

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مکانیک

گروه مکاترونیک

عنوان:

طراحی مسیر دینامیکی و پیاده‌سازی ناوبری برای ربات سیار در یک محیط

ناشناخته با حضور موانع

دانشجو:

مهدی نظری سرچشمه

استاد راهنما:

دکتر مهدی بامداد

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

دی ۱۳۹۳



دانشگاه علمی کاربردی

مدیریت تحصیلات تکمیلی

فرم شماره (۶)

باسمه تعالی

شماره:

تاریخ:

ویرایش:

فرم صورت جلسه دفاع از پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) نتیجه ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای مهدی نظری سرچشمه رشته مکترونیک گرایش میکاترونیک تحت عنوان: طراحی مسیر دینامیکی و پیاده‌سازی ناوبری برای ربات سیار در یک محیط ناشناخته با حضور مواع که در تاریخ ۱۳۹۳/۱۰/۳۰ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با درجه بسیار خوب امتیاز ۱۸/۸۵)   دفاع مجدد  مردود

۱- عالی (۲۰ - ۱۹)

۲- بسیار خوب (۱۸/۹۹ - ۱۸)

۳- خوب (۱۷/۹۹ - ۱۶)

۴- قابل قبول (۱۵/۹۹ - ۱۴)

۵- نمره کمتر از ۱۴ غیر قابل قبول

امضاء	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
		دکتر مهدی بامداد	۱- استاندارها
			۲- استاد مشاور
	استاد	دکتر حبیب احمدی	۳- نماینده شورای تحصیلات تکمیلی
	دانشیار	دکتر علیرضا احمدی فرد	۴- استاد ممتحن
	دانشیار	دکتر سید علی سلیمانی ابوری	۵- استاد ممتحن

رئیس دانشکده: امضاء

ش. مردان



تقدیم به پدر و مادر عزیزم

که شمع وجودشان

گرمابخش و مشوق من

در تمامی مراحل زندگی بوده است.

## تشکر و قدردانی

با شکر به درگاه خداوند متعال که آنچه انجام شده با لطف و عنایت او بوده است. از استاد راهنمای ارجمندم آقای دکتر مهدی بامداد که در انجام این پایان نامه مشوق و راهنمای من بودند صمیمانه سپاسگذاری می کنم.

## تعهد نامه

اینجانب مهدی نظری سرچشمه دانشجوی دوره کارشناسی ارشد رشته مکترونیک دانشکده مکانیک دانشگاه صنعتی شاهرود نویسنده پایان نامه طراحی مسیر دینامیکی و پیاده‌سازی ناوبری برای ربات سیار در یک محیط ناشناخته با حضور موانع تحت راهنمایی دکتر مهدی بامداد متعهد می‌شوم.

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

تاریخ ۱۳۹۳/۱۰/۳۰

امضای دانشجو

### مالکیت نتایج و حق نشر

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

## چکیده

ناوبری در محیط پیرامونی برای هر وسیله متحرکی مهم است. اجتناب از موقعیت‌های پرخطر نظیر برخورد‌ها و شرایط غیر ایمن به یک مسئله بهینه‌سازی مسیر منجر می‌شود. این مسئله، طراحی مسیری است که تعدادی معیار را با رعایت شرایط مرزی کمینه و یا بیشینه می‌نماید.

مسئله طراحی مسیر بدون برخورد برای ربات‌های سیار در این رساله مطرح می‌شود و روش‌های تکاملی متعددی برای حل مسئله بهینه‌سازی به کار گرفته می‌شود. بعلاوه برای اجتناب از موانع، گرادیان تابع پتانسیل واقع شده روی موانع بر پایه الگوریتم میدان پتانسیل در نظر گرفته می‌شود.

این مسئله بهینه‌سازی می‌تواند با اثرات شیب سطح حرکت و دینامیک ربات تلفیق گردد. به‌منظور ارزیابی بازدهی انرژی مسئله جدیدی تدوین می‌گردد. تابع هدف بر پایه انرژی ربات و به‌منظور افزایش زمان اجرای عملیات ربات و شارژ باتری کمینه می‌شود. این انرژی اغلب در موتورها هدر می‌رود که این هدر رفت به منحنی سرعت ورودی ربات سیار وابسته است.

الگوریتم‌های ارائه شده برای تولید مسیر حرکت ربات‌های سیار چرخ‌دار، کلیت و بازدهی مناسبی دارند. آزمایشات عملی به‌منظور اعتبار سنجی رویه پیشنهاد شده انجام می‌شود. نتایج گزارشات با داده‌های حرکتی از ربات رومبای شرکت آی‌ربات جمع‌آوری می‌شود. تلاش‌ها روی بهبود این ربات خانگی به یک سامانه تحقیقاتی رباتیک متمرکز می‌شود تا از حسگرهای ربات بهره‌بردار. شبیه‌ساز گرافیکی برای دسترسی به جزئیات داده‌های مورد نیاز الگوریتم‌ها طراحی می‌شود.

شبیه‌سازی و آزمایشات تجربی صورت گرفته و مقایسه می‌شوند. انطباق آن‌ها نشان می‌دهد که عملکرد خوبی در حل مسائل بهینه‌سازی برای اجتناب از کمینه‌های محلی محقق شده است.

**کلمات کلیدی:** ناوبری، بهینه‌سازی، الگوریتم میدان پتانسیل، دینامیک ربات، انرژی، ربات سیار چرخ‌دار، منحنی سرعت، ربات رومبا، شبیه‌ساز گرافیکی

## فهرست فصل‌ها

فصل اول: مقدمه	۱
۱-۱- مقدمه	۲
۲-۱- مسیریابی به روش برون خط و در حالت جهانی برای ربات سیار	۳
۱-۲-۱- روش میدان پتانسیل	۴
۲-۲-۱- مسیریابی از طریق برنامه‌ریزی پویا	۴
۳-۲-۱- الگوریتم‌های تکاملی در مسیریابی	۴
۳-۱- مشکلات و محاسن مسیریابی در حالت برون خط	۵
۴-۱- مسیریابی به روش برخط (حالت محلی)	۵
۵-۱- ابزارهای محقق کننده مسیریابی برخط	۶
۶-۱- انرژی در ربات‌های سیار	۸
۷-۱- ساختار پایان نامه	۹
فصل دوم: معادلات حرکت ربات سیار چرخ‌دار	۱۱
۱-۲- مقدمه	۱۲
۲-۲- سیستم‌های مختصات	۱۲
۳-۲- مدل سینماتیکی ربات سیار	۱۳
۱-۳-۲- مدل سینماتیک مستقیم	۱۳
۲-۳-۲- قیود سینماتیکی	۱۴
۴-۲- مدل سازی دینامیکی ربات سیار	۱۶
۱-۴-۲- روش لاگرانژ	۱۶
۵-۲- مدل موتور ربات سیار دوچرخ	۲۲
فصل سوم: مسیریابی برای ربات سیار به روش الگوریتم میدان پتانسیل	۲۳
۱-۳- مقدمه	۲۴
۲-۳- میدان پتانسیل مصنوعی سنتی	۲۴



۳-۳- میدان پتانسیل مصنوعی بهبود داده شده	۲۸
۳-۳-۱- تعریف تابع پتانسیل جذب	۲۸
۳-۳-۲- تعریف تابع پتانسیل دافعه	۳۰
۳-۴- نتایج شبیه‌سازی	۳۰
۳-۴-۱- شبیه‌سازی با میدان پتانسیل بهبود داده شده	۳۱
۳-۴-۲- شبیه‌سازی با میدان پتانسیل سنتی	۳۲
فصل چهارم: مسیریابی برای ربات سیار با الگوریتم‌های تکاملی	۳۵
۴-۱- مقدمه	۳۶
۴-۲- اجزای مسئله طراحی مسیر	۳۶
۴-۲-۱- منحنی‌های بی‌اسپلین	۳۸
۴-۲-۲- الگوریتم‌های تکاملی و نحوه بهینه‌سازی	۳۹
۴-۳- نتایج شبیه‌سازی	۳۹
فصل پنجم: طراحی مسیر برای ربات با هدف کمینه انرژی مصرفی	۴۹
۵-۱- مقدمه	۵۰
۵-۲- مدل ربات سیار	۵۰
۵-۳- مصرف انرژی ربات سیار چرخ‌دار	۵۱
۵-۴- کمینه انرژی برای حرکت ربات سیار	۵۲
۵-۴-۱- حرکت در امتداد خط مستقیم	۵۲
۵-۴-۲- حرکت در امتداد مسیر منحنی	۵۳
۵-۴-۳- بیان مسئله	۵۳
۵-۵- نتایج شبیه‌سازی	۵۵
۵-۵-۱- شبیه‌سازی بر روی مسیر مستقیم	۵۵
۵-۵-۲- شبیه‌سازی حرکت منحنی‌الخط ربات	۵۷
فصل ششم: پیاده‌سازی‌های عملی	۶۳
۶-۱- مقدمه	۶۴

۶۴	۲-۶- اجزای داخلی و خارجی ربات رومبا .....
۶۵	۳-۶- شبیه‌ساز رومبا .....
۶۵	۱-۳-۶- هدف از شبیه‌ساز .....
۶۵	۲-۳-۶- فرضیات شبیه‌ساز .....
۶۶	۴-۶- پیاده‌سازی .....
۶۶	۱-۴-۶- محدودیت‌های عملی ربات رومبا .....
۶۶	۲-۴-۶- محیط ناشناخته .....
۶۶	۱-۲-۴-۶- محیط شبیه‌ساز .....
۶۷	۲-۲-۴-۶- پیاده‌سازی در محیط واقعی .....
۶۸	۳-۴-۶- محیط شناخته‌شده .....
۷۰	۵-۶- محاسبات انرژی مصرفی ربات .....
۷۰	۱-۵-۶- محاسبه انرژی مصرفی ربات بر روی سطوح صاف و شیب‌دار .....
۷۵	۲-۵-۶- بازده انرژی ربات برای مسیرهای مختلف .....
۷۸	۱-۲-۵-۶- شبیه‌سازی .....
۷۹	۲-۲-۵-۶- محاسبه بازده انرژی در روش مستطیلی .....
۸۱	فصل هفتم: نتیجه‌گیری و پیشنهادات .....
۸۲	۱-۷- نتیجه‌گیری .....
۸۴	۲-۷- پیشنهادات .....
۸۵	پیوست الف .....
۹۵	پیوست ب .....
۱۰۶	مراجع: .....

## فهرست جدول‌ها

- جدول ۴-۱: عبور منحنی از درون موانع و افزایش هزینه آن در مقایسه با مسیر مماس بر مانع ..... ۳۸
- جدول ۲-۲: مشخصات سیستم ..... ۴۰
- جدول ۴-۳: تأثیر افزایش نسل‌ها در پاسخ ..... ۴۱
- جدول ۴-۴: نتایج مسیر ربات بدون مانع ..... ۴۲
- جدول ۴-۵: نتایج مسیر ربات با یک مانع ..... ۴۲
- جدول ۴-۶: مقایسه الگوریتم‌های تکاملی با تعداد تکرار ۱۰۰ ..... ۴۶
- جدول ۴-۷: مقایسه الگوریتم‌های تکاملی با تعداد تکرار ۵۰ ..... ۴۶
- جدول ۵-۱: انرژی مصرفی ربات برای  $t_f$  و  $x_f$  مختلف ..... ۵۶
- جدول ۵-۲: انرژی مصرفی ربات برای  $t_f$  مختلف با سه مانع ..... ۵۸
- جدول ۵-۳: انرژی مصرفی ربات برای مسیرهای مختلف ..... ۵۹
- جدول ۵-۴: طول مسیر حرکت با الگوریتم‌های متنوع ..... ۵۹
- جدول ۶-۱: پارامترهای ربات سیار ..... ۷۰
- جدول الف-۱: مقایسه الگوریتم‌های تکاملی ..... ۹۱

## فهرست شکل ها

- شکل ۱-۲: سیستم مختصات ربات ..... ۱۲
- شکل ۲-۲: ساختار ربات سیار..... ۱۴
- شکل ۳-۲: قيود غير هولونوميك ربات سيار..... ۱۵
- شکل ۴-۲: مؤلفه سرعت ربات..... ۱۷
- شکل ۱-۳: محدودیت‌های میدان پتانسیل مصنوعی..... ۲۷
- شکل ۲-۳: تابع پتانسیل جاذبه..... ۲۹
- شکل ۳-۳: میدان پتانسیل جاذبه..... ۲۹
- شکل ۴-۳: مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل بهبود داده شده با یک مانع ..... ۳۱
- شکل ۵-۳: مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل بهبود داده شده ..... ۳۲
- شکل ۶-۳: مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل سنتی ..... ۳۳
- شکل ۷-۳: قرار گرفتن ربات در کمینه محلی..... ۳۳
- شکل ۱-۴: منحنی درون مانع نزدیک‌ترین حالت به مرکز..... ۳۷
- شکل ۲-۴: منحنی درون مانع دور از مرکز..... ۳۷
- شکل ۳-۴: منحنی خارج از مانع و تقریباً مماس به مرز آن ..... ۳۸
- شکل ۴-۴: ۲۰۰ نسل با طول مسیر ۱۴.۹ ..... ۴۱
- شکل ۵-۴: ۱۰۰ نسل با طول مسیر ۱۵.۲۳ ..... ۴۱
- شکل ۶-۴: مسیر ربات خالی از مانع..... ۴۱
- شکل ۷-۴: مسیر ربات با یک مانع..... ۴۲
- شکل ۸-۴: مسیر ربات با هفت مانع..... ۴۳
- شکل ۹-۴: مقایسه مسیر با هفت مانع براساس تابع هزینه..... ۴۳

- شکل ۴-۱۰: مقایسه مسیر با هفت مانع براساس انحراف از معیار..... ۴۳
- شکل ۴-۱۱: مسیر ربات با سه مانع..... ۴۴
- شکل ۴-۱۲: مقایسه مسیر با سه مانع براساس تابع هزینه..... ۴۴
- شکل ۴-۱۳: مقایسه مسیر با سه مانع براساس انحراف از معیار..... ۴۴
- شکل ۴-۱۴: مسیر ربات با چهار مانع..... ۴۵
- شکل ۴-۱۵: مقایسه مسیر با چهار مانع براساس تابع هزینه..... ۴۵
- شکل ۴-۱۶: مقایسه مسیر با چهار مانع براساس انحراف از معیار..... ۴۵
- شکل ۴-۱۷: مسیر خروجی BA..... ۴۷
- شکل ۴-۱۸: مسیر خروجی GA..... ۴۷
- شکل ۴-۱۹: مسیر خروجی ICA..... ۴۷
- شکل ۴-۲۰: مسیر خروجی PSO..... ۴۷
- شکل ۴-۲۱: مراحل مسیریابی با الگوریتم‌های تکاملی..... ۴۸
- شکل ۵-۱: بلوک دیاگرام دینامیک ربات سیار چرخ‌دار..... ۵۱
- شکل ۵-۲: منحنی سرعت برای  $\gamma = \tau_v/t_f$  متنوع..... ۵۵
- شکل ۵-۳: منحنی سرعت برای زمان‌ها و مسافت‌های متنوع..... ۵۶
- شکل ۵-۴: شبیه‌سازی برای  $x_f = 5m$  و  $t_f = 10s$ ..... ۵۷
- شکل ۵-۵: مسیر حرکت ربات با سه مانع..... ۵۸
- شکل ۵-۶: منحنی سرعت برای زمان‌های مختلف..... ۵۸
- شکل ۵-۷: حرکت ربات در مسیرهای متنوع..... ۵۹
- شکل ۵-۸: منحنی سرعت برای مسیرهای متنوع..... ۵۹
- شکل ۵-۹: مسیر حرکت ربات با الگوریتم‌های متنوع..... ۵۹

- شکل ۵-۱۰: مسیر حرکت بات با یک مانع ..... ۶۰
- شکل ۵-۱۱: منحنی سرعت ..... ۶۰
- شکل ۵-۱۲: ورودی‌های کنترلی موتور راست و چپ ..... ۶۰
- شکل ۵-۱۳: جریان آرمیچر چرخ‌های راست و چپ ..... ۶۱
- شکل ۵-۱۴: گشتاور بهینه وارد بر چرخ راست و چپ ..... ۶۱
- شکل ۵-۱۵: مسیر حرکت ربات با سه مانع ..... ۶۱
- شکل ۵-۱۶: منحنی سرعت برای دو مسیر A و B ..... ۶۲
- شکل ۵-۱۷: گشتاور بهینه وارد بر چرخ راست و چپ برای مسیر A ..... ۶۲
- شکل ۵-۱۸: گشتاور بهینه وارد بر چرخ راست و چپ برای مسیر B ..... ۶۲
- شکل ۶-۱: نمایش دیاگرام داخلی رومبا ..... ۶۴
- شکل ۶-۲: حرکت ربات در محیط شبیه‌ساز با سه مانع ..... ۶۷
- شکل ۶-۳: مسیر حرکت ربات در محیط شبیه‌ساز ..... ۶۷
- شکل ۶-۴: حرکت ربات در محیط واقعی با فیدبک گرفتن از حسگرها ..... ۶۷
- شکل ۶-۵: مراحل حرکت ربات رومبا با فیدبک گرفتن از سنسورها در محیط واقعی ..... ۶۸
- شکل ۶-۶: مسیر حرکت ربات در محیط شبیه‌ساز ..... ۶۹
- شکل ۶-۷: رگرسیون خطی ..... ۶۹
- شکل ۶-۸: مسیر حرکت ربات در محیط واقعی ..... ۶۹
- شکل ۶-۹: پنل کنترلی رومبا در محیط سی‌شارپ ..... ۷۰
- شکل ۶-۱۰: پروفیل سرعت اعمالی به ربات ..... ۷۱
- شکل ۶-۱۱: جریان مصرفی باتری در دو حالت شبیه‌سازی و تجربی ..... ۷۱
- شکل ۶-۱۲: انرژی مصرف‌شده باتری در دو حالت شبیه‌سازی و تجربی ..... ۷۱

- شکل ۶-۱۳: حرکت ربات روی سطح شیب‌دار در محیط واقعی ..... ۷۲
- شکل ۶-۱۴: پروفیل سرعت اعمالی به ربات روی سطح شیب‌دار با  $x_f = 1m$  و  $t_f = 4sec$  ..... ۷۲
- شکل ۶-۱۵: جریان مصرفی باتری ربات روی سطح شیب‌دار با زوایای متنوع ..... ۷۲
- شکل ۶-۱۶: انرژی مصرف‌شده باتری ربات روی سطح شیب‌دار با زوایای متنوع ..... ۷۳
- شکل ۶-۱۷: پروفیل سرعت اعمالی به ربات بر روی سطح شیب‌دار با  $x_f = 1m$  و  $t_f = 6sec$  ..... ۷۳
- شکل ۶-۱۸: جریان مصرفی باتری ربات بر روی سطح شیب‌دار با زوایای متنوع ..... ۷۳
- شکل ۶-۱۹: انرژی مصرف‌شده باتری ربات روی سطح شیب‌دار با زوایای متنوع ..... ۷۴
- شکل ۶-۲۰: حرکت ربات روی سطح شیب‌دار در محیط واقعی ..... ۷۴
- شکل ۶-۲۱: جریان مصرفی باتری ربات در راستای سطح شیب‌دار ..... ۷۴
- شکل ۶-۲۲: انرژی مصرف‌شده باتری ربات در راستای سطح شیب‌دار ..... ۷۴
- شکل ۶-۲۳: توان برحسب سرعت ..... ۷۶
- شکل ۶-۲۴: نواحی پوشش داده شده ..... ۷۷
- شکل ۶-۲۵: بازدهی انرژی برحسب ناحیه تحت پوشش ..... ۷۸
- شکل ۶-۲۶: بازدهی انرژی برحسب سرعت ربات ..... ۷۹
- شکل ۶-۲۷: بازدهی انرژی برحسب ارتفاع در روش اسکن خطوط ..... ۷۹
- شکل ۶-۲۸: نمودار توان مصرفی ربات در حرکت مستطیلی ..... ۸۰
- شکل الف-۱: مراحل الگوریتم ازدحام ذرات ..... ۹۲
- شکل الف-۲: مراحل الگوریتم ژنتیک ..... ۹۲
- شکل الف-۳: مراحل الگوریتم رقابت استعماری ..... ۹۳
- شکل الف-۴: مراحل الگوریتم زنبور عسل ..... ۹۴
- شکل ب-۱: نمایش سنسور دیوار ..... ۹۸
- شکل ب-۲: محیط سی‌شارپ ..... ۹۹

شکل ب-۳: تنظیمات مربوط به ارتباط سریال ..... ۹۹

شکل ب-۴: کدهای اعمالی به ربات ..... ۱۰۰

شکل ب-۵: بسته‌های اطلاعاتی سنسور رومبا ..... ۱۰۴



## فهرست علائم

ثابت برگشتی موتور	$K_b$	مختصات ربات	$q$
ضریب گشتاور موتور	$K_t$	زاویه دوران	$\theta$
نسبت چرخ‌دنده	$n_g$	سرعت زاویه‌ای ربات	$w$
جریان آرمیچر	$I_a$	سرعت انتقالی ربات	$v$
ورودی کنترل	$u$	ماتریس دوران	$R$
ماتریس اینرسی موتور	$J$	شعاع چرخ	$r$
ضریب اصطکاک ویسکوز موتور	$F_v$	فاصله مرکز جرم ربات تا چرخ‌ها	$a$
تابع پتانسیل	$U$	فاصله هر چرخ تا مرکز ربات	$b$
نیرو پتانسیل	$F$	جرم ربات	$m$
فاصله ربات تا هدف	$\rho_g$	ماتریس اینرسی	$M$
تابع جریمه	$W$	ماتریس مرتبط با شتاب کوریولیس	$V$
تابع هزینه	$C$	ماتریس اصطکاک سطح	$F$
شعاع مانع	$r_{ob}$	ماتریس جاذبه	$G$
فاصله هر نقطه از مرکز مانع	$R_{dis}$	ماتریس انتقال سرعت	$B$
انرژی کل مصرف‌شده باتری ربات	$E_W$	گشتاور ورودی	$\tau$
انرژی تلف‌شده در مقاومت آرمیچر	$E_R$	ماتریس قیود	$A$
انرژی جنبشی ذخیره‌شده در ربات	$E_K$	بردار نیرو مرتبط با قیود	$\lambda$
توان	$p$	اینرسی ربات	$I$
		مقاومت آرمیچر	$R_a$



# فصل اول

مقدمه

## ۱-۱- مقدمه

رباتیک علمی است که باهدف راحتی انسان و افزایش وقت مفید او به وجود آمده است. دنیای ربات تلفیقی از الکترونیک، مکانیک، نرم‌افزار و سخت‌افزار هست که روزه‌روز در حال گسترش و تکامل است. یکی از مسائل مشترک در کاربردهای رباتیک جابجایی ربات است. زمانی که ربات می‌خواهد از یک نقطه به نقطه دیگری حرکت کند، ملاحظات مختلفی از جمله عدم برخورد با اجسام دیگر و نیز صرف انرژی کمتر اهمیت دارد.

یک ربات سیار<sup>۱</sup> باید قادر به انتخاب مسیر به‌گونه‌ای باشد که با حرکت از یک نقطه مشخص شروع و به سمت نقطه معین و معلوم هدف حرکت کرده و با موانع موجود در محیط برخورد نکند. به‌عبارت‌دیگر در این فرآیند سعی می‌کند پاسخ این سؤال را بیابد که چگونه می‌تواند با ایمنی کامل به هدف برسد؛ پس در طراحی مسیر مسئله اساسی یافتن یک حرکت ایمن و انجام‌پذیر برای رسیدن به هدف با در نظر گرفتن ملاحظات و قیود خاص است.

در یک مسئله رباتیک، بسته به عملیاتی که ربات موظف به انجام آن است کمیت‌ها و پارامترهای مختلفی بایستی مدنظر قرار گیرد. تعدادی از این کمیت‌های مهم که در ساخت نقشه بایستی مورد توجه قرار گیرند؛ عبارت است از: انرژی مصرف‌شده، مسافت طی شده و کیفیت نقشه تولید شده است. در بعضی از الگوریتم‌ها ممکن است هدف طی کردن کوتاه‌ترین مسیر و انجام عملیات مربوطه باشد و در بعضی دیگر هدف طی کردن مسیریابی توسط ربات که انرژی مصرفی ربات در آن مسیرها به حداقل برسد. ربات‌های سیار و ماشین‌های هدایت شونده‌ی خودکار<sup>۲</sup> در کارخانه‌ها برای کارهای مختلف از جمله انتقال مواد مورد استفاده قرار می‌گیرند. این ماشین‌ها برای حرکت و ناوبری<sup>۳</sup> نیاز به طرح‌ریزی مسیر<sup>۴</sup> و حرکت<sup>۵</sup> دارند. طرح‌ریزی و مسیریابی برای ربات سیار در یک محیط دوبعدی که مشخصات موانع موجود برای

---

<sup>1</sup> - Mobile Robot

<sup>2</sup> - Automated Guide Vehicle

<sup>3</sup> - Navigation

<sup>4</sup> - Path Planning

<sup>5</sup> - Motion Planning

آن، معلوم است بدین صورت تعریف می شود که با فرض موقعیت یا وضعیت های اولیه و همچنین وضعیت نهایی برای سیستم ربات، هدف این است که مشخص شود آیا یک مسیر پیوسته بدون برخورد با موانع برای سیستم ربات از وضعیت اولیه به وضعیت نهایی وجود دارد؟ اگر وجود دارد، چنین حرکتی طرح-ریزی شود. این مسئله که به عنوان مسیریابی استاندارد شناخته شده است، می تواند به روش ها و حالت های دیگری نیز تعمیم یابد، برای مثال اگر مشخصات محیط کاملاً برای سیستم ربات شناخته شده نباشد، در این حالت تولید مسیر به مرور زمان و با در نظر گرفتن اطلاعات از محیط و بروز در آوردن دینامیک مدل کلی (جهانی)<sup>۱</sup>، بهنگام<sup>۲</sup> می شود [۱]. مسئله ی جالب دیگر بسط و توسعه مسئله مسیریابی استاندارد به حالتی است که محیط شامل موانع متحرک شناخته شده و یا قابل پیش بینی باشد [۱، ۲]. مسیریابی برای ربات متحرک به دو حالت تقسیم می گردد. یک حالت مسیریابی جهانی است که بر پایه اطلاعات کامل اولیه از محیط می باشد. حالت دیگر مسیریابی محلی<sup>۳</sup> است که بر پایه اطلاعات حسگرهای گرفته شده از محیط است؛ به طوری که محل و شکل موانع، ناشناخته است. ناوبری ربات سیار که بر پایه این نوع مسیریابی انجام می گیرد، ناوبری به شکل برخط (محلی) نام گرفته است. هر کدام از شکل های فوق ناوبری ربات سیار دارای معایب و محاسنی است که در ادامه به آن پرداخته می شود.

## ۱-۲- مسیریابی به روش برون خط<sup>۴</sup> و در حالت جهانی برای ربات سیار

زمانی که مشخصات محیط برای ربات سیار خودکار کاملاً شناخته شده باشد و محیط یک وضعیت ساکن و مشخص داشته باشد، می توان به صورت برون خط به ایجاد و تولید مسیر حرکت برای ربات سیار پرداخت. این حالت از مسیریابی را مسیریابی جهانی نامیده اند. در این حالت به سبب مشخص بودن کل محیط می توان یک مسیر بهینه از نقطه مبدا به نقطه ی مقصد را پیدا نمود که شامل محدودیت هایی نیز باشد. در ادامه به بررسی برخی از روش های انجام شده در حالت برون خط پرداخته می شود.

---

<sup>1</sup> - Global

<sup>2</sup> - Update

<sup>3</sup> - Local

<sup>4</sup> - Off\_Line

## ۱-۲-۱- روش میدان پتانسیل<sup>۱</sup>

در این شیوه، تابع پتانسیل در فضای آزاد به صورت جمع پتانسیل‌های جاذب که ربات را به سمت موقعیت هدف می‌رانند و پتانسیل‌های دافع که آن را از موانع دور می‌کنند، تعریف می‌شوند و در هر تکرار، نیرویی مصنوعی که توسط تابع پتانسیل در موقعیت فعلی القا می‌شود، به عنوان جهت مناسب حرکت در نظر گرفته می‌شود. از مزیت‌های روش پتانسیل قابلیت تعمیم آن در فضاهای با ابعاد بالاتر است، اما امکان افتادن در کمینه محلی از نواقص آن است. امروزه از این روش با ترکیب روش‌های دیگر استفاده می‌شود. مثلاً مقالات [۳، ۴] ترکیبی از دو روش میدان پتانسیل و یادگیری تقویتی<sup>۲</sup> را برای فرار از قرار گرفتن در کمینه محلی استفاده می‌کنند. روش میدان پتانسیل بهبود داده شده می‌تواند این مشکل را برطرف کند.

