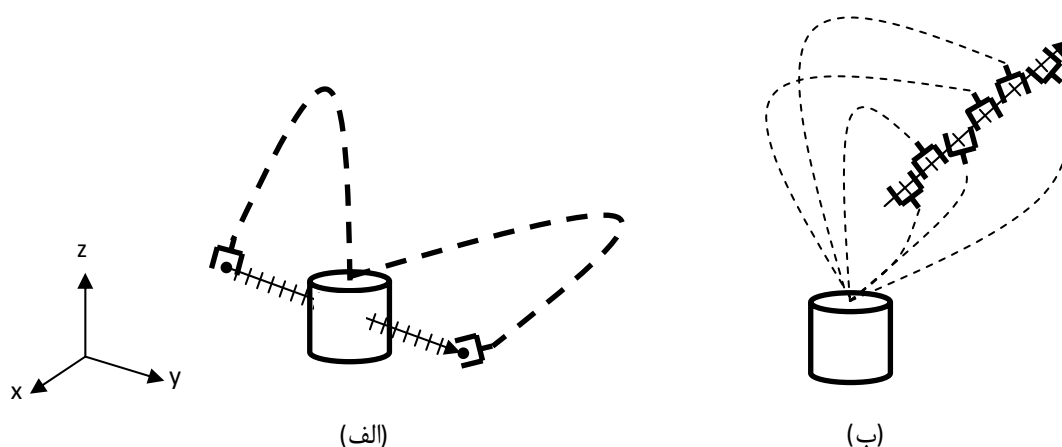


شکل ۳-۳

راست حرکت نماید. معادلات سینماتیک معکوس می‌بایست برای تک‌تک نقاط انتخاب شده محاسبه شوند. بنابراین مجموعه‌ای از متغیرهای مفصلی به وجود می‌آید که ربات می‌بایست به ترتیب به آنها دست یابد. زمانی که ربات این مجموعه را بپیماید به نقطه مطلوب در راستای یک خط راست دست می‌یابد. در این حالت مسیر حرکت ربات جهت رسیدن به نقطه مورد نظر شناخته شده می‌باشد و نقاط میانی در دستگاه کارتزین تعریف می‌شوند که این نوع بیان از حرکت ربات، بیان کارتزین نامیده می‌شود. بیان کارتزین مجموعه‌ای چند بیان مفصلی می‌باشد زیرا مسیر ربات میان هر دو نقطه میانی پشت سر هم به صورت مفصلی بیان می‌شود. بنابراین حجم محاسبات در بیان کارتزین بسیار بیشتر از حجم محاسبات در بیان مفصلی می‌باشد. در صورتی که بیان کارتزین بر خلاف بیان مفصلی به یک مسیر قابل پیش‌بینی می‌انجامد هر چند که مسیر تنها در نقاط میانی مشخص می‌باشد و مسیر مابین آنها غیر قابل پیش‌بینی است. هر چند که بیان مفصلی و کارتزین دارای مزایا و معایبی هستند اما بسیار مفید می‌باشند و در صنعت دارای کاربرد گسترده‌ای هستند.

هر منحنی در فضا می‌تواند یک مسیر کارتزین (توصیف شده در دستگاه کارتزین) توصیف شود و تنها کافی است کشیده شود. آسانی در نقاشی یک مسیر کارتزین یکی از مزایای بیان کارتزین می‌باشد و هر شخصی می‌تواند به طور دلخواهانه مسیری را در فضا بکشد. هر چند که مسیرهای کارتزین در رسم و بسیار آسان می‌باشند و به راحتی قابل تصور هستند اما دارای حجم محاسباتی بالایی هستند. مسیرهای کارتزین حتی نیازمند پردازش‌گر پر سرعت‌تری نسبت به مسیرهایی هستند که در بیان مفصلی توصیف می‌شوند. علاوه بر آن، در بیان کارتزین اجتناب از نقاط تکین امری مشکل می‌باشد و طراح می‌بایست در طول طراحی مسیر به نقاط تکین نیز توجه داشته باشد. در حالیکه در بیان مفصلی هیچ‌گونه نگرانی نسبت به نقاط تکین وجود ندارد زیرا ربات جهت رسیدن به نقطه مطلوب وادار به حرکت در مسیری خاص نمی‌شود و تنها در مسیری ممکن حرکت می‌نماید. شکل‌های ۳-۴ الف و ۳-۴ ب موارد نامطلوب

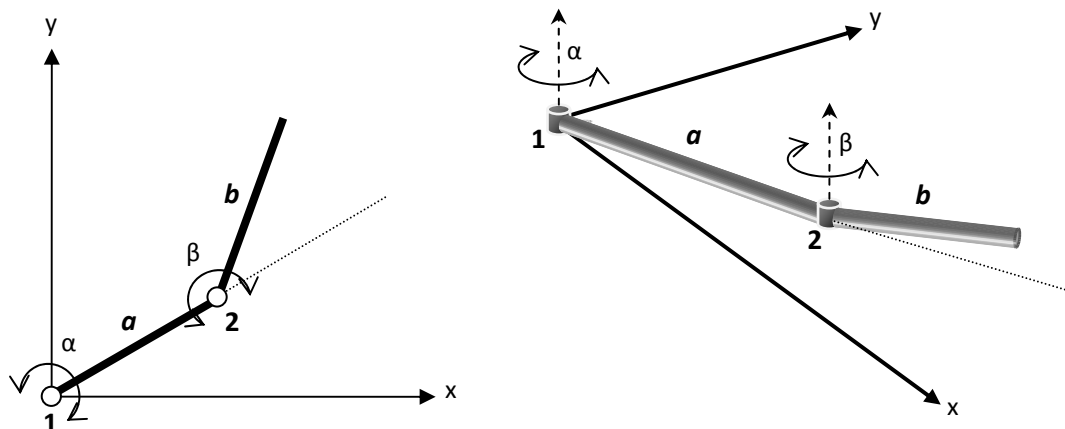
ممکنی را نشان می‌دهند که در بیان کارترین ممکن است روی دهند. این موارد نامطلوب زمانی روی می‌دهند که توجهی نسبت به اینکه ربات چگونه مسیر را طی می‌کند، وجود نداشته باشد. شکل ۳-۴ الف حالتی را نشان می‌دهد که ربات جهت رسیدن به موقعیت جدید می‌بایست مسیری را طی نماید که از یک جسم یا از قسمتی از ربات عبور می‌کند. همچنین شکل ۳-۴ ب حالتی را نمایش می‌دهد که ربات جهت رسیدن به موقعیت جدید از مسیر طراحی شده می‌بایست تغییرات ناگهانی در موقعیت مفاصل خود دهد.



شکل ۳-۴ موارد نامطلوب در بیان کارترین

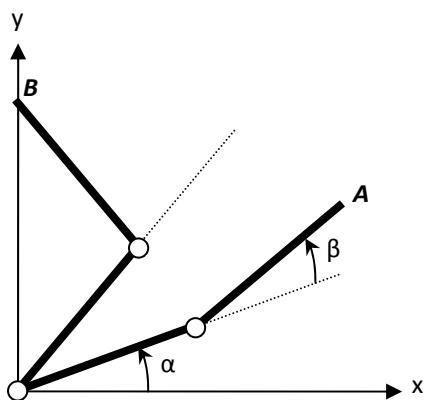
۳-۴- طراحی مسیر

یک بازوی RR ساده را به منظور توضیح اساس طراحی مسیر در بیان مفصلی و کارترین، در نظر بگیرید این بازو در شکل ۳-۵ نمایش داده شده است. چنین بازویی، مکانیزمی با دو درجه آزادی می‌باشد و مفاصل آن دارای محورهای موازی می‌باشند. جهت آسانی در بیان مسئله فرض کنید لینک‌های a ، b دارای سایز مساوی هستند.



شکل ۳-۵ بازوی کارتیزین دو لینکی

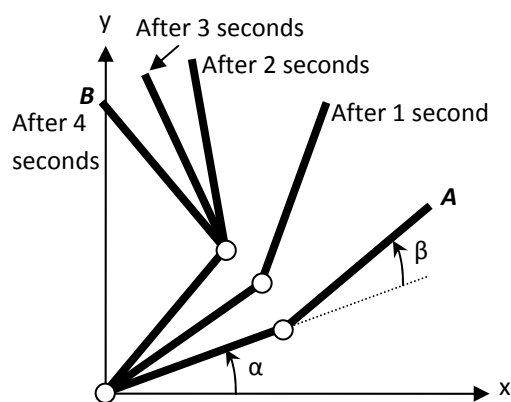
همانطور که در شکل ۳-۶ نشان داده شده است، مطلوب می‌باشد که ربات از موقعیت A به موقعیت جدید B حرکت کند. در موقعیت A متغیرهای مفصلی α , β به در موقعیت زاویه‌ای ۲۰ درجه می‌باشند و می‌بایست این موقعیتهای زاویه‌ای به ۵۰ و ۸۰ در موقعیت B برسند. محرکهای مفاصل یکسان بوده و می‌توانند حداکثر با سرعت ۱۵ درجه بر ثانیه بچرخند.



شکل ۳-۶

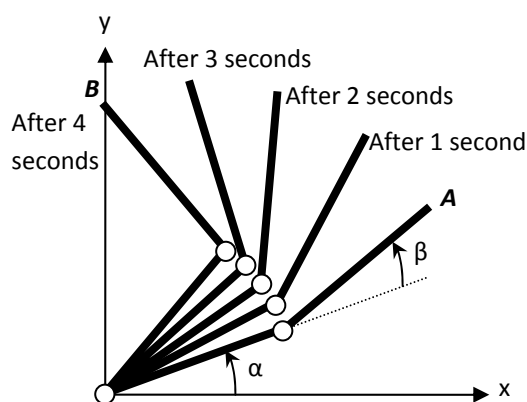
در ابتدا هر مفصل را با سرعت زاویه‌ای ۱۵ درجه بر ثانیه حرکت می‌دهیم. جهت دستیابی به موقعیت B، مفصل دو می‌بایست به اندازه ۳۰ درجه بیشتر از مفصل یک بچرخد. بنابراین، در این حالت،

همانطور که در شکل ۳-۷ نمایش داده شده است لینک a زودتر از لینک b متوقف می‌شود. لینک a به مدت ۲ ثانیه و لینک b به مدت ۴ ثانیه حرکت خواهند نمود، در نتیجه بازو، ۴ ثانیه پس از شروع حرکت به نقطه B دست خواهد یافت. همانطور که در شکل؟ مشهود می‌باشد، مسیر پیموده شده ناموزون (بی-شکل) می‌باشد و فواصل طی شده توسط نقطه نهایی ربات یکنواخت نمی‌باشند.



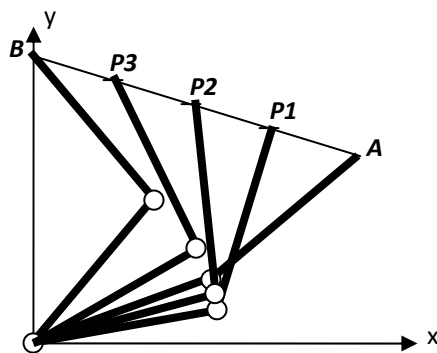
شکل ۳-۷

حال فرض کنید که مفصل یک با سرعت $7/5$ متر بر ثانیه بچرخد در حالیکه مفصل دو با نهایت سرعت خود خواهد چرخید. در این حالت هر دو لینک یا مفصل در یک زمان به نقطه B خواهد رسید که این امر مستلزم این بود که سرعت مفاصل در یک ضرایب مناسب ضرب شود. بنابراین بازو در عرض ۴ ثانیه به نقطه B دست خواهد یافت. هر چند که فواصل طی شده در این حالت بسیار شبیه به یکدیگر می‌باشند، اما مسیر طی شده همچنان ناموزون می‌باشد. توجه نمایید که هر دو این مسیرها در بیان مفصلی طراحی شده‌اند و تنها محاسبات انجام شده، حل معادله سینماتیک معکوس در حالت‌های نشان داده شده در شکل ۳-۸ و ضرایب سرعت مفاصل می‌باشند.



شکل ۳-۸

در آخر فرض کنید که بازو می‌بایست از نقطه A تا نقطه B از مسیر خاصی حرکت نماید این مسیر خاص، یک خط مستقیم از نقطه A به نقطه B می‌باشد، که در شکل ۳-۹ نشان داده شده است. همانطور که مشاهده می‌نمایید این مسیر به چهار قسمت تقسیم شده است یعنی بازو می‌بایست دستیابی به نقطه B از سه نقطه میانی گذر نماید. در این حالت معادلات سینماتیک معکوس می‌بایست علاوه بر نقطه B برای نقاط میانی نیز محاسبه شود. متغیرهای مفصلی و کارتیزین این نقاط، با این فرض که طول بازوها ۱۰ سانتیمتر می‌باشد، در جدول ۳-۱ آورده شده‌اند. هر چند که مسیر یک خط مستقیم می‌باشد، اما متغیرهای مفصلی به صورت یکنواخت تغییر نمی‌کنند. آشکار است که در فاصله میان نقاط



شکل ۳-۹

ربات مسیر مستقیمی را طی نمی کند، بنابراین به منظور دقت بیشتر می بایست نقاط بیشتری را در طول مسیر انتخاب نمود، که منجر به حل معادلات سینماتیک معکوس بیشتری می شود که مطلوب نمی باشد. در نتیجه می بایست تعاملی مابین تعداد نقاط و دقت به وجود آید.

جدول ۳-۱

Point	Cartesian values $\begin{pmatrix} x \\ y \end{pmatrix}$	Joint values $\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{y}{x}\right) - \cos^{-1}\left(\frac{0.5 \vec{P}_i }{a}\right) \\ 2\cos^{-1}\left(\frac{0.5 \vec{P}_i }{a}\right) \end{pmatrix}$
A	$\begin{pmatrix} 17.06 \\ 9.85 \end{pmatrix}$	$\begin{pmatrix} 20^\circ \\ 20^\circ \end{pmatrix}$
P1	$\begin{pmatrix} 12.795 \\ 11.218 \end{pmatrix}$	$\approx \begin{pmatrix} 10^\circ \\ 63^\circ \end{pmatrix}$
P2	$\begin{pmatrix} 8.53 \\ 12.586 \end{pmatrix}$	$\approx \begin{pmatrix} 15^\circ \\ 81^\circ \end{pmatrix}$
P3	$\begin{pmatrix} 4.265 \\ 13.953 \end{pmatrix}$	$\approx \begin{pmatrix} 30^\circ \\ 86^\circ \end{pmatrix}$
B	$\begin{pmatrix} 0 \\ 15.32 \end{pmatrix}$	$\begin{pmatrix} 50^\circ \\ 80^\circ \end{pmatrix}$

همانطور که تغییرات متغیرهای مفصلی در جدول ۳-۱ نشان می دهد مفاصل می بایست به صورت لحظه ای شتاب گیرند تا بازو را در موقعیت های مطلوب قرار دهند که در نقاط در دو نقطه ای ابتدایی مسیر شدت بیشتری دارد. که این عمل می تواند منجر به آسیب جدی به مفاصل و محرک های آنها شود. در غیر این صورت نیز ربات در مسیر دیگری که مطلوب ما نیست حرکت خواهد نمود. جهت حل این مشکل یا بهبود آن می بایست نقاط دیگری را در طول این خط راست انتخاب نمود.

۵-۳- آموزش ربات

در صنعت معمولاً ربات ها عملیات تکراری انجام می دهند، عملیاتی همچون رنگ پاشی، دریل کاری، جابجایی قطعات و ... ، در این عملیات مسیر مشخصی توسط ربات پشت سر هم پیموده می شود. در این صورت ربات ها را آموزش می دهند تا در مسیری دلخواه حرکت کنند. به این منظور ترمزهایی ربات را خلاص می کنند و ربات را از نقطه شروع مسیر به حرکت در می آورند و در طول مسیر متغیرهای

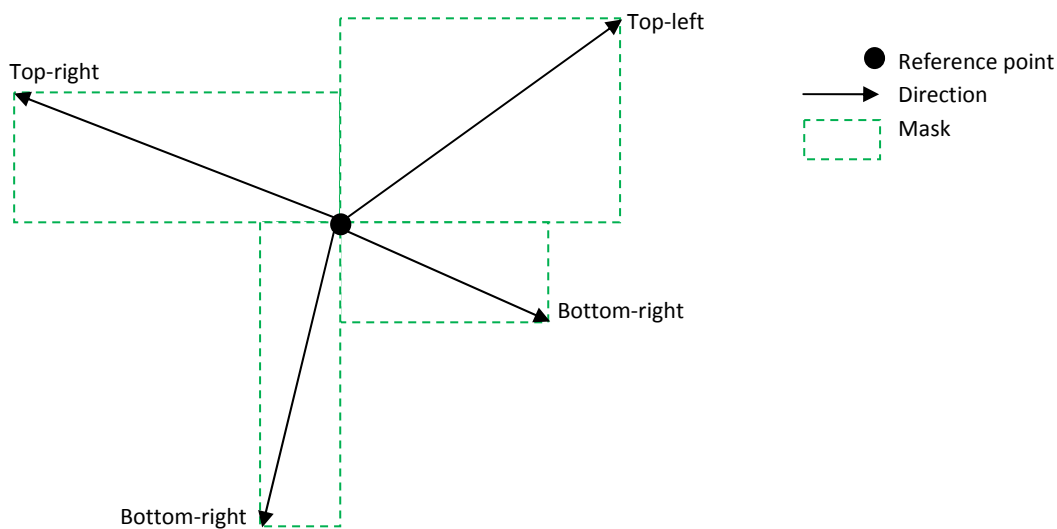
مفصلی را هر چند ثانیه یک بار ضبط می‌کنند تا به نقطه نهایی برسند. سپس ربات را در مسیر دلخواه برمی‌گرداند تا به نقطه شروع برسد در این حالت نیز عملیات ضبط کردن متغیرهای مفصلی انجام می‌گیرد. پس از آن ترمزها را فعال می‌نمایند و ربات را برای حرکت اتومات آماده می‌سازند. در این حالت ربات می‌تواند شروع به حرکت کرده و پشت سر هم به نقاط ضبط شده دست یابد. البته این نوع آموزش ربات مخصوص ربات‌هایی است که کاربر (آموزش‌دهنده) بتواند آن‌ها را به راحتی حرکت دهد. در طول آموزش باید دقت شود که ربات با دقت خوبی در مسیر مورد نظر حرکت داده می‌شود به طوری که عملیات مورد نظر را با کیفیت انجام دهد. آشکار است که این تکنیک بسیار ساده و پرکاربرد می‌باشد و حجم محاسبات کمی دارد و دیگر نیازی به محاسبه معادلات سینماتیک معکوس نمی‌باشد چرا که در هر لحظه متغیرهای شناخته شده ربات ضبط می‌شود. در واقع به این معنی است که اطلاعات از فیدبک موتورها یا فیدبک متغیرهای مفصلی خوانده و ضبط می‌شود.