## ۱-۲-۲- مسیریابی از طریق برنامه‌ریزی پویا<sup>۳</sup>

روش تغییراتی یا به عبارتی روشی که شامل رشد یک مسیر اولیه تا حد امکان ممکن و بدون برخورد با موانع برای مسیریابی است، توسط فوئرباخ استفاده شده است [۵]. این روش از استاندارد محاسباتی بر پایه گرادیان گرفته شده که در هر بار تکرار مسیر جاری به وسیله‌ی جستجوی برنامه‌ریزی پویای کلاسیک استفاده شده، در حالی که تا حد امکان سعی می‌شود از رشد نمایی حافظه جلوگیری شود.

در اینجا به جزئیات این الگوریتم اشاره نمی‌شود ولی به طور کلی در این روش یک مسیر اولیه تولید شده و با در نظر گرفتن تابع هزینه و نقاط برخورد مسیر با موانع، سعی در رشد و بهبود این مسیر اولیه است.

## ۱-۲-۳- الگوریتم‌های تکاملی<sup>۴</sup> در مسیریابی

برای پایین آوردن حجم محاسبات در روش برنامه‌ریزی پویا، می‌توان از الگوریتم‌های تکاملی استفاده کرد. نتایج به دست آمده نشان می‌دهد که حجم محاسبات به مراتب کمتر شده و مسیر به دست آمده نیز

<sup>1</sup> - Potential Field

<sup>2</sup> - Reinforcement Learning

<sup>3</sup> - Dynamic Programing

<sup>4</sup> - Evolutionary

خیلی نزدیک به مسیر بهینه‌ی کلی است. علاوه بر آن نیز می‌توان پارامترها و عوامل دیگری را در تابع هزینه برای بهینه نمودن آن پارامتر منظور کرد [۶].

### ۱-۳- مشکلات و محاسن مسیریابی در حالت برون خط

مسیریابی در حالت برون خط برای ناوبری برنامه‌ریزی شده<sup>۱</sup> مورد استفاده قرار می‌گیرد. نتیجتاً این ناوبری بر پایه مدل‌سازی محیط و تخمین موقعیت سیستم است. برای این حالت از ناوبری و بالطبع برای مسیریابی برون خط چند مشکل وجود دارد؛ اول اینکه یک توصیف از نقشه محیط در هر لحظه مورد نیاز است و این توصیف در هر لحظه باید بهنگام شود. در عمل، داشتن نقشه‌ی دقیق از محیط و بهنگام نگه داشتن آن به صورت مداوم امکان‌پذیر نیست برای آنکه شکل طرح‌ها و نقشه‌ها برای مدل کردن خیلی سخت است و اندازه‌گیری با دقت کل محیط مشکل است. دوم اینکه حتی اگر یک نقشه و نگاشت<sup>۲</sup> هم از محیط وجود داشته باشد، ربات به یک سیستم تخمین موقعیت دقیق نیاز دارد. اطلاعاتی که از حسگرهای داخلی سرعت و شتاب به دست می‌آید، برای به دست آوردن موقعیت، کافی نیستند زیرا با جمع شدن خطاها ممکن است مسئله برخورد پیش آید.

ولی از طرف دیگر در مسیریابی به صورت جهانی که به صورت برون خط انجام می‌شود، سعی در جستجو بر یک مسیر بهینه بر پایه محدودیت‌های ثابت (از جمله کوتاه‌ترین مسیر) که معمولاً داری هزینه است، صورت می‌گیرد و از طرف دیگر با اختلاط و کامل نمودن این حالت از مسیریابی با مسیریابی برخط، می‌توان به یک سرعت و قابلیت اطمینان بالایی برای ناوبری ربات سیار دست یافت.

### ۱-۴- مسیریابی به روش برخط (حالت محلی)

در ناوبری ربات سیار در محیط ناشناخته که نیاز به ناوبری به صورت محلی است، مسیریابی جهانی به آن صورت که قبلاً اشاره شد دیگر حائز اهمیت نیست و به علت ناشناخته بودن محیط پیدا کردن مسیر بهینه غیرممکن است. در واقع در این حالت مسیریابی همان عمل ناوبری ربات سیار از میان موانع در

<sup>1</sup> - Planned Navigation

<sup>2</sup> - Mapping

محیط می‌باشد و الگوریتم با ابزار هدایت ربات سیار، باید در هر لحظه جهت و سرعت ربات سیار را مشخص نماید که به این عمل طرح‌ریزی حرکت<sup>۱</sup> در زمان حقیقی<sup>۲</sup> گفته می‌شود. ولی با این حال چنانچه یک اطلاعات اولیه از محیط وجود داشته باشد، می‌توان با آن اطلاعات اولیه در ابتدای حرکت یک مسیر جهانی و برخط انجام داد و در حین حرکت در صورت نزدیک شدن به یک مانع و مشاهده آن که قبلاً ناشناخته بود، دوباره از آن موقعیت تا موقعیت نهایی مسیریابی دیگری انجام داد که مسلماً اگر زمان مسیریابی خیلی زیاد باشد، این روش به‌عنوان راه‌حل مناسبی نیست.

### ۱-۵- ابزارهای محقق‌کننده مسیریابی برخط

در ناوبری به‌صورت محلی، ربات سیار با دریافت اطلاعات از محیط به‌طور مداوم از میان موانع و بدون برخورد با آن‌ها، در مسیر و موقعیت هدف حرکت می‌کند؛ بنابراین توانایی ناوبری خودکار، بر پایه حسگرهای خارجی و نوع سیستم حسگری قرار دارد. به‌خصوص اگر یک‌راه حل با هزینه کمتر مطرح باشد. برحسب وضعیت مکانی<sup>۳</sup> محیطی که ربات در آن حرکت می‌کند، حسگرهای مختلفی مورد استفاده قرار می‌گیرند.

در بعضی از ناوبری‌های ربات سیار، در جاهای مختلف محیط، نشانه‌های طبیعی<sup>۴</sup> قرار می‌دهند که با تشخیص آن‌ها موقعیت ربات مشخص می‌گردد و یا اینکه ناوبری ربات سیار به روش دنبال کردن دیوار<sup>۵</sup> صورت می‌گیرد. که در صورت تشخیص مانع، آن را به‌عنوان یک دیوار دیگری در نظر می‌گیرد که نیاز به حسگرهایی برای مشخص کردن موقعیت سیستم یا تشخیص مانع در محیط می‌باشد [۷].

به سبب ایجاد سایه، در اشیاء نزدیک روی موانع دورتر نشانگر نصب‌شده، حسگرهایی با برد طولانی ممکن است مسئله کوری به وجود بیاورند و تنها حسگرهایی با برد کوتاه باعث کمتر شدن محدوده دید ربات، به‌خصوص در نواحی پهن و گسترده می‌گردد [۸].

---

<sup>۱</sup> - Motion Planning

<sup>۲</sup> - Real\_Time

<sup>۳</sup> - Topology

<sup>۴</sup> - Natural Landmark

<sup>۵</sup> - Wall Following



حسگرهای نوری همچون مادون قرمز<sup>۱</sup> یا حسگرهای عمقیاب لیزری<sup>۲</sup> کم‌هزینه، برای محیط‌های با نویز بیشتر، مورد اعتماد نیستند. همچنین استفاده از پردازش تصویر<sup>۳</sup> ممکن است یک روش و راه‌حل جالبی باشد. ولی نیاز به هزینه‌ی محاسباتی قابل توجه و شرایط روشن‌سازی<sup>۴</sup> مناسب دارد و از طرفی هرگز یک میدان دید کافی از محیطی که ربات در آن حرکت می‌کند، تهیه نمی‌کند [۸].

حسگرهای اولتراسونیک<sup>۵</sup> یک‌راه حل خوب و با هزینه کم پیشنهاد می‌کنند. با ادغام و استفاده چندین نوع از آن‌ها با محدوده‌های کاری مختلف، می‌توان مسافت‌های چند سانتیمتری و محدوده‌های متوسط شامل چند متر را تشخیص داد. بعضی از حسگرهای اولتراسونیک رفتار دیجیتالی دارند یعنی در صورت وجود جسمی در محدوده‌ی دید آن‌ها، سیگنال مبنی بر وجود مانع تولید می‌کنند و برخی دیگر نیز آنالوگ هستند که متناسب بافاصله مانع از ربات یک مقدار آنالوگ تولید می‌کنند [۹]. ولی این حسگرها نیز دارای مشکلاتی هستند که از آن جمله می‌توان به موارد زیر اشاره کرد:

۱- جهتی که این حسگرها از موانع می‌دهند ضعیف است که این امر دقت در تعیین موقعیت اصلی مانع را محدود می‌کند. مشکل دیگر، اطلاعات غلط مکرر حسگرها است که یا از نویز اولتراسونیک یا از انعکاس حسگرهای همسایه ناشی می‌شود. مشکل دیگر، انعکاس‌های بیش‌ازحد امواج اولتراسونیک از موانع می‌باشد. این امر زمانی اتفاق می‌افتد که زاویه اشعه و خط عمود بر سطح مانع، بزرگ باشد که در این صورت موجب عدم تشخیص موانع شده یا تنها قسمتی از سطح مانع تشخیص داده می‌شود.

۲- برای تخمین مسافت طی شده و سرعت ربات نیز می‌توان از انکودرهای افزایشی<sup>۶</sup> و نسبی<sup>۷</sup> استفاده کرد. انکودرهای افزایشی تنها به انتهای هر موتور محرک کوپل می‌شوند و از طریق آن

---

<sup>1</sup> - Infrared

<sup>2</sup> - Laser Range

<sup>3</sup> - Image Processing

<sup>4</sup> - Illumination

<sup>5</sup> - Ultrasonic Sensor

<sup>6</sup> - Incremental Encoder

<sup>7</sup> - Relative Encoder

می‌توان، کنترل روی سرعت موتور محرک داشت. انکودرهای نسبی نیز برای کنترل و اندازه‌گیری زاویه فرمان چرخ‌ها، بکار می‌رود. استفاده از شتاب‌سنج‌ها وژیروسکوپ سرعت زاویه‌ای که حساسیت بیشتری دارند، به‌منظور تخمین موقعیت سیستم ربات مناسب می‌باشند [۱۱].

## ۱-۶- انرژی در ربات‌های سیار

مصرف انرژی در ربات‌های سیار از راه‌های مختلف مانند استفاده از موتورهای دارای بازدهی بالا، بهبود دادن مدار راه‌انداز موتورها و پیدا کردن مسیرهای بهتر، می‌تواند کاهش یابد [۱۰، ۱۱]. موتورهای جریان مستقیم و میکروکنترلرها مصرف‌کننده اصلی انرژی هستند. تحقیقات نشان می‌دهد که مصرف توان در موتورهای جریان مستقیم بیش از هفتاد درصد توان کل است. سرعت موتور تا حد زیادی به تغییرات گشتاور مربوط می‌شود و انرژی تلف‌شده در موتور به منحنی سرعت بستگی دارد. از این رو برنامه‌ریزی حرکت با رویکرد بهینه‌سازی انرژی به‌وسیله بهینه‌سازی منحنی سرعت و کنترل ربات سیار در مسیرهای بهینه می‌تواند به زمان بیشتر برای انجام عملیات منجر شود [۱۲].

موتورهای جریان مستقیم برای تبدیل توان الکتریکی به مکانیکی استفاده می‌شوند؛ که این تبدیل در طی شتاب و سرعت گرفتن ایجاد می‌شود. چون موتورهای ایده آل نیستند؛ این تبدیل انرژی با تلفاتی که ناشی از مقاومت آرمیچر، اصطکاک ویسکوز می‌باشد؛ همراه است [۱۳]. بسیاری از تحقیقات بر روی کاهش دادن تلفات موتورهای جریان مستقیم تمرکز داشته است. در [۱۴، ۱۵] یک تابع هزینه که دارای اجزای تلفات انرژی در یک موتور جریان مستقیم بود توسعه داده شده است. اجزایی که در یک موتور جریان مستقیم تلفات ایجاد می‌کنند شامل تلفات مقاومت، تلفات آرمیچر، اصطکاک و تلفات سیم‌پیچ و... می‌باشد. چون اندازه‌گیری تلفات تمامی این اجزا مشکل است، پیاده‌سازی آن نیز پیچیده می‌شود. برای غلبه بر این مسئله بسیاری از محققان فقط تلفات ناشی از مقاومت آرمیچر را به‌عنوان یک تابع هزینه برای کمینه کردن در نظر می‌گیرند.

بسیاری از مسیرهایی که ربات‌های سیار می‌پیمایند شامل خطوط مستقیم و منحنی شکل می‌باشند [۱۶]. رید و شیپ نشان دادند که بهترین مسیر برای حرکت ربات مسیرهای مستقیم و قوسی شکل

هستند. از آنجایی که این مسیرها داری ناپیوستگی در اتصالات بین خطوط قوسی منحنی می‌باشند، در محیط واقعی ربات مجبور به توقف در هر انحنای ناپیوستگی می‌باشد. از این رو توقف‌های مکرر و چرخش‌های غیرضروری باعث می‌شود که مصرف انرژی باتری به‌طور قابل توجهی افزایش یابد. به‌منظور حذف این گسستگی در نقاط قوسی شکل، انواع مختلفی از منحنی‌ها پیشنهاد شده است [۱۷].

## ۱-۷- ساختار پایان‌نامه

در این پژوهش ابتدا معادلات دینامیک ربات سیار استخراج می‌شود. سپس روش میدان پتانسیل ارائه می‌شود و به بهبود این روش پرداخته شده است. به‌طوری که در مقایسه با روش کلاسیک ربات به‌طور مؤثری از کمینه‌های محلی واقع شده در مسیر می‌گریزد. مسیریابی با استفاده از این الگوریتم نشان می‌دهد که این روش هم خیلی سریع است و هم قادر به برخورد با کمینه‌های محلی که می‌تواند ربات‌های سیار را قبل از رسیدن به هدف به دام بیندازد، است. سپس مسئله مسیریابی برای یک ربات در یک محیط با مانع دوبعدی با تبدیل به یک مسئله بهینه‌سازی مقید و حل مسئله با استفاده از الگوریتم‌های تکاملی مطرح می‌شود. در ادامه به بررسی انرژی مصرفی در ربات‌های سیار با در نظر گرفتن قیود غیرهولونومیک چرخ‌ها، معادلات غیرخطی ربات متحرک پرداخته شده است. به‌طوری که مسئله طراحی مسیر بهینه ربات سیار، به‌عنوان یک مسئله کنترل بهینه، فرمولاسیون شده و با استفاده از روش غیرمستقیم کنترل بهینه شرایط بهینگی مسیر، به‌صورت یک مجموعه معادلات دیفرانسیل غیرخطی، استخراج می‌شود. سپس معادلات بهینگی استخراج شده به کمک روش‌های عددی حل شده و مسیرهای بهینه متنوعی برای ربات شبیه‌سازی می‌شود. در ادامه به‌منظور بررسی صحت نتایج تئوری، تحلیل‌های تجربی متعددی با استفاده از ربات سیار رومبا<sup>۱</sup> انجام پذیرفته و با نتایج شبیه‌سازی مقایسه می‌شود. تحلیل نتایج تجربی، نشان‌دهنده صحت و کارایی روش پیشنهادی در کاربردهای ربات سیار می‌باشد.

---

<sup>۱</sup> - Roomba



فصل دوم

معادلات حرکت ربات سیار

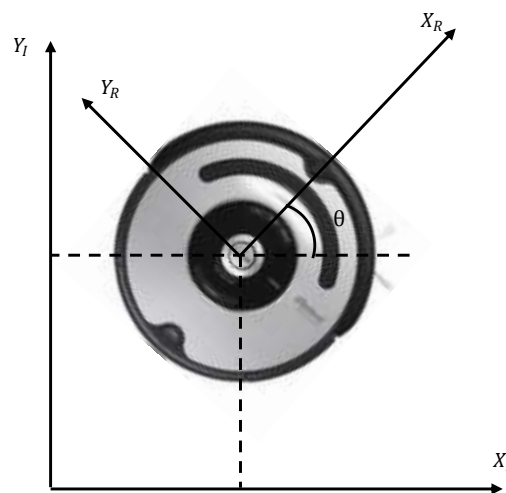
چرخ‌دار

## ۲-۱- مقدمه

طراحی، توسعه و کنترل یک سیستم مکترونیکی نیازمند داشتن درک مناسب از سیستم موردنظر است، که این خود نیازمند یک مدل خاص از سیستم است. یک سیستم ربات سیار دوچرخ شامل مدل‌سازی سینماتیکی و دینامیکی و مدل‌سازی محرک‌ها می‌باشد. مدل‌سازی سینماتیکی مرتبط با روابط هندسی حاکم بر سیستم و مطالعات ریاضی حرکت بدون در نظر گرفتن نیروهای مؤثر بر سیستم می‌باشد. مدل‌سازی دینامیکی، مطالعه حرکت با در نظر گرفتن نیروهای وارد بر سیستم است و مدل‌سازی محرک‌ها به پیدا کردن ارتباط بین سیگنال کنترل و ورودی سیستم مکانیکی مرتبط است [۱۸]. هر بخش از مدل‌سازی سیستم به‌طور جداگانه در این فصل توضیح داده می‌شود.

## ۲-۲- سیستم‌های مختصات

در شکل ۱-۲ سیستم مختصات موقعیت ربات نشان داده شده است. از دو سیستم مختصات برای مدل‌سازی و کنترل ربات استفاده می‌شود. مختصات ساکن  $(X_I, Y_I)$  سیستم مختصات ثابت برای بدنه ربات است و مختصات ربات  $(X_R, Y_R)$  سیستم مختصات متصل به ربات است.



شکل ۱-۲ سیستم مختصات ربات

موقعیت ربات در مختصات داخلی و مختصات ربات را می‌توان به‌صورت رابطه ۱-۲ و ۲-۲ تعریف کرد:

$$q_I = [x_I \ y_I \ \theta_I]^T \quad (1-2)$$

$$q_R = [x_R \ y_R \ \theta_R]^T \quad (2-2)$$

نگاشت بین این دو مختصات از طریق ماتریس دوران ۲-۴ حاصل می‌شود.

$$\dot{q}_R = R(\theta)\dot{q}_I \quad (۳-۲)$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (۴-۲)$$

### ۳-۲-۳ مدل سینماتیکی ربات سیار

هدف از مدل‌سازی سینماتیکی ربات، پیدا کردن سرعت ربات در مختصات ساکن به‌عنوان تابعی از سرعت چرخ و پارامترهای هندسی ربات است؛ به‌عبارت‌دیگر هدف ایجاد سرعت ربات،  $\dot{q} = [\dot{x} \ \dot{y} \ \dot{\theta}]^T$  به‌عنوان تابعی از سرعت چرخ‌های ربات  $w_r, w_l$  و پارامترهای هندسی ربات است. همچنین هدف پیدا کردن رابطه بین پارامترهای کنترلی  $(w_r, w_l)$  و رفتار سیستم در فضای حالت است. سینماتیک ربات به‌طور کلی دارای دو تحلیل است:

۱- سینماتیک مستقیم:

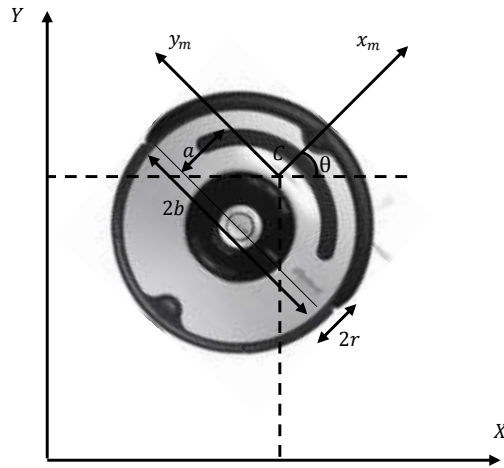
$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(w_r, w_l, b, r, \theta) \quad (۵-۲)$$

۲- سینماتیک معکوس:

$$\begin{bmatrix} w_r \\ w_l \end{bmatrix} = f(\dot{x}, \dot{y}, \dot{\theta}) \quad (۶-۲)$$

### ۳-۲-۱-۱ مدل سینماتیک مستقیم

یک ربات سیار که دارای دو چرخ با شعاع‌های  $r$  که بافاصله  $b$  نسبت به مرکز ربات قرار دارد در شکل ۲-۲ نشان داده شده است.



شکل ۲-۲ ساختار ربات سیار

مسئله سینماتیک مستقیم با تعریف رابطه ۲-۵ شرح داده شده است. سرعت انتقالی ربات به صورت میانگین سرعت‌های زاویه‌ای چرخ راست و چپ به صورت:

$$v = r \frac{w_r + w_l}{2} \quad (7-2)$$

و سرعت زاویه‌ای با رابطه ۲-۸ نشان داده می‌شود.

$$w = \frac{r}{2b} (w_r - w_l) \quad (8-2)$$

با توجه به اینکه  $\dot{q}_I = R(\theta)^{-1} \dot{q}_R$  مدل کلی سرعت ربات به صورت ۲-۹ است.

$$\dot{q}_I = R(\theta)^{-1} \frac{r}{2} \begin{bmatrix} w_r + w_l \\ 0 \\ \frac{w_r - w_l}{b} \end{bmatrix} \quad (9-2)$$

بنابراین سرعت ربات به صورت رابطه ۲-۱۰ است.

$$\dot{q}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{r}{2} \begin{bmatrix} w_r + w_l \\ 0 \\ \frac{w_r - w_l}{b} \end{bmatrix} = \begin{bmatrix} r \frac{w_r + w_l}{2} \cos\theta \\ r \frac{w_r + w_l}{2} \sin\theta \\ \frac{r}{2b} (w_r - w_l) \end{bmatrix} \quad (10-2)$$

### ۲-۳-۲-۲- قیود سینماتیکی

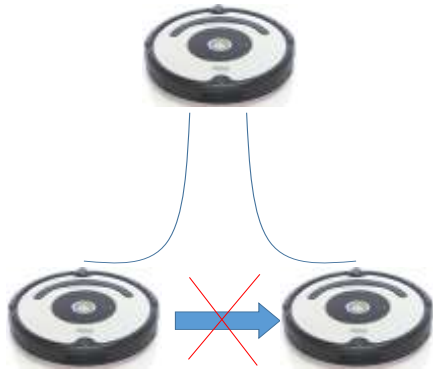
فرضیات زیر در حرکت چرخ‌ها، محدودیت سینماتیکی برای ربات ایجاد خواهد کرد.

- حرکت بر روی سطح افقی است.
- نقطه تماس، بین چرخ‌ها و زمین است.



- چرخ‌ها تغییر شکل ندارند.

با توجه به مفروضات فوق برای حرکت، ربات دارای نوع خاصی از قیود به نام قیود غیرهولونومیک می‌باشد. این قیود برای ربات سیار به این معنی است که ربات فقط بر روی خط مستقیم (به سمت جلو یا عقب) حرکت می‌کند (شکل ۲-۳).



شکل ۲-۳ قیود غیرهولونومیک ربات سیار

قیود غیرهولونومیک برای ربات سیار به صورت زیر است.

- قید عدم لغزش:

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - \dot{\theta}a = 0 \quad (11-2)$$

- قید غلتش خالص:

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta + b\dot{\theta} = rw_r \quad (12-2)$$

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta - b\dot{\theta} = rw_l \quad (13-2)$$

این قیود نشان می‌دهد که چرخ‌های ربات دارای لغزش نیستند. این سه محدودیت را می‌توان به صورت

رابطه ۱۴-۲ نشان داد.

$$A(q)\dot{q} = 0 \quad (14-2)$$

که در آن:

$$A(q) = \begin{bmatrix} -\sin\theta & \cos\theta & a & 0 & 0 \\ \cos\theta & \sin\theta & b & -r & 0 \\ \cos\theta & \sin\theta & -b & 0 & -r \end{bmatrix} \quad (15-2)$$

$$\dot{q} = [\dot{x}_c \quad \dot{y}_c \quad \dot{\theta} \quad w_r \quad w_l]^T \quad (16-2)$$

## ۲-۴- مدل سازی دینامیکی ربات سیار

فرمولاسیون معادلات حرکت یک ربات در تجزیه و تحلیل، طراحی و کنترل آن ضروری است. یک ربات سیار که دارای ساختار  $n$  بعدی در فضای  $L$  با مختصات کلی  $(q_1, q_2, \dots, q_n)$  و دارای  $m$  محدودیت است را می توان به صورت رابطه ۲-۱۷ نوشت:

$$M(q)\ddot{q} + V(q, \dot{q}) = B(q)\tau - A^T(q)\lambda \quad (۲-۱۷)$$

که در آن:

$M(q)$  ماتریس اینرسی که متقارن و مثبت است.  $V(q, \dot{q})$  ماتریس دارای ترم های شتاب کوریولیس می باشد.  $B(q)$  ماتریس انتقال ورودی است.  $\tau$  بردار ورودی است.  $A^T(q)$  ماتریس مرتبط با قیود می باشد.  $\lambda$  بردار نیرو مرتبط با قیود است.

## ۲-۴-۱- روش لاگرانژ

معادلات لاگرانژ، معادلات دیفرانسیل هستند که انرژی و کار انجام شده توسط سیستم را بررسی می کند. اگر  $m$  معادله با قید غیرهولونومیک وجود داشته باشد، می توان رابطه ۲-۱۸ را نوشت.

$$\sum_{i=1}^n a_{ij} dq_i + a_{jt} dt = 0, \quad j = 1, 2, \dots, m \quad (۲-۱۸)$$

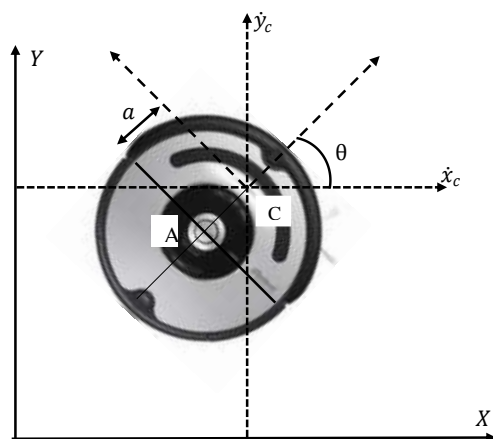
که در آن  $a_{ij}$  توابعی از مختصات تعمیم یافته است. پس معادله لاگرانژ به صورت:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum_{j=1}^m \lambda_j a_{ji} + Q_i \quad i = 1, 2, \dots, n \quad (۲-۱۹)$$

که در آن  $(q_1, q_2, \dots, q_n)$  مختصات های تعمیم یافته هستند.

$L = E_K - V$  لاگرانژین که اختلاف بین انرژی جنبشی و انرژی پتانسیل است.  $\lambda_j$  ضریب لاگرانژ که مربوط به قیود نیرو می باشد.  $Q_i$  شامل نیروهای غیرکنسرواتیو می باشد.

اولین قدم برای نوشتن معادلات دینامیک، پیدا کردن انرژی های جنبشی و پتانسیل سیستم است.