۶-۳- بیان ماسک

در این بخش می‌خواهیم بیان جدیدی از مسیر را ارائه دهیم که دارای مزایای زیادی نسبت به بیان مفصلی و بیان کارتیزین از مسیر می‌باشد. ما می‌خواهیم بیانی را معرفی کنیم که حجم محاسبات در آن همچون بیان مفصلی کم می‌باشد و همچون بیان کارتیزین دقیق می‌باشد. این بیان، بیان ماسک نام-گذاری شده است، چرا که اساس طراحی مسیر در آن با استفاده از ماسک انجام می‌شود که بعداً در این فصل توضیح خواهیم داد.

بر اساس اینکه محدوده کاری ربات چند بعدی است یک ماسک می‌تواند مستطیلی یا مکعب مستطیلی باشد. هر ماسک دارای یک نقطه به عنوان نقطه مرجع می‌باشد که یکی از گوشه‌های ماسک است. یک ماسک توسط نقطه مرجع و ابعاد آن در راستای محورهای x, y, z در دستگاه مختصات کارتیزین کاملاً مشخص می‌شود. یک بعد از یک ماسک گسسته می‌باشد و برای بیان اندازه آن می‌بایست

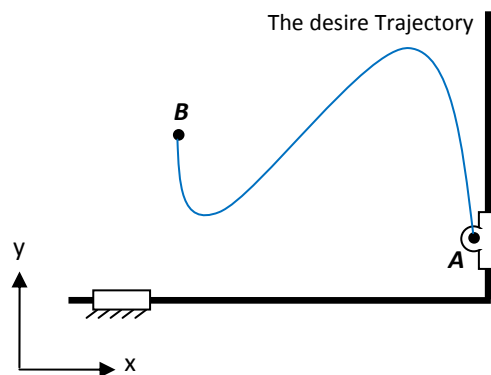
ابتدا قطعات اندازه را مشخص کرد. ماسک‌ها همگن می‌باشند به این معنی که قطعات اندازه در آن در هر راستا مساوی می‌باشد و کوچکترین واحد هر ماسک در ماسک‌های مستطیلی یک مربع به ابعاد اندازه قطعه و در ماسک‌های مکعب مستطیلی یک مکعب مستطیل به ابعاد اندازه قطعه است در صورتی که یک ماسک مکعب مستطیل شکل به ابعاد n داشته باشیم تعداد $4n(n-1)(n-2)$ ماسک حول یک نقطه مرجع ممکن است. همچنین اگر n ابعاد یک ماسک مستطیلی باشد تعداد $4n(n-1)$ ماسک حول یک نقطه مرجع ممکن می‌باشد. ماسکی با ابعاد 1×1 یا $1 \times 1 \times 1$ تعریف نمی‌شود، در واقع به بیان دیگر ماسکی با ابعاد اندازه قطعه وجود ندارد. شکل ۳-۱۰ چهار ماسک با ابعاد متفاوت حول یک نقطه مرجع که با یک نقطه مشکی نشان داده شده است، نمایش می‌دهد. در ماسک‌های دو بعدی، حول یک نقطه مرجع، چهار جهت متفاوت وجود دارد که بالا-راست، بالا-چپ، پایین-چپ و پایین-راست نامیده می‌شوند. همچنین در ماسک‌های سه بعدی، هشت جهت متفاوت حول یک نقطه مرجع وجود دارد. ربات می‌بایست در داخل ماسک در راستای قطر ماسک حرکت نماید. در این صورت، در یک ماسک، نقطه نهایی ربات از نقطه مرجع در راستای قطر شروع به حرکت کرده تا به نقطه گوشه‌ای مقابل برسد، که این نوع حرکت، حرکت



شکل ۳-۱۰

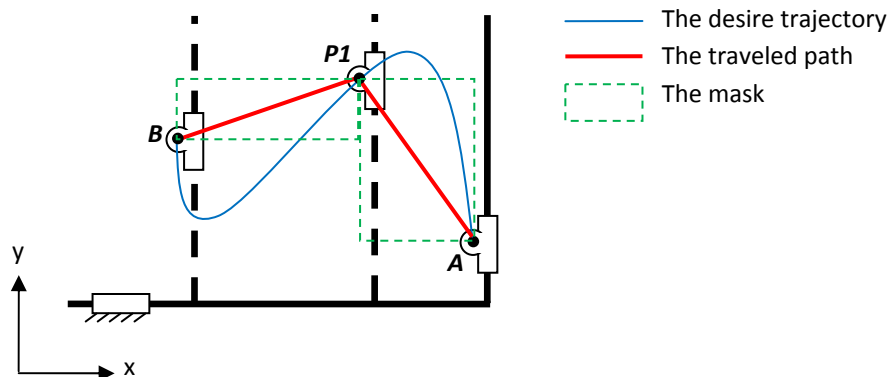
همه‌هنگ نامیده می‌شود. زمانی که یک ماسک بر روی یک مسیر مطلوب قرار می‌گیرد آن قسمت از مسیر که در داخل ماسک می‌باشد را به منظور آنالیز برش می‌دهد.

جهت انجام توضیحات تکمیلی، یک ربات کارتزین دو بعدی با دو مفصل کشویی را در نظر بگیرید. ربات مفروض یک مکانیزم دو درجه آزادی می‌باشد که فضای کاری آن یک صفحه‌ی دو بعدی است. یعنی ربات تنها می‌تواند به نقاطی در یک صفحه دسترسی یابد. شکل ۳-۱۱ ربات و مسیر مطلوبی را نمایش می‌دهد که ربات می‌بایست در راستای این مسیر نشان داده شده از نقطه A به نقطه B برسد. همچنین فرض کنید محرک‌های مفاصل ربات همسان بوده و دارای سرعت ماکزیمم مساوی می‌باشند.



شکل ۳-۱۱

حال نقطه P1 را بر روی مسیر در نظر بگیرید که ربات می‌بایست به این نقطه به طور کامل دست یابد. در این صورت، مسیر را می‌توان با دو ماسک طراحی نمود. ماسک‌ها و مسیر پیموده شده در شکل ۳-۱۲ نشان داده شده‌اند، که ماسک‌ها توسط خطوط خط‌چین و مسیر طی شده توسط خطوط مستقیم مشخص شده‌اند. توجه کنید که مسیر طی شده توسط ربات که توسط دو ماسک ساخته شده است، خطوط مستقیم می‌باشد که از پیکربندی ربات نتیجه می‌شود. همچنین توجه نمایید که در روش ماسک مفاصل می‌بایست در داخل ماسک همزمان شروع به حرکت کرده و همزمان در نقطه گوشه‌ای مقابل متوقف شوند، که خطوط مستقیم طی شده را نتیجه می‌دهد.

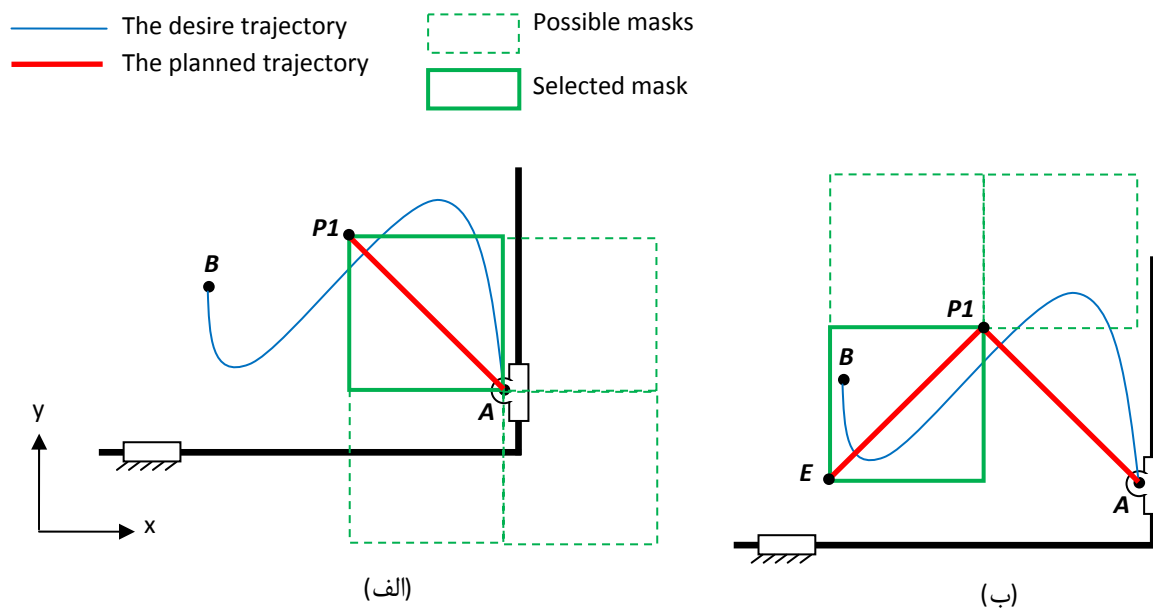


شکل ۳-۱۲

۱-۶-۳- بیان ماسک مربعی

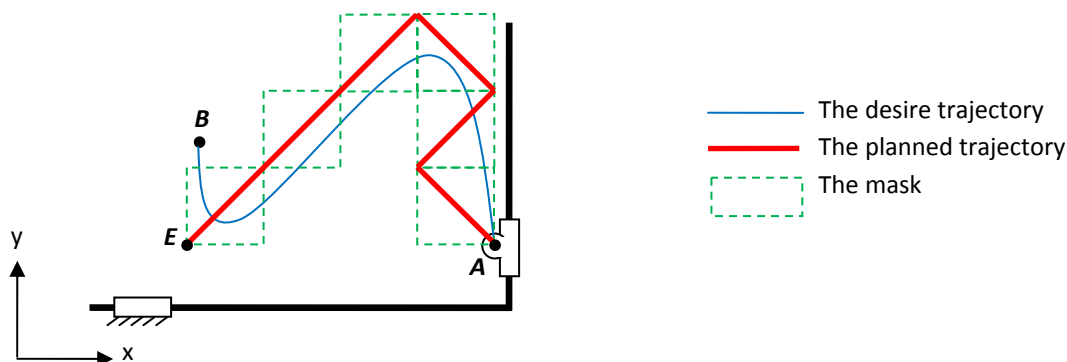
در بیان ماسک مربعی مسیر مطلوب توسط ماسک‌های مربعی طراحی می‌شود. در این بیان همه-ی ماسک‌ها هم‌اندازه بوده و اندازه ماسک در ابتدای حرکت توسط کاربر مشخص می‌شود. جهت طراحی مسیر توسط ماسک‌های مربعی ابتدا نقطه شروع مسیر را نقطه مرجع اولین ماسک قرار می‌دهیم. سپس چهار ماسک ممکن را روی مسیر قرار داده و ماسکی که دارای بیشترین قسمت از مسیر مطلوب می‌باشد جهت حرکت را مشخص می‌نماید. مفصل ربات می‌بایست در جهت مشخص شده هم‌زمان شروع به حرکت کرده و هم‌زمان در نقطه گوشه‌ای مقابل متوقف شوند. پس از آن نقطه گوشه‌ای مقابل، نقطه مرجع جدید برای ماسک بعدی در نظر گرفته می‌شود. عملیات تا زمانی ادامه داده می‌شود که نقطه گوشه‌ای ماسک و نقطه پایانی مسیر در داخل یک ماسک قرار گیرند. در این صورت نقطه نهایی ربات و نقطه پایانی مسیر با دقتی بیشتر از قطر ماسک نسبت به هم قرار می‌گیرند. شکل ۳-۱۳ نتیجه‌ی مثال شکل ۳-۱۱ را توسط روش ماسک مربعی نشان می‌دهد که فرآیند طراحی در زیر توضیح داده شده است. ابتدا چهار ماسک که نقطه مرجع آن بر نقطه A منطبق است روی مسیر قرار می‌گیرد. ماسکی که دارای بیشترین قسمت از مسیر است یعنی ماسک سمت چپ- بالا جهت حرکت ربات را نشان می‌دهد. سپس

ربات از نقطه A در راستای قطر ماسک شروع به حرکت کرده تا به نقطه گوشه‌ای مقابل دست یابد. پس از رسیدن ربات به نقطه P1 (نقطه گوشه‌ای مقابل در ماسک اول) سه ماسک دیگر با نقطه مرجع P1 بر روی مسیر قرار می‌گیرند. ماسکی که دارای بیشترین قسمت از مسیر است جهت مسیر را نشان می‌دهد، سپس ربات در راستای قطر ماسک حرکت نموده تا به نقطه E برسد. در این حالت چون نقطه پایانی مسیر و نقطه E در داخل یک ماسک قرار دارند عملیات متوقف شده و نقطه نهایی ربات با دقت بیشتر از اندازه قطر ماسک در نزدیکی نقطه B قرار گرفته است. همانطور که می‌بینید، در این روش معادلات سینماتیک معکوس، با فرض اینکه ربات در نقطه شروع قرار دارد، تنها برای دو نقطه P1 و E محاسبه شده است.



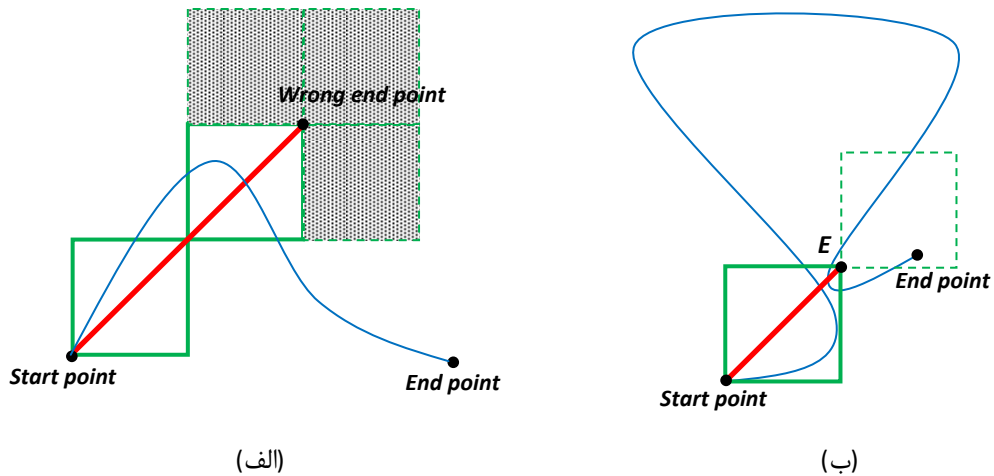
شکل ۳-۱۳

شکل ۳-۱۴ مسیر طراحی شده دیگری را نمایش می‌دهد که در آن اندازه ماسک‌ها کوچک‌تر از حالت قبل است. همانطور که می‌بینید، مسیر طراحی شده در این مثال توسط شش ماسک ساخته شده است. در مقایسه با مثال قبل، در این مثال دقت طراحی افزایش یافته، در حالیکه معادلات سینماتیک معکوس برای شش نقطه محاسبه شده است. واضح است که، دقت مسیر طراحی شده و تعداد معادلات سینماتیک معکوس محاسبه شده در این روش تحت تاثیر اندازه ماسک می‌باشد. توجه کنید در بیان ماسک‌های مربعی نیازی نیست که نقطه گوشه‌ای مقابل نقطه مرجع ماسک منطبق بر مسیر باشد. به این دلیل که در قسمت‌های بعد به تفصیل به آن می‌پردازیم، ضمانتی جهت رسیدن ربات به نقطه نهایی مسیر



شکل ۳-۱۴

مطلوب وجود ندارد. دو حالت اشتباه ممکن است روی دهد زمانی که نقطه گوشه‌ای منطبق بر مسیر نباشد یا به عبارت دیگر نقاط مرجع ماسک‌ها منطبق بر مسیر نباشند. در حالت اشتباه اول ربات طی یک مسیر اشتباه به نقطه پایانی مسیر دست می‌یابد. یعنی ربات بدون پیمودن مسیر از پیش تعریف شده به صورت اتفاقی به نقطه نهایی مسیر دست می‌یابد. در حالت اشتباه دوم ربات به نقطه نهایی مسیر دست نمی‌یابد. این دو حالت اشتباه به ترتیب در شکل‌های ۳-۱۵ الف و ۳-۱۵ ب نمایش داده شده‌اند. این حالت‌ها دلالت بر این دارد که دستیابی به نقطه نهایی بسیار اثرپذیر از اندازه ماسک می‌باشد.

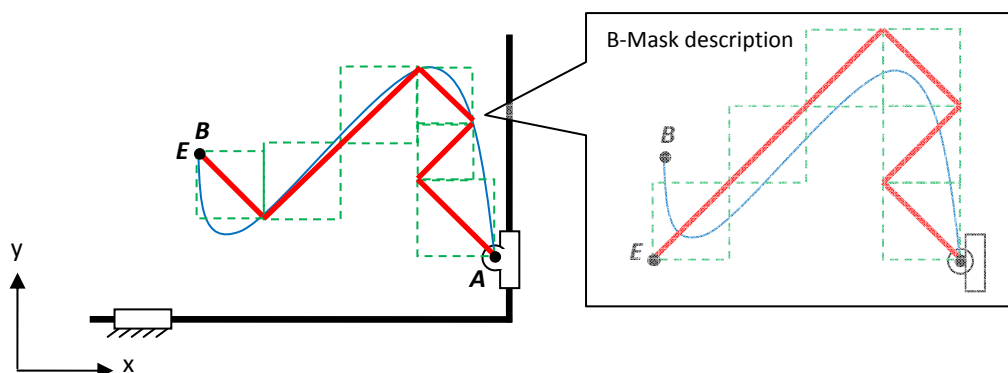


شکل ۳-۱۵

۲-۶-۳- بیان ماسک مربعی چسبنده

در این بیان ماسک‌ها به صورت ماسک‌های مربعی تعریف شده در بخش قبل می‌باشند. با این تفاوت که، در این حالت اندازه ماسک قابل تغییر می‌باشد. اندازه ماسک در تمامی ابعاد به یک اندازه می‌بایست تغییر نماید تا ماسک باز هم مربعی باقی بماند. اندازه ماسک در صورتی تغییر می‌نماید که قطر ماسک و مسیر از پیش تعریف شده همدیگر را قطع نمایند که در این صورت نقطه تقاطع نقطه گوشه‌ای ماسک قرار داده می‌شود و اندازه ماسک را تعیین می‌نماید. مزیت این بیان نسبت به بیان مربعی در این است که دقت مسیر طراحی شده افزایش می‌یابد و همچنین احتمال وقوع حالت‌های اشتباه ذکر شده در بخش قبل کاهش می‌یابد. شکل ۳-۱۶ مسیر طراحی شده توسط بیان ماسک مربعی چسبنده را نمایش می‌دهد. اندازه از پیش تعیین شده ماسک در این مثال مشابه اندازه از پیش تعیین شده در مثال شکل ۳-۱۴ می‌باشد. شکل ۳-۱۴ جهت مقایسه در کنار نتیجه حاصله از ماسک مربعی چسبنده در شکل ۳-۱۶ آمده است. افزایش دقت کاملاً محسوس می‌باشد. همانطور که مشاهده می‌کنید، سه ماسک دارای اندازه کوچک‌تر از اندازه از پیش تعیین شده برای ماسک می‌باشند که از متقاطع بودن قطر این ماسک‌ها و مسیر مطلوب نتیجه شده‌اند. تعداد معادلات سینماتیک معکوس محاسبه شده در این بیان تفاوت چندانی

نسبت به تعداد تعداد معادلات سینماتیک معکوس محاسبه شده در بیان ماسک مربعی ندارد. حالت‌های اشتباه نیز در این بیان مشابه بیان ماسک مربعی می‌باشد با این تفاوت که احتمال وقوع آن‌ها در این بیان کاهش یافته است.