شکل ۲-۴ مؤلفه سرعت ربات

برای پیدا کردن سرعت در نقاط  $A, C$  مطابق شکل ۲-۴ داریم:

$$x_c = x_A + a \cos \theta \quad (20-2)$$

$$y_c = y_A + a \sin \theta \quad (21-2)$$

$$\dot{x}_c = \dot{x}_A - a \dot{\theta} \sin \theta \quad (22-2)$$

$$\dot{y}_c = \dot{y}_A + a \dot{\theta} \cos \theta \quad (23-2)$$

سرعت ربات در مرکز چرخش آن در نقطه  $A$  به صورت رابطه ۲۴-۲ است.

$$v_A = \dot{x}_c \vec{i} + \dot{y}_c \vec{j} + a \dot{\theta} \sin \theta \vec{i} - a \dot{\theta} \cos \theta \vec{j} \quad (24-2)$$

$$v_A = (\dot{x}_c + a \dot{\theta} \sin \theta) \vec{i} + (\dot{y}_c - a \dot{\theta} \cos \theta) \vec{j} \quad (25-2)$$

انرژی جنبشی به صورت ۲۶-۲ به دست می آید.

$$E_k = \frac{1}{2} m v_A^2 + \frac{1}{2} I \dot{\theta}^2 \quad (26-2)$$

$$E_k = \frac{1}{2} m \left[ (\dot{x}_c + a \dot{\theta} \sin \theta)^2 + (\dot{y}_c - a \dot{\theta} \cos \theta)^2 \right] = \frac{1}{2} m \left[ \dot{x}_c^2 + 2 \dot{x}_c a \dot{\theta} \sin \theta + \right. \\ \left. a^2 \dot{\theta}^2 \sin^2 \theta + \dot{y}_c^2 - 2 \dot{y}_c a \dot{\theta} \cos \theta + a^2 \dot{\theta}^2 \cos^2 \theta \right] + \frac{1}{2} I \dot{\theta}^2 \quad (27-2)$$

$$E_k = \frac{1}{2} m \dot{x}_c^2 + \frac{1}{2} m \dot{y}_c^2 + m \dot{x}_c a \dot{\theta} \sin \theta - m \dot{y}_c a \dot{\theta} \cos \theta + \frac{1}{2} m a^2 \dot{\theta}^2 + \frac{1}{2} I \dot{\theta}^2 \quad (28-2)$$

از قضیه محورهای موازی می توان نوشت:

$$I_c + m a^2 = I \quad (29-2)$$

بنابراین انرژی جنبشی سیستم به صورت رابطه ۲-۳۰ است.

$$E_k = \frac{1}{2}m\dot{x}_c^2 + \frac{1}{2}m\dot{y}_c^2 + \frac{1}{2}I_c\dot{\theta}^2 + ma^2\dot{\theta}^2 + m\dot{x}_ca\dot{\theta}\sin\theta - m\dot{y}_ca\dot{\theta}\cos\theta \quad (۳۰-۲)$$

چون ربات روی سطح صاف است انرژی پتانسیل صفر در نظر گرفته شده است. بنابراین معادله لاگرانژ با رابطه ۲-۳۱ بیان می‌شود.

$$L = \frac{1}{2}m\dot{x}_c^2 + \frac{1}{2}m\dot{y}_c^2 + \frac{1}{2}I_c\dot{\theta}^2 + ma^2\dot{\theta}^2 + m\dot{x}_ca\dot{\theta}\sin\theta - m\dot{y}_ca\dot{\theta}\cos\theta \quad (۳۱-۲)$$

مختصات تعمیم یافته برای به کار بردن در معادله لاگرانژ عبارتند از:

$$q = [x_c, y_c, \theta] \quad (۳۲-۲)$$

$$\dot{q} = [\dot{x}_c, \dot{y}_c, \dot{\theta}] \quad (۳۳-۲)$$

در مرحله بعد معادلات دینامیکی با استفاده از مختصات تعمیم یافته به صورت:

$$\frac{\partial L}{\partial \dot{x}_c} = m\dot{x}_c + ma\dot{\theta}\sin\theta \quad (۳۴-۲)$$

$$\frac{\partial L}{\partial \dot{y}_c} = m\dot{y}_c - ma\dot{\theta}\cos\theta \quad (۳۵-۲)$$

$$\frac{\partial L}{\partial \dot{\theta}} = I_c\dot{\theta} + 2ma^2\dot{\theta} + m\dot{x}_c a \sin\theta - m\dot{y}_c a \cos\theta \quad (۳۶-۲)$$

با مشتق گرفتن نسبت به زمان از معادله ۲-۳۴ تا ۲-۳۶ داریم:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_c} \right) = m\ddot{x}_c + ma\ddot{\theta}\sin\theta + ma\dot{\theta}^2\cos\theta \quad (۳۷-۲)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{y}_c} \right) = m\ddot{y}_c - ma\ddot{\theta}\cos\theta + ma\dot{\theta}^2\sin\theta \quad (۳۸-۲)$$

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) &= I_c\ddot{\theta} + 2ma^2\ddot{\theta} + ma\ddot{x}_c\sin\theta + ma\dot{x}_c\dot{\theta}\cos\theta - ma\dot{y}_c\cos\theta + \\ & \quad m\dot{y}_ca\dot{\theta}\cos\theta \end{aligned} \quad (۳۹-۲)$$

$$\frac{\partial L}{\partial x_c} = 0 \quad (۴۰-۲)$$

$$\frac{\partial L}{\partial y_c} = 0 \quad (۴۱-۲)$$

$$\frac{\partial L}{\partial \theta} = m\dot{x}_ca\dot{\theta}\cos\theta + m\dot{y}_ca\dot{\theta}\sin\theta \quad (۴۲-۲)$$

با قرار دادن معادلات در فرمول لاگرانژ داریم:

$$m\ddot{x}_c + ma\dot{\theta}\sin\theta + ma\theta^2\cos\theta = F_x + C_1 \quad (۴۳-۲)$$

$$m\ddot{y}_c - ma\dot{\theta}\cos\theta + ma\theta^2\sin\theta = F_y + C_2 \quad (۴۴-۲)$$

$$(I_c + 2ma^2)\ddot{\theta} + ma\ddot{x}_c\sin\theta - ma\dot{y}_c\cos\theta = \tau + C_3 \quad (۴۵-۲)$$

که در آن نیروی محرک در جهت  $x$  و  $F_y$  نیروی محرک در جهت  $y$  و  $\tau$  گشتاور ربات و  $C_X, C_Y, C_\theta$  قیود نیرو در مختصات  $x, y, \theta$  می باشد.

$$\begin{bmatrix} m & 0 & masin\theta \\ 0 & m & -macos\theta \\ masin\theta & -macos\theta & I_c + 2ma^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} ma\theta^2\cos\theta \\ ma\theta^2\sin\theta \\ 0 \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} + \begin{bmatrix} C_X \\ C_Y \\ C_\theta \end{bmatrix} \quad (۴۶-۲)$$

نیروهای مرتبط در جهت های  $x, y$  و  $\theta$  برای گشتاورهای محرک هر چرخ به صورت:

$$F_R = \frac{\tau_R}{r} \quad (۴۷-۲)$$

$$F_L = \frac{\tau_L}{r} \quad (۴۸-۲)$$

$$F_X = \left(\frac{\tau_R}{r} + \frac{\tau_L}{r}\right) \cos\theta = \frac{\cos\theta}{r} (\tau_R + \tau_L) \quad (۴۹-۲)$$

$$F_Y = \left(\frac{\tau_R}{r} + \frac{\tau_L}{r}\right) \sin\theta = \frac{\sin\theta}{r} (\tau_R + \tau_L) \quad (۵۰-۲)$$

$$\tau = (F_R - F_L)b = \frac{b}{r} (\tau_R - \tau_L) \quad (۵۱-۲)$$

با توجه به معادلات ۴۷-۲ تا ۵۱-۲ ماتریس نیرو در سیستم دینامیک با رابطه ۵۲-۲ بیان می شود.

$$\begin{bmatrix} F_X \\ F_Y \\ \tau \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{r} (\tau_R + \tau_L) \\ \frac{\sin\theta}{r} (\tau_R + \tau_L) \\ \frac{b}{r} (\tau_R - \tau_L) \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (۵۲-۲)$$

با توجه به قیود غیرهولونومیک که در بخش ۲-۳-۲ برای ربات ذکر شد، قیود مرتبط با نیرو عبارتند از:

$$C_X = m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}\sin\theta \quad (۵۳-۲)$$

$$C_Y = -m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}\cos\theta \quad (۵۴-۲)$$

$$C_\theta = ma(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta} \quad (۵۵-۲)$$

ماتریس مرتبط با قیود نیروها عبارتند از:

$$\begin{bmatrix} C_X \\ C_Y \\ C_\theta \end{bmatrix} = \begin{bmatrix} m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \sin\theta \\ -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \cos\theta \\ ma(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \end{bmatrix} = A^T(q)\lambda \quad (56-2)$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -a \end{bmatrix} \quad (57-2)$$

$$\lambda = -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \quad (58-2)$$

با توجه به معادلات ذکر شده، فرم کلی معادله دینامیک ربات سیار با رابطه ۵۹-۲ بیان می شود.

$$\begin{bmatrix} m & 0 & m \sin\theta \\ 0 & m & -m \cos\theta \\ m \sin\theta & -m \cos\theta & I_c + 2ma^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} ma\dot{\theta}^2 \cos\theta \\ ma\dot{\theta}^2 \sin\theta \\ 0 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} + \begin{bmatrix} m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \sin\theta \\ -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \cos\theta \\ ma(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \end{bmatrix} \quad (59-2)$$

با مقایسه معادله با شکل کلی معادله دینامیک، پارامترهای سیستم ربات سیار دو چرخ به صورت روابط ۶۰-۲ تا ۶۵-۲ است.

$$M(q) = \begin{bmatrix} m & 0 & m \sin\theta \\ 0 & m & -m \cos\theta \\ m \sin\theta & -m \cos\theta & I_c + 2ma^2 \end{bmatrix} \quad (60-2)$$

$$V(q, \dot{q}) = \begin{bmatrix} ma\dot{\theta}^2 \cos\theta \\ ma\dot{\theta}^2 \sin\theta \\ 0 \end{bmatrix} \quad (61-2)$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \quad (62-2)$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -a \end{bmatrix} \quad (63-2)$$

$$\lambda = -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \quad (64-2)$$

$$B(q) = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ b & -b \end{bmatrix} \quad (65-2)$$

رابطه ۵۹-۲ را می توان برای اهداف کنترلی و شبیه سازی به صورت بهتری تبدیل کرد. در این تبدیل

بایستی راهی برای از بین بردن قیود از معادله پیدا کرد. دو ماتریس ۶۶-۲ و ۶۷-۲ را برای این تبدیل

در نظر گرفته شده است.

$$v_a(t) = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \dot{X}_R \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (۶۶-۲)$$

$$S(q) = \begin{bmatrix} \cos\theta & -a\sin\theta \\ \sin\theta & a\cos\theta \\ 0 & 1 \end{bmatrix} \quad (۶۷-۲)$$

ماتریس  $S(q)$  اصلاح شده ماتریس سینماتیک مستقیم است.

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = S(q)v_a(t) = \begin{bmatrix} \cos\theta & -a\sin\theta \\ \sin\theta & a\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (۶۸-۲)$$

اثبات می‌شود که ماتریس  $S(q)$  با ماتریس  $A(q)$  به صورت رابطه ۶۹-۲ ارتباط دارند.

$$S^T(q)A^T(q) = 0 \quad (۶۹-۲)$$

رابطه ۶۹-۲ برای حذف قیود از معادله دینامیک اصلی به کار می‌رود. با مشتق از رابطه ۶۹-۲ داریم:

$$\ddot{q} = \dot{S}(q)v_a(t) + S(q)\dot{v}_a(t) \quad (۷۰-۲)$$

با جایگذاری معادله ۶۹-۲ در رابطه ۱۷-۲ داریم:

$$M(q)\dot{S}(q)v_a(t) + M(q)S(q)\dot{v}_a(t) + V_m(q, \dot{q})S(q)v_a(t) = B(q)\tau - A^T(q)\lambda \quad (۷۱-۲)$$

$$V_m(q, \dot{q}) = \begin{bmatrix} 0 & 0 & ma\dot{\theta}\cos\theta \\ 0 & 0 & ma\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \quad (۷۲-۲)$$

در گام بعدی برای از بین بردن ماتریس قیود  $A^T(q)\lambda$ ، ماتریس  $S^T(q)$  در رابطه ۷۱-۲ ضرب می‌شود.

$$[S^T(q)M(q)S(q)]\dot{v}_a(t) + [S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S(q)]v_a(t) = S^T(q)B(q) - S^T(q)A^T(q)\lambda \quad (۷۳-۲)$$

می‌توان از رابطه ۷۳-۲ عبارت  $S^T(q)A^T(q)$  را با توجه به رابطه ۶۹-۲ برابر صفر قرار داد و قیود را از معادله دینامیک حذف کرد.

$$[S^T(q)M(q)S(q)]\dot{v}_a(t) + [S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S(q)]v_a(t) = S^T(q)B(q) \quad (۷۴-۲)$$

بنابراین می‌توان معادلات را به صورت ۲-۷۵ بازنویسی کرد.

$$\bar{M}(q)\dot{v}_a(t) + \bar{V}_m(q, \dot{q})v_a(t) = \bar{B}(q)\tau \quad (۷۵-۲)$$

$$\bar{M}(q) = S^T(q)M(q)S(q) \quad (۷۶-۲)$$

$$\bar{V}_m(q, \dot{q}) = S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S(q) \quad (۷۷-۲)$$

$$\bar{B}(q) = S^T(q)B(q) \quad (۷۸-۲)$$

$$S^T(q)A^T(q)\lambda = 0 \quad (۷۹-۲)$$

## ۲-۵- مدل موتور ربات سیار دوچرخ

در این بخش، به استخراج معادلات دینامیکی موتور ربات سیار چرخ‌دار با در نظر گرفتن قیود غیرهولونومیک پرداخته می‌شود. با فرض اینکه ربات سیار دارای دو موتور جریان مستقیم مشابه است،  $R_a$  مقاومت آرمیچر،  $K_b$  ثابت برگشتی موتور و  $n_g$  ثابت چرخ‌دنده و  $V_s$  ولتاژ باتری است. برای ساده‌سازی دینامیک موتور، از اندوکتانس آرمیچر به دلیل اینکه پاسخ‌های الکتریکی خیلی سریع‌تر از پاسخ دینامیکی است صرف‌نظر شده است. معادله الکتریکی مدار آرمیچر به صورت ۲-۸۰ تعریف می‌شود.

$$R_a I_a = V_s u - K_b n_g W \quad (۸۰-۲)$$

که در آن  $I_a = [i_r \ i_l]^T$  بردار جریان آرمیچر،  $W = [w_r \ w_l]^T$  بردار سرعت زاویه‌ای چرخ‌ها و  $u = [u_r \ u_l]^T$  بردار کنترل ورودی می‌باشد. علاوه بر این رابطه دینامیکی بین سرعت زاویه‌ای و جریان موتور، با توجه به اینرسی و اصطکاک ویسکوز با رابطه ۲-۸۱ بیان می‌شود.

$$J \frac{dW}{dt} + F_v W = K_t n_g I_a \quad (۸۱-۲)$$

در اینجا  $F_v$  ضریب اصطکاک ویسکوز و  $J$  ماتریس اینرسی موتورها می‌باشد.



فصل سوم

مسیریابی برای ربات سیار به

روش الگوریتم میدان

پتانسیل

### ۳-۱- مقدمه

در این فصل روش میدان پتانسیل مصنوعی APF<sup>۱</sup>، به عنوان یک روش سراسری، برای طراحی مسیر ربات استفاده می‌شود. در این روش، فرض می‌شود که موانع حامل بارهای الکتریکی هستند و پتانسیل‌های دفع‌کننده حول موقعیت‌های موانع (نواحی ممنوعه) برای وادار ساختن ربات به دور شدن و یک پتانسیل جذب‌کننده‌ی مناسب در نقطه‌ی هدف برای جذب ربات استفاده می‌شود [۱۹]. این‌گونه از تصادمات بین موانع و ربات توسط نیروی دفع‌کننده‌ی بین آن‌ها که برابر منفی پتانسیل گرادیان کل است، جلوگیری می‌شود. این نیرو ربات را در سراسیمی به سمت موقعیت هدف خود تا زمانی که به یک کمینه برسد و متوقف شود، سوق می‌دهد [۲۰]. اولین بار کاربرد میدان پتانسیل مصنوعی برای پرهیز برخورد با مانع توسط خطیب جهت کنترل بازوی مکانیکی ارائه شد [۲۱]. تورپ از یک تابع هزینه شبیه پتانسیل در طراحی یک مسیر بهینه برای یک ربات مدور در دو بعد استفاده کرده است [۲۲]. سو و شین یک الگوریتم مبتنی بر پتانسیل برای یافتن مسیر بهینه‌ی یک ربات نقطه‌ای در دو بعد گزارش کردند و یک طرح خلاصه برای حالت سه بعدی نیز ارائه کردند [۲۳]. اگرچه میدان پتانسیل مصنوعی یک روش ساده است، محدودیت‌هایی وجود دارد که ممکن است جلوی رسیدن به هدف را بگیرند. محدودیت عمده‌ی روش میدان پتانسیل مصنوعی موقعیت‌های تله است که ممکن است به علت قرار گرفتن در کمینه‌های محلی رخ دهند.

### ۳-۲- میدان پتانسیل مصنوعی سنتی

در نظریه اساسی روش میدان پتانسیل مصنوعی فرض شده که ربات به عنوان یک نقطه در یک میدان نیروی مصنوعی حرکت می‌کند. میدان پتانسیل مصنوعی شامل میدان پتانسیل جذب‌کننده به سمت هدف و میدان پتانسیل دفع‌کننده از موانع می‌باشد. پتانسیل جذب به وسیله هدف تولید شده و در جهت حرکت به سمت هدف افزایش می‌یابد. پتانسیل دفع از موانع مختلف ایجاد می‌شود. جهت پتانسیل دفع

---

<sup>1</sup> - Artificial Potential Field

ترکیب شده دافع موانع می‌باشد. از این رو تابع پتانسیل رابطه ۱-۳ به‌عنوان برآیندی از پتانسیل جذب و پتانسیل دفع تعریف شده است. مختصات ربات به‌صورت  $q = (x, y)^T$  است بدین ترتیب میدان پتانسیل مصنوعی به‌صورت رابطه ۱-۳ تعریف می‌شود.

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (۱-۳)$$

در رابطه ۱-۳،  $U(q)$  میدان پتانسیل مصنوعی،  $U_{att}(q)$  نشان‌دهنده پتانسیل جذب،  $U_{rep}(q)$  پتانسیل دفع را نشان می‌دهد. نیروهای مصنوعی ربات به‌صورت رابطه ۲-۳ است.

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = F_{att}(q) + F_{rep}(q) \quad (۲-۳)$$

پتانسیل جاذبه بین ربات و هدف ایجاد شده، ربات را به سمت هدف می‌کشاند. پتانسیل جاذبه ایجاد شده توسط هدف در رابطه ۳-۳ نشان داده شده است.

$$U_{att}(q) = \frac{1}{2}k(q - q_g)^2 = \frac{1}{2}k\rho_g^2(q) \quad (۳-۳)$$

در رابطه ۳-۳،  $k$  ضریب مثبت برای  $APF$  است و  $q_g = (x_g, y_g)^T$  بردار موقعیت تا هدف است.

$$\rho_g(q) = \|q - q_g\| \quad (۴-۳)$$

رابطه ۴-۳ فاصله اقلیدسی ربات از موقعیت ربات تا موقعیت هدف است.

نیروی جاذبه ربات در جهت منفی گرادیان پتانسیل جذب محاسبه شده که به‌صورت رابطه ۵-۳ است:

$$F_{att}(q) = -\nabla U_{att}(q) = -\frac{1}{2}k\nabla\rho_g^2(q) = -k(q - q_g) \quad (۵-۳)$$

در اینجا  $F_{att}(q)$  یک بردار به سمت  $q_g$  با مقداری است که به‌صورت خطی به فاصله بین  $q$  تا  $q_g$  مربوط است. اجزای سازنده  $F_{att}(q)$  گرادیان در جهت منفی پتانسیل جذب، در امتداد جهت‌های  $x$  و  $y$  می‌باشد. بنابراین هنگامی که پتانسیل جذب اعمال می‌شود اجزای سازنده آن را می‌توان به‌صورت روابط ۶-۳ و ۷-۳ معرفی کرد.

$$F_{att-x}(q) = -k(x - x_g) \quad (۶-۳)$$

$$F_{att-y}(q) = -k(y - y_g) \quad (۷-۳)$$

در اینجا  $F_{att-x}$  نیروی جاذبه در جهت  $x$  و  $F_{att-y}$  نیروی جاذبه در جهت  $y$  می‌باشد.

ربات باید از موانع دفع، اما زمانی که ربات دور از موانع است، ما نمی‌خواهیم موانع حرکت ربات را تحت تأثیر قرار دهد. خطیب رابطه ۳-۸ را به‌عنوان میدان پتانسیل دافعه تعریف کرد.

$$U_{rep}(q) = \begin{cases} 0 & \rho(q) \geq \rho_0 \\ \frac{1}{2}\eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) & \rho(q) \leq \rho_0 \end{cases} \quad (۸-۳)$$

در اینجا  $\eta$  ضریب مثبت است.  $q_c = (x_c, y_c)$  مربوط به پیکربندی مانع است.  $\rho_0$  بزرگ‌ترین فاصله برخورد با یک مانع است.  $\rho(q) = \|q - q_c\|$  کوتاه‌ترین فاصله بین ربات و موانع است. ربات زمانی بدون برخورد از موانع عبور می‌کند که فاصله بین ربات و موانع بیشتر از  $\rho_0$  است. به‌طور مشابه، نیروی دافعه به‌صورت روابط ۳-۹ و ۳-۱۰ تعریف می‌شود.

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} 0 & \rho(q) \geq \rho_0 \\ \eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left( \frac{1}{\rho^2(q)} \right) \nabla \rho(q) & \rho(q) \leq \rho_0 \end{cases} \quad (۹-۳)$$

$$F_{rep}(q) = \begin{cases} 0 & \rho(q) \geq \rho_0 \\ \eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left( \frac{1}{\rho^2(q)} \right) \frac{q - q_c}{\|q - q_c\|} & \rho(q) \leq \rho_0 \end{cases} \quad (۱۰-۳)$$

که  $F_{rep-x}$  و  $F_{rep-y}$  موقعیت دکارتی نیروی دافعه  $F_{rep}$  است. زمانی که پتانسیل دافعه بر روی ربات

اعمال می‌شود اجزا به‌صورت روابط ۳-۱۱ و ۳-۱۲ بیان می‌شود.

$$F_{rep-x}(q) = \begin{cases} 0 & \rho(q) \geq \rho_0 \\ \eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left( \frac{1}{\rho^2(q)} \right) \frac{x - x_c}{\|q - q_c\|} & \rho(q) \leq \rho_0 \end{cases} \quad (۱۱-۳)$$

$$F_{rep-y}(q) = \begin{cases} 0 & \rho(q) \geq \rho_0 \\ \eta \left( \frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \left( \frac{1}{\rho^2(q)} \right) \frac{y - y_c}{\|q - q_c\|} & \rho(q) \leq \rho_0 \end{cases} \quad (۱۲-۳)$$

اگر محیط دارای موانع زیادی باشد میدان پتانسیل دافعه کل، مجموع میدان پتانسیل دافعه همه موانع است. در میدان پتانسیل مصنوعی کل داریم:

$$U(q) = U_{att}(q) + \sum_{i=1}^{n_o} U_{rep}(q) \quad (۱۳-۳)$$

در رابطه (۱۳-۳)،  $i = 1, 2, \dots, n_o$  که  $n_o$  تعداد موانع است. میدان نیروی پتانسیل به‌صورت رابطه ۳-۱۴ بیان می‌شود.

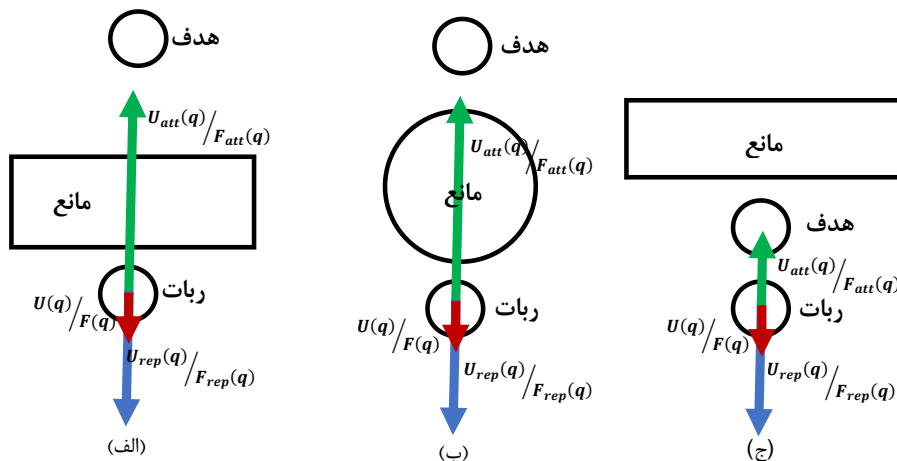
$$F(q) = F_{att}(q) + \sum_{i=1}^{n_o} F_{rep}(q) \quad (۱۴-۳)$$

اگرچه روش میدان پتانسیل مصنوعی سنتی می‌تواند یک مسیر صاف و مؤثر را برنامه‌ریزی کند اما دارای مشکلات عمده‌ای است. روش میدان پتانسیل مصنوعی سنتی که در برنامه‌ریزی مسیر استفاده شده ممکن است در کمینه محلی قرار بگیرد و مسئله نوسانات برای آن رخ دهد. مسئله کمینه محلی و نوسانات را به صورت روابط ۱۵-۳ و ۱۶-۳ تعریف می‌شود.

$$|U(q)| = |U_{att}(q) + \sum_{i=1}^{n_o} U_{rep}(q)| \leq \varepsilon \quad (15-3)$$

$$|F(q)| = |F_{att}(q) + \sum_{i=1}^{n_o} F_{rep}(q)| \leq \varepsilon \quad (16-3)$$

ربات در صورتی در کمینه محلی قرار می‌گیرد که میدان پتانسیل مصنوعی معادلات ۱۵-۳ و ۱۶-۳ را ارضاء کند. زمانی که نیرو یا پتانسیل جاذبه و نیرو یا پتانسیل دافعه معادل بر روی یک خط مستقیم واقع شوند میدان نیرو یا پتانسیل ربات صفر است که این باعث می‌شود ربات در کمینه محلی به دام بیفتد. علاوه بر این، هنگامی که موقعیت هدف بسیار نزدیک به موانع باشد، ربات نمی‌تواند به هدف برسد (شکل ۱-۳).



شکل ۱-۳ محدودیت‌های میدان پتانسیل مصنوعی. (الف) و (ب) مسئله کمینه محلی (ج) مسئله هدف غیرقابل دسترس

شکل ۱-۳ الف نمایانگر زمانی است که ربات و هدف بر روی یک خط مستقیم واقع شوند و یک مانع هم بین آن‌ها وجود داشته باشد. در این حالت ربات در کمینه محلی قرار می‌گیرد. شکل ۱-۳ ب نمایانگر زمانی است که نیرو یا پتانسیل جاذبه و نیرو یا پتانسیل دافعه با هم برابرند که در این حالت میدان نیرو یا پتانسیل مصنوعی ربات صفر می‌شود. در این حالت نیز ربات در کمینه محلی می‌گیرد. شکل ۱-۳ ج

نمایانگر زمانی است که موقعیت هدف خیلی نزدیک به مانع است و نیرو یا پتانسیل دافعه خیلی بزرگ تر از نیرو یا پتانسیل جاذبه می شود. تحت این شرایط ربات هرگز نمی تواند خود را به هدف برساند که در این صورت مسئله هدف غیرقابل دسترس ایجاد می شود.

### ۳-۳- میدان پتانسیل مصنوعی بهبود داده شده

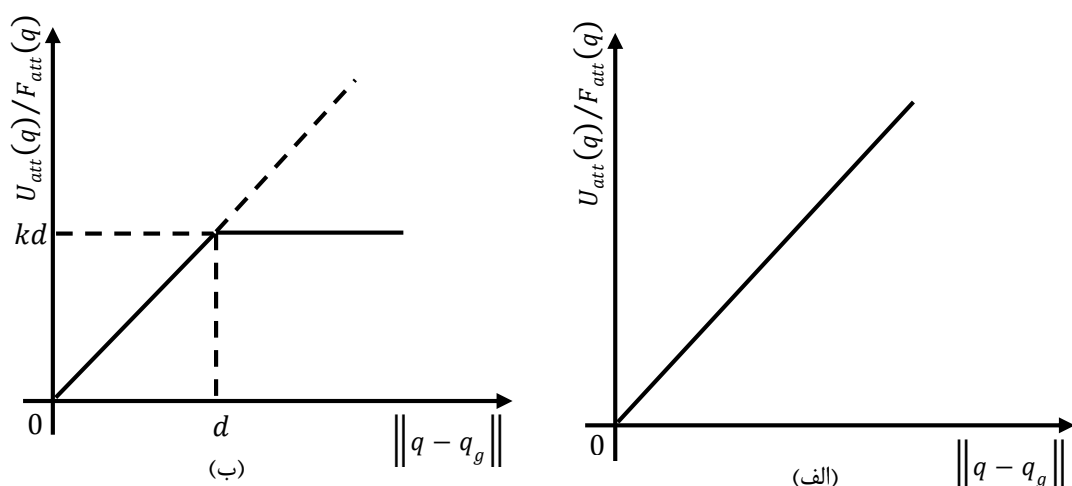
#### ۳-۳-۱- تعریف تابع پتانسیل جذب

در معادلات ۳-۳ و ۵-۳ نیرو یا پتانسیل جاذبه به صورت مستقیم وابسته به مسافت  $\rho_g(q)$  است. مقدار نیرو یا پتانسیل جاذبه بر طبق مسافت بین ربات و هدف مشخص شده است؛ که در تابع پتانسیل جاذبه سنتی پیشنهاد شده است. زمانی که  $\rho_g(q)$  خیلی بزرگ است مقدار نیرو یا پتانسیل جاذبه خیلی بزرگ خواهد شد، به عبارت دیگر مواقعی که ربات از هدف دور است، نیرو یا پتانسیل جاذبه به راحتی ربات را در نزدیکی موانع حرکت می دهد [۲۴]. بدین ترتیب در محیط واقعی که در شکل ۳-۳ دیده می شود ربات با خطر برخورد با موانع روبرو می شود، در نتیجه پتانسیل و نیروی جاذبه به صورت روابط ۳-۱۷ و ۳-۱۸ اصلاح می شوند.

$$U_{att}(q) = \begin{cases} \frac{1}{2}k\rho_g^2(q) & , \rho_g(q) \leq d \\ kd\rho_g(q) & , \rho_g(q) \geq d \end{cases} \quad (۱۷-۳)$$

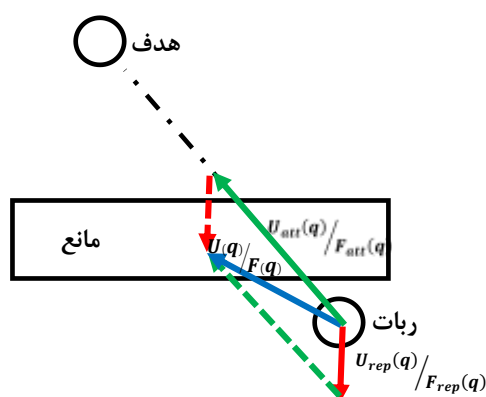
$$F_{att}(q) = \begin{cases} -k(q - q_g) & , \|q - q_g\| \leq d \\ -kd \frac{(q - q_g)}{\|q - q_g\|} & , \|q - q_g\| \geq d \end{cases} \quad (۱۸-۳)$$

زمانی که فاصله  $\rho_g(q)$  کمتر از  $d$  است نیرو و میدان پتانسیل جاذبه همان تعریف متداول را دارد. در غیر این صورت پتانسیل و نیرو جاذبه به صورت یک ثابت ارائه شده است (شکل ۳-۲). در رابطه ۳-۱۷ تابع پتانسیل مجدداً تعریف شده است. رابطه ۳-۱۸ ضمانت می کند که ربات از برخورد با موانع اجتناب می کند. زمانی که ربات در نزدیکی موانع حرکت می کند، نیرو یا پتانسیل دفع کننده از موانع به صورت مؤثر بزرگ تر از  $kd$  است و ربات را از موانع دور می کند (شکل ۳-۲).



شکل ۳-۲ تابع پتانسیل جاذبه. (الف) تابع پتانسیل جذب کننده سنتی (ب) تابع پتانسیل جذب کننده بهبود داده شده

شکل ۳-۲ الف تعریف تابع پتانسیل سنتی، ارتباط بین نیرو یا پتانسیل جاذبه و فاصله ربات تا هدف است. با افزایش فاصله ربات تا هدف مقدار پتانسیل یا نیرو جاذبه افزایش می یابد. شکل ۳-۲ ب تعریف تابع پتانسیل بهبود داده شده است. در اینجا خطر برخورد و خطا برنامه ریزی ربات ارزیابی می شود و تابع نیرو یا پتانسیل جاذبه بهبود داده می شود و آستانه  $d$  تعریف شده است. اگر فاصله کمتر از  $d$  باشد، مقدار نیرو یا پتانسیل جاذبه به صورت خطی متناسب با فاصله ربات تا هدف افزایش می یابد، همانند تعریف میدان پتانسیل سنتی، در غیر این صورت نیرو یا پتانسیل جاذبه مقداری ثابت دارد.



شکل ۳-۳ میدان پتانسیل جاذبه

زمانی که فاصله ربات تا هدف خیلی زیاد است، نیروی جاذبه خیلی بزرگ تر از نیروی دافعه است در نتیجه ربات خیلی نزدیک به مانع حرکت می کند. ربات در نزدیکی مانع بر روی برآیند نیروها حرکت می کند.

و در محیط‌های واقعی، ربات با خطر برخورد با مانع روبرو می‌شود.

### ۳-۳-۲- تعریف تابع پتانسیل دافعه

در بسیاری از مقالات که توصیف شده‌اند، خطیب [۲۱] و ژانگ [۲۵] زمانی که هدف به شدت به مانع نزدیک می‌شود نیرو یا پتانسیل دافعه خیلی بزرگ‌تر از نیرو یا پتانسیل جاذبه است.

$$|U_{att}(q)| \ll \left| \sum_{i=1}^n U_{rep}(q) \right| \quad (۱۹-۳)$$

$$|F_{att}(q)| \ll \left| \sum_{i=1}^n F_{rep}(q) \right| \quad (۲۰-۳)$$

در این حالت ربات نمی‌تواند هیچ وقت به موقعیت هدف برسد. این شرایط به‌عنوان مسئله هدف غیرقابل دسترس نام‌گذاری شده است. که این امر برای مسئله برنامه‌ریزی ربات نامطلوب است. در اینجا توابع پتانسیل ۳-۲۱ و ۳-۲۲ برای حل مسئله هدف غیرقابل دسترس تعریف شده است. حال مجدداً توابع نیرو یا پتانسیل جاذبه، نیرو یا پتانسیل دافعه، اگر دو شرط  $\rho(q) \leq d_{ob}$  و  $\rho_g \leq d_{gr}$  به صورت همزمان برقرار باشد؛ تعریف می‌شوند. در اینجا  $d_{ob}$  و  $d_{gr}$  به ترتیب نشان‌دهنده ضرایب مثبت می‌باشند. اگر فاصله بین هدف و موانع کمتر از  $d_{ob}$  و به‌صورت همزمان فاصله بین هدف و ربات کمتر از  $d_{gr}$  باشد، ربات بجای اینکه برآیند نیرو یا پتانسیل جاذبه و نیرو یا پتانسیل دافعه را برای رسیدن به هدف محاسبه کند، فقط در نیرو یا پتانسیل جاذبه حرکت می‌کند. زمانی که هر دو شرط بالا برقرار باشد دیگر نیرو پتانسیل دافعه وجود ندارد و ربات فقط به سمت هدف جذب می‌شود [۲۶].

$$U(q) = \begin{cases} U_{att}(q) & , \rho(q) \leq d_{ob} \text{ and } \rho_g \leq d_{gr} \\ U_{att}(q) + U_{rep}(q) & , \text{ otherwise} \end{cases} \quad (۲۱-۳)$$

$$F(q) = \begin{cases} F_{att}(q) & , \rho(q) \leq d_{ob} \text{ and } \rho_g \leq d_{gr} \\ F_{att}(q) + F_{rep}(q) & , \text{ otherwise} \end{cases} \quad (۲۲-۳)$$

### ۳-۴- نتایج شبیه‌سازی

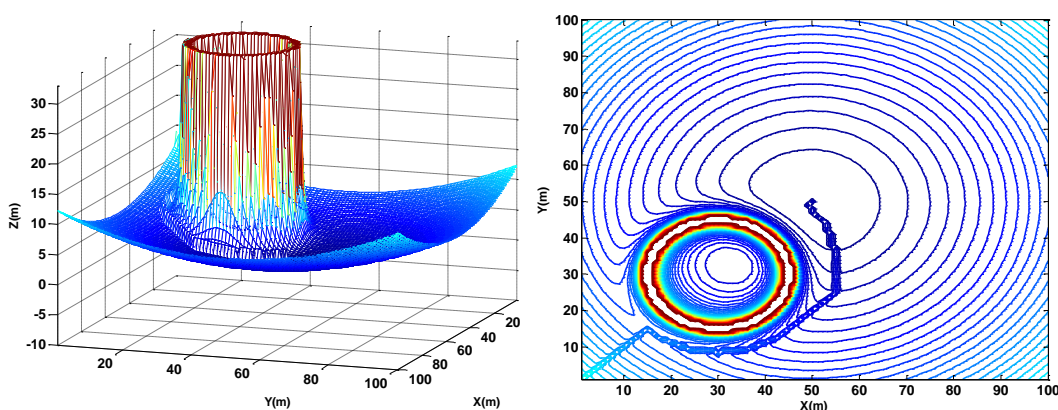
در این بخش نتایج به‌دست آمده از آزمایش‌های مختلف بر روی میدان پتانسیل بهبود داده شده و میدان پتانسیل سنتی توصیف شده و مشاهده خواهد شد که مشکلات میدان پتانسیل سنتی در میدان پتانسیل



بهبود داده شده حل می‌شود. در این شبیه‌سازی‌ها ربات سیار به صورت نقطه در نظر گرفته شده است. که این نقطه دارای جرم و اینرسی نیست و معادلات دینامیک ربات در شبیه‌سازی‌ها وارد نشده است.

### ۳-۴-۱- شبیه‌سازی با میدان پتانسیل بهبود داده شده

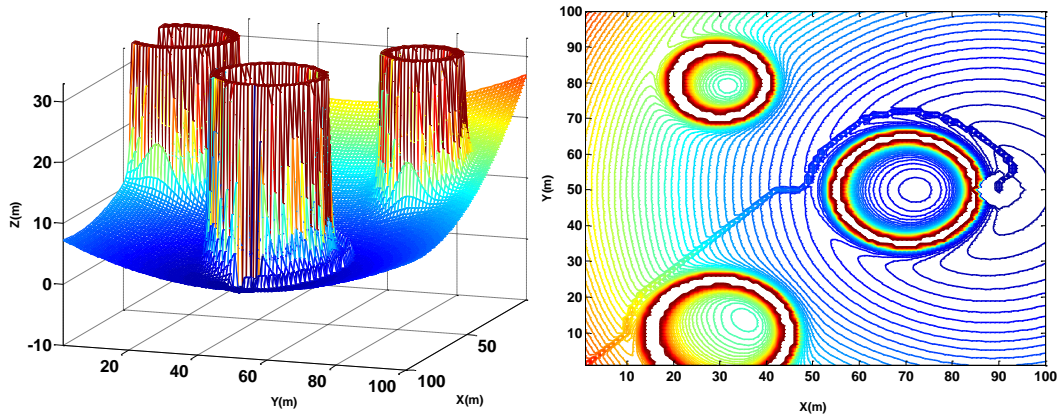
شبیه‌سازی یک: همان‌طور که در شکل ۳-۴ نمایانگر است طراحی مسیر برای ربات سیار با یک مانع صورت گرفته است، مانع در مختصات  $(30, 30)$  قرار دارد و ربات از مبدا  $(0, 0)$  شروع به حرکت کرده و به نقطه هدف در مختصات  $(50, 50)$  می‌رسد. ربات و هدف بر روی یک خط مستقیم واقع شده‌اند و یک مانع هم بین آن‌ها وجود دارد. در این حالت با استفاده از روش‌های مرسوم ربات در کمینه محلی قرار می‌گیرد ولی با استفاده از میدان پتانسیل بهبود داده شده این مسئله حل شده است.



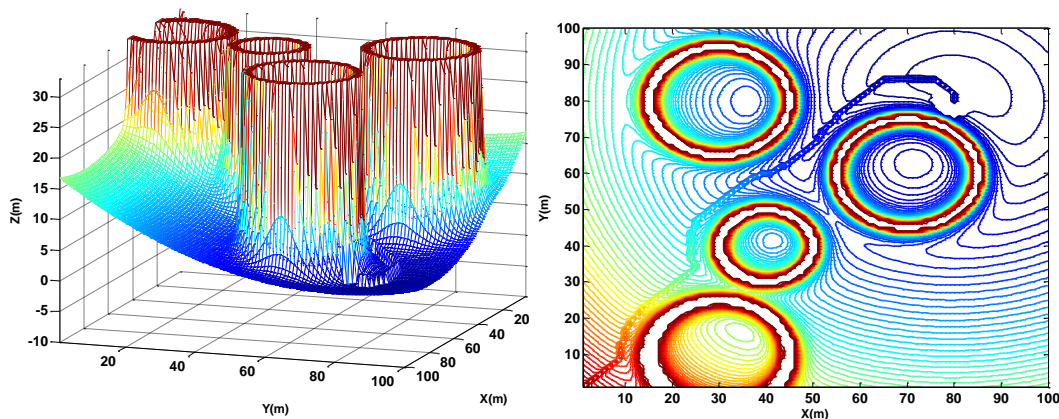
شکل ۳-۴ مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل بهبود داده شده با یک مانع و حل مسئله کمینه محلی برای ربات سیار

شبیه‌سازی دو: در شکل‌های ۳-۵ الف و ۳-۵ ب طراحی مسیر با سه و چهار مانع برای ربات سیار انجام شده است. به طوری که مسیر ربات از نقطه مبدا  $(0, 0)$  شروع و به نقاط هدف به ترتیب در  $(90, 50)$  و  $(80, 80)$  می‌رسد. با توجه به تعریف پتانسیل جاذبه، با افزایش فاصله ربات از هدف مقدار نیرو پتانسیل جاذبه افزایش می‌یابد و ربات با خطر برخورد با موانع روبرو می‌شود که در شکل ۳-۵ الف مشاهده می‌شود که ربات بدون برخورد با موانع به هدف رسیده است. زمانی که موقعیت هدف خیلی نزدیک به مانع است و نیرو یا پتانسیل دافعه خیلی بزرگ‌تر از نیرو یا پتانسیل جاذبه می‌شود؛ تحت این شرایط ربات هرگز نمی‌تواند خود را به هدف برساند. که در این صورت مسئله هدف غیرقابل دسترس

ایجاد می‌شود، در این حالت همانطور که در شکل ۳-۵ ب مشاهده می‌شود، هدف نزدیک مانع قرار دارد و ربات به نقطه هدف رسیده است.



(ب)



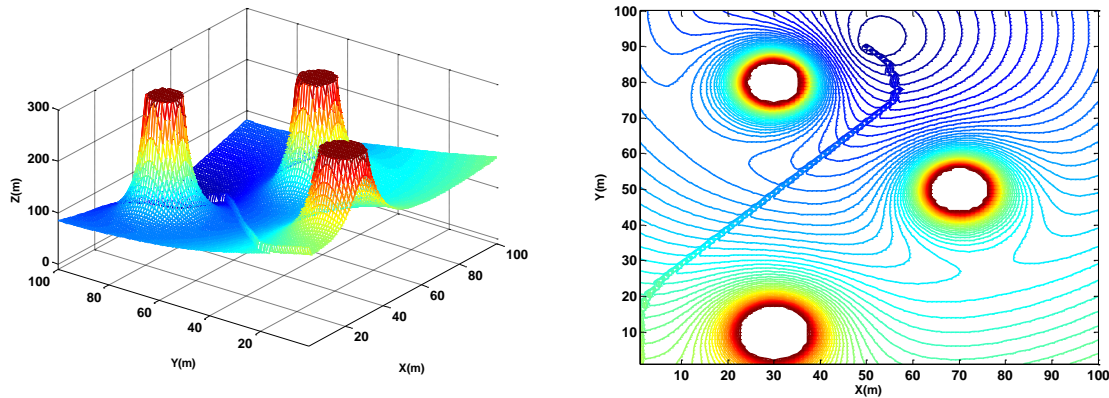
(ج)

شکل ۳-۵ مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل بهبود داده شده (الف) و (ب) حل مسئله هدف غیرقابل دسترس و حرکت ربات از نزدیک موانع

### ۳-۴-۲- شبیه‌سازی با میدان پتانسیل سنتی

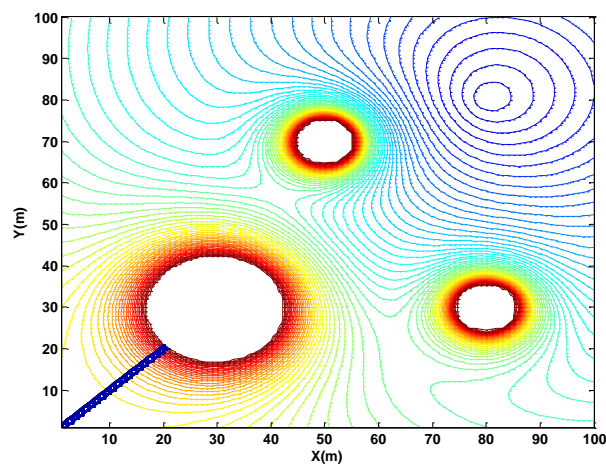
شبیه‌سازی یک: در شکل ۳-۶ با قرار داشتن سه مانع در مسیر حرکت ربات و حرکت از مبدا و رسیدن به نقطه هدف در (۵۰،۹۰) برای ربات سیار با استفاده از روش میدان پتانسیل سنتی شبیه‌سازی شده است. در این روش اگر هدف نزدیک مانع قرار بگیرد مسئله هدف غیرقابل دسترس ایجاد می‌شود و در برنامه خطا ایجاد شده و ربات به نقطه هدف نمی‌رسد. همچنین ربات با فاصله زیاد از موانع حرکت

می‌کند به طوری که ممکن است اگر هدف نزدیک مانع باشد، به دلیل نیروی دافعه مانع، ربات نتواند به هدف برسد.



شکل ۳-۶ مسیر حرکت ربات با استفاده از الگوریتم میدان پتانسیل سنتی

شبیه‌سازی دو: شکل ۳-۷ طراحی مسیر با سه مانع برای ربات سیار را نشان می‌دهد به طوری که نقطه مبدا  $(0,0)$  و به مختصات هدف  $(80,80)$  است. همان طور که دیده می‌شود ربات و هدف بر روی خط مستقیم واقع شده‌اند و یک مانع هم بین آن‌ها وجود دارد و ربات در کمینه محلی قرار گرفته است.



شکل ۳-۷ قرار گرفتن ربات در کمینه محلی



## فصل چهارم

مسیریابی برای ربات سیار با

الگوریتم‌های تکاملی

#### ۴-۱- مقدمه

در این فصل مسیر طراحی شده از منحنی‌های بی‌اسپلاین<sup>۱</sup> تشکیل شده است. به این معنی که چند جمله‌ای درجه سه بوده و همچنین در دو مشتق اول منحنی که معرف سرعت و شتاب ربات است، نیز دارای پیوستگی است [۲۷]. همچنین برای تعیین تعداد و محل نقاط کنترل به گونه‌ای که مسیر حاصل بدون برخورد با موانع<sup>۲</sup> دارای کمترین طول باشد از الگوریتم‌های تکاملی استفاده شده است. دلیل استفاده از این الگوریتم‌ها همگرا شدن جواب‌ها به پاسخ‌های بهینه محلی است. در ادامه ابتدا به نحوه تعریف موانع در فضای دو بعدی پرداخته می‌شود و تابع هزینه به منظور کمینه کردن طول مسیر معرفی می‌شود و سپس به معرفی نمودارهای بی‌اسپلاین پرداخته شده است. در انتها با مثال‌های ساده در فضای دو بعدی و مقایسه با حل، تحلیل درستی روش را مورد بررسی قرار داده و در نهایت به طراحی مسیرهای مختلف در فضای دوبعدی برای ربات پرداخته می‌شود.

#### ۴-۲- اجزای مسئله طراحی مسیر

تابع هزینه در نظر گرفته شده برای بهینه‌سازی طول مسیر با اجتناب از برخورد با موانع به صورت روابط ۱-۴ و ۲-۴ تعریف می‌شود [۲۸]:

$$C = \int W \cdot ds \quad (1-4)$$

$$ds = \sqrt{dx^2 + dy^2} \quad (2-4)$$

که  $ds$  معرف المان طول مسیر است. همچنین  $W$  به منظور وزن دادن به تابع هزینه در صورت برخورد و عبور مسیر از موانع است و به عنوان جریمه‌های مختلف به طوری که فاصله‌های طی شده به حداقل برسد اضافه شده است. جریمه‌ها در نتیجه محدودیت‌های اعمال شده برای اجتناب از برخورد با موانع در نظر گرفته شده است. با در نظر گرفتن این نکته که منحنی‌های بی‌اسپلاین به دست آمده دارای نقاط  $[x_i, y_i]$   $i = 1, \dots, N$  است، می‌توان رابطه ۳-۴ را به صورت گسسته و به فرم ۳-۴ بیان کرد.

<sup>1</sup> - B-spline Curves

<sup>2</sup> - Obstacle Avoidance

$$L = \sum_{i=1}^N \sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2]} \quad (3-4)$$

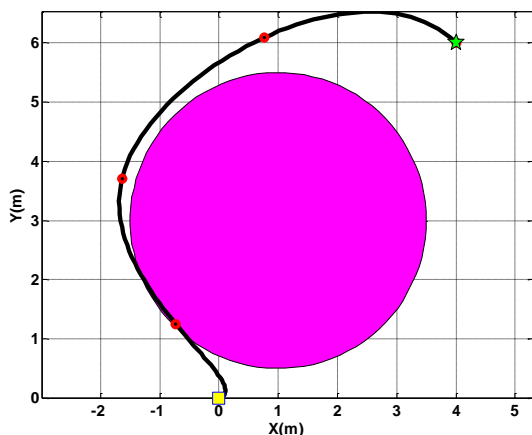
$$C = L * (1 + B * W_{xy}) \quad (4-4)$$

در رابطه ۳-۴،  $N$  تعداد نقاط منحنی است که تابع هدف برای کمینه کردن طول مسیر می‌باشد. رابطه ۴-۴ تابع هزینه کل با در نظر گرفتن قیود مربوط به موانع نوشته شده است که در آن  $B$  ضریب مثبت است و  $W_{xy}$  تابع جریمه برای زمانی که منحنی وارد موانع می‌شود. اگر  $W_{xy}$  صفر باشد تغییری در تابع هزینه ایجاد نمی‌شود. یعنی منحنی با موانع برخوردی نداشته است. در روابط ۴-۵ و ۴-۶،  $r_{ob}$  شعاع موانع و  $[x_{ob}, y_{ob}]$  مختصات مرکز موانع و  $R_{dis}$  فاصله هر نقطه از مرکز موانع است.

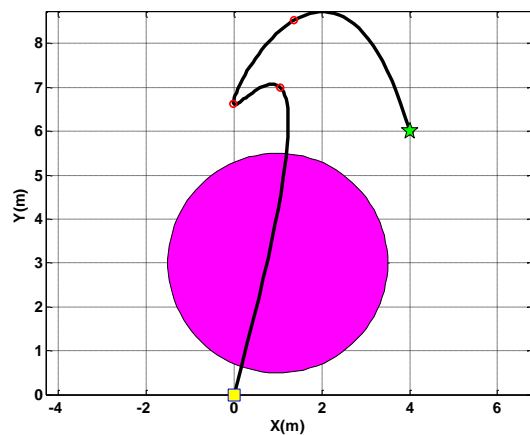
$$R_{dis} = \sqrt{(x_i - x_{ob})^2 + (y_i - y_{ob})^2} \quad (5-4)$$

$$W_{xy} = \begin{cases} 0 & R_{dis} \geq r_{ob} \\ \left(1 - \frac{R_{dis}}{r_{ob}}\right)^2 & R_{dis} < r_{ob} \end{cases} \quad (6-4)$$

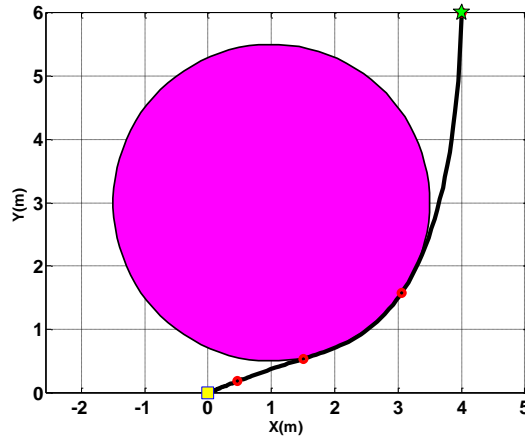
در اینجا با تعریف یک مثال به تأثیر قرار گرفتن منحنی مسیر درون موانع و تابع جریمه تولید شده پرداخته شده است. همان‌طور که در شکل‌های ۴-۱ تا ۴-۳ مشخص است یک موانع در مختصات (۱,۳) با شعاع ۲,۵ متر قرار دارد و مشخصات مسیر به این صورت است که مسیر از نقطه (۰,۰) شروع و به نقطه هدف در (۴,۶) می‌رسد. با تغییر جزئی نقطه کنترل که توسط الگوریتم تکاملی به دست آمده است منحنی وارد موانع می‌شود. جدول ۴-۱ جریمه‌ها و تابع هزینه به دست آمده را نشان می‌دهد.



شکل ۲-۴ منحنی درون موانع دور از مرکز



شکل ۱-۴ منحنی درون موانع نزدیک‌ترین حالت به مرکز



شکل ۳-۴ منحنی خارج از مانع و تقریباً مماس به مرز آن

جدول ۱-۴ عبور منحنی از درون موانع و افزایش هزینه آن در مقایسه با مسیر مماس بر مانع

وضعیت منحنی نسبت به مانع	تابع هزینه	جریمه	ردیف
درون مانع نزدیک‌ترین حالت به مرکز	۲۹۰.۵۷۳	۰.۰۳۹	۱
درون مانع دور از مرکز	۱۲.۲۴۳	۰.۰۰۰۷	۲
خارج از مانع و تقریباً مماس به مرز آن	۸.۰۸۲	۰	۳

#### ۱-۲-۴ منحنی‌های بی‌اسپلاین

منحنی‌های بی‌اسپلاین دارای رفتار محلی هستند و توسط نقاطی به نام نقاط کنترل، کنترل می‌شوند. به عبارت دیگر با تغییر و جابه‌جا کردن یک نقطه کنترل تنها قسمت کوچکی از منحنی تحت تأثیر قرار گرفته و سایر قسمت‌های منحنی بدون تغییر باقی می‌مانند [۲۹].

همچنین این منحنی‌ها دارای سازگاری مناسبی با الگوریتم‌های تکاملی می‌باشند که در این کار از روش بهینه‌سازی تصادفی به منظور به دست آوردن نقاط کنترل منحنی‌های بی‌اسپلاین به گونه‌ای که طول مسیر با اجتناب از برخورد با موانع کمینه شود استفاده می‌شود [۳۰].

در بسیاری از پیاده‌سازی‌های رایج رباتیک مسیر حرکت ربات را به صورت مسیر درجه‌دو در نظر می‌گیرند. در حالی که در این پژوهش منحنی‌های اسپلاین مرتبه سه<sup>۱</sup> مورد استفاده قرار گرفته است. از مزایای این منحنی‌ها بر منحنی‌های یادشده می‌توان به موارد زیر اشاره نمود:

<sup>۱</sup> - Cubic Spline



- طراحی حرکت ربات به صورت مسیر همواری که قابل پیاده‌سازی عملی باشد.
- این منحنی برای مسائلی که دارای نقطه شروع و پایان مشخصی بوده و نقاط کنترلی میانی وجود دارد که ربات باید از آن‌ها عبور کند؛ بسیار پرکاربرد است.

#### ۴-۲-۲- الگوریتم‌های تکاملی و نحوه بهینه‌سازی

الگوریتم‌های تکاملی یکی از ابزارهای تصادفی بهینه‌سازی است [۳۱]. در اینجا از چهار الگوریتم تکاملی استفاده شده است که در پیوست الف به اختصار هر کدام توضیح داده شده است. در اینجا هدف از الگوریتم‌های تکاملی به دست آوردن نقاط کنترل منحنی‌های بی‌اسپلین به‌گونه‌ای است که تابع هزینه معرفی شده در قسمت قبل کمینه گردد. با توجه به اینکه نقاط ابتدا و انتهای مسیر مشخص است الگوریتم‌های تکاملی وظیفه تعیین  $n_s - 2$  نقطه کنترل را بر عهده دارند، همان‌طور که در قسمت نتایج خواهیم دید، تعداد نقاط کنترل برای منحنی‌ها  $n_s = 5$  در نظر گرفته شده است که به این معناست که الگوریتم‌های تکاملی وظیفه پیدا کردن محل سه نقطه کنترل میانی را به‌گونه‌ای که طول مسیر کمینه شود را بر عهده دارند. هر چه تعداد نسل‌ها بیشتر باشد پاسخ نهایی به پاسخ واقعی نزدیک‌تر خواهد بود اما این امر به‌طور قابل‌ملاحظه‌ای باعث افزایش هزینه محاسباتی نیز می‌گردد. روش‌های گرادینتی بهینه‌سازی بسیار حساس به حدس اولیه بوده و در صورتی که حدس اولیه نامناسب باشد سرعت همگرایی به جواب بسیار پایین خواهد بود. اما یکی از برتری‌های الگوریتم‌های تکاملی عدم نیاز به حدس اولیه می‌باشد [۳۲].

#### ۴-۳- نتایج شبیه‌سازی

در این قسمت به منظور اطمینان از درستی الگوریتم، آن را برای حالت ساده آزمایش کرده و پاسخ به دست آمده، با پاسخ تحلیلی مسئله مقایسه می‌شود. سپس به طراحی مسیر در محیط دوبعدی ساده شامل یک مانع تا محیط‌هایی شامل موانع متعدد پرداخته می‌شود. مشخصات سیستم برای پیاده‌سازی

الگوریتم‌ها مطابق جدول ۲-۴ است.

جدول ۲-۴ مشخصات سیستم

CPU	Intel® Core™ i5-2450M CPU @ 2.50GHz
RAM	4.00 GB (3.84 GB usable)
Motherboard	Acer SPIRE E1-571 G
Graphics	NVIDIA® GeForce® GT 620M

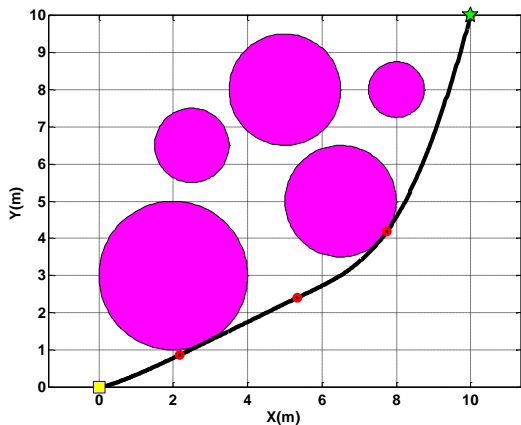
مسئله طراحی مسیر با چندین مانع برای محاسبه کوتاه‌ترین مسیر از نقطه شروع تا نقطه پایان بدون برخورد با موانع ارائه شده است. محیط نمونه شامل یک فضای دو بعدی با موانع ثابت است. اطلاعات محیط شامل محدودیت‌های فضای کاری و شکل، موقعیت و موانع ثابت در فضای کاری می‌باشد. در فضای کاری موانع بدون حرکت فرض شده است.