شکل ۳-۱۶

۳-۶-۳- بیان ماسک مستطیلی در ماسک مربعی

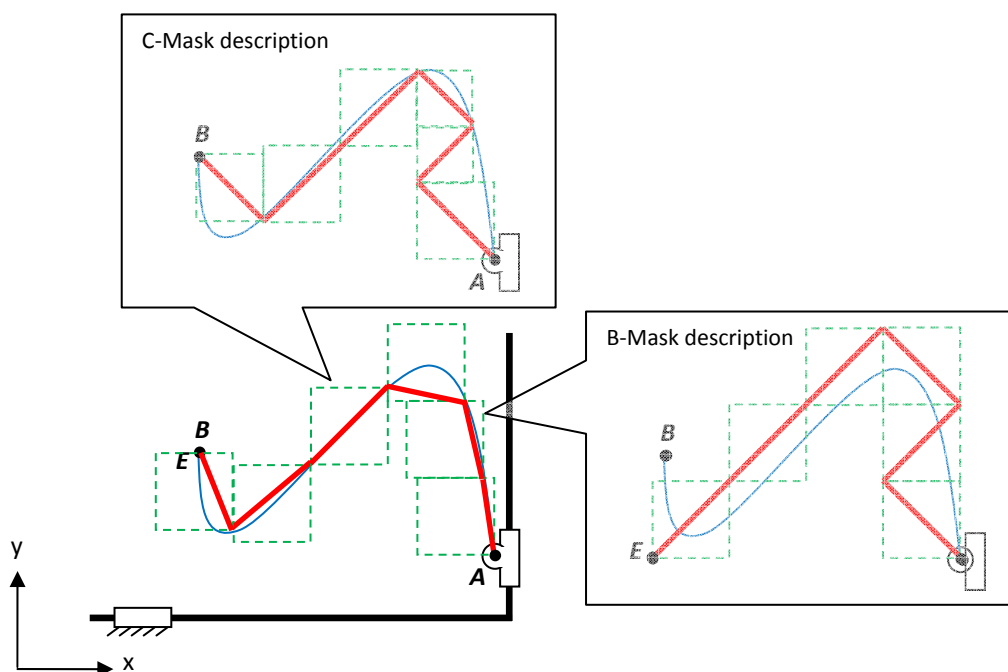
در این حالت ماسک‌ها می‌توانند به صورت مستطیل باشند. روند تشکیل این نوع ماسک به صورت زیر می‌باشد.

۱- یک ماسک مربعی با اندازه از پیش تعیین شده روی مسیر قرار می‌گیرد.

۲- محل تقاطع مسیر مطلوب و مرز ماسک نقطه گوشه‌ای مقابل را تشکیل می‌دهد.

در این صورت ماسک‌های مستطیلی در یک راستا دارای اندازه‌ای معادل اندازه از پیش تعیین شده ماسک مربعی می‌باشند. در این روش همیشه نقطه گوشه‌ای بر مسیر منطبق می‌باشد به عبارت دیگر نقاط مرجع ماسک‌ها بر مسیر از پیش تعریف شده منطبق هستند. این بیان نسبت به دو بیان قبلی بسیار دقیق‌تر می‌باشد. در این حالت نیز هر چند نقاط مرجع ماسک‌ها منطبق بر مسیر است اما ضمانتی نسبت به رسیدن به نقطه نهایی مسیر در مسیری درست وجود ندارد. در واقع شرط انطباق نقاط مرجع بر مسیر

یک شرط لازم جهت رسیدن به نقطه پایانی در مسیر درست می‌باشد. شکل ۳-۱۷ نتیجه طراحی مسیر بر اساس بیان ماسک مستطیلی در ماسک مربعی را نمایش می‌دهد. همچنین دو مسیر طراحی شده بر اساس دو بیان قبلی نیز در این شکل نمایش داده شده‌اند. همانطور که می‌بینید دقت مسیر طراحی شده شدیداً افزایش یافته است. اندازه پیش فرض ماسک‌ها نیز در این مثال مشابه دو مثال تکرار شده می‌باشد. در این بیان نیز تعداد معادلات سینماتیک معکوس تفاوت چندانی نسبت به تعداد معادلات سینماتیک معکوس در بیان قبلی ندارد. همچنین در این بیان نیز مشابه بیان‌های قبل دو حالت اشتباه ذکر شده روی می‌دهند اما احتمال وقوع این حالت‌های اشتباه در این بیان بسیار کاهش یافته است.

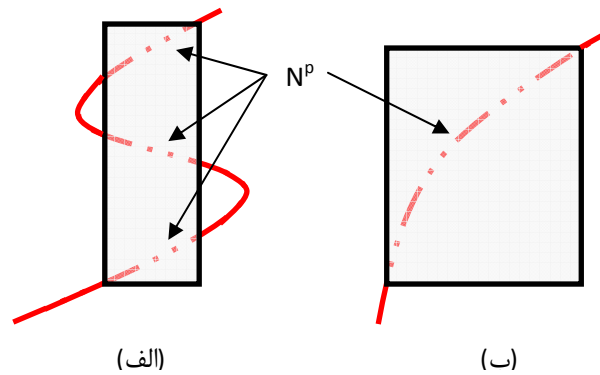


شکل ۳-۱۷

۴-۶-۳- بیان ماسک تطبیقی

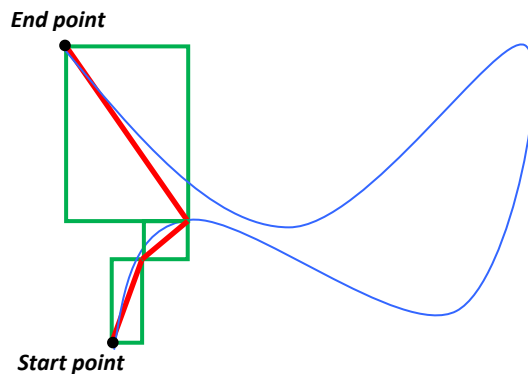
در این روش ماسک‌ها مستطیلی بوده و اندازه آن متغیر می‌باشد. روند تعیین اندازه ماسک‌ها به صورت تطبیقی با معیار شباهت انجام می‌شود که توضیح خواهیم داد. در این روش تضمین خواهیم نمود که ربات مسیر از پیش تعریف شده را به درستی طی نموده و به نقطه نهایی دست خواهد یافت. به این منظور تعاریف جدیدی انجام خواهد شد تا با استفاده از آن‌ها شرایطی که منجر به اشتباه می‌شود را حذف نماییم. همچنین می‌خواهیم یک روش قابل تنظیم را ارائه دهیم تا کاربر قادر باشد دقت مسیر طراحی شده را تغییر دهد. بنابراین، دقت می‌بایست به گونه‌ای تعریف شود که به راحتی قابل محاسبه باشد و به راحتی کاربر بتواند آن را تغییر دهد. در این راستا مفاهیم دقت و N_p را تعریف خواهیم نمود. همچنین توجه نمایید که تمامی نقاط مرجع می‌بایست بر مسیر منطبق باشند. در حالت کلی، بر اساس این روش، ربات‌ها قادر خواهند بود با سرعت و دقت بالا وظایف خود را انجام دهند. معیار دقت با محاسبه واریانس خطای محلی مابین مسیر طراحی شده و مسیر از پیش تعریف شده بدست می‌آید. بنابراین معیار طراحی تطبیقی این روش، واریانس‌های محلی خواهد بود. همچنین این روش قابل پیاده‌سازی بر روی هر ربات با هر پیکربندی می‌باشد.

حرف N_p تعداد قطعات مجزا از مسیر از پیش تعریف شده در محدود ماسک تعریف شده است. شکل ۳-۱۸ الف حالتی را نمایش می‌دهد که در آن سه قطعه جدا از هم از مسیر در داخل یک ماسک قرار دارد، در این صورت مقدار N_p برای ماسک مورد نظر سه می‌باشد. همچنین در شکل ۳-۱۸ ب، N_p مساوی یک می‌باشد. به منظور این که ربات بتواند مسیر را با موفقیت طی نماید N_p می‌بایست در تمام ماسک‌ها مقدار یک داشته باشد. توجه نمایید که دو حالت اشتباه در مثالهای قبلی زمانی پیش می‌آیند که شرط انطباق و N_p برآورده نشده‌اند. در صورتی که شرط انطباق نیز برآورده شود تنها احتمال رخ دادن این اشتباهات کمتر خواهد شد و این اشتباهات به طور کامل برطرف نمی‌شود. شکل ۳-۱۹ الف و ۳-۱۹



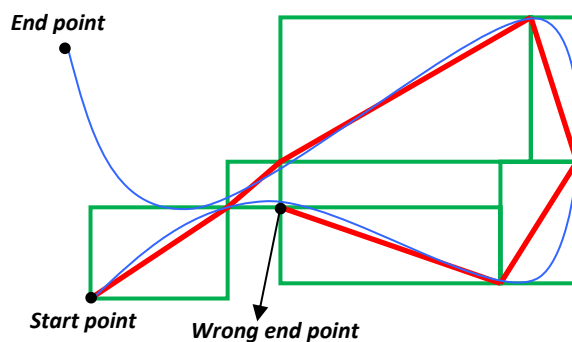
شکل ۳-۱۸

ب حالاتی را نمایش می‌دهند که در آن‌ها شرط انطباق در طراحی مسیر رعایت شده است اما N_p در بعضی ماسک‌ها یک نمی‌باشد. همانطور که ملاحظه می‌کنید اشتباه در ماسک‌هایی روی می‌دهد که در آن‌ها N_p مخالف یک است. در شکل ۳-۱۹ الف ربات به نقطه نهایی طی مسیر اشتباه دست یافته است. توجه نمایید که N_p زمانی یک خواهد بود که نقطه مرجع توسط یک قطعه از مسیر از پیش تعریف شده در داخل ماسک به نقطه گوشه‌ای مقابل متصل باشد. در شکل ۳-۱۹ ب ربات به نقطه نهایی اشتباه دست یافته است.



(الف)

شکل ۳-۱۹



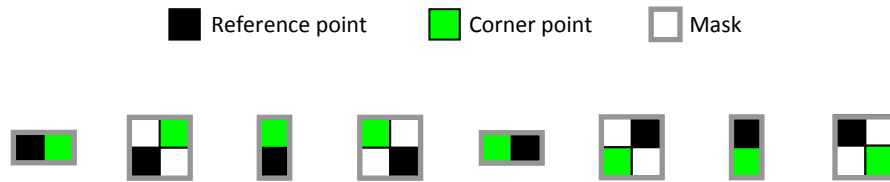
(ب)

شکل ۳-۱۹

ماسک و N_p مشخص می‌نمایند که یک مسیر چگونه می‌بایست طراحی شود. روند طراحی یک مسیر بر اساس مسیر از پیش تعریف شده به صورت زیر می‌باشد.

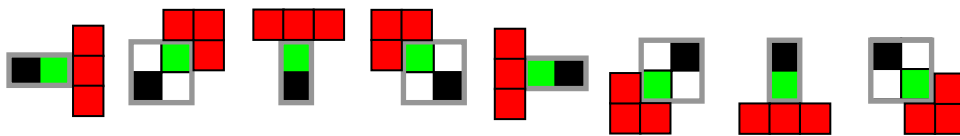
۱- یک ماسک که نقطه مرجع آن منطبق بر نقطه شروع مسیر است بر روی مسیر قرار می‌گیرد. اندازه ابتدایی ماسک‌ها 1×2 یا 2×1 می‌تواند باشد. که 1×2 به عنوان اندازه شروع انتخاب شده است که نقطه مرجع نقطه سمت چپ ماسک می‌باشد. اندازه ماسک به صورت پادساعتگرد به شکل 1×2 ، 2×2 و 1×2 تغییر خواهد نمود. پس از آن نقطه مرجع به نقطه پایین سمت راست تبدیل خواهد شد و اندازه ماسک دوباره به صورت پادساعتگرد تغییر می‌نماید و این عملیات تکرار خواهد یافت تا به یک ماسک با اندازه 2×2 برسیم که نقطه مرجع آن نقطه سمت چپ بالا باشد. این روند در شکل ۳-۲۰ نمایش داده شده است. همانطور که مشاهده می‌نمایید نقطه مرجع در ماسک‌ها نیز به صورت پادساعتگرد جابه‌جا می‌شود.

۲- طی فرآیند بالا در صورتی که نقطه گوشه‌ای مقابل بر روی مسیر قرار گیرد عملیات متوقف شده و ماسک ابتدایی بدست خواهد آمد. در این حالت جهت افزایش اندازه ماسک مشخص شده و فرآیند به صورت مقابل ادامه می‌یابد.



شکل ۳-۲۰

۳- شکل ۳-۲۱ انتخاب‌های ممکن جهت افزایش اندازه ماسک را نمایش می‌دهد در صورتی که یک از سه انتخاب در هر ماسک بر روی مسیر قرار بگیرد، اندازه ماسک در آن جهت افزایش می‌یابد.



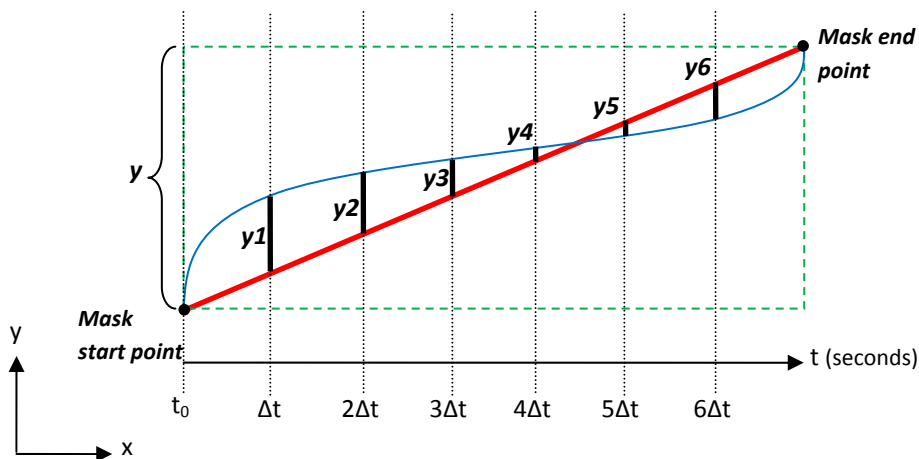
شکل ۳-۲۱

۴- این روند ادامه می‌یابد تا زمانی که نقطه جدیدی یافت نشود. توجه نمایید که در طول فرآیند اندازه ماسک نمی‌تواند در هیچ راستایی کوچک شود. این فرآیند تضمین می‌نماید که N_p همواره یک باقی می‌ماند و همچنین نقطه گوشه‌ای مقابل همیشه بر مسیر از پیش تعریف شده منطبق است. حال می‌توانیم ضمانت نماییم که بر اساس فرآیند بالا مسیر طراحی شده همواره به نقطه نهایی مسیر از پیش تعریف شده در مسیر درست دست می‌یابد. فرآیند بالا زمانی متوقف می‌شود که نقطه مرجع آخرین ماسک بر مسیر منطبق باشد.

نقطه متوقف کننده فرآیند ساخته شدن ماسک وابسته به دو عامل می‌باشد، N_p و معیار دقت. حال به تعریف دقت و چگونگی محاسبه آن می‌پردازیم. همانطور که عنوان کرده بودیم، روش ما بر اساس فرآیند تطبیقی استوار است. این روش یک روش قابل تنظیم می‌باشد. در نتیجه معیار دقت، معیاری

جهت فرآیند تطبیقی و یک کمیت جهت تنظیم طراحی مسیر می‌باشد. دقت به صورت شباهت مسیر از پیش تعیین شده و مسیر طراحی شده در داخل یک ماسک تعریف می‌شود. جهت محاسبه دقت، اختلاف مابین این دو مسیر می‌بایست محاسبه شود. واریانس این اختلاف معیاری از دقت می‌باشد.