فرضیات:

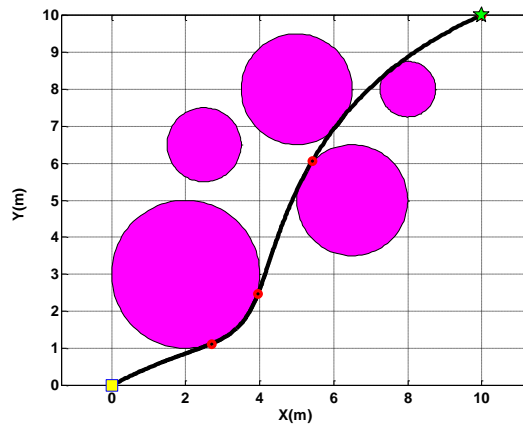
- موقعیت موانع با توجه به سیستم مختصات مرجع شناخته شده است.
- موانع به صورت دایره ای می باشند.
- موانع می توانند در مجاورت هم باشند اما با هم تداخل نداشته باشند.
- الگوریتم برنامه ریزی مسیر اجرا می شود تا ربات به هدف مشخص شده برسد.

در مسئله طراحی مسیر موقعیت شروع، موقعیت هدف و موقعیت موانع به عنوان ورودی‌ها داده و کوتاه‌ترین مسیر بدون برخورد با موانع به عنوان خروجی مسئله می باشند.

**شبیه‌سازی یک:** در جدول ۳-۴ مقایسه‌ای برای یافتن پاسخ بهینه بین نقطه مبدا و نقطه (۱۰,۱۰) با در نظر گرفتن پنج مانع صورت گرفته است. همین‌طور که مشخص است با کاهش تعداد نسل هزینه محاسباتی به‌طور چشمگیری کاهش یافته است اما از طرفی این امر منجر به ایجاد مسیرهایی با طول بیشتر شده است.



شکل ۴-۵ نسل با طول مسیر ۱۵.۲۳

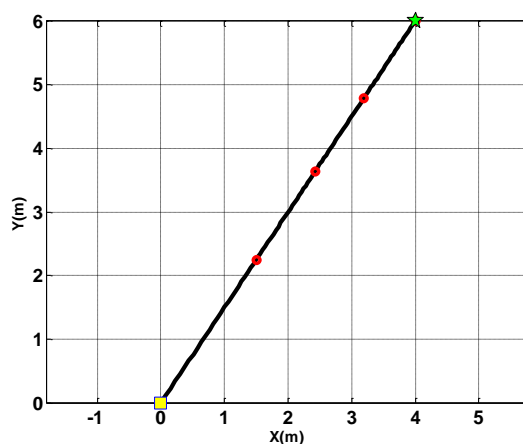


شکل ۴-۴ نسل با طول مسیر ۱۴.۹

جدول ۳-۴ تأثیر افزایش نسل‌ها در پاسخ

شماره شکل	تعداد نسل‌ها	طول مسیر (متر)	هزینه محاسباتی (ثانیه)
۴	۲۰۰	۱۴.۹۴۱	۴۷.۴۷۰
۵	۱۰۰	۱۵.۲۳۰	۲۴.۱۲۷

شبیه‌سازی دو: در این مرحله برای اطمینان از جواب‌های به‌دست‌آمده توسط الگوریتم، مسیر در فضای دوبعدی بدون مانع و یک مانع به‌دست آمده است. می‌دانیم کوتاه‌ترین فاصله بین دو نقطه خط راست است. در اینجا از نقطه شروع  $(0,0)$  به نقطه پایان  $(4,6)$  می‌رویم. حل تحلیلی این مسئله برابر است با  $7.2111$  که با مقایسه این مقدار با مقدار به‌دست آمده از الگوریتم معرفی شده در جدول ۴-۴ دارای خطای تقریباً صفر است.

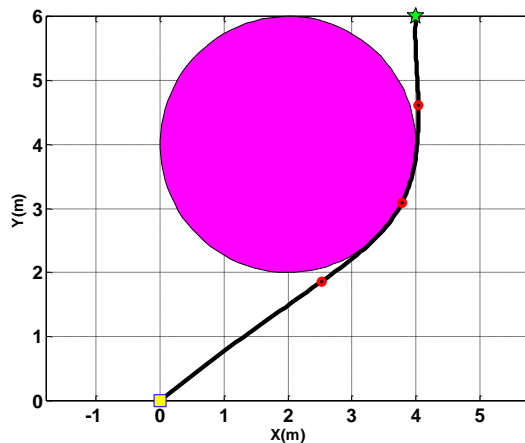


شکل ۴-۶ مسیر ربات خالی از مانع

جدول ۴-۴ بدون مانع

تعداد نقاط اسپلین	طول منحنی (متر)	تعداد نسلها	هزینه محاسباتی (ثانیه)
۵	۷.۲۱۱۱	۱۰۰	۳۲.۲۳

حال با قرار دادن مانع مطابق شکل ۷-۴ به بررسی نتایج حاصل پرداخته شده است.



شکل ۷-۴ مسیر ربات با یک مانع

جدول ۴-۵ یک مانع

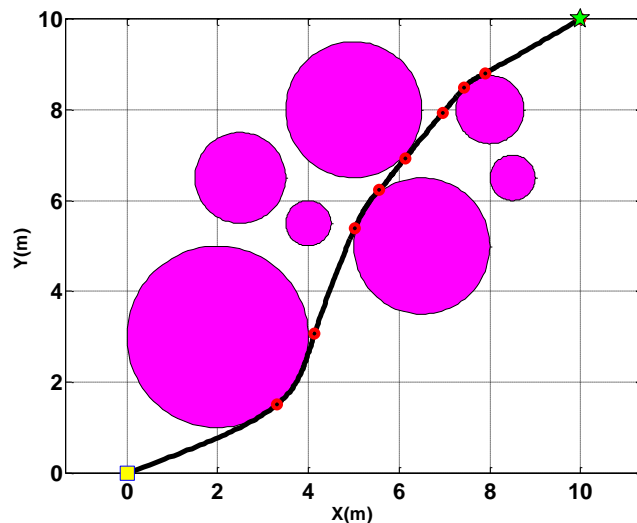
تعداد نقاط اسپلین	نقطه شروع	نقطه پایان	مختصات مانع (متر)	شعاع مانع (متر)	طول منحنی (متر)	تعداد نسلها	هزینه محاسباتی (ثانیه)
۵	(۰،۰)	(۴،۶)	(۲،۴)	۲	۷.۸۶	۱۰۰	۱۸.۶۴

مسیر با استفاده از منحنی‌های اسپلین ایجاد شد، حال با استفاده از چهار الگوریتم بهینه‌سازی که شامل  $PSO^1$ ,  $ICA^2$ ,  $GA^3$ ,  $BA^4$  می‌باشد، بهینه‌سازی بر روی مسیر طی شده انجام خواهد شد. در اینجا تابع هزینه طول مسیر طی شده برای ربات در نظر گرفته شده است و در نهایت به مقایسه الگوریتم‌های بهینه‌سازی پرداخته شده است.

شبیه‌سازی سه: در ابتدا چهار الگوریتم برای مسیر دوبعدی با هفت مانع در شکل ۴-۸ مقایسه شده

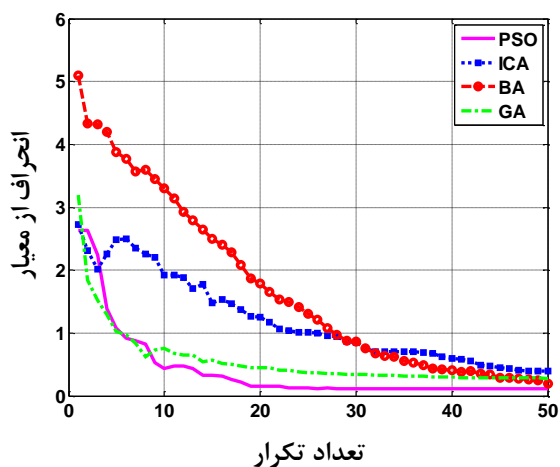
1 - Particle Swarm Optimization  
 2 - Imperialist Competitive Algorithm  
 3 - Genetic Algorithm  
 4 - Bees Algorithm

است. شکل‌های ۹-۴ و ۱۰-۴ بر اساس میانگین و انحراف از معیار الگوریتم‌ها را مقایسه کرده است.

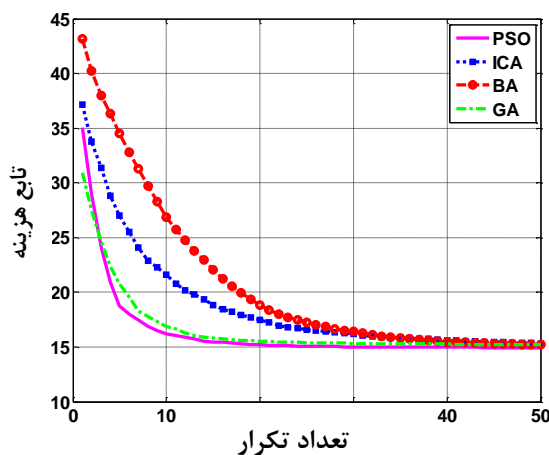


شکل ۴-۸ مسیر ربات با هفت مانع

در شکل‌های ۹-۴ و ۱۰-۴ مشاهده می‌شود که الگوریتم PSO نسبت به سایر الگوریتم‌ها، زمانی که تعداد موانع هفت عدد است دارای تابع هزینه کمتری است.

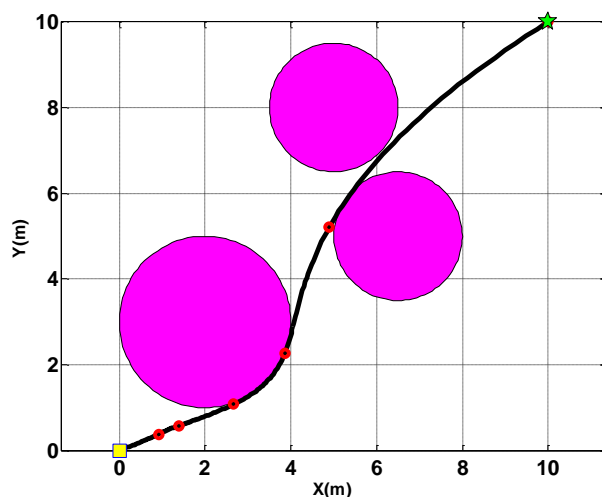


شکل ۱۰-۴ مقایسه مسیر با هفت مانع براساس انحراف از معیار

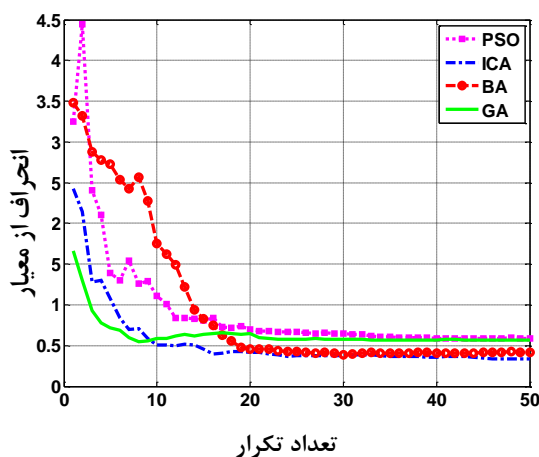


شکل ۹-۴ مقایسه مسیر با هفت مانع براساس تابع هزینه

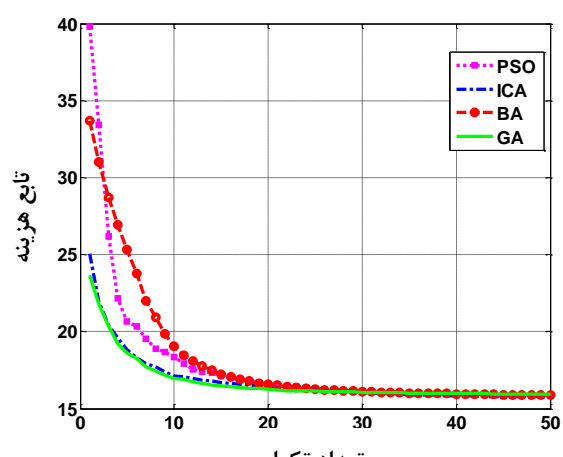
شبیه‌سازی چهار: حال بین چهار الگوریتم، برای مسیر دوبعدی با سه مانع در شکل ۱۱-۴ مقایسه صورت گرفته است. در شکل‌های ۱۲-۴ و ۱۳-۴ نیز بر اساس میانگین و انحراف از معیار مقایسه صورت گرفته است.



شکل ۱۱-۴ مسیر ربات با سه مانع



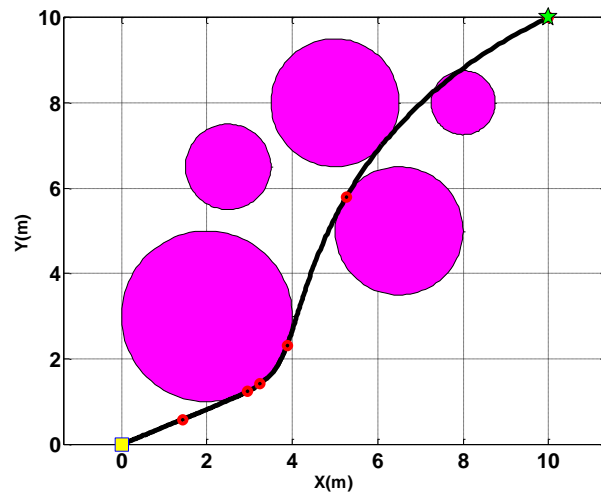
شکل ۱۳-۴ مقایسه مسیر با سه مانع براساس انحراف از معیار



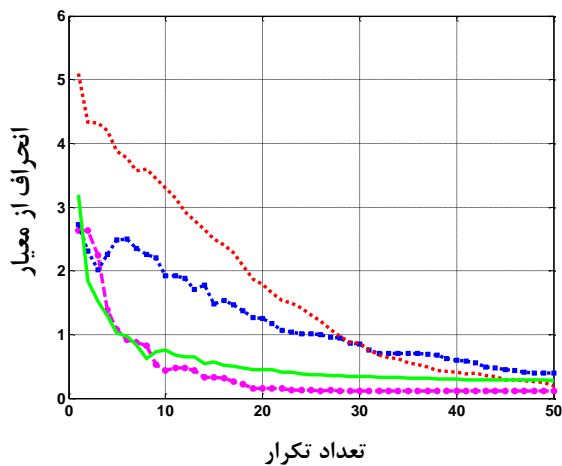
شکل ۱۲-۴ مقایسه مسیر با سه مانع براساس تابع هزینه

در شکل‌های ۱۲-۴ و ۱۳-۴ مشاهده می‌شود که الگوریتم ICA نسبت به سایر الگوریتم‌ها، زمانی که تعداد موانع سه عدد است دارای تابع هزینه کمتری است.

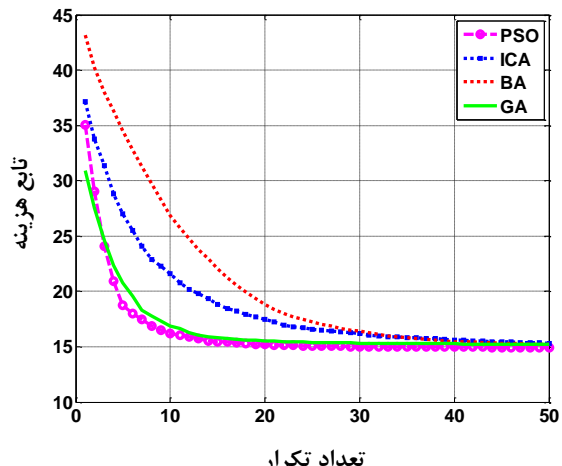
شبه‌سازی پنج: حال مقایسه بین چهار الگوریتم بهینه‌سازی با پنج مانع به صورت شکل ۱۴-۴ انجام شده است. در اینجا به ترتیب الگوریتم PSO و GA و ICA و BA دارای تابع هزینه کمتر (طول مسیر کوتاه‌تر) هستند.



شکل ۴-۱۴ مسیر ربات با چهار مانع



شکل ۴-۱۰ مقایسه مسیر با چهار مانع براساس انحراف از معیار



شکل ۴-۹ مقایسه مسیر با چهار مانع براساس تابع هزینه

شبیه‌سازی شش: حال در این مرحله هرکدام از الگوریتم‌ها را بر اساس تعداد موانع و تکرار مختلف اجرا کرده و در جداول ۴-۶ و ۴-۷ باهم مقایسه شده است. در شکل‌های ۴-۱۷ تا ۴-۲۰ نیز تعدادی از شبیه‌سازی‌های انجام شده با مسیرها و تعداد موانع مختلف نشان داده شده است. در این مقایسه دیده می‌شود که الگوریتم PSO در تعداد موانع پنج و تعداد تکرار ۵۰ و ۱۰۰ نسبت به سایر الگوریتم‌ها به مسیر کوتاه‌تری منجر شده است و در تعداد موانع هفت و تعداد تکرار ۱۰۰ این الگوریتم رقابت استعماری است که نسبت به سایر الگوریتم‌ها طول مسیر کوتاه‌تری دارد. از نظر هزینه محاسباتی نیز الگوریتم BA نسبت به الگوریتم‌های دیگر دارای وضعیت بهتری است.

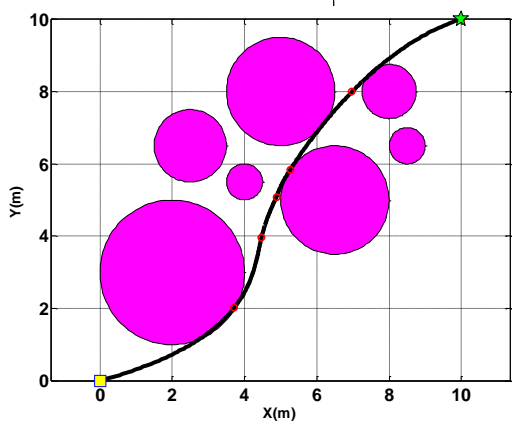
جدول ۴-۶ مقایسه الگوریتم‌های تکاملی با تعداد تکرار ۱۰۰

نوع الگوریتم	تعداد تکرار	زمان اجرای الگوریتم (ثانیه)	مسافت طی شده (متر)	تعداد موانع
PSO	۱۰۰	۳۷.۶۳۵	۱۴.۹۰۳	۵
ICA	۱۰۰	۴۳.۰۹۲	۱۴.۹۲۵	۵
BA	۱۰۰	۲۵.۵۹۶	۱۴.۹۶۸	۵
GA	۱۰۰	۴۳.۴۳۱	۱۴.۹۶۳	۵
PSO	۱۰۰	۳۷.۷۴۲	۱۴.۸۹۴	۷
ICA	۱۰۰	۴۱.۷۵۹	۱۴.۸۸۳	۷
BA	۱۰۰	۲۴.۹۱۸	۱۵.۲۹۴	۷
GA	۱۰۰	۴۴.۴۸۶	۱۵.۰۹۳	۷

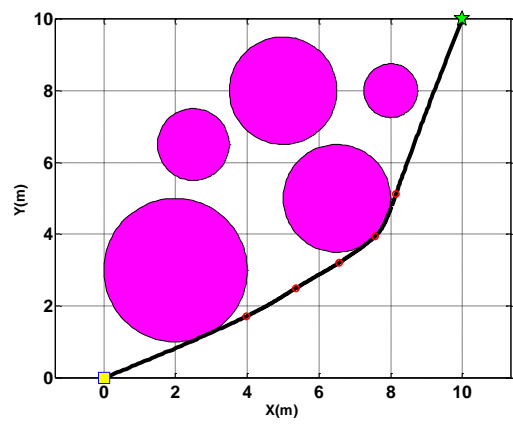
جدول ۴-۷ مقایسه الگوریتم‌های تکاملی با تعداد تکرار ۵۰

نوع الگوریتم	تعداد تکرار	زمان اجرای الگوریتم (ثانیه)	مسافت طی شده (متر)	تعداد موانع
PSO	۵۰	۱۸.۶۰۹	۱۴.۹۱۳	۵
ICA	۵۰	۲۲.۴۲	۱۴.۹۸۱	۵
BA	۵۰	۱۲.۷۸۶	۱۴.۹۴۸	۵
GA	۵۰	۲۲.۵۸۸	۱۵.۱۲۶	۵

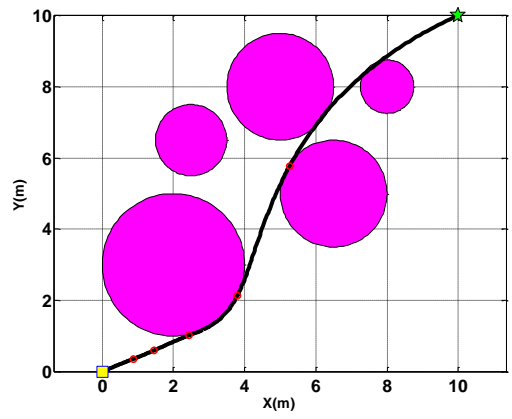




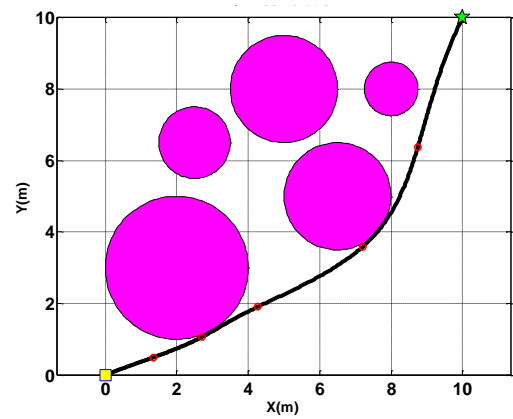
شکل ۴-۱۸ مسیر خروجی GA



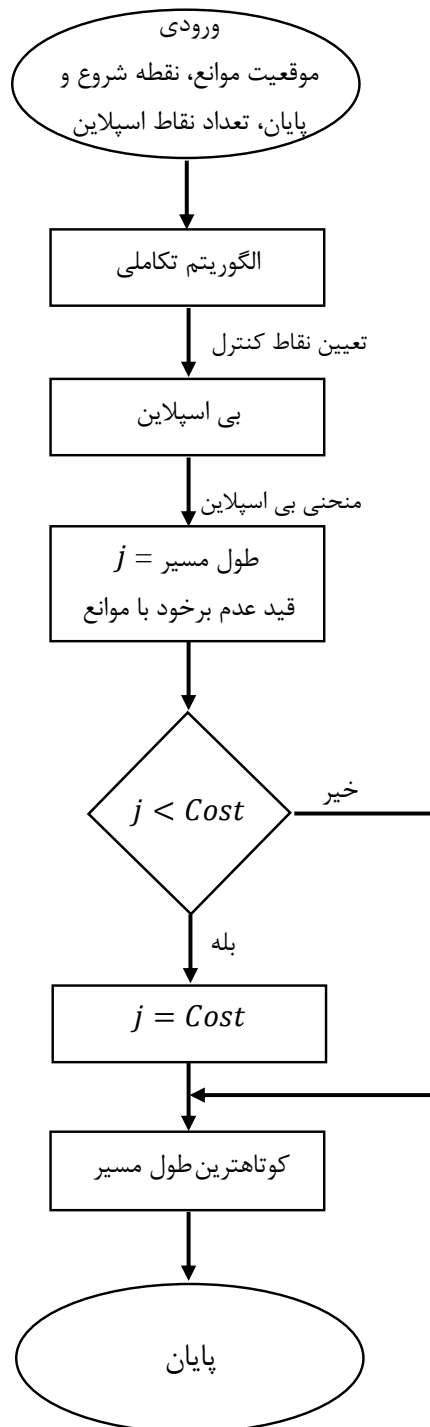
شکل ۴-۱۷ مسیر خروجی BA



شکل ۴-۲۰ مسیر خروجی PSO



شکل ۴-۱۹ مسیر خروجی ICA



شکل ۴-۲۱ مراحل مسیریابی با الگوریتم‌های تکاملی

## فصل پنجم

طراحی مسیر برای ربات با

هدف کمینه انرژی مصرفی

## ۵-۱ مقدمه

با پیشرفت قابل توجه در علم رباتیک، کاربرد ربات‌های سیار در بسیاری از زمینه‌ها که شامل جست‌وجو و شناسایی در محیط‌های ناشناخته، نظامی، توان‌بخشی، تمیز کردن و سرویس‌های شخصی می‌باشد، گسترش یافته است. آن‌ها برای انجام مأموریت‌های خود به منبع انرژی نیاز دارند. بسیاری از ربات‌های سیار به همراه خود یک باتری که دارای انرژی محدود است حمل می‌کنند. بنابراین کاربردهای آن‌ها به میزان انرژی موجود در باتری آن‌ها هم محدود شده است؛ زیرا جایگزینی یک منبع انرژی جدید برای آن‌ها در انجام مأموریت غیرممکن یا پرهزینه است [۳۳]. همچنین برای کمینه کردن انرژی باتری یک ربات سیار چرخ‌دار، انرژی مصرفی باتری به صورت یک تابع هزینه در نظر گرفته می‌شود [۳۴].

در ادامه با در نظر گرفتن قیود غیرهولونومیک چرخ‌ها، معادلات دینامیکی ربات سیار استخراج می‌شود. این معادلات در فضای حالت بیان شده و به عنوان قیود مسئله کنترل بهینه در نظر گرفته می‌شود. سپس با استفاده از اصل مینیمم پونتریاگن<sup>۱</sup> و حل غیرمستقیم مسئله کنترل بهینه، معادلات بهینگی به صورت یک مجموعه معادلات دیفرانسیل غیرخطی استخراج شده که با استفاده از روش‌های عددی حل می‌شود.

## ۵-۲- مدل ربات سیار

از معادلات ۲-۸۰ و ۲-۸۱، معادله دیفرانسیل ۵-۱ به دست می‌آید.

$$\dot{W} + A W = B u \quad (۱-۵)$$

$$A = \begin{bmatrix} a_1 & a_2 \\ a_2 & a_1 \end{bmatrix} = J^{-1} \left( F_v + \frac{K_t K_b n g^2}{R_a} \right), B = \begin{bmatrix} b_1 & b_2 \\ b_2 & b_1 \end{bmatrix} = J^{-1} \frac{V_s K_t n g}{R_a} \quad (۲-۵)$$

بردار حالت به صورت  $z = [v \ W]^T$  تعریف می‌شود. بنابراین رابطه  $v$  و  $W$  با  $w_R, w_L$  به صورت ۵-۳ است:

$$z = \begin{bmatrix} v \\ W \end{bmatrix} = T_q \begin{bmatrix} w_R \\ w_L \end{bmatrix} = T_q W, T_q = \begin{bmatrix} r/2 & r/2 \\ r/2b & -r/2b \end{bmatrix} \quad (۳-۵)$$

با استفاده از معادلات ۵-۱ و ۵-۳ داریم:

$$\dot{z} = -\bar{A} z + \bar{B} u \quad (۴-۵)$$

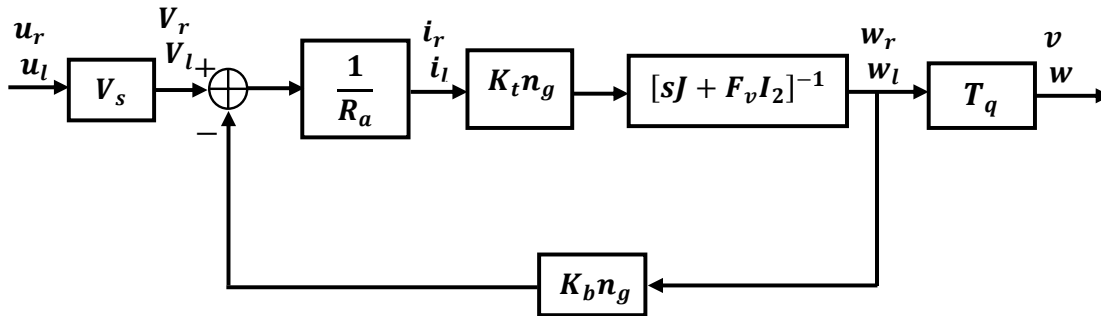
<sup>۱</sup> - Pontryagin's Minimum Principle

که در آن:

$$\bar{A} = T_q A T_q^{-1} = \begin{bmatrix} \pi_v & 0 \\ 0 & \pi_w \end{bmatrix} = \begin{bmatrix} a_1 + a_2 & 0 \\ 0 & a_1 - a_2 \end{bmatrix} \quad (5-5)$$

$$\bar{B} = T_q B = \begin{bmatrix} B_1 & B_1 \\ B_2 & -B_2 \end{bmatrix} = \begin{bmatrix} r(b_1 + b_2)/2 & r(b_1 + b_2)/2 \\ r(b_1 - b_2)/2b & -r(b_1 - b_2)/2b \end{bmatrix} \quad (6-5)$$

دینامیک کلی موتور ربات‌های چرخ‌دار به صورت شکل ۵-۱ می‌باشد.