خطای میان مسیر از پیش تعیین شده و مسیر طراحی شده می‌تواند در راستای هر یک از محورهای دستگاه کارتزین محاسبه شود. همانطور که در شکل ۳-۲۲ نشان داده شده است، در یک ماسک دو بعدی محور x را مبنا قرار داده و خطا را در راستای محور y محاسبه کرده‌ایم. خطا برای نقاط



شکل ۳-۲۲

مرجع و گوشه‌ای ماسک محاسبه نمی‌شود زیرا در این روش این نقاط حتماً بر مسیر منطبق می‌باشند. میزان خطا می‌بایست در هر قطعه اندازه از محور x محاسبه شود. علامت خطا مهم نمی‌باشد زیرا به دنبال محاسبه واریانس خطا می‌باشیم. شکل ۳-۲۲ نمایش‌دهنده این فرآیند می‌باشد. واریانس طبق معادله ۳-۱ محاسبه می‌شود. حرف i بیانگر قطعات اندازه در راستای محور x می‌باشد که فاصله قطعه از نقطه مرجع را عنوان می‌نماید و y اندازه ماسک در راستای محور y می‌باشد. همچنین y_i ها تعداد قطعات اندازه مابین مسیر واقعی و مسیر طراحی شده در قطعه i می‌باشد. واریانس بیانگر این واقعیت می‌باشد که مسیر تعریف شده و مسیر طراحی شده چقدر به هم نزدیک می‌باشند. به عبارت دیگر واریانس معیاری برای روی هم-

افتادگی می‌باشد. در داخل یک ماسک، اگر واریانس صفر باشد. مسیر واقعی و مسیر طراحی شده در قطعات اندازه بر هم منطبق می‌باشند که در این حالت بالاترین درجه روی هم‌افتادگی برآورده می‌شود. این بدان معنی نیست که مسیر طراحی شده کاملاً منطبق بر مسیر واقعی می‌باشد. چرا که در نقاط غر از قطعات اندازه واریانس محاسبه نمی‌شود. در نتیجه درجه روی هم‌افتادگی دو مسیر توسط واریانس تنظیم می‌شود. زمانی که واریانس صفر می‌باشد یعنی در بالاترین درجه روی هم‌افتادگی ربات کمترین سرعت را در پیمایش مسیر دارا می‌باشد. در واقع هر چه واریانس به سمت صفر نزدیک شود سرعت پیمایش مسیر کمتر خواهد شد. در این حالت معادله سینماتیک معکوس برای بیشترین نقاط از مسیر محاسبه خواهد شد. روش پیشنهادی یک مسیر را با توجه به مسیری از پیش تعریف شده بر اساس معیار روی هم‌افتادگی طراحی می‌نماید. با استفاده از این روش تعداد معادلات سینماتیک معکوس به شدت کاهش یافته، معادلاتی که بسیار زمان‌بر و حافظه‌بر می‌باشند. تغییر در اندازه ماسک منجر به تغییر در خطا و واریانس می‌شود. بنابراین اندازه ماسک علاوه بر N_p از واریانس نیز تاثیر می‌پذیرد. واریانس در طول افزایش اندازه ماسک می‌تواند کاهش یا افزایش یابد. بنابراین مطلوب است که ماسک در صورتی افزایش اندازه دهد که علاوه بر یک ماندن N_p ، واریانس محاسبه شده در ماسک نیز کمتر از حد قابل قبول باشد. در غیر اینصورت اندازه ماسک ثابت شده و ماسک مطلوب بدست می‌آید.

$$var = \sum_{i=1}^n \frac{(y_i)^2}{n} \quad (1-3)$$

در بیان ماسک تطبیقی دست‌یابی ربات به نقطه نهایی مسیر تضمین می‌شود. به این معنی که ربات در طول مسیری صحیح در نقطه پایانی متوقف می‌شود. همچنین معادلات سینماتیک معکوس نسبت به بیان‌های غیر ماسکی بسیار کاهش یافته و این معادلات تنها برای نقاط گوشه‌ای ماسک‌ها محاسبه می‌شوند. به عنوان مزیت دیگری از بیان ماسک، تمامی مسیرهای طراحی شده در این بیان که

دارای معیار دقت مشابه می‌باشند، یکسان بوده و همچنین مسیرهای زمان‌دار (تراژکتوری) نیز مشابه مسیرهای بدون زمان طراحی می‌شوند.

۷-۳- بیان ماسک در مقایسه با بیان مفصلی و کارت‌زین

بیان ماسک دارای مزایای بسیاری نسبت به بیان مفصلی و بیان کارت‌زین می‌باشد. بیان ماسک دارای حجم محاسبات کمی همچون بیان مفصلی و دارای دقت بالا همچون بیان کارت‌زین می‌باشد. همچنین مسیرهای طراحی شده در بیان ماسک منجر به مسیرهای یکسانی در طراحی مسیرهای زمان‌دار و بدون زمان می‌شود، در حالیکه در بیان‌های دیگری مسیرها غیر قابل پیش‌بینی می‌باشند. در مسیرهای زمان‌دار، بیان‌های مفصلی و کارت‌زین با تغییر در زمان‌بندی مسیر منجر به مسیرهای متفاوتی می‌شوند، در حالیکه در بیان ماسک در صورتی که دقت و یا اندازه پیش فرض ماسک ثابت باشد، مسیرهای طراحی شده در زمان‌بندی‌های متفاوت یکسان می‌باشند.

فصل چہارم

۱-۴- مقدمه

در این فصل شبکه‌های پتری را معرفی خواهیم نمود، شبکه پتری یک ابزار مدل‌سازی ریاضی و گرافیکی می‌باشد که قابل کاربرد در بسیاری از سیستم‌هاست. نگاه گسسته‌رخداد به سیستم‌ها نگاهی قابل قبول می‌باشد. از این دیدگاه فرآیند ساخت ماسک‌ها و کنترل حرکت ربات‌ها را توسط شبکه پتری که یک ابزار قدرتمند مدل‌سازی سیستم‌های گسسته‌رخداد می‌باشد، بررسی خواهیم نمود. در این فصل ابتدا، سیستم‌های گسسته رخداد را بررسی خواهیم نمود. سپس، شبکه‌های پتری را معرفی خواهیم کرد. پس از آن، به منظور مدل‌سازی روش پیشنهادی، از شبکه‌های پتری استفاده خواهیم نمود. در آخر فرآیند ساخت ماسک‌ها و کنترل حرکت ربات را توسط شبکه پتری پیاده‌سازی می‌نماییم.

۲-۴- سیستم‌های گسسته رخداد

یک سیستم گسسته رخداد، سیستمی پویا است که با توجه به رخدادهای ناگهانی از حالتی به حالت دیگر گذار می‌کند. امروزه نگاه گسسته رخداد به سیستم، در حوزه‌های زیادی وارد شده است. حوزه‌هایی همچون تولید، رباتیک، کامپیوتر، شبکه‌های ارتباطی و غیره. رخدادها در سیستم‌های گسسته رخداد می‌توانند در هر لحظه از زمان و در فاصله‌های اتفاقی روی دهند و در حالت کلی رخدادها دارای محدودیت زمانی نمی‌باشند. به عنوان مثال یک رخداد می‌تواند ورود و خروج از یک صف، کامل شدن یک پروسه یا خرابی یک دستگاه در سیستم‌های تولید، رسیدن یک بسته در سیستم‌های ارتباطی و اغتشاش یا تغییر در ورودی مرجع در سیستم‌های کنترلی باشد.

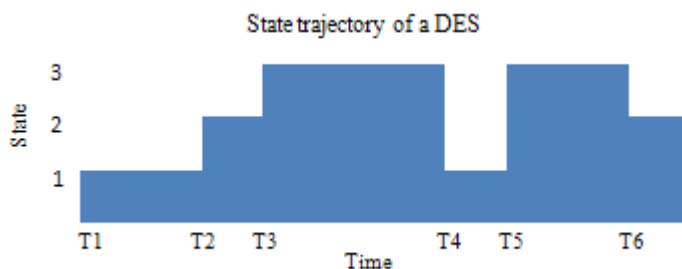
در گذشته بیان گسسته رخداد یک سیستم، به دلیل سادگی به چند مورد کوچک محدود می‌شد. اما امروزه با پیشرفت تکنولوژی و استفاده گسترده از کامپیوتر سیستم‌های پیچیده نیز به صورت گسسته رخداد بیان، مدل، آنالیز و طراحی شده‌اند.

۳-۴- مدل سازی سیستم های گسسته رخداد

یک سیستم گسسته رخداد می تواند به صورت یک سیستم پویا با فضای حالت گسسته و نمودار حالت پیوسته تکه ای مشخص، تعریف شود که فاصله زمانی در گذار از حالت ها در حالت کلی غیر قابل پیش بینی می باشد.

گذار از حالت در یک سیستم گسسته رخداد یک رخداد نامیده می شود. رخدادها می توانند با استفاده از حروف الفبا یا برچسب هایی که آن ها را بیان می کنند، علامت گذاری شوند. برای مثال در یک پروسه تولید، رخدادها می توانند با "قطعه مورد قبول قرار گرفت"، "قطعه دریل کاری شد"، "دستگاه تعمیر شد" و ... بیان شوند.

در شکل ۱، یک نمودار حالت مثال زده شده است. همانطور که مشاهده می شود ماشین مربوطه دارای سه حالت ۱، ۲ و ۳ می باشد و در حالت اولیه ۱ نشان داده شده است که در زمان t_2 به حالت ۲ گذار کرده است و ...



شکل ۴-۱

مدل های گسسته رخداد زیادی، به دلیل ورود نگاه گسسته رخداد به زمینه های متنوع، توسعه یافته اند. در واقع گونه های رفتاری متنوع در زمینه های گوناگون منجر به بیان و توسعه این مدل ها شده است. دو دسته کلی از مدل های سیستم های گسسته رخداد.

۱. مدل‌های زمان

۲. مدل‌های منطقی

نامیده می‌شوند. بیان این مدل‌ها در نوع برخورد با عنصر زمان متفاوت می‌باشد و این نوع برخورد منجر به بیان و توسعه این مدل‌ها شده است.

۱-۳-۴- مدل‌های زمانی

نمودار گذار حالت یک سیستم، یک بیان رفتاری بسیار مهم در انواع وسیعی از سیستم‌ها می‌باشد، به این معنی که، مهم است بدانیم یک سیستم در چه زمانی از یک حالت به حالت دیگر گذار کرده است یا در چه مدت زمانی در یک حالت باقی مانده است. در نتیجه رفتار زمانی در سیستم‌ها منجر به بیان مدل‌هایی شده است که مدل‌های زمانی نامیده می‌شوند. مدل‌های زمانی به دو دسته وسیع تصادفی و غیر تصادفی کلاس‌بندی می‌شوند.

مدل‌های تصادفی و غیر تصادفی زمان در سیستم‌های گسسته رخداد مربوط به نوع شناخت ما از زمان در این سیستم‌ها می‌باشد. زمانی که زمان رخداد هر اتفاق کاملاً شناخته شده باشد، مدل یک مدل غیر تصادفی و زمانی که خواص آماری زمان، الگویی از بیان زمان باشد، مدل یک مدل تصادفی است.

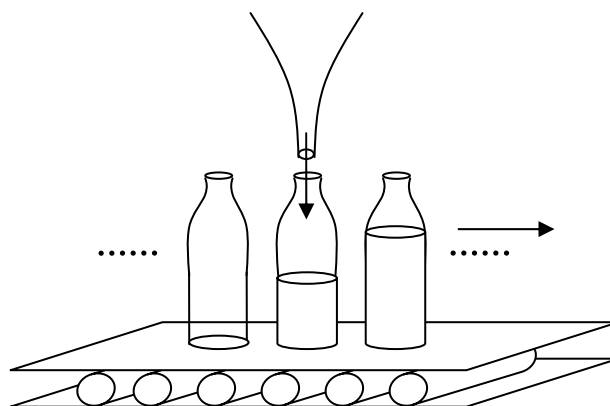
۲-۳-۴- مدل‌های منطقی

دسته دیگری از مدل‌ها با حذف عنصر زمان در بیان اتفاق‌ها و سیستم به وجود می‌آید. این نوع مدل‌ها که تنها به ترتیب رخدادها می‌پردازند، مدل‌های منطقی نامیده می‌شوند. یک مدل منطقی، رخدادها را طبق ترتیب روی‌دادشان، فهرست می‌کند. در نتیجه بیان رفتار سیستم به رشته‌ای از رخدادها کاهش می‌یابد. این ساده‌سازی در صورتی قابل انجام است که دینامیک رخدادها مستقل از بیان زمانی

آنها باشد. زمانی که یک ماشین توسط مدل‌های منطقی بیان می‌شود، هیچ نوع اهمیتی در زمان رخدادها وجود ندارد و تنها دنباله رویدادها بیان‌گر رفتار سیستم می‌باشد.

۳-۳-۴- مقایسه مدل‌های زمانی و منطقی

در این بخش توسط یک مثال مدل‌های زمانی و منطقی را با هم مقایسه می‌نماییم. یک خط تولید را فرض کنید که در آن قرار است توسط یک غلتک و قیف بطری‌های شیشه‌ای از آبمیوه پر شوند. همانطور که در شکل ۲ نشان داده شده است، زمانی که یک بطری زیر قیف قرار می‌گیرد، دهانه قیف باز شده و شروع به پر کردن بطری می‌نماید. پس از پر شدن بطری، غلتک شروع به گردش کرده تا بطری

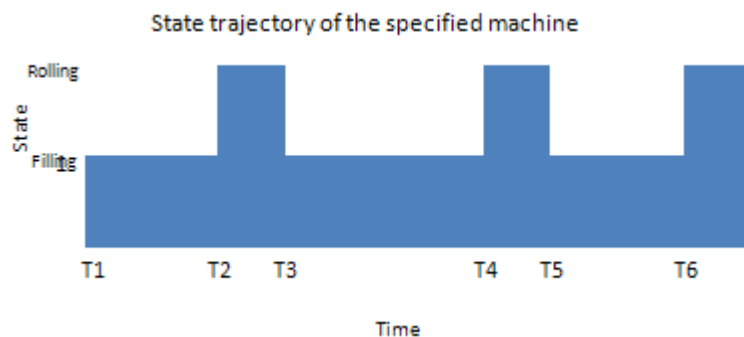


شکل ۴-۲

بعدی را زیر قیف مستقر کند. در این حالت می‌توانیم در یک نگاه کلی به سیستم، سیستم را با دو حالت "پر کردن" و "چرخاندن" و دو اتفاق "پر شد" و "بطری قرار گرفت" بیان کنیم. در این صورت حالت‌ها با برچسب‌های R (چرخاندن) و F (پر کردن) و رخدادها با برچسب‌های Filled (پر شد) و Bottle Positioned (بطری قرار گرفت) بیان شده‌اند.

۳-۳-۳-۱- بیان زمانی مسئله

در این مثال هیچ نوع قیدی در مورد فاصله زمانی رخدادها بیان نشده است و همانطور که در شکل ۳-۴ نشان داده شده است، تنها وابسته به زمان انجام شدن رخدادها می‌باشد.



شکل ۳-۴

۲-۳-۳-۴- بیان منطقی مسئله

در این مدل، هیچ نوع اهمیتی نسبت به زمان رخدادها وجود ندارد و تنها دنباله رویدادها بیان‌گر رفتار سیستم می‌باشد، که در این مثال به صورت زیر نمایش داده می‌شود.

... Filled, Bottle Positioned, Filled ...

۴-۴- بیان ریاضی مدل

در این بخش تنها به ریاضیات مدل منطقی می‌پردازیم. بیان ریاضی مدل‌های زمانی به راحتی از ریاضیات مدل منطقی قابل استنباط می‌باشد. در یک مدل منطقی ما علاقه‌مند به بیان دنباله‌ای از رخدادها هستیم که یک ماشین می‌تواند تولید نماید. در این راستا تعاریف زیر را انجام می‌دهیم.

- Λ مجموعه‌ای است حاوی کلیه اتفاق‌های ممکن در سیستم. در نتیجه هر عضو از مجموعه Λ یک رخداد نامیده می‌شود.

- Λ° مجموعه‌ای است شامل کلیه ترکیبات از اتفاق‌ها روی مجموعه Λ ، که شامل مجموعه تهی ε می‌باشد.

- Φ بیان‌گر مجموعه حالت‌هایی می‌باشد که سیستم می‌تواند به آنها گذار نماید یا در آن حالت‌ها قرار داشته باشد.

- یک رشته به صورت زیر یک مسیر از سیستم را بیان می‌کند.

$$(\sigma_1 \in \Lambda^0) u = \sigma_1 \sigma_2 \dots \sigma_n \quad (۱ - ۴)$$

- زیرمجموعه L از مجموعه Λ° ، مجموعه‌ای است از رشته‌هایی که قابل پذیرش یا اتفاق می‌باشد و به عنوان مثال قابلیت روی دادن فیزیکی دارند.

- هر زیر مجموعه از Λ به طور معمول یک زبان روی مجموعه الفبایی Λ نامیده می‌شود. که زیرمجموعه L بیان‌گر یک زبان فیزیکی می‌باشد.

- رشته u یک پیشوند برای رشته v از مجموعه Λ° می‌باشد در صورتی که w از مجموعه Λ° وجود داشته باشد که $v = u w$ باشد. در صورتی که v یک رشته قابل اتفاق باشد تمامی پیشوندهای آن نیز دارای قابلیت اتفاق می‌باشند.

- L یک زبان بسته روی پیشوندها می‌باشد در صورتی که

$$\bar{L} = \{u: uv \in L \text{ for some } v \in \Lambda^\circ\} \quad (۲ - ۴)$$

- $|A|_a$ بیانگر تعداد رخداد‌های a در رشته A می‌باشند. به عنوان مثال L زبانی است که بیانگر سیستم گسسته رخدادی می‌باشد که نقش بافر دارد همانطور که در تعریف L نشان داده شده است همیشه باید تعداد رویدادهای α (خالی کردن بافر) کمتر مساوی تعداد رویدادهای β (بارگزاری بافر) باشد تا زبان L از لحاظ فیزیکی معنی‌دار باشد.

$$L = \{w \in \Lambda^\circ : \text{for each prefix } u \text{ of } w \ |u|_\alpha \leq |u|_\beta\} \quad (۳ - ۴)$$

۴-۵- بیان زبانی

به منظور پیاده‌سازی این هدف، یک سیستم گسسته رخداد را با استفاده از بیان حالت‌ها یا

معماری گذارهایش بیان می‌کنیم. برای بیان رفتار L به صورت زیر عمل می‌کنیم:

یک ماشین M با مجموعه حالت‌های Φ و حالت اولیه q_0 و تابع گذار $\delta: \Lambda \times \Phi \rightarrow \Phi$ را در نظر

بگیرید با توجه به اتفاقیهای ممکن در حالت q ، نتیجه $\Lambda(q)$ تحت رخداد σ به صورت $\delta(\sigma, q)$ بیان می‌شود

که حاصل q' می‌باشد یعنی در حالت q تحت رخداد σ سیستم از حالت q به حالت q' گذار می‌کند.

می‌توان ماشین M را به صورت یک گراف در نظر گرفت که حالت‌های ماشین گره‌های آن و گذار

از $q \rightarrow q'$ لبه‌های آن می‌باشند و $q' = \delta(\sigma, q)$ بیانگر سه‌تایی مرتب (σ, q, q') است.