شکل ۵-۱ بلوک دیاگرام دینامیک ربات سیار چرخ‌دار

### ۵-۳- مصرف انرژی ربات سیار چرخ‌دار

انرژی گرفته‌شده از باتری به انرژی مکانیکی برای راه‌اندازی موتور و تلفات حرارتی در مقاومت آرمیچر تبدیل می‌شود. در ربات‌های سیار انرژی به وسیله مقاومت داخلی باتری، مقاومت تقویت‌کننده موجود در راه‌انداز موتور، مقاومت آرمیچر و اصطکاک ویسکوز موتورها تلف می‌شود. استفاده از مدولاسیون عرض پالس برای کنترل سرعت موتور در مقایسه با کنترل ولتاژ به دلیل تولید گرمای کم دارای بازدهی بیشتر است [۳۵]. بنابراین انرژی کل باتری ربات سیار، برای کمینه کردن آن به صورت تابع هزینه‌ی ۷-۵ تعریف می‌شود.

$$E_W = \int I_a^T V dt = V_s \int I_a^T u dt \quad (7-5)$$

که در آن  $V = [V_r \ V_l]^T$  ولتاژ ورودی که برای موتورها به کار گرفته‌شده است و به صورت رابطه  $u = \frac{V}{V_s} = [u_r \ u_l]^T$  است. همچنین محدودیت‌های ۵-۸ و ۵-۹ در ولتاژ و گشتاور ربات‌های سیار در نظر گرفته شده است.

$$-u_{max} \leq u_r \leq u_{max}, -u_{max} \leq u_l \leq u_{max} \quad (۸-۵)$$

$$-\tau_{max} \leq \tau_r \leq \tau_{max}, -\tau_{max} \leq \tau_l \leq \tau_{max} \quad (۹-۵)$$

از معادلات ۲-۸۰ و ۵-۳،  $E_W$  به صورت معادله ۵-۱۰ نوشته می شود.

$$E_W = \int (K_1 u^T u - K_2 z^T T_q^{-T} u) dt \quad (۱۰-۵)$$

که در آن  $K_1 = V_s^2 / R_a$  و  $K_2 = K_b n_g V_s / R_a$  می باشد.

از معادلات ۲-۸۰ و ۲-۸۱ تابع هزینه  $E_W$  به صورت ۵-۱۱ تعریف می شود.

$$E_W = R_a \int I_a^T I_a dt + F_v \frac{K_b}{K_t} \int z^T T_q^{-T} T_q^{-1} z dt + \frac{K_b}{K_t} \int \dot{z}^T T_q^{-T} J^T T_q^{-1} z dt \quad (۱۱-۵)$$

که قسمت اول،  $E_R (= R_a \int I_a^T I_a dt)$  انرژی تلف شده به وسیله مقاومت آرمیچر در موتورها، قسمت

دوم  $E_F (= F_v \frac{K_b}{K_t} \int z^T T_q^{-T} T_q^{-1} z dt)$  مربوط به تلفات سرعت ناشی از اصطکاک ویسکوز است. قسمت

سوم  $E_K (= \frac{K_b}{K_t} \int \dot{z}^T T_q^{-T} J^T T_q^{-1} z dt)$  انرژی جنبشی ذخیره شده در ربات سیار است و زمانی که ربات

با سرعت ثابت حرکت می کند و یا سرعت اولیه و نهایی آن باهم برابرند، صفر می باشد. این بدین معناست

که سهم خالص مصرف انرژی از آخرین ترم صفر است.

## ۵-۴-۴- کمینه انرژی برای حرکت ربات سیار

مسیر حرکت یک ربات سیار معمولاً از خطوط مستقیم و قوسی شکل تشکیل شده است. در این قسمت

به بررسی انرژی ربات سیار بر روی خط مستقیم و حرکت در امتداد مسیر منحنی پرداخته می شود.

### ۵-۴-۱- حرکت در امتداد خط مستقیم

در این قسمت به کمینه کردن حرکت انتقالی ربات سیار در امتداد خط مستقیم که منجر به حداقل

کردن انرژی مصرفی متناسب با منحنی سرعت می باشد؛ پرداخته شده است. هدف از حل مسئله کنترل

بهینه تعیین متغیرهای کنترلی و به حداقل رساندن تابع هزینه با توجه به قیود داده شده می باشد. تحت

حرکت ربات، بردار  $q(t) = [x(t) \ y(t) \ \theta(t)]^T$  موقعیت و بردار  $Z(t) = [v(t) \ w(t)]^T$  سرعت ربات

می باشد. با توجه به زمان اولیه و نهایی  $t_0, t_f$  و سرعت  $v(t)$  و ورودی  $u(t)$ ، هدف به حداقل رساندن

تابع هزینه ۵-۱۲ می باشد.

$$E_W = \int_{t_0}^{t_f} (K_1 u^T u - K_2 z^T T_q^{-T} u) dt \quad (12-5)$$

که قيود آن عبارتند از:

$$1- \text{ موقعیت اولیه و نهایی: } q(t_0) = [x_0 \ 0 \ 0]^T, q(t_f) = [x_f \ 0 \ 0]^T$$

$$2- \text{ سرعت اولیه و نهایی: } z(t_0) = [v_s \ 0]^T, z(t_f) = [v_f \ 0]^T$$

۳- محدوده تغییرات ولتاژ باتری  $u_{max}$  و محدوده تغییرات گشتاور با  $\tau_{max}$  نشان داده شده است.

### ۵-۴-۲- حرکت در امتداد مسیر منحنی

در این قسمت به کمینه کردن حرکت ربات سیار در امتداد مسیر منحنی که منجر به حداقل کردن انرژی مصرفی متناسب با منحنی سرعت می باشد؛ پرداخته شده است. هدف به حداقل رساندن تابع هزینه ۵-۱۲ می باشد. که قيود آن عبارتند از:

$$1- \text{ موقعیت اولیه و نهایی: } q(t_0) = [x_0 \ y_0 \ \theta_0]^T, q(t_f) = [x_f \ y_f \ \theta_f]^T$$

$$2- \text{ سرعت اولیه و نهایی: } z(t_0) = [v_s \ w_s]^T, z(t_f) = [v_f \ w_f]^T$$

۳- محدوده تغییرات ولتاژ باتری  $u_{max}$  و محدوده تغییرات گشتاور با  $\tau_{max}$  نشان داده شده است.

### ۵-۴-۳- بیان مسئله

با فرض اینکه سرعت اولیه و نهایی صفر و موقعیت اولیه نیز صفر است، مسئله کمینه سازی انرژی به صورت ۵-۱۳ تعریف می شود.

$$E_W = \int_0^{t_f} (K_1 u^T u - K_2 z^T T_q^{-T} u) dt \quad (13-5)$$

که قيود آن عبارتند از:

$$\dot{z} = -\bar{A} z + \bar{B} u \quad (14-5)$$

$$z(0) = z(t_f) = [0 \ 0]^T \quad (15-5)$$

$$q_f = \int_0^{t_f} T_p z dt = [x_f \ 0 \ 0]^T \quad (16-5)$$

$$\begin{bmatrix} -u_{max} \\ -u_{max} \end{bmatrix} \leq u = \begin{bmatrix} u_r \\ u_l \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ u_{max} \end{bmatrix} \quad (17-5)$$

برای کمینه کردن معادله ۵-۱۳ از اصل مینیمم پونتریاگن استفاده شده است. در اینجا کمینه‌سازی معادله ۵-۱۲ با قیود ۵-۱۵ تا ۵-۱۷ بر روی سیستم با معادله ۵-۱۴ انجام می‌شود. ضرایب لاگرانژ برای قیود حالت معادله ۵-۱۶ به صورت  $a = [a_x \ a_y \ a_\theta]$  و ضرایب برای معادله ۵-۱۴ به صورت بردار  $\lambda = [\lambda_v \ \lambda_w]^T$  تعریف می‌شود. در نتیجه همیلتون آن به صورت رابطه ۵-۱۸ است.

$$H = K_1 u^T u - K_2 z^T T_q^{-T} u - a^T T_p z + \frac{a^T q_f}{t_f} + \lambda^T (-\bar{A} z + \bar{B} u) \quad (18-5)$$

که شرایط لازم برای سرعت بهینه  $z$  و کنترل ورودی  $u$  عبارتند از:

$$\frac{\partial H}{\partial u} = 2K_1 u - K_2 T_q^{-1} z + \bar{B}^T \lambda = 0 \quad (19-5)$$

$$\frac{\partial H}{\partial z} = -K_2 T_q^{-T} u - T_p^T a - \bar{A}^T \lambda = -\dot{\lambda} \quad (20-5)$$

$$\frac{\partial H}{\partial \lambda} = -\bar{A} z + \bar{B} u = \dot{z} \quad (21-5)$$

که از معادلات ۵-۱۹ تا ۵-۲۱ معادله دیفرانسیل ۵-۲۲ را می‌توان نوشت.

$$\dot{z} - \left( \bar{B} \bar{B}^T \bar{A} \bar{B}^{-T} \bar{B}^{-1} \bar{A} - \frac{k_2}{k_1} \bar{B} \bar{B}^T T_q^{-T} \bar{B}^{-1} \bar{A} \right) z + \frac{1}{2k_1} \bar{B} \bar{B}^T T_p^T a = 0 \quad (22-5)$$

چون  $\bar{B} \bar{B}^T$  و  $\bar{A}$  ماتریس‌های قطری هستند، معادله ۵-۲۲ به فرم معادله دیفرانسیل مرتبه دو کاهش می‌یابد.

$$\dot{z} - Q Q^T z + R^T T_p^T a = 0 \quad (23-5)$$

$$Q Q^T = \bar{A}^T \bar{A} - \frac{k_2}{k_1} \bar{B} \bar{B}^T T_q^{-T} \bar{B}^{-1} \bar{A}, \quad Q = \begin{bmatrix} \frac{1}{\tau_v} & 0 \\ 0 & \frac{1}{\tau_w} \end{bmatrix}, \quad R = \frac{\bar{B}^T \bar{B}}{2k_1} \quad (24-5)$$

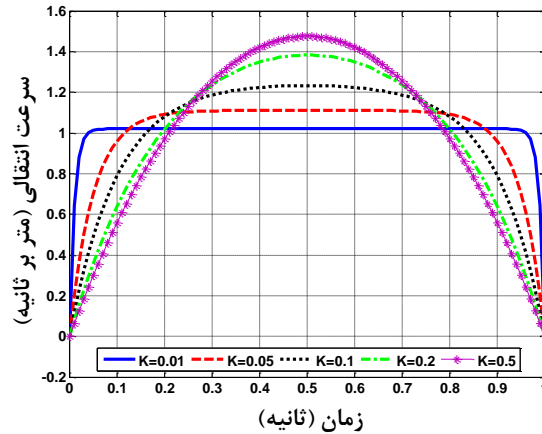
$$\tau_v = (J_1 + J_2) / \sqrt{F_v (F_v + k_t k_b n^2 / R_a)} \quad (25-5)$$

$$\tau_w = (J_1 - J_2) / \sqrt{F_v (F_v + k_t k_b n^2 / R_a)} \quad (26-5)$$

که  $\tau_v$  ثابت زمانی مکانیکی برای حرکت انتقالی و  $\tau_w$  ثابت زمانی برای حرکت چرخشی ربات سیار است. برای بررسی مشخصه منحنی سرعت در مدت زمان  $t_f$ ، کمینه انرژی منحنی سرعت برای حرکت انتقالی



به ازای ثابت زمانی‌های متنوع در شکل ۲-۵ نشان داده شده است. برای  $\gamma = \tau_v/t_f$  متنوع منحنی سرعت به دست آمده است.



شکل ۲-۵ منحنی سرعت برای  $\gamma = \tau_v/t_f$  متنوع

شکل ۲-۵ نشان می‌دهد که برای  $\gamma$  های نزدیک به صفر منحنی سرعت به شکل دوزنقه‌ای می‌باشد. در حالی که برای مقادیر  $\gamma > 0.2$  منحنی سرعت به سمت سهموی همگرا می‌شود.

## ۵-۵- نتایج شبیه‌سازی

در ادامه تعدادی مسئله برای ارزیابی انرژی ذخیره شده با در نظر گرفتن تابع هدف معادله ۵-۱۲ شبیه‌سازی شده است. مسئله در دو بخش شبیه‌سازی شده است. که شامل شبیه‌سازی روی مسیر مستقیم و شبیه‌سازی بر روی مسیر منحنی‌الخط ربات می‌باشد.

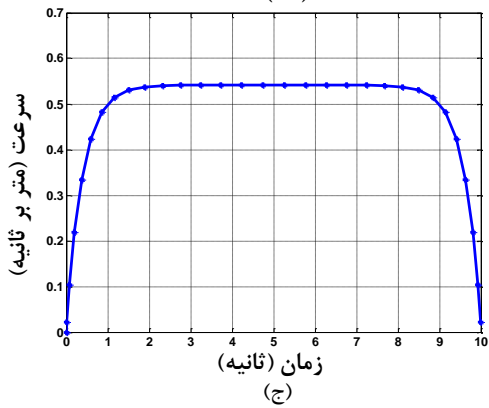
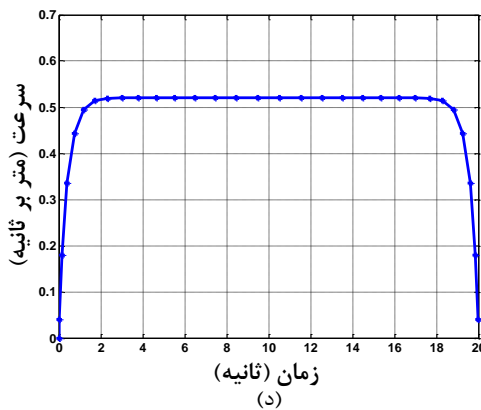
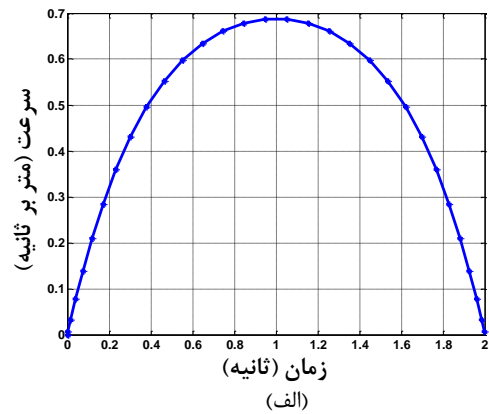
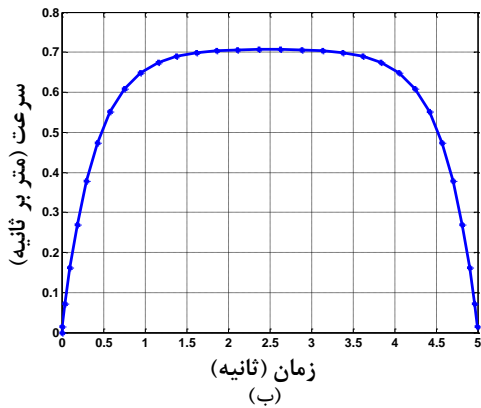
### ۵-۵-۱- شبیه‌سازی بر روی مسیر مستقیم

در این قسمت حرکت انتقالی ربات سیار در امتداد خط مستقیم که منجر به حداقل کردن انرژی مصرفی متناسب با منحنی سرعت می‌باشد؛ شبیه‌سازی شده است.

شبیه‌سازی یک: در جدول ۵-۱ نرخ انرژی مصرف شده برای هرکدام از اجزای معادله ۵-۱۰ با توجه به کل انرژی گرفته شده از باتری نشان داده شده است. با توجه به اینکه سرعت اولیه و نهایی صفر است ترم  $E_K$  در انرژی مصرفی تأثیری ندارد. در شکل ۳-۵ منحنی‌های مختلف سرعت به ازای مقادیر جدول ۵-۱ نشان داده شده است.

جدول ۱-۵ انرژی مصرفی ربات برای  $t_f$  و  $x_f$  مختلف

$t_f(s)$	$x_f(m)$	$E_w(J)$	$E_F(J)$	$E_R(J)$	$E_K(J)$
۲	۱	۷.۲۶۷	۴.۹۵۷	۲.۳۱۰	۰.۰
۵	۳	۱۸.۴۹۱	۱۶.۷۱۶	۱.۷۷۵	۰.۰
۱۰	۵	۲۴.۱۷۳	۲۲.۴۳۴	۱.۷۳۹	۰.۰
۲۰	۱۰	۴۶.۰۳۳	۴۴.۰۴۵	۱.۹۵۸	۰.۰
۳۰	۱۵	۶۸.۳۱۴	۶۵.۶۵۲	۲.۶۶۱	۰.۰

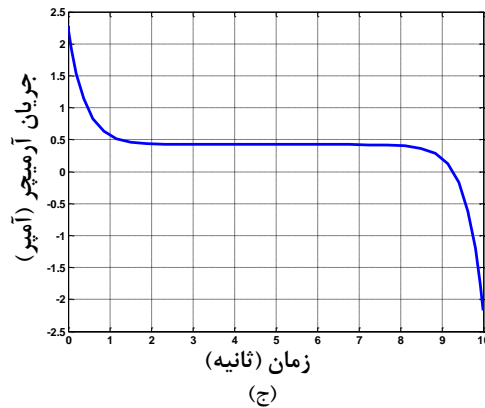
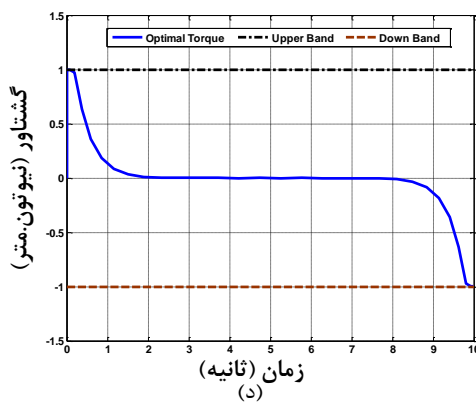
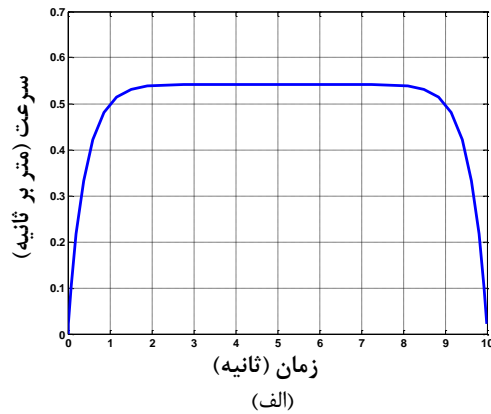
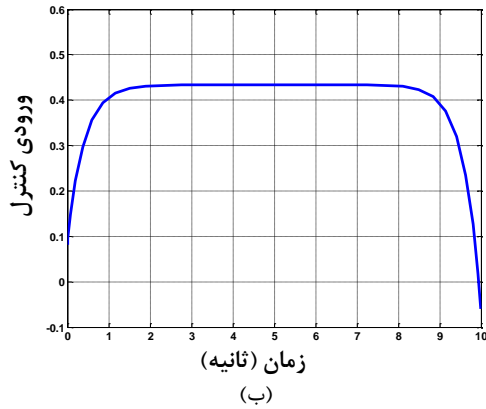


شکل ۳-۵ الف) منحنی سرعت برای  $t_f = 2s$  و  $x_f = 1m$  ب) منحنی سرعت برای  $t_f = 5s$  و  $x_f = 3m$

ج) منحنی سرعت برای  $t_f = 10s$  و  $x_f = 5m$  د) منحنی سرعت برای  $t_f = 20s$  و  $x_f = 10m$

شکل ۴-۵ نتایج شبیه‌سازی برای ربات با مختصات اولیه  $(0,0)$  و مختصات نقطه هدف  $(5,0)$  در مدت

زمان ۱۰ ثانیه را نشان می‌دهد.



شکل ۴-۵ شبیه‌سازی برای  $x_f = 5m$  و  $t_f = 10s$ . الف) منحنی سرعت ب) ورودی‌های کنترلی ج) تغییرات جریان آرمیچر د) گشتاور بهینه وارد بر چرخ راست و چپ  $\tau_r = \tau_l$

همانطور که در شکل ۴-۵ دیده می‌شود، گشتاورهای وارد بر چرخ‌های ربات هموار بوده و کمتر از مقادیر اکستریم موتورهای محرک چرخ‌های ربات است. لذا در طی مسیر بهینه موتورهای ربات به اشباع نرسیده است.

### ۵-۵-۲- شبیه‌سازی حرکت منحنی الخط ربات

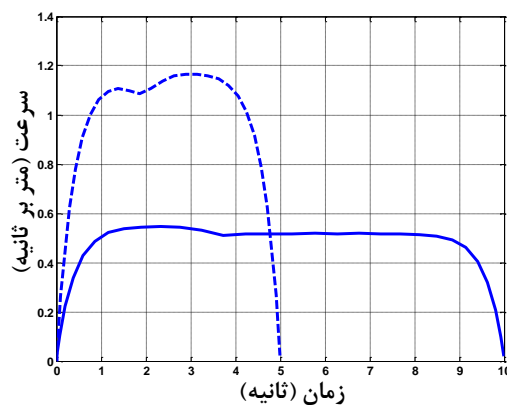
در این قسمت حرکت ربات سیار روی مسیر منحنی الخط که منجر به حداقل کردن انرژی مصرفی متناسب با منحنی سرعت می‌باشد؛ شبیه‌سازی شده است.

شبیه‌سازی یک: در شکل ۵-۵ مسیر حرکت ربات با سه مانع نشان داده شده است. جدول ۵-۲ انرژی مصرف‌شده ربات برای حرکت از نقطه مبدا  $(0,0)$  و رسیدن به نقطه هدف  $(3,3)$ ، با اضافه کردن قیود مربوط به موانع و به ازای زمان‌های مختلف  $t_f$  نشان می‌دهد. شکل ۶-۵ منحنی سرعت برای زمان‌های

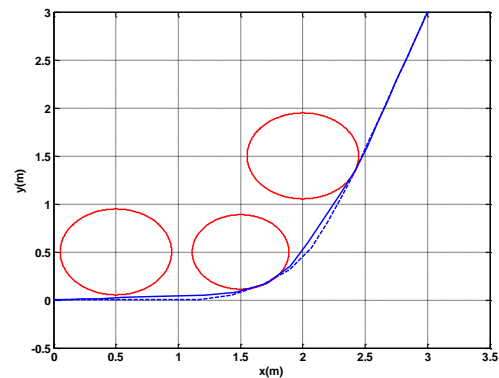
مختلف با سه مانع را نشان می‌دهد. مشاهده می‌شود هر چه ربات در مدت زمان کمتری بخواند به نقطه هدف برسد مطابق شکل ۵-۶ سرعت ربات افزایش یافته که متناظر با این افزایش سرعت انرژی مصرفی (جدول ۵-۲) نیز افزایش می‌یابد.

جدول ۵-۲ انرژی مصرفی ربات برای  $t_f$  مختلف با سه مانع

$t_f(s)$	$E_R(J)$	$E_F(J)$	$E_W(J)$
۵	۱.۸	۴۴.۱	۴۶.۱
۱۰	۳.۵	۲۱.۲	۲۴.۸



شکل ۵-۶ منحنی سرعت برای زمان‌های مختلف

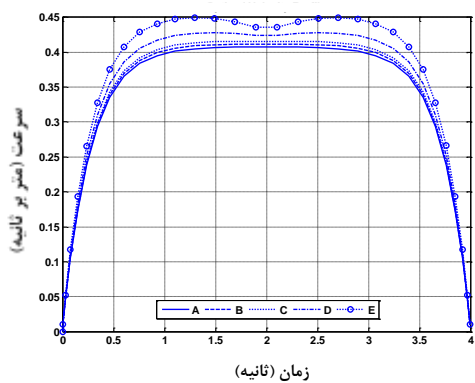


شکل ۵-۵ مسیر حرکت ربات با سه مانع

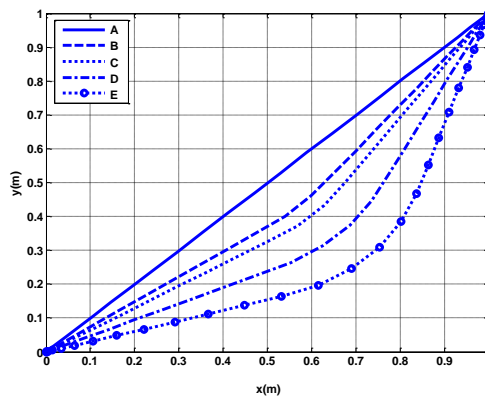
**شبیه‌سازی دو:** در ادامه به بررسی انرژی مصرف شده ربات در مدت زمان ۴ ثانیه برای مسیرهای مختلف مطابق شکل ۵-۷ که از نقطه مبدا (۰,۰) شروع و به نقطه (۱,۱) می‌رود پرداخته می‌شود. در جدول ۵-۳ برای پنج مسیر، مقادیر انرژی تلف شده ربات و طول مسیر پیموده شده با هم مقایسه شده است.

جدول ۵-۳ انرژی مصرفی ربات برای مسیرهای مختلف

مسیر	$L(m)$	$E_W(J)$	$E_R(J)$	$E_F(J)$
A	۱.۴۱۴	۱۹.۱	۱.۴۲	۱۷.۷۱
B	۱.۴۲۸	۱۹.۴۹	۱.۴۲	۱۸.۰۷
C	۱.۴۴۲	۱۹.۸۶	۱.۴۲	۱۸.۴۴
D	۱.۴۸۷	۲۱	۱.۴۲	۱۹.۵۹
E	۱.۵۶۱	۲۳	۱.۴۲	۲۱.۵۸

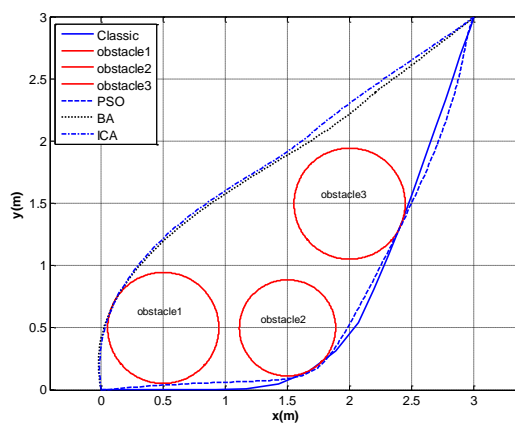


شکل ۸-۵ منحنی سرعت برای مسیرهای متنوع



شکل ۷-۵ حرکت ربات در مسیرهای متنوع

شبیه‌سازی سه: در شکل ۹-۵ به مقایسه مسیرهای مختلف با دیگر الگوریتم‌ها پرداخته شده است. مسیر دارای سه مانع است که از نقطه مبدا  $(0,0)$  شروع و به نقطه  $(3,3)$  می‌رود. در اینجا هدف کمینه کردن طول مسیر می‌باشد. مطابق جدول ۳-۵ مشاهده می‌شود که الگوریتم‌های ICA و BA دارای طول مسیر کمتری نسبت به الگوریتم‌های PSO و کنترل بهینه است.