تابع گذار δ روی M می‌تواند به یک تابع قطعه‌ای روی فضای $\Lambda^\circ \times \Phi$ توسط

$$\begin{aligned} \delta(\varepsilon, q) &= q, \quad \text{and} \\ \delta(w\sigma, q) &= \delta(\sigma, \delta(w, q)) \end{aligned} \quad (۴ - ۴)$$

توسعه می‌یابد. زمانی که $q' = \delta(\sigma, q)$ و $\delta(\sigma, q')$ تعریف شده باشند. به این منظور $\delta(w, q)!$ بیانگر عبور

از حالت q تحت رخداد‌های رشته w می‌باشد که منجر به حالت q' می‌شود. در صورتی که کلیه گذارهای

میانی تعریف شده باشند و در حالت کلی در یک ماشین M ، $\delta(w, q)!$ به این معنی است که، مسیری در

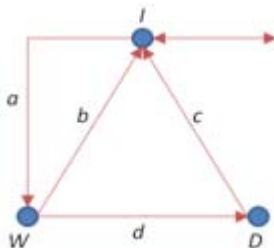
گراف M از q به q' تحت نام w وجود دارد. از این پس M به عنوان یک ماشین می‌باشد که از حالت اولیه

q_0 در مسیر گرافش شروع به گذار می‌نماید. ماشین M نقش Plant را در تئوری کنترل ایفا می‌نماید.

گراف نشان داده شده در شکل ۴-۴ نمایش‌دهنده یک ماشین با سه حالت و چهار رویداد متفاوت

می‌باشد. حالتها عبارتند از Idle، Working و Down که به ترتیب با حروف الفبایی I, W, D برچسب

شده‌اند. حالت اولیه ماشین توسط جهت ورودی به حالت I مشخص شده است. رخدادها همانطور که در گراف نشان داده شده‌اند با حروف a, b, c, d برچسب شده‌اند.



شکل ۴-۴

ماشین مزبور به صورت زیر بیان می‌شود:

$$\Phi = \{I, W, D\} \bullet$$

$$\Lambda = \{a, b, c, d\} \bullet$$

رفتار بسته M توسط رشته‌هایی از رخدادها که از حالت ابتدایی I شروع شده باشند بیان می‌شوند. در حالت متداول این رفتار به صورت زیر نمایش داده می‌شود.

$$L(M) = \{ab + adc\} * \{\epsilon + a + ad\} \quad (۵ - ۴)$$

همچنین رفتار مارک شده از ماشین M که توسط حالت‌های مارک شده بیان می‌شود به صورت

زیر داده می‌شود:

$$L_m(M) = \{s: s \in L \text{ and } \delta(w, q) \in \Phi_m\} \quad (۶ - ۴)$$

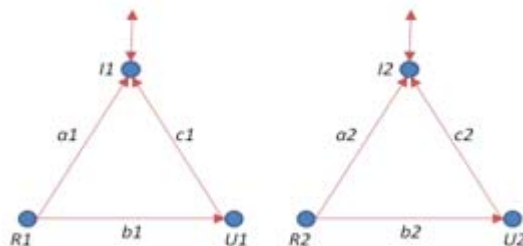
که Φ_M مجموعه حالت‌های مارک شده است. ما L_m را به صورت یک زیرمجموعه از رفتارهای

ممکن ماشین M بیان می‌کنیم که نمایش دهنده یک Task کامل شده یا زنجیره‌ای از Task های کامل شده هستند.

حالت‌های مارک شده توسط پیکان‌های تنها نمایش داده می‌شوند که همانطور که در گراف مزبور مشاهده می‌شود حالت I یک حالت مارک شده می‌باشد. فرم نمایش داده شده بیانگر این موضوع است که رفتارهای مارک شده می‌بایست از حالت I شروع به حرکت کرده و دوباره به حالت I برسند که یک پروسه کامل از ماشین را نمایش می‌دهند. واضح می‌باشد که هر رشته در مجموعه L_m می‌تواند به یک پروسه کامل توسعه یابد. رفتارهای مارک شده در این مثال توسط L_m در زیر بیان شده‌اند.

یکی از مسائل مهم در تئوری کنترل بیان سیستم‌های سری، موازی، همزمان و ... می‌باشد که با ذکر یک مثال به بیان یک مورد از این سیستم‌ها که سیستم‌های غیر همزمان مستقل می‌باشند می‌پردازیم.

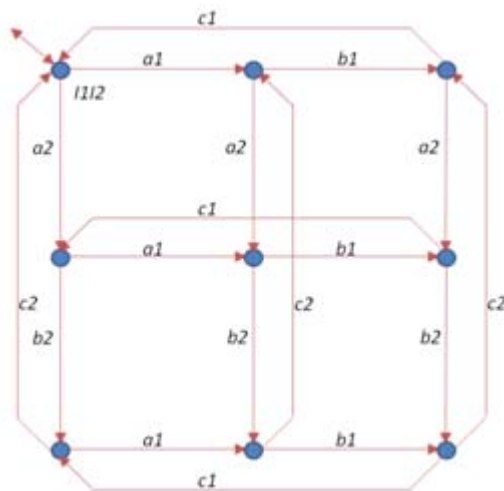
دو ماشین غیر همزمان مستقل M_1 و M_2 در شکل ۴-۵ نمایش داده شده‌اند.



شکل ۴-۵

هر ماشین سه حالت I (Idle) ، R (Request) و U (Use) به همراه گذارهای نمایش داده شده می‌باشند. ماشین‌هایی از نوع توسط عملگر Shuffle Product بیان می‌شوند به صورت $M = M_1 \parallel M_2$ نمایش داده می‌شود. ماشین M یک سیستم گسسته رخداد می‌باشد که حاصل Shuffle Product ، M_1 و M_2 است تحت فرض غیر همزمان و مستقل بودن این دو ماشین.

حالت‌های ماشین M حاصل از ضرب دکارتی مجموعه حالت‌های M_1 و M_2 می‌باشد. این مجموعه حالت (x, y) که x حالتی از مجموعه حالت M_1 و y حالتی از مجموعه حالت M_2 است، تنها می‌تواند به صورت‌های $(x, y) \rightarrow (x, y')$ یا $(x, y) \rightarrow (x', y)$ گذار نماید. که همان بیانگر غیر همزمان بودن دو ماشین است. شکل ۴-۶ ضرب Shuffle این دو ماشین یا ماشین M را نمایش می‌دهد.



شکل ۴-۶

استقلال این دو ماشین در شکل ۴-۶ مشهود می‌باشد. به این دلیل که هر کدام از ماشین‌ها توانسته‌اند بدون هیچ محدودیتی نسبت به ماشین دیگر به هر حالت ممکن خود دسترسی داشته باشند. همچنین گذارها تنها با یک دخداد برچسب شده‌اند که نمایش دهنده غیر همزمان بودن دو سیستم می‌باشد.

مجموعه رخداد‌های ماشین M توسط Λ به صورت

$$\Lambda = \{a_1, b_1, c_1, a_2, b_2, c_2\} \quad (۷-۴)$$

نمایش داده می‌شود که حاصل اجتماع رخدادها در ماشینهای M_1 و M_2 می‌باشد. همانطور که مشاهده می‌شود حالت I_1I_2 یک حالت مارک می‌باشد یعنی ماشین M از حالت اولیه I_1I_2 شروع به کار کرده و به حالت نهایی I_1I_2 برای انجام یک پروسه کامل باید دسترسی یابد.

این مثال استفاده از Shuffle Product را در مدل کردن سیستم‌های غیر همزمان نمایش داد. این نوع سیستمها منجر به گرافهای کوچکی می‌شوند. اما سایر سیستمها به دلیل پیچیدگی‌های بیشتر می‌توانند منجر به گرافهای بزرگتر و پیچیده شوند.

۴-۶- کنترل سیستم‌های گسسته رخداد

تمامی سیستم‌هایی که تا کنون مورد بحث قرار گرفته‌اند فاقد معنی کنترلر به عنوان یک ماشین می‌باشند. به این منظور برای تعریف کنترلر به ارائه مباحث زیر می‌پردازیم.

۴-۶-۱- کنترل پذیری و نظارت سیستم‌های گسسته رخداد

جهت کنترل یک سیستم گسسته رخداد در نظر داریم رخدادهای خاصی را در حالت‌های مطلوب غیر فعال سازیم. در نتیجه به منظور مدل‌سازی ساختمان یک کنترلر رخدادها در مجموعه رخداد Plant به دو زیرمجموعه کنترل پذیر و کنترل ناپذیر تقسیم می‌شوند.

رخدادها در مجموعه Λ_c می‌توانند در هر لحظه از زمان غیر فعال شوند در صورتی که رخدادها در مجموعه Λ_{nc} متأثر از کنترلر نمی‌باشند. به عنوان مثال رخدادهایی چون شکستن قطعه‌ای از ماشین در سیستمهای تولید، از دست دادن یک بسته در کانالهای ارتباطی و اغتشاش و... رخدادهایی در مجموعه کنترل ناپذیر می‌باشند زیرا ما نمی‌توانیم هیچ دخل و تصرفی در آنها داشته باشیم.

یک ورودی کنترلر به ماشین M ، شامل رخدادهایی است $\lambda \subseteq \Lambda$ که شرط $\Lambda_{nc} \subseteq \lambda$ را برآورده سازند. در نتیجه اگر σ عنصری از λ باشد آنگاه σ توسط λ فعال شده است (اجازه دارد که اتفاق بیفتد) در

غیر اینصورت σ غیر فعال شده است. شرط $\Lambda_{ii} \subseteq \lambda$ به این معنا می‌باشد که رخدادهای کنترل ناپذیر همیشه فعال می‌باشند.

در صورتی که Γ بیانگر مجموعه ورودی‌های کنترل باشد یک سیستم گسسته رخداد که مجهز به Γ باشد یک سیستم گسسته رخداد کنترل شده نامیده و با سیستم گسسته رخداد کنترل شده CDES نمایش داده می‌شود. کنترل در یک سیستم گسسته رخداد کنترل شده شامل سوئیچ کردن مابین اعضای مجموعه Γ می‌باشد. عملیات سوئیچینگ براساس رشته‌هایی که توسط ماشین تولید شده‌اند انجام می‌شود. در نتیجه در این حالت کنترلر ما نقش نظارتی را ایفا می‌کند.

یک ناظر یک نگاشت می‌باشد به صورت

$$f: L \rightarrow \Gamma \quad (۸ - ۴)$$

و تعریف می‌شود به صورت $f(w)$ که w رخداد مشاهده شده و $f(w)$ خروجی از مجموعه ورودی کنترل می‌باشد.

هدف ما طراحی یک ناظر می‌باشد که ورودی‌های کنترل مناسب را جهت رفتار مناسب ماشین به ماشین اعمال می‌نماید. به این منظور تعاریف زیر را انجام می‌دهیم.

رابطه $T = (\Lambda, X, \delta, x_0)$ یک ماشین می‌باشد و نگاشتی به صورت

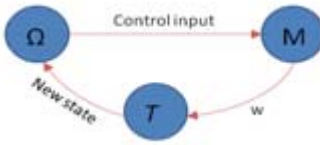
$\Omega: X \rightarrow \Gamma$ نگاشتی است از فضای حالت‌ها به فضای ورودی کنترل. جفت (T, Ω) یک ناظر را تحقق می‌-

دهد در صورتی که برای هر w داشته باشیم

$$\Omega(\delta(w, x_0)) = f(w) \quad (۹ - ۴)$$

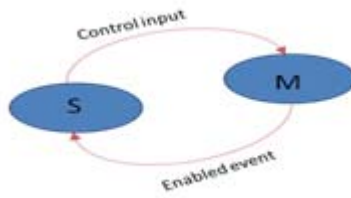
در اصطلاحات تئوری کنترل، M نقش Plant، T رؤیت‌گر و Ω نقش فیدبک را ایفا می‌کنند که

در شکل زیر نمایش داده شده‌اند.



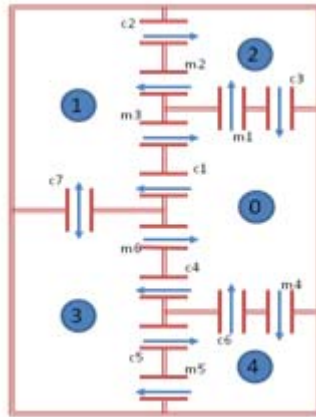
شکل ۴-۷

همچنین یک ناظر در شکل ۴-۸ نمایش داده شده است. ناظر و ماشین به صورت موازی عمل می‌کنند به این معنا که $S \times M$ شامل مجموعه حالت‌های (x, q) می‌باشد و رخداد σ امکان‌پذیر است در صورتی که در هر دو ماشین امکان‌پذیر باشد. در نتیجه پس از رویداد σ نتیجه فضای حالت به صورت (x', q') خواهد بود.



شکل ۴-۸

جهت روشن شدن مطالب به مثال زیر می‌پردازیم. یک موش و گربه در لابیرنت نشان داده شده در شکل ۴-۹ قرار دارند. همانطور که نشان داده شده است درها تنها در جهت نشان داده شده می‌توانند باز شوند و درهایی که تنها با یک جهت نشان داده شده‌اند قابلیت کنترل پذیری را دارا می‌باشند. درهایی که با برچسب M نشانه‌گذاری شده‌اند تنها به موش اجازه گذر می‌دهند و درهایی که با c نشانه‌گذاری شده‌اند تنها به گربه اجازه عبور می‌دهند.

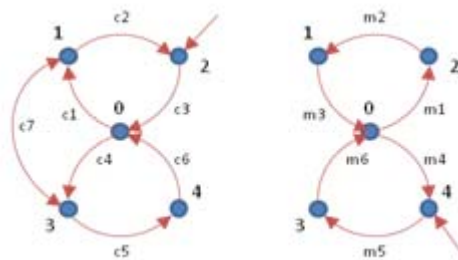


شکل ۴-۹

مجموعه رخدادها در این ماشین به صورت

$$A = \{m_j, c_i : 1 \leq i \leq 7, 1 \leq j \leq 6\} \quad (۴-۱۰)$$

می‌باشند. موش و گربه در لایبرنت مزبور توسط ماشین‌های M_1 و M_2 نشان داده شده در شکل ۴-۱۰ مدل شده‌اند.



شکل ۴-۱۰

شکل ۴-۱۱ نتیجه Shuffle Product این دو ماشین می‌باشد.

	00	01	02	03	04	10	11	12	13	14	20	21	22	23	24	30	31	32	33	34	40	41	42	43	44
00	.	.	M1	.	M4	C1	C4
01	M3	C1	C4
02	.	.	M2	C1	C4
03	M5	C1	C4
04	.	.	.	M5	C1	C4
10	M1	.	M4	C2	C7
11	M3	C2	C7
12	M2	C2	C7
13	M6	C2	C7
14	C3	M5	C2	C7
20	.	C3	M1	.	M4
21	.	.	C3	M3
22	.	.	.	C3	M2
23	C3	M6
24	C7	M5
30	C7	M1	.	M4	C5
31	C7	M3	C5	.	.	.
32	C7	M2	C5	.	.
33	C7	M6	C5	.	.
34	M5	C5	.
40	C5	M1	.	M4	.
41	.	C5	M3
42	.	.	C5	M2	.	.	.
43	.	.	.	C5	M6	.	.	.
44	C5	M5	.

شکل ۴-۱۱

هدف ما این است که ناظری طراحی کنیم تا

۱. موش و گربه هیچ‌گاه در یک اتاق به هم نرسند.

۲. موش و گربه به ترتیب در خانه‌های ۴ و ۲ قرار دارند و هدف ما این است که بتوانند بعد از خروج از آن‌ها به این حالت اولیه باز گردند.

در صورتی که M نتیجه ضرب دو ماشین باشد، $\Phi_m = \{(2, 4)\}$ و $\Lambda_c = \{c_7\}$ می‌باشند. همچنین

مجموعه ورودی کنترل به صورت $L = \{\lambda : c_7 \in \lambda\}$ بیان می‌شود.

عملیات کنترلی که ناظر ما ایفا می‌کند به این صورت است که زمانی که هر کدام از موش یا گربه

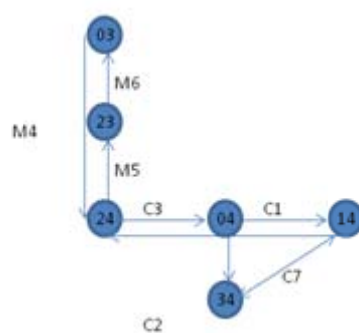
از خانه خود خارج شد، دیگری تا بازگشت عنصر خارج شده به خانه اولیه خود، نتواند از خانه خود خارج شود. در نتیجه استراتژی کنترل به صورت زیر می‌باشد.

	C1	C2	C3	C4	C5	C6	C7	M1	M2	M3	M4	M5	M6
Home	---	---	1	---	---	---		---	---	---	---	1	---
Room	1	1	0	1	0	---		0	---	---	1	0	1

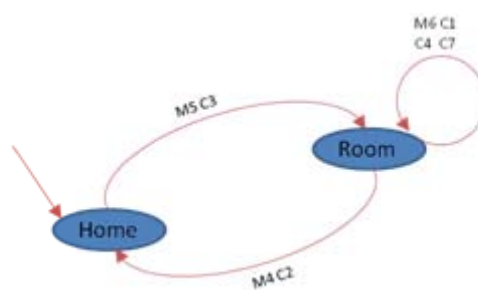
شکل ۴-۱۲

ناظر طراحی شده و مسیرهای ممکن (گراف) به ترتیب در شکل‌های ۴-۱۳ و ۴-۱۴ نشان داده

شده‌اند.



شکل ۴-۱۳



شکل ۴-۱۴

۲-۶-۴- رؤیت پذیری

برای مدل‌سازی یک سیستم گسسته رخداد توسط رؤیت‌گرها زیرمجموعه Λ_0 که شامل رخدادهای رؤیت پذیر می‌باشد، تعریف می‌شود. همچنین یک Projector (P) به صورت زیر تعریف می‌شود.

$$P: \Lambda \rightarrow (\Lambda_0 \cup \{\varepsilon\}) \quad (۴-۱۱)$$

نظریه به این صورت می‌باشد که $P(\sigma)$ رخدادی است که در گذار ماشین تحت رخداد σ رؤیت شده است.

رخدادها در مجموعه Λ تحت نظر P به صورت نگاشت زیر رؤیت می‌شوند.

$$P(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Lambda_0 \\ \varepsilon & \text{if } \sigma \notin \Lambda_0 \end{cases} \quad (۴-۱۲)$$

در این صورت $P(\sigma) = \varepsilon$ نمایش دهنده این موضوع می‌باشد که رخداد σ پاک شده است و قابل رؤیت نمی‌باشد. همچنین α و β در حالتی که $P(\alpha) = P(\beta)$ باشد متمایز از هم تشخیص داده نمی‌شوند.

۷-۴- زمان در سیستم

در بسیاری از سیستم‌ها اطلاعات زمانی از سیستم بسیار مهم بوده و آنالیز سیستم توسط این اطلاعات زمانی انجام می‌شود. در این سیستم‌ها، اطلاعات زمانی می‌بایست در مدلی که از سیستم ارائه می‌شود گنجانده شوند، که مدل‌های زمانی نامیده می‌شوند. مدل‌های زمانی به دو دسته غیر اتفاقی و اتفاقی کلاس‌بندی می‌شوند.

در سیستم‌های کیفی، سیستم توسط مدل‌های منطقی مدل می‌شود. پیچیدگی و حجم محاسبات در مدل‌های ارائه شده یک مساله اساسی می‌باشد. افزایش خطی تعداد پارامترهای سیستم منجر به افزایش تصاعدی حالات سیستم می‌شود. ساده‌ترین روش‌های پردازش و آنالیز نیز در این حالت

به حجم محاسبات بالایی نیازمند می‌شوند. روش‌های زیادی جهت کاهش این پیچیدگی در مدل‌ها موجود می‌باشد که در این پایان‌نامه از ساختارهای ناهمگن استفاده شده است.

مدل زمانی غیر اتفاقی بسیار مشابه مدل‌های منطقی می‌باشند، با این تفاوت که در این مدل‌ها زمان تغییر حالات می‌بایست در مدل گنجانده شود. جهت فرموله کردن این نوع مدل، تراژکتوری‌های زیادی می‌بایست بررسی شوند که می‌توان از ساختار گذار همچون شبکه‌های پتری استفاده نمود. این نوع مدل‌ها در سیستم‌های صنعتی، ساختارهای کنترل زمان زنده و پردازش سیگنال به منظور آنالیز رفتارهای متناوب مورد استفاده قرار می‌گیرند.

مدل‌های زمانی اتفاقی مشابه مدل‌های زمانی غیر اتفاقی می‌باشند و همچنین محدودیت‌های آن‌ها را نیز دارا هستند. زمان وقوع اتفاقات و گذر از حالت در این نوع مدل‌ها یا در قسمتی از آن‌ها غیر قابل پیش‌بینی می‌باشد. بسیاری از سیستم‌های صنعتی، شبکه‌های ارتباطی و شبکه‌های صف تحت این مدل‌ها، مدل می‌شوند.

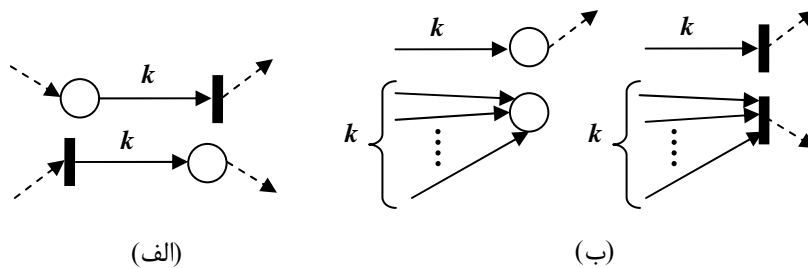
۸-۴- شبکه‌های پتری

شبکه‌های پتری ابزار مدل‌سازی ریاضی و گرافیکی قدرتمندی می‌باشند. به عنوان یک زبان برنامه‌ریزی سطح بالا، شبکه‌های پتری قادر هستند، محدوده وسیعی از سیستم‌ها را بیان نمایند. همچنین شبکه‌های پتری قادر به مدل‌سازی سیستم‌های دارای خواص همزمانی، موازی، غیر هم‌زمان، غیر اتفاقی و اتفاقی می‌باشد. شبکه پتری به طور گسترده در مدل‌سازی سیستم‌های دینامیک به کار برده می‌شود. این واقعیت‌ها شبکه پتری را کاندیدای مناسبی جهت آنالیز رفتاری و کیفی سیستم‌های رباتیکی نموده است. شبکه پتری به صورت گرافیکی، ساختارهای سیستم را توسط گراف جهت‌دار و وزن‌دار مدل می‌نماید. در مدل شبکه پتری دو نوع گره وجود دارد که این گره‌ها توسط کمان‌های یک جهته وزن‌دار به هم متصل می‌شوند. نوع اول از این گره‌ها، حالت نامیده می‌شود. که در شکل ۴-۱۵ الف نمایش داده شده است.



شکل ۴-۱۵

توکن‌ها بیانگر پر بودن بافرها می‌باشند و به عبارت دیگر هر توکن نشانه وجود نیازمندی‌ها هستند. شکل ۴-۱۵ ب یک حالت را نمایش می‌دهد که یک توکن در آن وجود دارد. هر دایره، حالتی از سیستم را نمایش می‌دهد که توکن نشانه حالتی است که سیستم در آن قرار دارد. نوع دیگر گره، گذار نامیده می‌شود که در شکل ۴-۱۵ ج نمایش داده شده است. گذار، رخدادهای یک سیستم را مدل می‌کند. بر اساس قانون آتش، گذارها می‌توانند توکن‌های جدیدی بسازند و یا توکن‌ها را از بین ببرند. کمان‌ها تنها می‌توانند از یک حالت خارج شده و به یک گذار وارد شوند و یا از یک گذار خارج شده و به یک حالت وارد شوند که در شکل ۴-۱۶ الف نشان داده شده است. همچنین کمان‌ها توسط وزن‌هایشان برچسب می‌شوند که در صورتی که وزن کمان یک باشد، کمان برچسب نمی‌شود. به عبارت دیگر کمان بدون برچسب، کمانی با وزن یک می‌باشد. وزن کمان بیانگر تعداد کمان‌های خروجی از یک حالت به یک گذار و یا تعداد کمان‌های خروجی از یک گذار به یک حالت می‌باشد که در شکل ۴-۱۶ ب نشان داده شده است. مارک دلالت بر حالت سیستم دارد و به هر حالت از سیستم یک عدد نامنفی اختصاص می‌دهد. موقعیت ابتدایی سیستم مارک ابتدایی نامیده و توسط M_0 نمایش داده می‌شود. مارک توسط حرف M



شکل ۴-۱۶

نشان داده می‌شود و یک n بردار می‌باشد که n تعداد کل حالت‌های سیستم می‌باشد. هر درایه از بردار M توسط $M(p)$ نشان‌گذاری می‌شود که تعداد توکن‌های موجود در حالت p می‌باشد. اگر یک مارک عدد نامنفی k را به حالت p اختصاص دهد در این صورت تعداد k توکن (نقاط مشکلی) در داخل p موجود می‌باشد. در این حالت گفته می‌شود که p توسط k توکن، مارک شده است.

شبکه پتری توسط یک پنج‌تایی مرتب $PN = (P, T, F, W, M_0)$ بیان می‌شود که:

- مجموعه $P = \{p_1, p_2, p_3, \dots\}$ مجموعه محدودی از حالت‌ها می‌باشد.
- مجموعه $T = \{t_1, t_2, t_3, \dots\}$ مجموعه محدودی از گذارها می‌باشد.
- مجموعه $F \subseteq (P \times T) \cup (T \times P)$ از لبه‌ها می‌باشد که ارتباط گذارها را بیان می‌کند.
- تابع $W: F \rightarrow \{1, 2, 3, \dots\}$ تابع وزن می‌باشد و $w(n_i, n_j)$ وزن لبه از n_i به n_j را نشان می‌دهد.
- مارک $M_0: P \rightarrow \{0, 1, 2, \dots\}$ مارک ابتدایی می‌باشد.
- $P \cup T \neq \emptyset, P \cap T = \emptyset$

۱-۸-۴- فعال و آتش شدن گذار

در تئوری شبکه پتری تنها قانون، قانون فعال شدن و آتش شدن گذار می‌باشد. هر چند این قانون در ظاهر ساده می‌باشد اما در تئوری شبکه پتری مفهومی عمیق و پیچیده دارد. در مدل‌سازی، با استفاده از تعریف شرایط، شرایط همچون حالت‌ها می‌باشند و با استفاده از تعریف رخداد، گذار همچون رخداد می‌باشد. ورودی‌ها و خروجی‌های یک گذار همچون شرایط پیشین و پسین یک رخداد می‌باشند. اگر یک حالت دربردارنده یک توکن باشد در این صورت می‌توان اینگونه بیان نمود که، شرط موجود در حالت برآورده شده است.

شبکه‌های پتری خصوصاً در مدل‌سازی سیستم‌های گسسته‌رخداد پیچیده که توسط حالت‌ها و رفتار آن‌ها بیان می‌شوند پرکاربرد می‌باشد. رفتار سیستم در یک مدل پتری به وسیله قانون زیر بیان می‌شود.

- گذار t فعال می‌باشد، اگر تمامی حالت‌های ورودی آن (P_i, t) حداقل به اندازه $W(P_i, t)$ توکن داشته باشند.

- یک گذار زمانی آتش می‌شود که رخداد مربوط به آن روی دهد.

- اگر گذار فعال شده t آتش شود، $W(P_i, t)$ توکن از حالت‌های P_i خارج می‌شوند (از بین می‌روند) و $W(P_i, t)$ توکن به حالت‌های P_i که (t, P_i) وارد می‌شوند (ساخته می‌شوند).

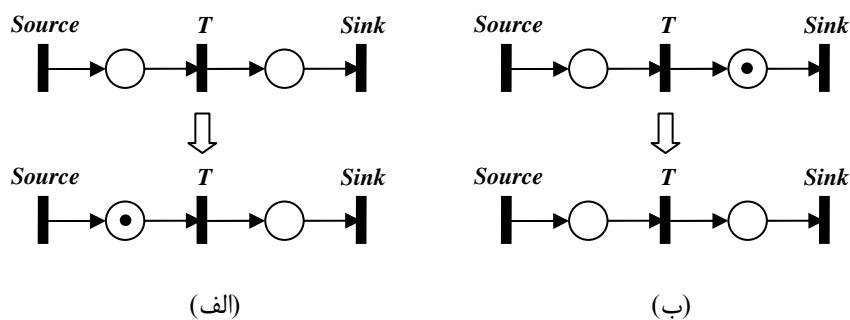
گذاری که هیچ حالت ورودی ندارند منبع نامیده می‌شود و گذاری که هیچ حالتی در خروجی ندارد حفره نامیده می‌شود. توجه نمایید که گذار منبع همواره بدون شرط فعال می‌باشد و گذار حفره پس از آتش شدن تمامی توکن‌های ورودی را نابود می‌سازد و هیچ توکنی به وجود نمی‌آورد.

۲-۸-۴- شبکه‌های پتری زمانی

در تعریف اصلی شبکه‌های پتری تعریف کاملی از زمان نیامده است. (مرجع) لازم و مفید می‌باشد که تاخیرهای زمانی جهت انجام مدل‌های زمانی و حل مسائل زمان‌بندی شده در سیستم‌های دینامیک به طور کامل تعریف شوند. تاخیرهای زمانی می‌توانند با حالت‌ها یا گذارها همراه شوند. در صورتی که زمان به صورت کامل مشخص باشد، مدل‌های پتری، شبکه‌های زمانی نامیده می‌شوند. در غیر این صورت در صورتی که تاخیرها اتفاقی باشند مدل پتری، یک شبکه اتفاقی نامیده می‌شود.

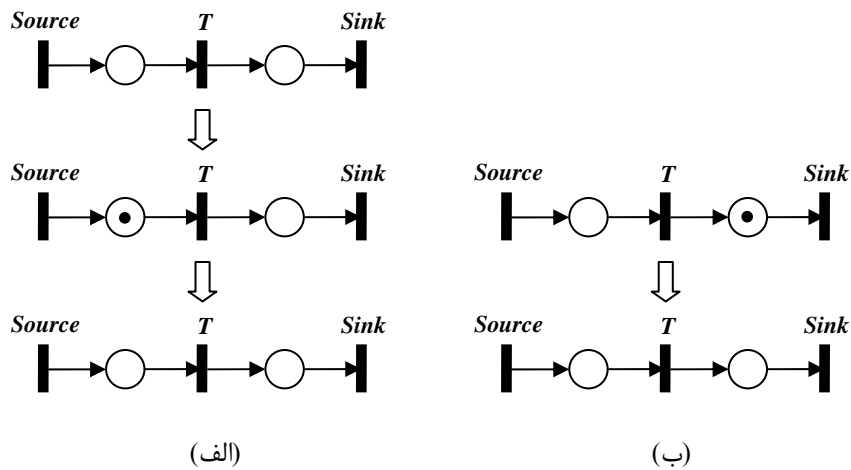
یک شبکه پتری ساده و مارک ابتدایی آن را که در شکل ۴-۱۷ الف نمایش داده شده است را در

نظر بگیرید. شبکه پتری مورد نظر شامل سه گذار، منبع، T و حفره و همچنین دارای دو حالت $P1$ و $P2$ می‌باشد و وزن تمامی کمان‌های موجود در شبکه، یک می‌باشد. در نتیجه کمان‌ها فاقد برچسب می‌باشند.



شکل ۴- ۱۷

گذار سمت راست یک منبع می‌باشد چرا که هیچ حالت ورودی ندارد. همچنین گذار سمت چپ در شبکه یک حفره است چون هیچ حالتی در خروجی ندارد. گذار میانی یک گذار زمان‌دار می‌باشد که توسط T برچسب شده است. در این حالت تاخیر زمانی (d) با گذار T همراه شده است. به این معنی که زمانی که T فعال می‌شود، یک توکن در P1 برای d ثانیه وجود خواهد داشت و پس از آن از بین خواهد رفت. به عبارت دیگر، گذار T، d ثانیه پس از فعال شدن، آتش خواهد شد. شکل ۴- ۱۷ ب دلالت بر حالتی دارد که در آن d ثانیه از فعال شدن T گذشته و T آتش شده است. گذار حفره، به محض فعال شدن آتش می‌شود و پس از آن توکنی در شبکه وجود نخواهد داشت. این مثال کلاسی از گذارهای زمانی را نشان می‌دهد که گذارهای زمانی بدون مصرفی نامیده می‌شوند. در این کلاس از گذارهای زمانی، پس از فعال شدن گذار، توکن‌ها در حالت‌های ورودی باقی خواهند ماند و پس از زمان مشخصی که گذار، آتش شد، در حالت‌های ورودی از بین خواهند رفت و در حالت‌های خروجی به وجود خواهند آمد. در کلاس دیگری از گذارهای زمانی که گذارهای زمان مصرفی نامیده می‌شوند، به محض فعال شدن گذار زمان‌دار، توکن‌ها در حالت‌های ورودی از بین خواهند رفت. پس از گذشتن زمان تاخیر گذار، گذار آتش شده و توکن در حالت‌های خروجی به وجود خواهد آمد. شکل ۴- ۱۸ مثال بالا را در حالتی که گذار میانی، یک گذار زمانی مصرفی می‌باشد را نشان می‌دهد.



شکل ۴-۱۸

ساختارهایی که شامل حالتی هستند که حالت مورد نظر، حالت ورودی چند گذار می باشد، منجر به ناسازگاری در شبکه پتری می شوند. در این ناسازگاری ممکن است شبکه مجبور به انتخاب شود، یعنی یک یا چند تا از گذارها آتش شوند نه همه آنها. در این صورت نیاز به مکانیزمی داریم که این ناسازگاری را برطرف نماید. به این منظور گذارهای زمانی می توانند منجر به ساختارهای سازگار یا بدون انتخاب شوند.

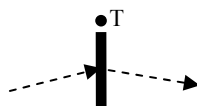
در بخش بعد به تعریف کلاس جدیدی از شبکه های پتری خواهیم پرداخت که سازگار یا بدون انتخاب می باشند.

به منظور هماهنگی در حرکت مفاصل ربات و پیمایش مسیر به صورت هماهنگ، مجبوریم تا حرکت مفاصل را زمان بندی نماییم. شبکه های پتری زمانی انتخاب مناسبی جهت پیاده سازی این هدف می باشند. دنباله ای از حالت ها و گذارهای زمانی نقش مهمی را در کنترل مفاصل و مدیریت هماهنگی در آنها را به عهده دارند که در بخش بعد به بیان آنها خواهیم پرداخت. این کلاس از شبکه های پتری زمانی ما را قادر می نمایند تا به راحتی حرکت هماهنگ ربات را در راستای مسیر طراحی شده را با استفاده از شبکه پتری پیاده سازی نماییم. پس از آن، به پیاده سازی روش تطبیقی طراحی مسیر که در

فصل ما را قادر می‌نمایند تا به راحتی حرکت هماهنگ ربات را در راستای مسیر طراحی شده را با استفاده از شبکه پتری پیاده‌سازی نماییم. پس از آن، به پیاده‌سازی روش تطبیقی طراحی مسیر که در فصل ۳ تعریف شد، توسط شبکه‌های پتری می‌پردازیم.

۳-۸-۴- گذارهای زمانی دست‌کاری شده

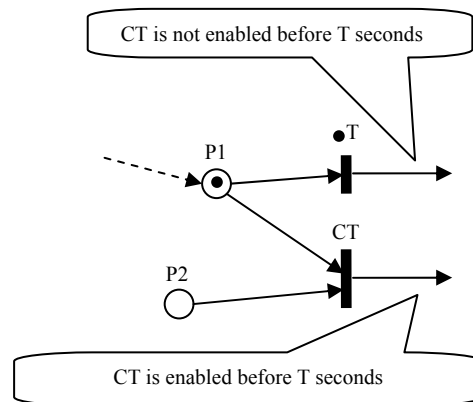
به منظور پیاده‌سازی هماهنگی مابین مفاصل ربات، می‌خواهیم کلاس جدیدی از شبکه پتری را معرفی نماییم که آن را گذار زمانی دست‌کاری شده (MTT) نام‌گذاری نموده‌ایم. گذار معرفی شده یک گذار زمانی غیر اتفاقی می‌باشد که با یک متغیر تاخیر همراه شده است. متغیر تاخیر می‌تواند در هر لحظه به مقداری دلخواه تغییر داده شود. این نوع گذار از لحاظ ظاهری مانند گذارهای دیگر است، با این تفاوت که، یک نقطه سیاه در بالای آن قرار دارد و این نقطه نشان‌دهنده یک گذار زمانی دست‌کاری شده می‌باشد.



شکل ۴-۱۹

نوع دیگری از گذار که به تعریف آن خواهیم پرداخت، گذار شرط نامیده می‌شود، که آن را با CT نمایش می‌دهیم. گذار CT دارای یک حالت ورودی است که شرط گذار نامیده می‌شود. گذار شرط نیز از لحاظ ظاهری مانند سایر گذارها می‌باشد و توسط CT متمایز می‌شود. کلاس مورد نظر در شکل ۴-۲۰ نمایش داده شده است. دو گذار نمایش داده شده در شکل که پیشتر آن‌ها را معرفی نموده‌ایم به صورت یک XOR عمل می‌نمایند که توسط حالت P2 کنترل می‌شود. حالت P2 در هر زمان می‌تواند دارای توکن شود، در این حالت گذار شرط فعال شده و به محض فعال شدن آتش می‌شود. در این صورت توکن

از MTT خارج شده و MTT غیر فعال خواهد شد. این اتفاقات در صورتی رخ خواهد داد که MTT آتش نشده باشد، زیرا در صورت آتش شدن MTT و از بین رفتن توکن در P1، گذار شرط نمی‌تواند فعال شود. در نتیجه، در صورتی که P2 خالی باشد، نتیجه XOR، آتش شدن MTT خواهد بود در غیر این صورت گذار شرط آتش خواهد شد. زمان MTT می‌تواند در هر لحظه به مقدار جدیدی تغییر داده شود که از خواص این نوع گذار می‌باشد. این کلاس از شبکه‌های پتری منجر به یک شبکه سازگار یا بدون انتخاب می‌شود که نتیجه انتخاب توسط P2 کنترل می‌شود.