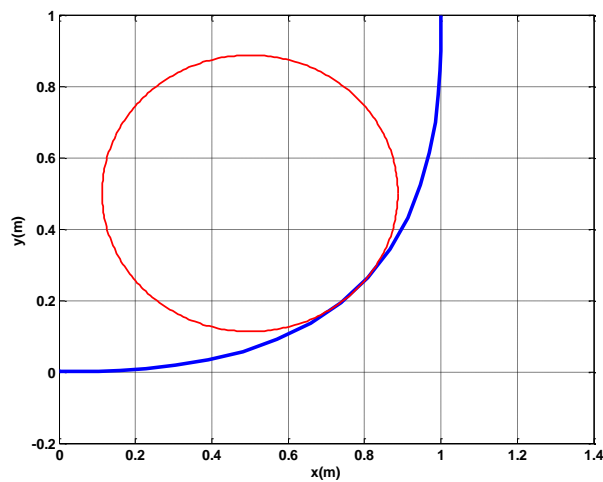
شکل ۹-۵ مسیر حرکت ربات با الگوریتم‌های متنوع

جدول ۳-۵ طول مسیر حرکت با الگوریتم‌های متنوع

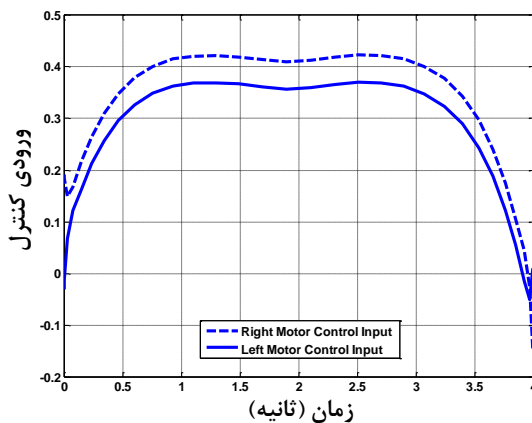
نوع الگوریتم	PSO	ICA	BA	کنترل بهینه
طول مسیر	۴.۸۷۸	۴.۴۵۱	۴.۴۵۰	۴.۸۸۷

شبیه‌سازی چهار: در اینجا به طراحی مسیر با در نظر گرفتن قيود مربوط به موانع و محدودیت‌هایی که شامل ترمزهای ولتاژ و گشتاور وارد بر هر چرخ می‌باشد، با هدف کمینه کردن انرژی و طول مسیر

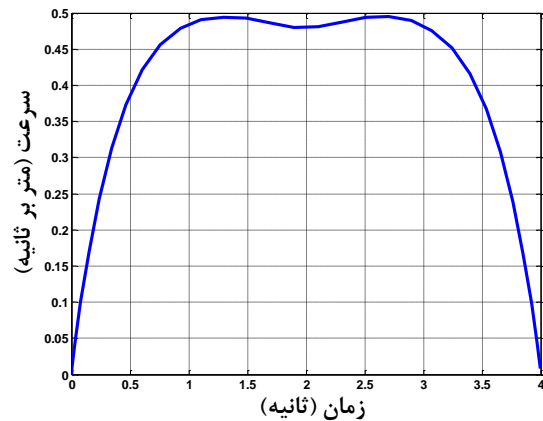
پرداخته می‌شود. مطابق شکل ۵-۱۰ مسیر دارای یک مانع است که از نقطه مبدا  $(0,0)$  شروع و به نقطه  $(1,1)$  می‌رود. در شکل ۵-۱۱ منحنی سرعت برای پیمودن مسیر نشان داده شده است. شکل ۵-۱۲ ورودی‌های کنترل چرخ‌های راست و چپ ربات است. شکل ۵-۱۳ مقدار جریان مصرفی موتورهای راست و چپ ربات را نشان می‌دهد و شکل ۵-۱۴ مقدار گشتاور بهینه وارد بر موتور راست و چپ را نشان می‌دهد. انرژی مصرفی باتری در این حالت برابر  $58.2$  ژول و طول مسیر پیموده شده در مدت زمان  $4$  ثانیه برابر  $1.65$  متر است.



شکل ۵-۱۰ مسیر حرکت بات با یک مانع

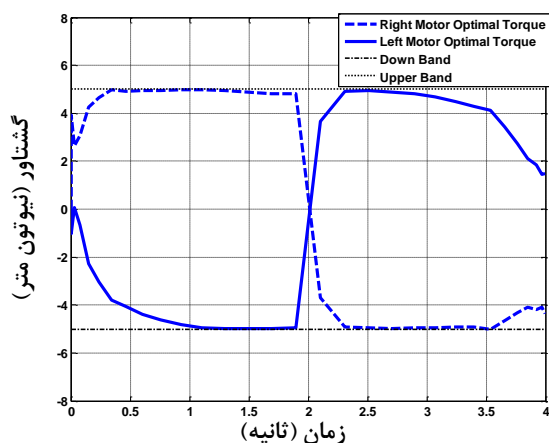


شکل ۵-۱۲ ورودی‌های کنترلی موتور راست و چپ

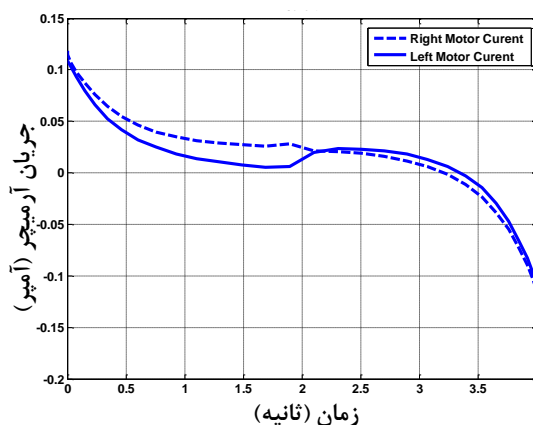


شکل ۵-۱۱ منحنی سرعت

شبیه‌سازی پنچ: در این قسمت نیز به طراحی مسیر با در نظر گرفتن قیود مربوط به موانع و محدودیت‌هایی که شامل ترم‌های ولتاژ و گشتاور وارد بر هر چرخ می‌باشد، با هدف کمینه کردن انرژی

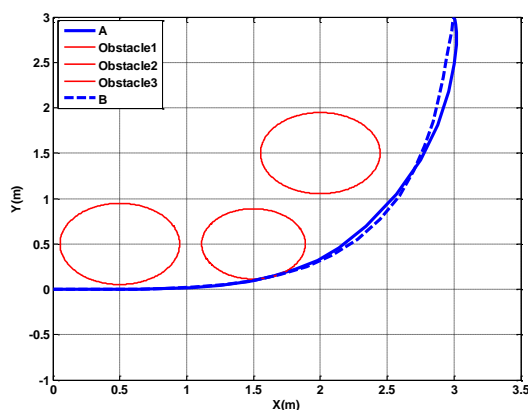


شکل ۵-۱۴ گشتاور بهینه وارد بر چرخ راست و چپ

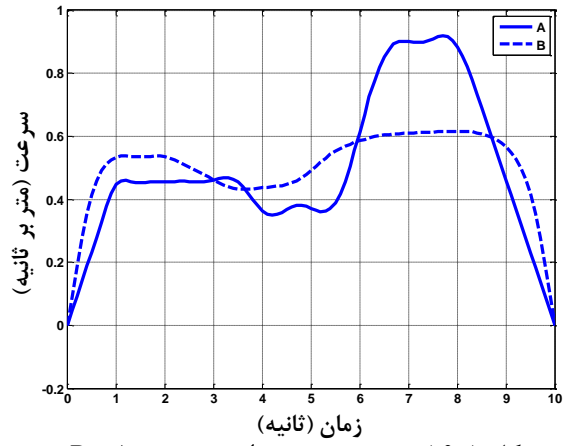


شکل ۵-۱۳ جریان آرمیچر چرخ‌های راست و چپ

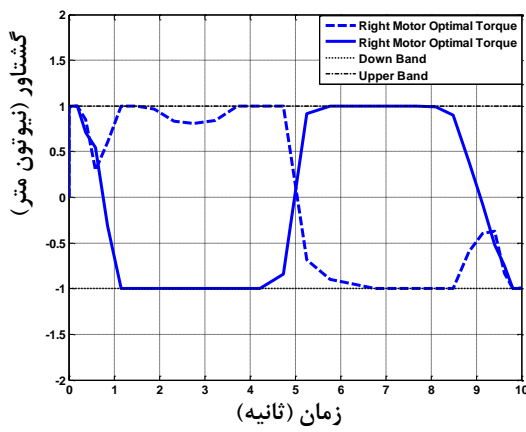
و طول مسیر پرداخته می‌شود. مطابق شکل ۵-۱۵ مسیر دارای سه مانع است که از نقطه مبدا  $(0,0)$  شروع و به نقطه  $(3,3)$  می‌رود. دو مسیر A و B با محدودیت‌های مختلف بر روی گشتاور وارد بر چرخ‌ها با هم مقایسه شده‌اند. منحنی‌های سرعت متناظر با هر کدام از مسیرها در شکل ۵-۱۶ نشان داده شده است. به طوری که مسیر A با در نظر گرفتن محدودیت گشتاور بین ۱ و -۱ دارای انرژی مصرفی بیشتری، با مقدار ۵۸.۲ ژول و طول مسیر ۵.۱۶ متر است. برای مسیر B با محدودیت گشتاور بین ۰.۵ و -۰.۵ انرژی مصرفی ۳۷.۸ ژول و طول مسیر ۵.۰۰ متر به دست آمده است. مشاهده می‌شود که با وارد کردن محدودیت بر گشتاور چرخ‌ها می‌توان به سطح‌های انرژی مختلف رسید.



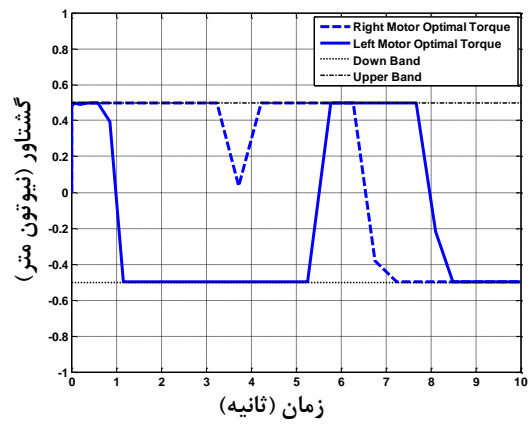
شکل ۵-۱۵ مسیر حرکت ربات با سه مانع



شکل ۱۶-۵ منحنی سرعت برای دو مسیر A و B



شکل ۱۸-۵ گشتاور بهینه وارد بر چرخ راست و چپ برای مسیر B



شکل ۱۷-۵ گشتاور بهینه وارد بر چرخ راست و چپ برای مسیر A



## فصل ششم

### پیاده‌سازی‌های عملی

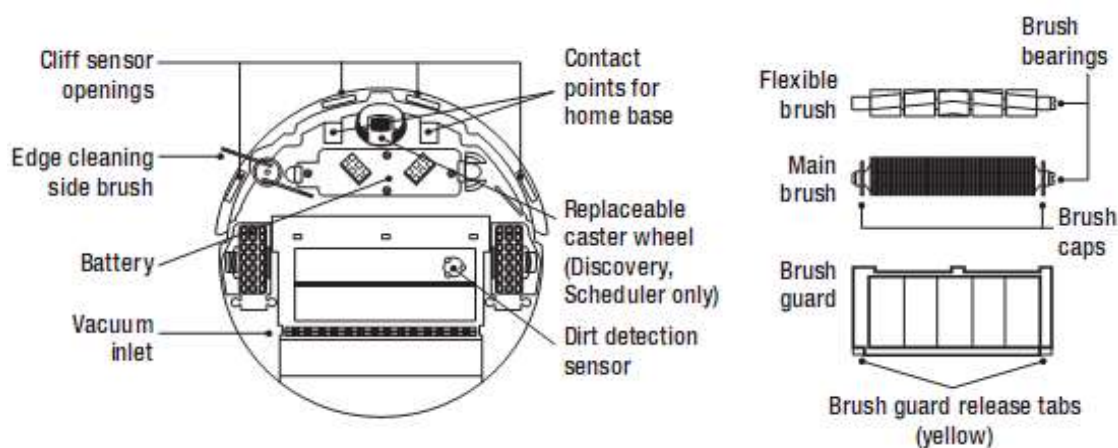
## ۶-۱- مقدمه

در این فصل به پیاده‌سازی‌های عملی بر روی ربات سیار رومبا از شرکت آی‌روبات<sup>۱</sup> است؛ پرداخته می‌شود. ابتدا مشخصه‌های ربات رومبا را معرفی کرده و سپس بر روی پیاده‌سازی‌های مربوطه بحث می‌شود.

## ۶-۲- اجزای داخلی و خارجی ربات رومبا

تقریباً تمامی حسگرها جلو قرار دارند. انواع حسگرهای جلو عبارت‌اند از:

- ۱- شناسایی دست‌انداز یا صخره
- ۲- شناسایی دیوار
- ۳- شناسایی تماس‌ها



شکل ۶-۱ نمایش دیاگرام داخلی رومبا [۳۶]

در واقع بیشتر حسگرها بر روی سپر یا ضربه‌گیر جلوی متحرک نصب شده است. این ضربه‌گیر متحرک باعث می‌شود که ربات تماس را اندازه‌گیری کند، شوک را جذب نموده و آسیب را به حداقل برساند. سیستم عامل دستگاه رومبا همیشه برای حرکت رو به جلو طراحی شده است. موتور وسط، درایو اصلی موتور است. موتور مکش و قلم تمیزکننده و باتری‌ها در وسط دستگاه هستند. رومبا در مرکز ثقلی نزدیک به مرکز جرم قرار دارد که باعث حرکت پایدار آن می‌شود.

<sup>1</sup> - Irobot

## ۶-۳- شبیه‌ساز رومبا

شبیه‌ساز رومبا برای اهداف آموزشی و پژوهشی مورد استفاده قرار می‌گیرد، برای این منظور به راحتی قابل برنامه‌ریزی و قابل تغییر است که شامل حسگرهای داخلی رومبا و در صورت لزوم می‌توان از حسگر خارجی نیز استفاده کرد [۳۷]. شبیه‌ساز توسط کاربر قابل کنترل است و از طریق پورت سریال با رومبا ارتباط برقرار می‌کند.

### ۶-۳-۱- هدف از شبیه‌ساز

هدف از استفاده شبیه‌ساز این است که به کاربر اجازه دهد تا کدهایی که به‌طور معمول در محیط واقعی اجرا می‌کند در شبیه‌ساز نیز اجرا کند. کاربر با نوشتن یک تابع متلب، می‌تواند با استفاده از جعبه‌ابزار متلب که برای ربات نوشته شده است برنامه کنترلی خود را در محیط شبیه‌ساز آزمایش کند. برنامه کنترلی را می‌توان در محیط شبیه‌ساز اجرا کرد و رفتار مشابه رومبا را در محیط واقعی توسط شبیه‌ساز مشاهده کرد.

### ۶-۳-۲- فرضیات شبیه‌ساز

در شبیه‌ساز فرض می‌شود که اصطکاک فقط در دیوارها وجود دارد نه در چرخ‌ها؛ بنابراین اگر ربات با هیچ دیواری در تماس نباشد و بخواهد به سمت جلو با سرعت معینی حرکت کند با همان سرعت حرکت می‌کند. در شبیه‌ساز فرض می‌شود که چرخ‌ها گیر نمی‌کند، آن‌ها هرگز لغزش ندارند به جز زمانی که به دیوارها برخورد می‌کنند.

حسگر ضربه‌گیر و سیستم تعیین موقعیت بدون داشتن نویز مدل شده‌اند. حسگر ضربه‌گیر ممکن است در یک برخورد زودگذر و اجمالی با یک دیوار فعال شود و ممکن است که فعال نیز نشود؛ همچنین ممکن است هر دو حسگر ضربه‌گیر یا یکی از آن‌ها در برخورد با دیوار بسته به اینکه چگونه سپر ضربه‌گیر تحت فشار قرار گیرد؛ فعال شود.

## ۴-۶- پیاده‌سازی

در این قسمت به‌وسیله نوشتن برنامه‌های کنترلی با استفاده از توابع موجود در پیوست ب به اجرای آن در محیط شبیه‌ساز و محیط واقعی پرداخته می‌شود.

### ۴-۶-۱- محدودیت‌های عملی ربات رومبا

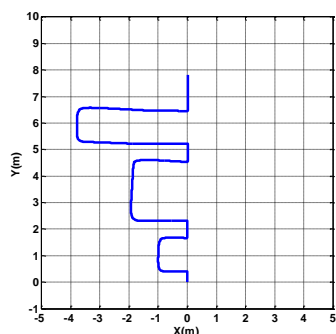
حسگرهای زاویه و مسافت رومبا در محیط واقعی دارای دقت مناسبی نیستند به طوری‌که برای ایجاد چرخش یا مسافت‌های خاص در رومبا، بایستی از راهکارهایی که در پیوست ب توضیح داده شده است استفاده شود.

### ۴-۶-۲- محیط ناشناخته

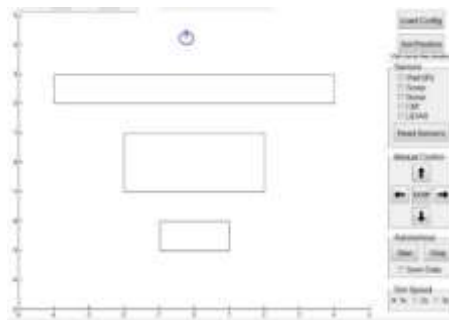
در گام اول محیط به‌صورت محلی یا ناشناخته فرض می‌شود و با استفاده از فیدبک گرفتن از حسگرهای ربات جاروبرقی رومبا، که این حسگرها در پیوست ب توضیح داده شده است، ربات از نقطه مشخص شروع به حرکت می‌کند و به نقطه هدف می‌رسد. حسگرهایی که برای تشخیص موانع و دور زدن آن‌ها استفاده شده شامل حسگر ضربه‌گیر رومبا که در صورت برخورد با مانع مقدار آن یک شده و تصمیم موردنظر کاربر را اجرا می‌کند و دیگری حسگر دیوار رومبا است که شامل یک حسگر مادون‌قرمز با برد خیلی کم است و در سمت راست حسگر ضربه‌گیر وجود دارد. این حسگر مقدار منطقی صفر یا یک را دارا است. اگر در مجاورت دیوار (مانع) باشد یک در غیر این صورت مقدار صفر را برمی‌گرداند. بعد از برخورد رومبا با مانع با دنبال کردن مانع و با کمک گرفتن از حسگرهای مسافت و زاویه اقدام به دور زدن مانع می‌کند.

### ۴-۶-۲-۱- محیط شبیه‌ساز

این قسمت در محیط شبیه‌ساز پیاده‌سازی شده است. در اینجا با در نظر گرفتن سه مانع به‌صورت مستطیل (شکل ۶-۲) ربات از مبدا نقطه  $(0,0)$  شروع به حرکت کرده و به سمت هدف که در مختصات  $(8,0)$  است؛ می‌رود. شکل ۶-۵ مراحل حرکت رومبا با فیدبک گرفتن از حسگرها را نشان می‌دهد.



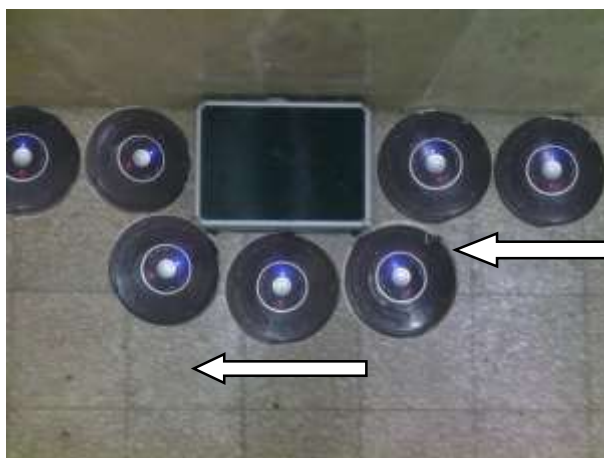
شکل ۳-۶ مسیر حرکت ربات در محیط شبیه‌ساز



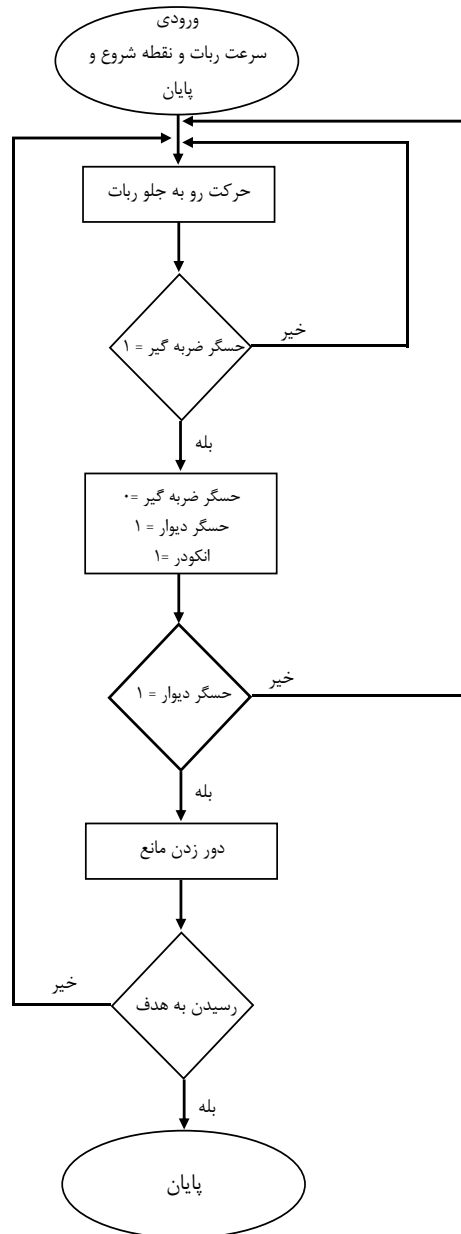
شکل ۲-۶ حرکت ربات در محیط شبیه‌ساز با سه مانع

#### ۲-۲-۴-۶- پیاده‌سازی در محیط واقعی

حال همین عمل در محیط واقعی انجام می‌شود، در محیط شبیه‌ساز به دلیل اینکه محیط ایده آل در نظر گرفته شده است (از لیز خوردن چرخ‌ها صرف‌نظر شده، نوع سطح پیمایش موردنظر نیز لحاظ نشده است) ربات به راحتی به نقطه هدف می‌رسد. زمانی که برنامه کنترلی برای ربات در محیط واقعی اجرا شد، به دلیل اینکه حسگرهای مسافت و زاویه ربات از انکودرهای موجود در چرخ رومبا استفاده کرده و در محیط واقعی نیز چرخ‌ها لیز خورده و سطح‌های مختلف وجود دارد؛ در بعضی مواقع اطلاعات غلط از طریق این حسگرها به رومبا رسیده و در نتیجه عمل رسیدن رومبا از نقطه مبدا به نقطه هدف به درستی انجام نمی‌شود (شکل ۴-۶).



شکل ۴-۶ حرکت ربات در محیط واقعی با فیدبک گرفتن از حسگرها



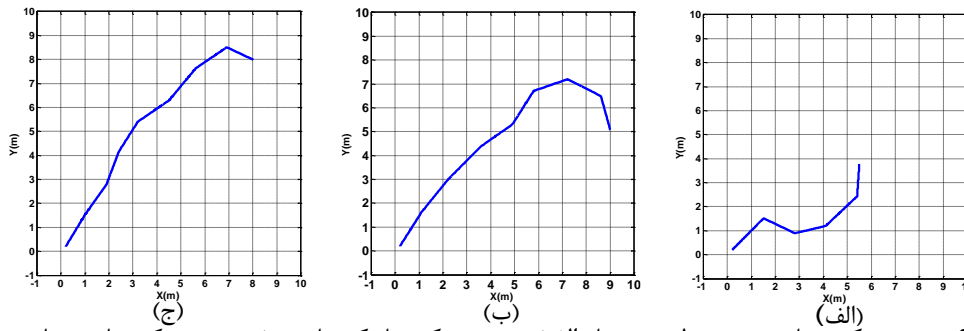
شکل ۵-۶ مراحل حرکت ربات رومبا با فیدبک گرفتن از حسگرها در محیط واقعی

### ۶-۴-۳- محیط شناخته شده

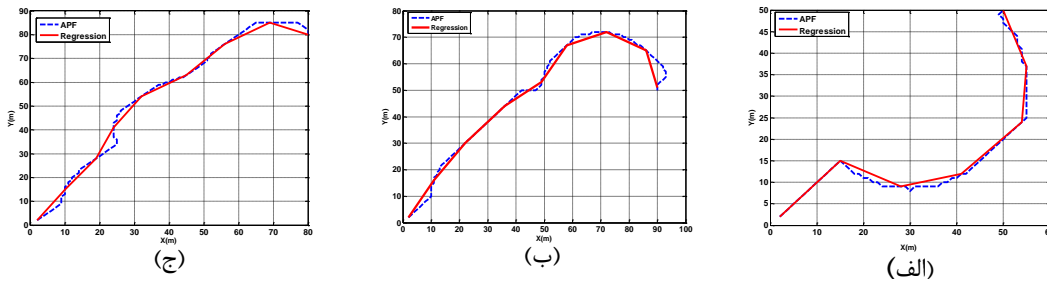
در گام دوم محیط شناخته شده در نظر گرفته می شود. در این حالت نقشه محیط از قبل به دست آمده است. در اینجا از الگوریتم میدان پتانسیل بهبود داده شده که در فصل سوم مطرح شد استفاده شده است. از شکل ۳-۵ ب فصل سوم (الگوریتم میدان پتانسیل) برای اجرای الگوریتم در محیط شبیه ساز مطابق شکل ۶-۶ استفاده شده است.

رگرسون خطی برای ایجاد مسیری بهینه بر روی ربات انجام شده تا ربات در محیط واقعی بتواند مسیر

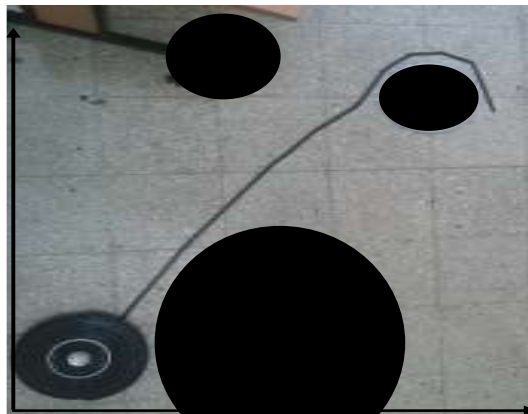
را به راحتی طی کند ( شکل ۶-۷).



شکل ۶-۶ مسیر حرکت ربات در محیط شبیه‌ساز الف) مسیر حرکت با یک مانع ب) مسیر حرکت با سه مانع ج) مسیر حرکت با چهار مانع



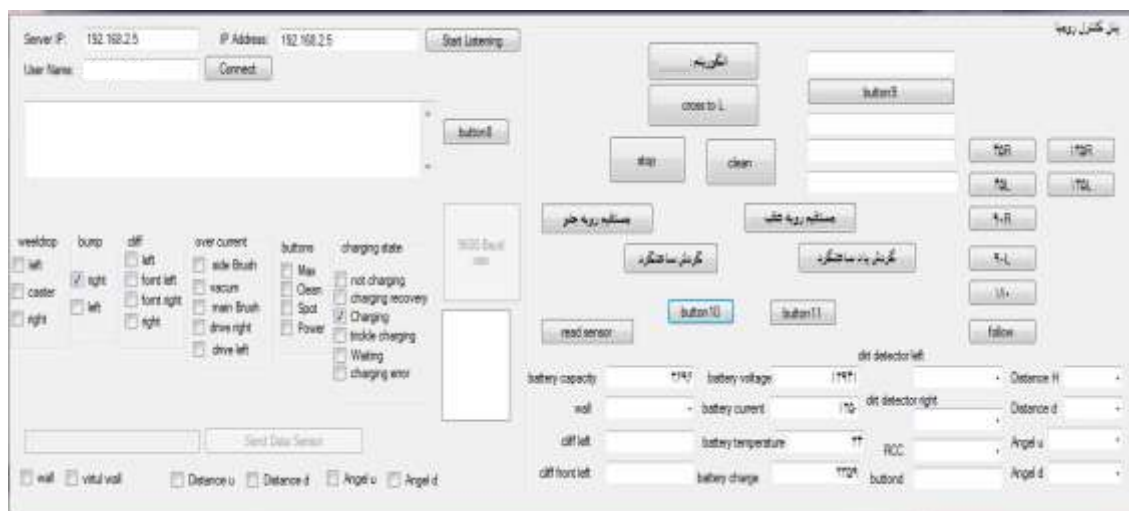
شکل ۶-۷ رگرسیون خطی الف) مسیر حرکت با یک مانع ب) مسیر حرکت با سه مانع ج) مسیر حرکت با چهار مانع  
 شکل ۶-۸ مسیر حرکت ربات رومبا در محیط واقعی را نشان می‌دهد. این مسیر شامل سه مانع (الگوریتم میدان پتانسیل (شکل ۳-۵ ب)) می‌باشد. ربات از نقطه مبدا  $(0,0)$  حرکت کرده و به نقطه هدف در مختصات  $(80,50)$  می‌رسد. البته به دلیل لیز خوردن چرخ‌های ربات در بعضی از موارد ربات به درستی به نقطه هدف نمی‌رسد.



شکل ۶-۸ مسیر حرکت ربات در محیط واقعی

## ۶-۵- محاسبات انرژی مصرفی ربات

در این قسمت با خواندن اطلاعات حسگر ربات رومبا که شامل ۲۶ بایت می‌باشد آزمایش‌های عملی را بر روی رومبا پیاده‌سازی کرده و به مقایسه با نتایج حاصل از شبیه‌سازی پرداخته می‌شود. لازم به ذکر است که این آزمایش‌ها با خواندن اطلاعات حسگرهای مربوط به جریان کشیده شده از باتری و مقدار ولتاژ باتری با استفاده از پنل کنترلی رومبا (شکل ۶-۹) که در محیط سی‌شارپ (C#) طراحی شده است، انجام شده است.