شکل ۴-۲۰

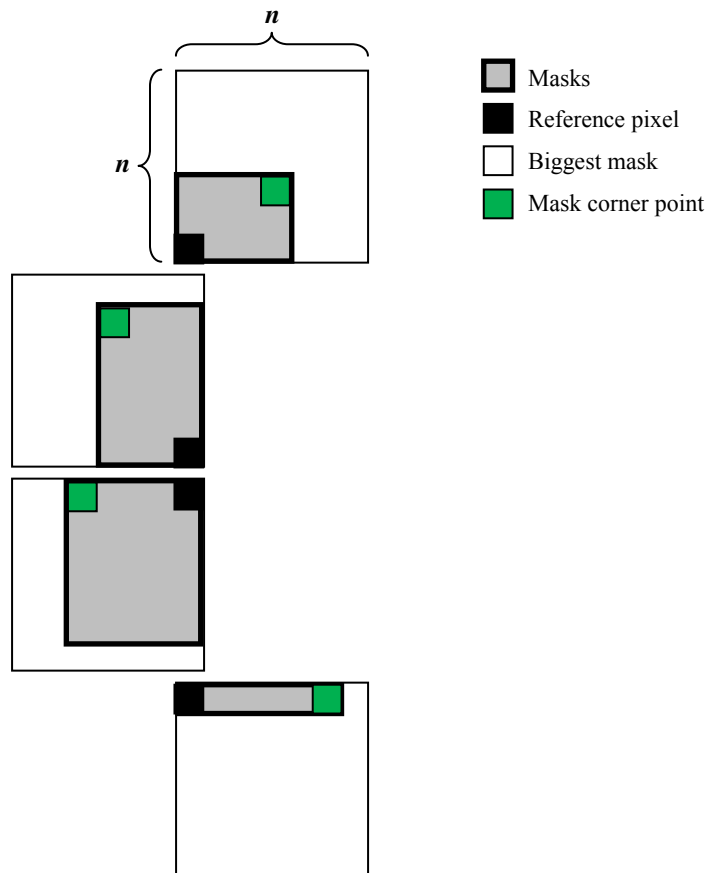
۴-۹- پیاده‌سازی پتری

برنامه‌ریزی سطح بالای ربات‌های صنعتی به منظور کاربردهای رباتیکی پیچیده بسیار توسعه یافته است. در این بخش می‌خواهیم، روش طراحی مسیر پیشنهادی و کنترل حرکت ربات را بر اساس شبکه پتری، پیاده‌سازی نماییم. روش پیشنهادی مسیری را بر اساس مسیر از پیش تعیین شده طراحی می‌نماید که منجر به کاهش زیاد معادلات سینماتیک معکوس می‌شود. در نتیجه ربات می‌تواند به صورت زنده و سریع عملیات مطلوب را انجام دهد. همچنین، مساله اساسی در حرکت ربات، هماهنگی در مفاصل می‌باشد. یعنی هر مفصل می‌بایست به صورت هماهنگ و وابسته به مفاصل دیگر حرکت نماید، در حالیکه

هیچ‌گونه وابستگی مکانیکی مابین این مفاصل وجود ندارد. بر اساس روش پیشنهادی زمانی که این هماهنگی به وجود آید، ربات‌ها می‌توانند عملیاتی چون، دنبال کردن هدف، جابه‌جایی اجسام، جاروب نمودن اجسام و ... را سریع و دقیق انجام دهد. روش پیشنهادی بر روی یک ربات کارتزین دو لینیکی پیاده‌سازی شده است که در دو بخش بعد به توضیح آن می‌پردازیم. پیش از توضیح پیاده‌سازی روش پیشنهادی به تعریف دیگری می‌پردازیم. کوچکترین حرکت یک مفصل، یک قطعه حرکت نامیده می‌شود، که در محرکه‌های پله‌ای و یا در محرکه‌های دیگر رزولوشن فیدبک می‌تواند به عنوان قطعه حرکت انتخاب شود. قطعه حرکت زمانی برآورده می‌شود که گذار مربوط به آن آتش شده باشد.

۱-۹-۴- پیاده‌سازی فرآیند ساخت ماسک

قبل از هر گونه توضیحی در رابطه با پیاده‌سازی فرآیند ساخت ماسک، فرضیاتی را انجام می‌دهیم که در ساده‌سازی پیاده‌سازی مؤثر می‌باشند. ابتدا فرض می‌نماییم که مسیر مطلوب به صورت یک تصویر می‌باشد. در این صورت، P پس از گسسته‌سازی مسیر مطلوب و تبدیل آن به مسیری با ضخامت یک پیکسل بدست خواهد آمد که، پیکسل‌های مسیر همسایه‌های هشت‌تایی می‌باشند. دوم، ماسک یک آرایه دو بعدی از پیکسل‌ها می‌باشد که نقطه مرجع آن یکی از چهار پیکسل گوشه‌ای آن است. یک ماسک توسط پیکسل مرجع و اندازه ماسک (تعداد پیکسل) در طول محورهای دستگاه مختصات کارتزین، به طور کامل مشخص می‌شود. اگر n بیشترین اندازه ماسک در راستای محورهای x , y باشد، تعداد $4 \times n \times (n-1)$ ماسک ممکن حول یک نقطه مرجع وجود خواهد داشت. توجه نمایید که ماسکی با ابعاد 1×1 وجود ندارد. شکل ۴-۲۱ چهار ماسک توسط مربع سفید نمایش داده شده و همچنین نقطه گوشه‌ای مقابل با رنگ سبز نشان داده شده است. زمانی که یک ماسک بر روی مسیر قرار می‌گیرد، قسمتی از مسیر که زیر ماسک قرار گرفته مورد پردازش قرار می‌گیرد. سوم، خطای مسیر می‌تواند در راستای هر کدام از محورهای x یا y محاسبه شود. اگر بخواهیم خطا را در راستای محور x محاسبه نماییم، می‌بایست خطا



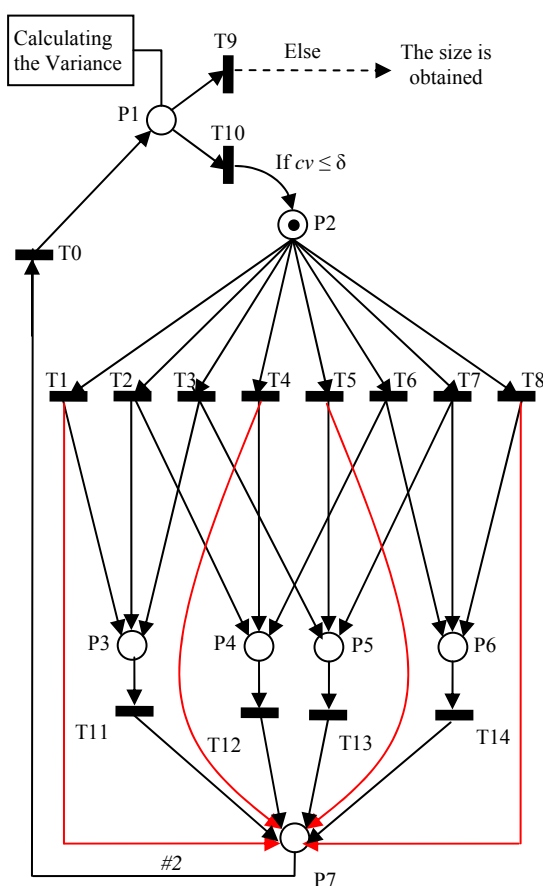
شکل ۴- ۲۱

در راستای هر پیکسل محاسبه شود، در این صورت تعداد پیکسل‌ها مابین مسیر اصلی و مسیر طراحی شده در راستای محور y خطا در پیکسل مورد نظر می‌باشد. در این حالت نیز علامت خطا مهم نمی‌باشد و در نظر گرفته نمی‌شود. همچنین خطا در راستای پیکسل‌های مرجع و گوشه‌ای مقابل محاسبه نمی‌شود، زیرا در این نقاط مسیر اصلی و مسیر طراحی شده هم‌پوشانی کامل دارند. در این حالت واریانس به صورت زیر محاسبه می‌شود.

$$Var = \sum_{i=1}^{y-2} \frac{\left(\frac{error_i}{x}\right)^2}{y-2} \quad (۴-۱۳)$$

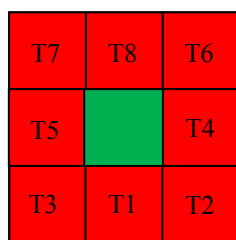
جایی که، I فاصله پیکسل از پیکسل مرجع را مشخص می‌نماید و Y تعداد پیکسل‌های ماسک در راستای محور Y و X تعداد پیکسل‌های ماسک در راستای محور X می‌باشند. چهارم، روش پیشنهادی را بر روی یک ربات کارت‌زین دو لینی با مفاصل کشویی پیاده‌سازی می‌نماییم. ربات مفروض، یک مکانیزم دو درجه آزادی می‌باشد که محدوده کاری آن یک صفحه دو بعدی است.

فرآیند ساخت ماسک توسط شبکه پتری که در شکل ۴-۲۲ پیاده‌سازی شده است. مدل شبکه پتری جهت پیاده‌سازی فرآیند بدست آوردن اندازه ماسک بر اساس واریانس از پیش تعیین شده، طراحی شده است. بر اساس واریانس محاسبه شده به ازای هر تغییر در ماسک، شبکه تصمیم می‌گیرد که آیا



شکل ۴-۲۲

اندازه ماسک می‌تواند افزایش یابد یا خیر. اندازه ماسک زمانی بدست می‌آید که واریانس محاسبه شده در ماسک بیشتر از واریانس از پیش تعیین شده باشد. در غیر این صورت، شبکه اندازه ماسک را افزایش می‌دهد تا زمانی که واریانس محاسبه شده بیشتر از واریانس از پیش تعیین شده باشد. عملکرد گذارها T1 تا T8 بر اساس شکل ۴-۲۳ می‌باشد، که نسبت به جهت انتخاب شده برای افزایش اندازه ماسک، یکی از گذارها آتش خواهد شد. حالت‌های شبکه و توضیحات آن‌ها در جدول ۴-۱ آورده شده است.



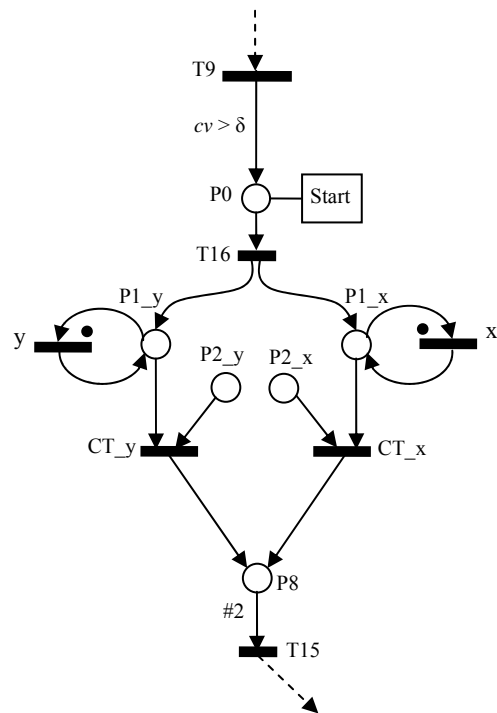
شکل ۴-۲۳

جدول ۴-۱

حالت	توضیح
P1	variance calculated
P2	$cv \leq \delta$
P3	mask size increase will happen in the third or fourth quadrant along y-axis
P4	mask size increase will happen in the first or fourth quadrant along x-axis
P5	mask size increase will happen in the second or third quadrant along x-axis
P6	mask size increase will happen in the first or second quadrant along y-axis
P7	mask size calculated

۲-۹-۴- پیاده‌سازی کنترل حرکت ربات

روش کنترل حرکت بازوی مورد نظر توسط شبکه پتری نشان داده شده در شکل ۴-۲۴ پیاده‌سازی شده است. شبکه شامل دو گذار MTT و دو گذار CT متناظر می‌باشد. ساختار شبکه به صورت شکل ۴-۲۰ پیکربندی شده است. گذار MTT و CT متناظر با آن توسط حروف X و Y برچسب شده‌اند که هر کدام حرکت در راستای محور برچسبشان را کنترل می‌نمایند. حالت P2 و برچسب متناظر با آن نظارت بر اندازه ماسک در راستای برچسب حالت دارد. حالت‌ها و توضیح آن‌ها در جدول ۴-۱ بیان شده‌اند. الگوریتم زیر روش پیشنهادی را بیان می‌کند.



شکل ۴-۲۴

۱- شروع

۲- مقدار پیش فرض واریانس معین شود.

- ۳- اندازه شروع ماسک 1×2 می باشد.
- ۴- نقطه مرجع ماسک روی نقطه شروع قرار گیرد.
- ۵- نقطه گوشه‌ای مقابل بدست می آید.
- ۶- واریانس محلی محاسبه شود.
- ۷- اگر واریانس از مقدار پیش فرض کوچکتر بوده به ۴ برو.
- ۸- در غیر این صورت ربات در راستای قطر حرکت می نماید.
- ۹- نقطه مرجع ماسک بعدی نقطه گوشه‌ای ماسک حاضر می باشد.
- ۱۰- در صورتی که نقطه مرجع ماسک جدید و نقطه نهایی مسیر بر هم منطبق نباشند به ۲ برو.
- ۱۱- در غیر این صورت، پایان.

جدول ۴-۲

حالت	توضیح
P0	mask obtained
P1_x	robot moves a "bit" along x-axis
P2_x	robot reaches the mask corner pixel along x-axis
P1_y	robot moves a "bit" along y-axis
P2_y	robot reaches the mask corner pixel along y-axis
P8	robot reaches the mask corner pixel

روش پیشنهادی رسیدن به نقطه پایان را در مسیر درست بر اساس واریانس از پیش تعیین شده ضمانت می نماید. همچنین معادلات سینماتیک معکوس را بسیار کاهش می دهد و حرکت همزمان را محقق می سازد.

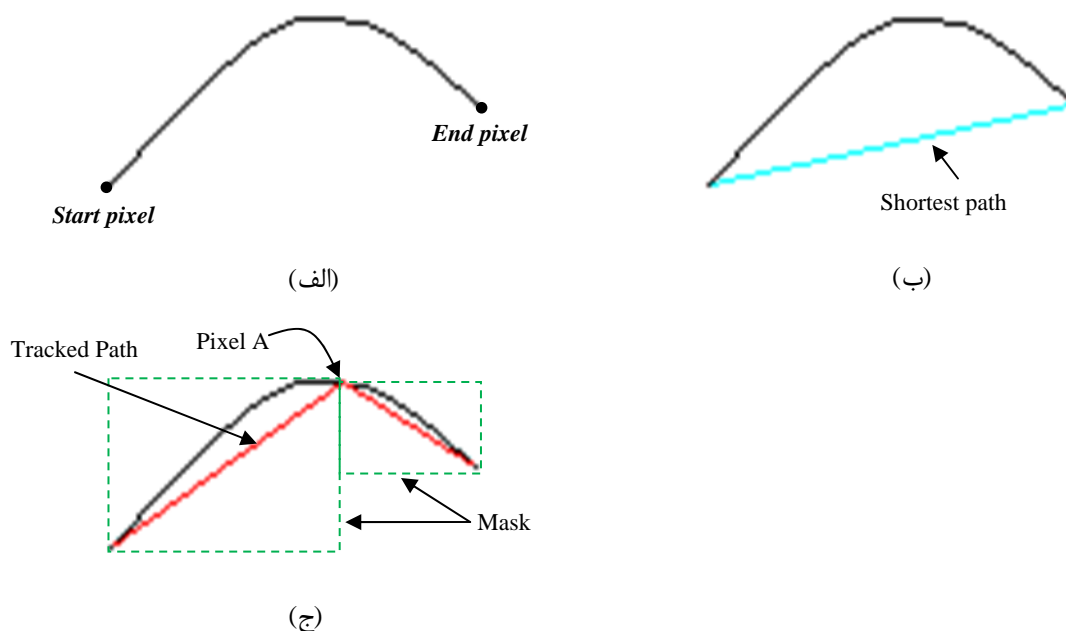
فصل پنجم

۱-۵- مقدمه

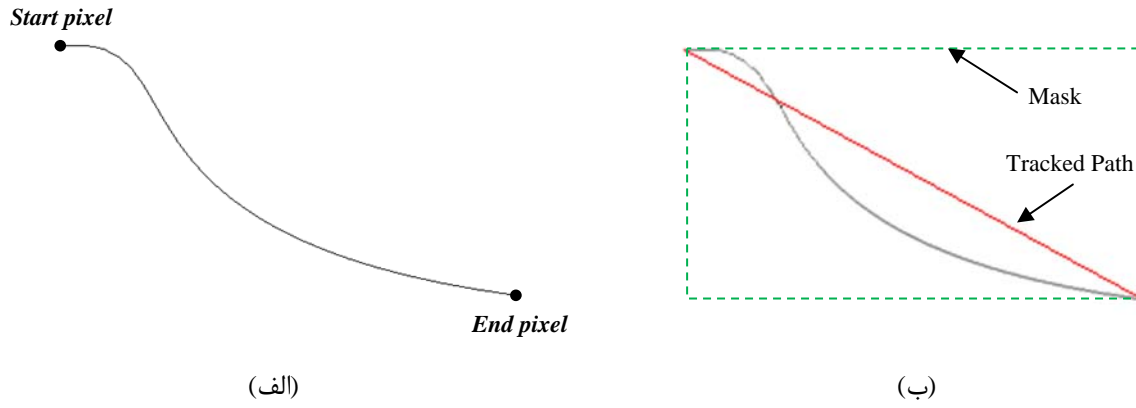
در این فصل به شبیه‌سازی روش‌های بیان شده در دو فصل اخیر و شرح نتایج تجربی آن‌ها می‌پردازیم. نتایج تجربی نشان می‌دهد که روش پیشنهادی بسیار کارآمد می‌باشد. همچنین، تعداد معادلات سینماتیک معکوس فوق‌العاده کاهش یافته است، در حالیکه دقت مسیر قابل تنظیم می‌باشد. نتایج نشان می‌دهد که، تعداد معادلات سینماتیک معکوس از دقت مطلوب بسیار تاثیر پذیر می‌باشد. بر اساس روش پیشنهادی، مسیر با هر شکل و رزولوشنی قابل پیمایش می‌باشد.

۲-۵- شبیه‌سازی و نتایج تجربی

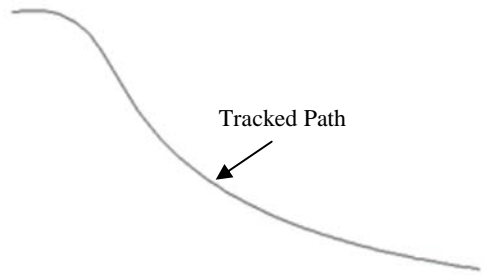
در این بخش روش پیشنهادی را بر روی مسیرهایی چون کمان ساده، خط مستقیم و مسیره‌ای پیچیده‌ای پیاده‌سازی می‌نماییم. همانطور که نتایج نشان می‌دهد، معادلات سینماتیک معکوس برای خطوط مستقیم تنها برای یک نقطه محاسبه می‌شود. شکل‌های نشان داده شده حاکی از دقت و سرعت روش پیشنهادی می‌باشد.



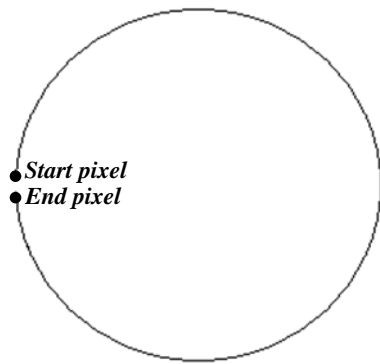
شکل ۱-۵- ۱ (الف) مسیر مطلوب (ب) کوتاه‌ترین مسیر (ج) مسیر پیموده شده



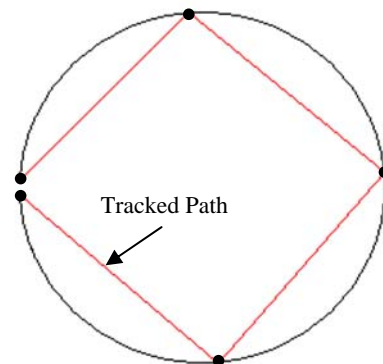
شکل ۲-۵ (الف) مسیر مطلوب (ب) مسیر پیموده شده



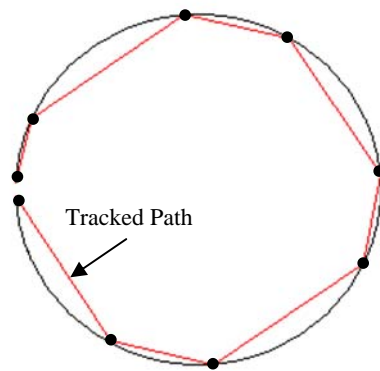
شکل ۳-۵ مسیر پیموده شده



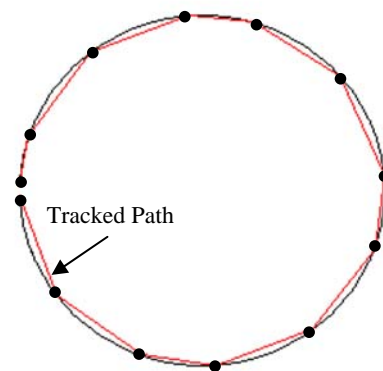
(الف)



(ب)



(ج)

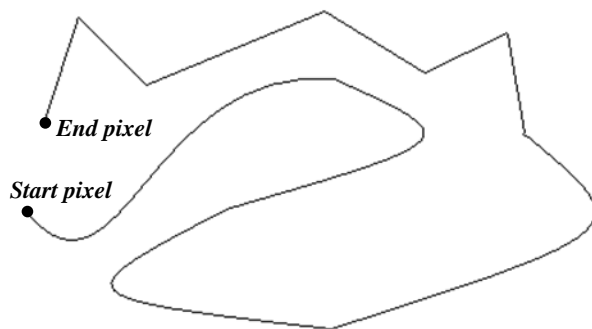


(د)

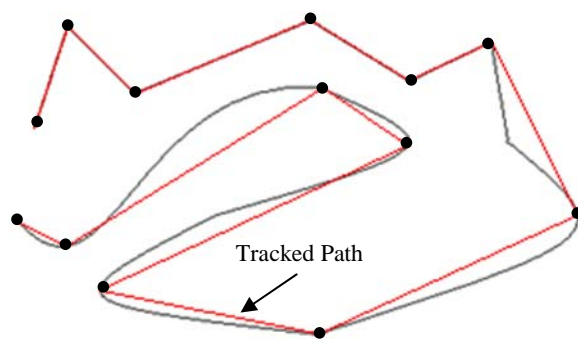


(ه)

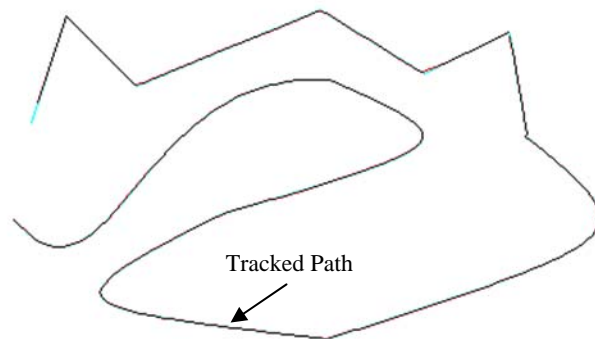
شکل ۴-۵ (الف) مسیر مطلوب ۲۲۹ پیکسل (ب) واریانس = $0/15$ بزرگترین واریانس محاسبه شده = $0/11$ (ج) واریانس = $0/078$ بزرگترین واریانس محاسبه شده = $0/0757$ (د) واریانس = $0/055$ بزرگترین واریانس محاسبه شده = $0/053$ (ه) واریانس = $0/046$ بزرگترین واریانس محاسبه شده = $0/045$



(الف)

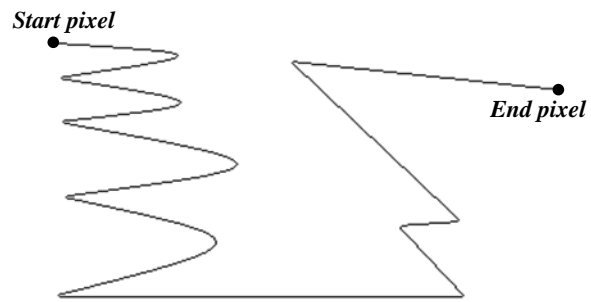


(ب)

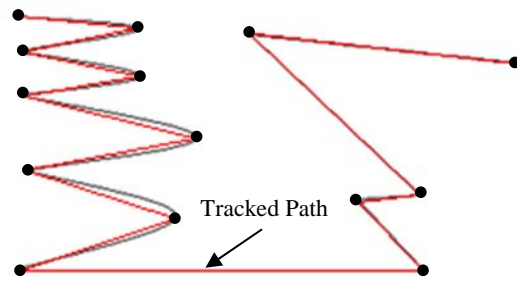


(ج)

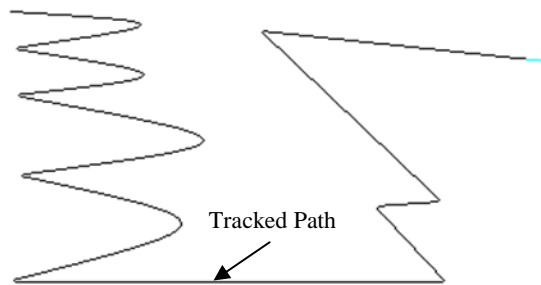
شکل ۵-۵ (الف) مسیر مطلوب ۹۶۶ پیکسل (ب) واریانس
 $= 0.08$ = بزرگترین واریانس محاسبه شده $= 0.0728$ (ج)
 واریانس $= 0.005$ = بزرگترین واریانس محاسبه شده $= 0.0046$



(الف)



(ب)



(ج)

شکل ۶-۵ (الف) مسیر مطلوب ۱۱۵۳ پیکسل (ب)
 واریانس = ۰/۰۵ بزرگترین واریانس محاسبه شده = ۰/۰۳
 (ج) واریانس = ۰/۰۰۵ بزرگترین واریانس محاسبه شده =
 ۰/۰۰۴۴

منابع

- [1] S. B. Niku, *Introduction to Robotics Analysis, Systems, Applications*, Upper Saddle River, NJ: Prentice-Hall, 2001.
- [2] G. V. Arnold, P. R. Henriques, and J. C. Fonseca, "A development approach to industrial robots programming," *Proc. 2003 ICAR*, Coimbra, Portugal, Jun-Jul, 2003.
- [3] H. J. Genrich and K. Lautenbach, "System modelling with high-level Petri nets," *Theoret. Comp. Sci.*, vol. 13, pp. 109-136, 1991.
- [4] R. David and H. Alla, "Petri nets for modelling of dynamic systems," *Automotica*, vol. 30, no. 2, pp. 175-202, 1994.
- [5] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *IEEE*, vol. 77, no. 1, pp. 81-97, Jan, 1989.
- [6] M. Lundgren, "Path tracking and obstacle avoidance for a miniature robot," M. S. thesis, Dept. Comput. Sci., Umea Univ., Umea, Sweden, 2003.
- [7] A. Stoytchev and R. Arkin, "Incorporating motivation in a hybrid robot architecture," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 8, no. 3, pp. 269-274, May, 2004.
- [8] K. H. Hong and T. G. Kim, "DEVSpecL: DEVS specification language for modeling, simulation and analysis of discrete event systems," *ELSEVIER Information and Software Technology*, vol. 48, pp. 221-234, 2006.
- [9] P. Makowski, A. Ravn, "Architecture and frameworks for autonomous robots," *Agrobotics Workshop*, Nov, 2004.
- [10] T. Murata, "Petri nets: properties, analysis and applications," *IEEE*, vol. 77, no. 4, pp. 541-580, 1989.
- [11] M. Adamou, A. Bourljault, and S. Zerhouni, "Modelling and control of flexible manufacturing assembly systems using object oriented Petri nets," *IEEE Int. Workshop on Emerging Technol. and Factory Automation*, Palm-Cove, Cairns, Australia, pp. 164-168, 1993.

- [12] S. A. Ben, M. Moalla, M. Courvoisier, and R. Valette, "Flexible manufacturing production system modelling using object Petri nets and their analysis," *IMACS Symp. MCTS*, Lille, France, pp. 553-560, 1991.
- [13] M. Akaza, D. I. Lee, S. Kumagai, and S. Kodama, "Application of timed marked graphs to a scheduling problem of production systems including repetitive processes with set-up times," *Modern Tools for Manufacturing Systems*, R. Zourawski, T. Dillon, and Eds. Amsterdam: Elsevier, 1993, pp. 263-277.
- [14] R. Y. Al-Jaar and A. A. Desrochers, "Performance evaluation of automated manufacturing systems using generalized stochastic Petri nets," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 621-639, 1990.
- [15] F. DiCesare and A. A. Desrochers, "Modelling, control, and performance analysis of automated manufacturing systems using Petri nets," *Control and Dynamic Systems*, C. T. Leondes, Ed. New York: Academic, vol. 47, pp. 121-172, 1991.
- [16] P. Freedman, "Time, Petri nets, and Robotics," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 417-433, 1991.
- [17] B. Berthomeiu, and M. Diaz, "Modelling and verification of time dependent systems," *IEEE Trans. Software Eng.*, vol. 17, no. 3, pp. 259-273, 1991.
- [18] N. Maragos, D. N. Kleftouris, and C. Ziogon, "A formal and executable model for path finding," in *Proc. 1st South-East European Workshop on Formal Methods (SEEFM)*, Thessaloniki, Greece, 2003.
- [19] D. A. Handelman, "Adaptive collaborative control of highly redundant robots," *Proc. SPIE Defense and security 2008*, vol. 6962, 69620Y, 2008.
- [20] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Comparative experimental evaluations of task space control with redundancy resolution," Aug. 2006.
- [21] O. B. Bayazit, D. Xie, and N. M. Amato, "Iterative relaxation of constraints: A framework for improving automated motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2005.
- [22] L. Han, "Hybrid probabilistic roadmap – Monte Carlo motion planning for closed chain systems with spherical joints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2004.
- [23] Y. Ting, W. I. Lei, and H. C. Jar, "A path planning algorithm for industrial robots," *Computer and Industrial Engineering*, vol. 42, pp. 299-308, 2002.
- [24] J. Trinkle and R. Milgram, "Complete path planning for closed kinematic chains with spherical joints," *Int. J. Robot. Res.*, vol. 21, no. 9, pp. 773-789, 2002.

- [25] R. Milgram, and J. Trinkle, “The geometry of configuration spaces for closed chains in two and three dimensions,” *Homology Homotopy and Applications*, 2002.
- [26] L. Han, L. Rudolph, J. Bluementhal, and I. Valodzin, “The inverse kinematics of a serial chain with spherical joints,” 2007, submitted.
- [27] ———, “Satisfied configuration space and path planning for a planner closed chain with revolute joints,” in *Proc. 7th International Workshop on Algorithmic Foundation of Robotics*, 2006.
- [28] H. Wang, Y. T. Chin, T. T. Ng, and S. Y. Ong, “Target tracking and pursue using laser scanner,” *BITEC. 2nd Asian Symposium on Industrial Automation and Robotics*, Bangkok, Thailand, May, 2001.
- [29] H. Toossian Shandiz, S. Arastehfar, “Two links robot controlling based on time Petri net,” *CSNDSP 6th International Symposion*, Graz, Austria, Jul, 2008, pp. 609-612.
- [30] R. Berlin, and G. L. Bergqvist, “Accurate robot and workcell simulation based on 3D laser scanning,” *Proc. ISR, 33rd International Symposium on Robotics*, Oct, 2002.
- [31] S. Roumeliotis and G. Bekey, “Distributed multi-robot localization,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 781-795, Oct, 2002.
- [32] O. Khatib, L. Sentis, J. Park, and J. Warren, “Whole-body dynamic behavior and control of human-like robots,” *Int. J. of Humanoid Robotics*, vol. 1, no. 1, pp. 29-43, 2004.
- [33] J. Nielsen and B. Roth, “On the kinematic analysis of robotic mechanisms,” *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1147-1160, 1999.
- [34] M. Raghavan and B Roth, “Inverse kinematics of the general 6R manipulator and related linkages,” *J. Mechanical Design*, vol. 115, pp. 502-508, 1993.
- [35] D. Manocha and J. Canny, “Efficient inverse kinematics for general 6R manipulators,” *IEEE Trans. Robot. Automat.*, vol. 10, pp. 648-657, 1994.
- [36] J. Porta, L. Rose, and F. Thomas, “Inverse kinematics by distance matrix completion,” *Proc. International Workshop on Computational Kinematics*, 2005.
- [37] F. G. Pin, L. J. Love, and D. L. Jung, “Automate kinematic equations generation and constrained motion planning resolution for modular and reconfigurable robots,” *27th ACS National Meeting*, Anaheim, California, Mar-Apr, 2004.
- [38] ———, “Multiple-loop position analysis via iterated linear programming,” *Robotics: Science and Systems*, 2006.

- [39] M. C. cavusoglu, I. Villanueva, and F. Tendick, "Workspace analysis of robotic manipulators for a teleoperated suturing task," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robot. Syst.*, Maui, Hawaii, USA, Oct-Nov, 2001.
- [40] Capek, Karel, *Rossum's Universal Robots*, Translated by Paul Selver, Doubleday, New York, 1923.
- [41] Denavit, J., R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics*, pp. 215-221, Jun, 1955.
- [42] Wiitala, Jared M., B. J., Rister, and J. P. Schmiedler, "A more flexible robotic wrist," *Mechanical Engineering*, pp. 78-80, Jul, 1997.
- [43] C. A. Petri, "Communication with automata," New York: Griffiss Air force Base, RADCTR, vol. 1, 1966.
- [44] ———, "Fundamentals of a theory of asynchronous information flow," *IFIP Congress 62*, pp. 386-390, 1963.
- [45] A. W. Holt, R. Shapiro, and S. Warshall, Final report of the information systems theory project, New York: Griffiss Air force Base, Sep, 1968.
- [46] A. W. Holt and F. Commoner, "Events and conditions," *MAC Conf.*, pp. 3-52, 1970.
- [47] A. W. Holt, "Introduction to occurrence systems," *American Elsevier*, pp. 175-203, 1971.
- [48] F. Commoner, A. W. Holt, and A. Pnueli, "Marked directed graphs," *J. Comput. Syst. Sci.*, vol. 5, pp. 511-523, 1971.
- [49] J. L. Peterson, *Petri net Theory and the Modeling of Systems*, Prentice Hall, Inc., 1986.
- [50] W. Reisig, *Petri nets*, *EATCS*, vol. 4, 1985.
- [51] T. Agerwala, "Putting Petri nets to work," *Computer*, vol. 12, no. 12, pp. 85-94, Dec, 1979.
- [52] R. Johnsonbaugh and T. Murata, "Petri nets and marked graphs—Mathematical models of concurrent computation," *The American Math. Monthly*, vol. 89, no. 8, pp. 552-566, Oct, 1982.
- [53] J. L. Peterson, "Petri nets," *ACM*, vol. 9, no. 3, pp. 223-252, Sep, 1977.
- [54] W. Brauer, Ed., *Net Theory and Applications*, *LNCS*, vol. 84, 1980.

Abstract

This thesis describes a new adaptive time based control Petri net model for robot path tracking. It introduces a new description of robot motions, which is as fast as Joint-space description and as accurate as Cartesian-space description. A new description of path planning named mask description has been introduced, which reduces inverse kinematics calculations.

Masks are constructed adaptively based on desired accuracy to obtain adjustable path planning. We describe how to specify a path based on original path in order to reduce the inverse kinematics calculations that are quite complicated and extremely time consuming. Kinematics redundant solutions and singular points are skipped based on the proposed method. Consequently, the robot can be moved online and faster. The method is independent of the path resolution. Based on the new description any path shape can be tracked in real time, by eliminating the complicated and time consuming inverse kinematics calculations in robot local positioning. To run actuators for coordinating robot motion a new class of timed Petri nets has been introduced. The proposed method can also be applied to other robotics control issues such as trajectory planning, velocity control and dynamics. The path planning process is adjusted based on variance of error, which is the criterion of our adaptive control method.