شکل ۶-۹ پنل کنترلی رومبا در محیط سی شارپ

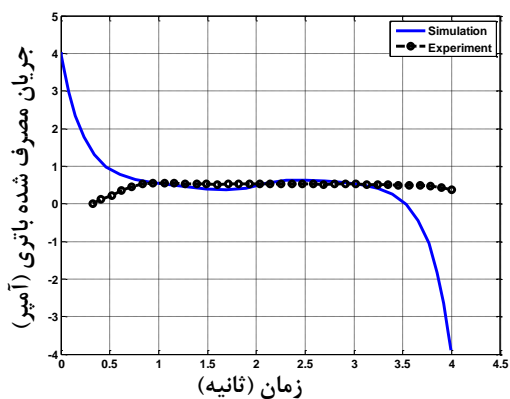
## ۶-۵-۱- محاسبه انرژی مصرفی ربات بر روی سطوح صاف و شیب‌دار

در اینجا با اعمال منحنی سرعت (شکل ۶-۱۰) به ربات و حرکت ربات بر روی مسیر مستقیم با  $x_f = 2m$  و  $t_f = 4sec$ ، به بررسی انرژی مصرف‌شده توسط رومبا پرداخته شده است.

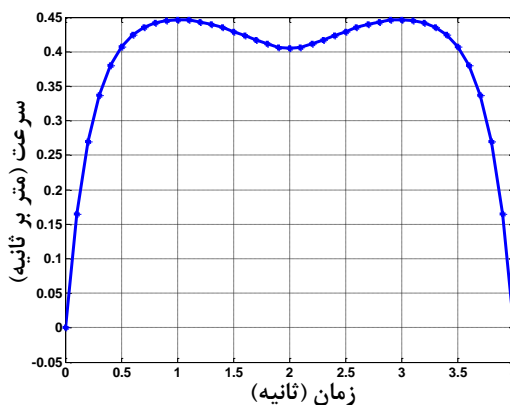
جدول ۶-۱ پارامترهای ربات سیار

پارامتر	مقدار	پارامتر	مقدار
$R_a$	$0.07k\Omega$	$K_t$	$0.0225Nm/A$
$K_b$	$0.0214V/(rad/s)$	$n$	25
$V_s$	13V	$u_{max}$	1
$F_v$	$0.02Nm/(rad/s)$	$r$	0.03m
$b$	0.129m	$J = \begin{bmatrix} J_1 & J_2 \\ J_2 & J_1 \end{bmatrix}$	$\begin{bmatrix} 0.1935 & 0.00135 \\ 0.00135 & 0.1935 \end{bmatrix}$

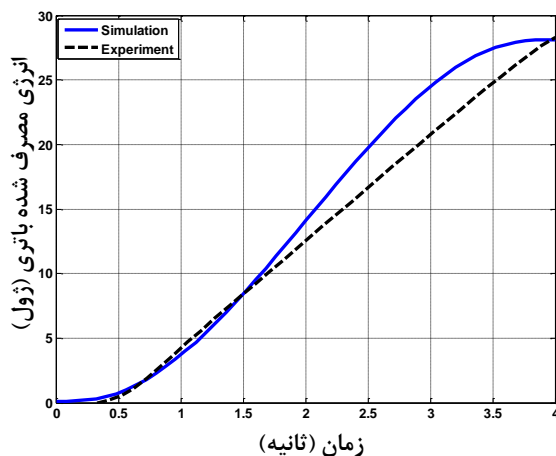




شکل ۱۱-۶ جریان مصرفی باتری در دو حالت شبیه‌سازی و تجربی



شکل ۱۰-۶ پروفیل سرعت اعمالی به ربات



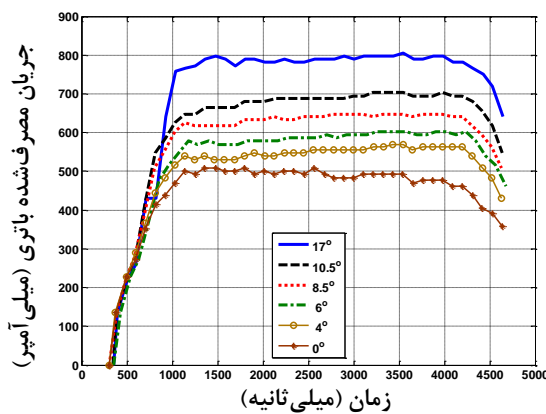
شکل ۱۲-۶ انرژی مصرف شده باتری در دو حالت شبیه‌سازی و تجربی

شکل ۱۰-۶ منحنی سرعت در طول مسیر حرکت ربات رومبا را نشان می‌دهد. بایستی توجه داشت که منحنی سرعت ربات به عنوان ورودی و برای طی کردن مسیر ربات در نظر گرفته شده است. لذا، برای انجام تست تجربی با استفاده از ربات رومبا، بایستی سرعت‌های بهینه به دست آمده از شبیه‌سازی را به عنوان ورودی‌های بخش کنترلی ربات به آن اعمال کرد. در مرحله بعد، بردار سرعت به برد کنترلی ربات اعمال شده و ربات شروع به حرکت می‌کند. در طول مسیر حرکت، با فیدبک گرفتن از حسگرهای جریان و ولتاژ ربات و ارسال این اطلاعات از طریق پورت سریال، مقدار انرژی مصرف شده باتری را می‌توان به دست آورد. همان‌طور که دیده می‌شود، نتایج تجربی با دقت مناسبی شبیه‌سازی را دنبال می‌کند. نزدیکی نتایج شبیه‌سازی و تجربی به علت دقت خوب حسگرهای جریان و ولتاژ رومبا است.

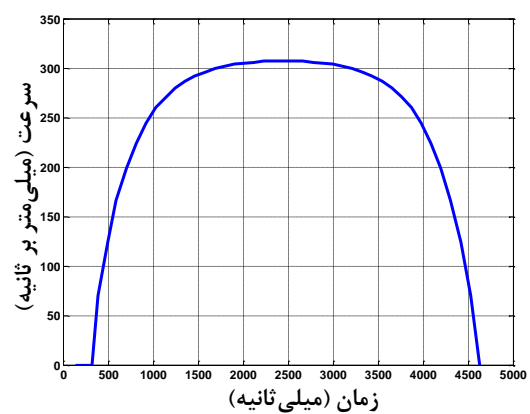
حال ربات را بر روی سطح شیب‌دار قرار داده و به بررسی انرژی مصرف‌شده ربات بر روی سطح شیب‌دار مطابق شکل ۱۳-۶ پرداخته می‌شود. ابتدا با اعمال منحنی سرعت با  $x_f = 1m$  و  $t_f = 4sec$  (شکل ۱۴-۶) به بررسی انرژی مصرف‌شده توسط ربات رومبا بر روی سطوح شیب‌دار مختلف پرداخته شده است. شکل‌های ۱۵-۶ و ۱۶-۶ نشان می‌دهند که با افزایش زاویه سطح شیب‌دار جریان و انرژی مصرف‌شده توسط باتری افزایش می‌یابد.



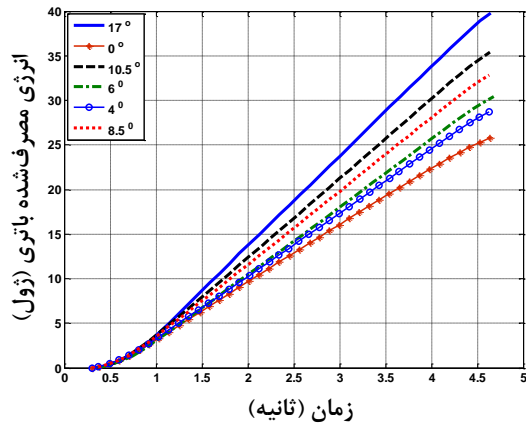
شکل ۱۳-۶ حرکت ربات روی سطح شیب‌دار در محیط واقعی



شکل ۱۵-۶ جریان مصرفی باتری ربات روی سطح شیب‌دار با زوایای متنوع



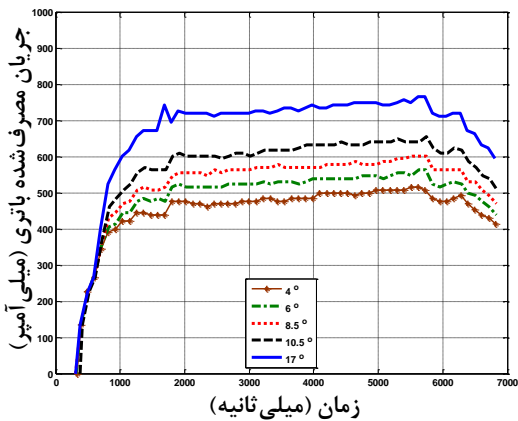
شکل ۱۴-۶ پروفیل سرعت اعمالی به ربات روی سطح شیب‌دار با  $x_f = 1m$  و  $t_f = 4sec$



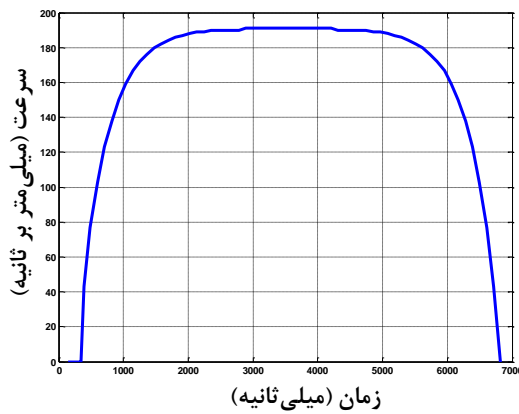
شکل ۱۶-۶ انرژی مصرف شده باتری ربات روی سطح شیبدار با زوایای متنوع

حال با در نظر گرفتن  $t_f = 6sec$  و  $x_f = 1m$  به بررسی انرژی مصرفی ربات بر روی سطح شیبدار پرداخته می‌شود. شکل‌های ۱۸-۶ و ۱۹-۶ نیز نشان می‌دهند که با افزایش زاویه سطح شیبدار جریان و انرژی مصرف شده توسط باتری افزایش می‌یابد.

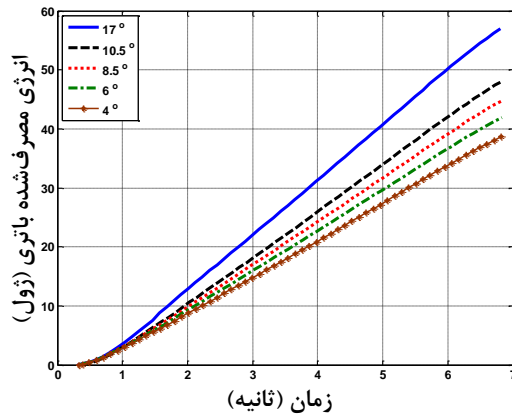
همان‌طور که در شکل ۲۰-۶ مشاهده می‌شود این بار ربات در راستای سطح شیبدار برای حالت  $t_f = 4sec$  و  $x_f = 1m$  با شیب‌های ۱۰ و ۲۰ درجه آزمایش شده است. نتایج به دست آمده از شکل‌های ۲۱-۶ و ۲۲-۶ نشان می‌دهد که در این حالت چون تغییرات انرژی پتانسیل ربات صفر است، تغییر زاویه سطح شیبدار بر انرژی مصرفی ربات تأثیری ندارد.



شکل ۱۸-۶ جریان مصرفی باتری ربات بر روی سطح شیبدار با زوایای متنوع



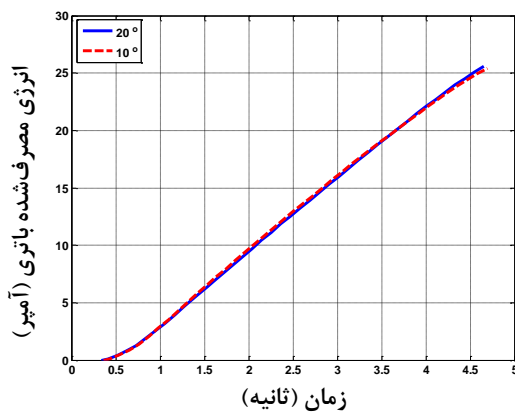
شکل ۱۷-۶ پروفیل سرعت اعمالی به ربات بر روی سطح شیبدار با  $t_f = 6sec$  و  $x_f = 1m$



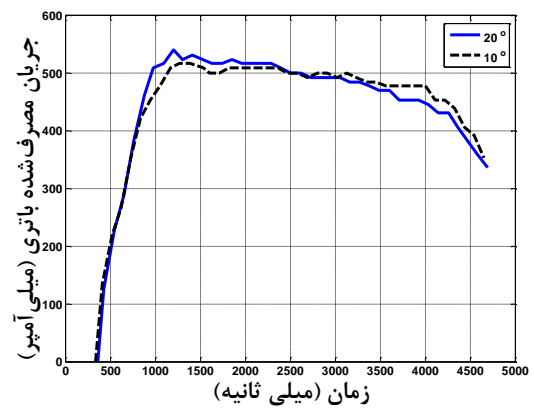
شکل ۶-۱۹ انرژی مصرف شده باتری ربات روی سطح شیب‌دار با زوایای متنوع



شکل ۶-۲۰ حرکت ربات در راستای سطح شیب‌دار در محیط واقعی



شکل ۶-۲۲ انرژی مصرف شده باتری ربات در راستای سطح شیب‌دار با  $x_f = 1m$  و  $t_f = 4sec$



شکل ۶-۲۱ جریان مصرفی باتری ربات در راستای سطح شیب‌دار با  $x_f = 1m$  و  $t_f = 4sec$

نتایج عملی بر روی ربات رومبا نشان می‌دهد که در حرکت مستقیم رو به بالا سطح شیب‌دار حداکثر نیروی جاذبه بر روی موتورها ایجاد می‌شود. حال اگر ربات موازی سطح شیب‌دار حرکت کند هیچ نیروی جاذبه‌ای بر روی موتورها اعمال نمی‌شود.

### ۲-۵-۶ بازده انرژی ربات برای مسیرهای مختلف

در این بخش به بازدهی انرژی حرکت، برای سه مسیر مختلف پرداخته شده است. بازدهی در اینجا به صورت، ناحیه تحت پوشش توسط ربات به انرژی مصرفی توسط آن تعریف می‌شود. در اینجا عرض ناحیه تحت پوشش به اندازه قطر ربات رومبا در نظر گرفته شده است به گونه‌ای که ربات با حرکت در هر کدام از سه مسیر، ناحیه مورد نظر را به طور کامل پیمایش می‌کند.

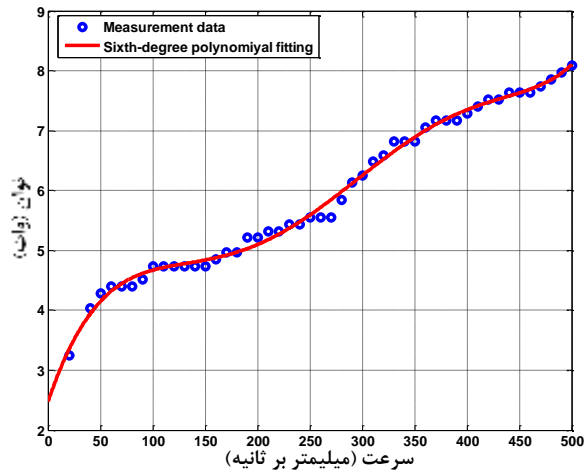
$$\text{بازدهی} = \frac{\text{ناحیه تحت پوشش}}{\text{انرژی مصرفی}} \quad (1-6)$$

مدل توان ربات به صورت تجربی تعریف شده، بنابراین مصرف توان برای ربات به صورت ۲-۶ فرموله می‌شود:

$$\text{power} = \sum_{i=1}^K P \left( v_i(t), \frac{dv_i(t)}{dt} \right) \quad (2-6)$$

که  $K$  تعداد موتورهای ربات و  $v_i(t)$  سرعت موتور  $i$  ام با شعاع  $r$  است. به طور کلی مصرف انرژی موتور با سرعت زاویه‌ای مرتبط است. بنابراین شتاب زاویه‌ای می‌تواند در محاسبات لحاظ نشود.

در اینجا برای محاسبه توان ربات رومبا با استفاده از آزمون تجربی مطابق شکل ۲۳-۶ منحنی توان بر حسب سرعت ربات را به دست آورده و با یک چند جمله‌ای درجه شش تقریب زده شده است به طوری که در محاسبات توان از رابطه ۳-۶ استفاده شده است.



شکل ۶-۲۳ توان بر حسب سرعت

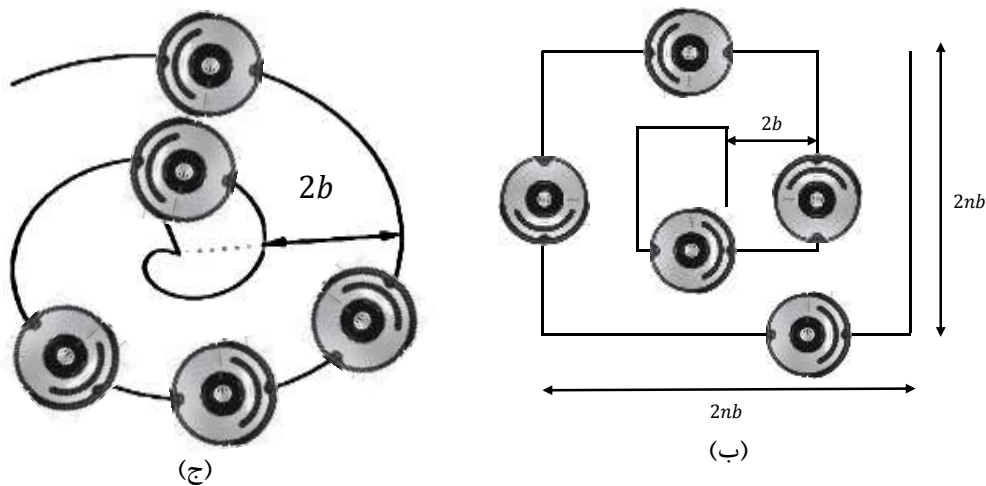
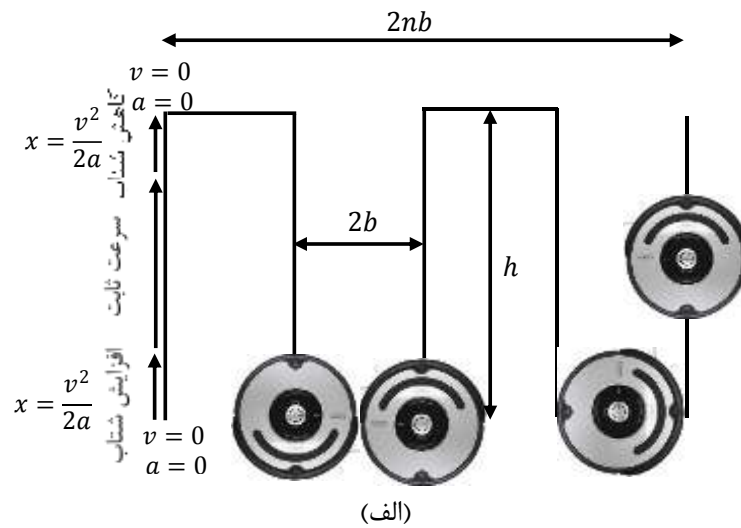
$$power = 2.258e^{-15} * v^6 - 2.561e^{-13} * v^5 - 2.576e^{-9} * v^4 + 1.822e^{-6} * v^3 - 0.0004582 * v^2 + 0.05216 * v + 2.472 \quad (3-6)$$

در این قسمت مسیرهای مختلف برای حرکت ربات بررسی می‌شود. این مسیرها می‌تواند برای یک ربات جاروبرقی که محیط اطراف را تمیز می‌کند به کار گرفته شود. فرض کنید ربات می‌خواهد یک ناحیه مشخص را تمیز کند. ربات این ناحیه را به چند راه مختلف می‌تواند تمیز کند. در این پژوهش سه روش اسکن خطوط<sup>۱</sup> (مستطیلی)، مارپیچی<sup>۲</sup> و مارپیچ مربعی<sup>۳</sup> بررسی شده است. در شکل ۶-۲۴ این سه حرکت نشان داده شده است.

در روش مستطیلی، حرکت ربات از یک گوشه شروع می‌شود و با سرعت ثابت حرکت می‌کند تا زمانی که به مرز برسد. سپس ربات چرخش ۹۰ درجه انجام می‌دهد و به اندازه  $2b$  حرکت و دوباره چرخش ۹۰ درجه و با سرعت ثابت به مسیر خود ادامه می‌دهد. فاصله بین دو خط اسکن  $2b$  است بنابراین ربات تمامی ناحیه را پیمایش می‌کند. در روش مارپیچی، ربات بر روی یک منحنی حرکت کرده به طوری که دائماً شعاع منحنی افزایش می‌یابد. با استفاده از مختصات قطبی، شعاع و زاویه به صورت خطی افزایش می‌یابد. حرکت مربعی، شبیه به حرکت مارپیچی است. حرکت ربات از مرکز شروع می‌شود، تفاوت این

1 - Scan Lines  
2 - Spiral  
3 - Square Spiral

حرکت با حالت مارپیچی این است که حرکت ربات بر روی خط مستقیم است نه بر روی منحنی و در حرکت مربعی چرخش‌های ۹۰ درجه وجود دارد.



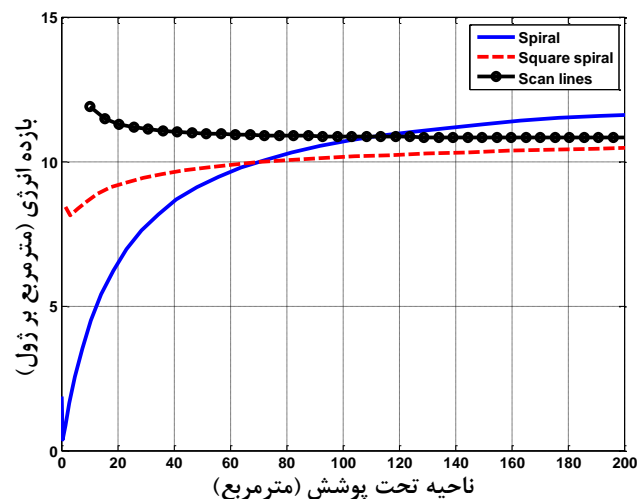
شکل ۶-۲۴ نواحی پوشش داده شده (الف) مستطیلی (ب) مربعی (ج) مارپیچی

برای هرکدام از سه مسیر، ناحیه تحت پوشش محاسبه شده است [۳۸]. انرژی مصرفی نیز با استفاده از مدل توان به دست آمده در قسمت قبل به دست می‌آید. در نتیجه بازده انرژی برای هرکدام از مسیرها را می‌توان با رابطه ۶-۱ به دست آورد.

## ۶-۵-۲-۱ شبیه‌سازی

در این بخش به بررسی نتایج به‌دست‌آمده از شبیه‌سازی با استفاده از مدل توان به‌دست آمده از ربات رومبا پرداخته شده است. که در آن شتاب زاویه‌ای  $5/3$  رادیان بر مجذور ثانیه و سرعت زاویه‌ای  $2/3$  رادیان بر ثانیه و شتاب ثابت  $0.2$  متر بر مجذور ثانیه و سرعت  $0.2$  متر بر ثانیه برای ناحیه  $200$  متر مربع انجام شده است. ربات دارای شعاع  $0.129$  متر و شعاع چرخ  $0.03$  متر است. همچنین در شبیه‌سازی برای حرکت مستطیلی مقدار ارتفاع  $0.8$  متر در نظر گرفته شده است.

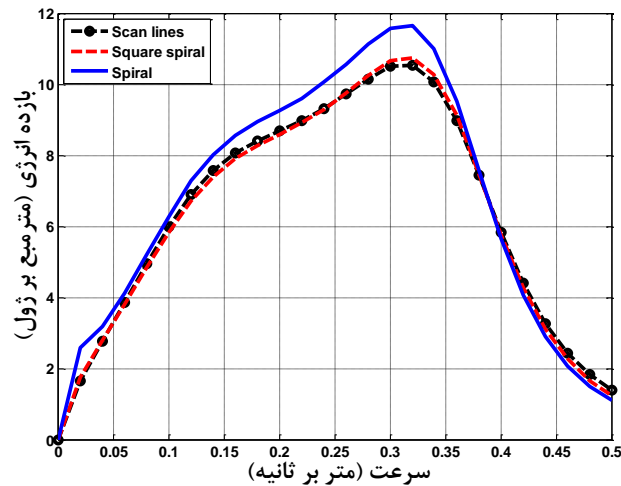
شکل ۶-۲۵ بازده انرژی را برای سه مسیر بررسی شده؛ نشان می‌دهد. زمانی که ناحیه تحت پوشش کوچک است حرکت اسکن خطوط بازدهی بهتری دارد. زیرا حرکت ربات تنها در مسیر مستقیم و بدون چرخش است. زمانی که ناحیه تحت پوشش افزایش می‌یابد، حرکت مارپیچی دارای بازدهی بهتری است. دلیل آن این است که در حرکت اسکن خطوط زمانی که ناحیه حرکت افزایش می‌یابد، در حرکت کاهش شتاب، چرخش و افزایش شتاب رخ می‌دهد که در نتیجه ربات  $2n$  چرخش را در مدت زمانی که سطح  $(2nb + 2b) \times (h + 2b)$  را پیمایش می‌کند؛ انجام می‌دهد. شکل ۶-۲۵ رویکرد تحلیلی را برای این سه مسیر به خوبی نشان می‌دهد.



شکل ۶-۲۵ بازدهی انرژی برحسب ناحیه تحت پوشش

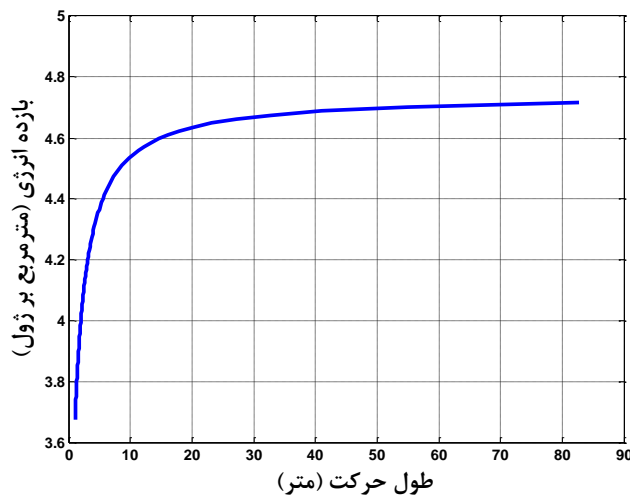


در شکل ۶-۲۶ نمودار سرعت ربات برحسب بازدهی رسم شده است که نشان می‌دهد در سرعت‌های بین ۰.۲۵ و ۰.۳۵ ربات داری بازدهی بیشتر است.



شکل ۶-۲۶ بازدهی انرژی برحسب سرعت ربات

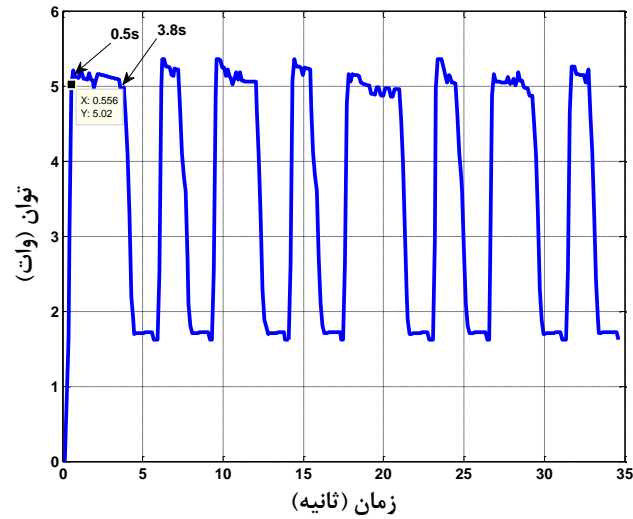
شکل ۶-۲۷ بازده روش مستطیلی را برای طول‌های مختلف نشان می‌دهد. زمانی که ناحیه تحت پوشش ثابت است، با افزایش  $h$  تعداد کاهش شتاب‌ها و چرخش‌ها و شتاب گرفتن‌ها کاهش می‌یابد. همان‌طور که در شکل نشان داده شده بازدهی می‌تواند از ۳.۸ به ۴.۶ متر مربع بر ژول افزایش پیدا کند.



شکل ۶-۲۷ بازدهی انرژی برحسب ارتفاع در روش اسکن خطوط

#### ۶-۵-۲-۲- محاسبه بازده انرژی در روش مستطیلی

در اینجا با توجه به فرضیات مربوط به بخش ۶-۵-۲-۱، نتایج شبیه‌سازی و نتایج به‌دست آمده از آزمایش‌ها بر روی ربات رومبا باهم مقایسه شده‌اند.



شکل ۶-۲۸ نمودار توان مصرفی ربات در حرکت مستطیلی

برای به دست آوردن بازده انرژی در بازه زمانی ۰.۵ تا ۳.۸ ثانیه، میانگین توان مصرفی در این بازه را محاسبه کرده و در زمان سپری شده ضرب کرده تا انرژی مصرفی به دست آید. سپس از رابطه ۶-۱ بازده انرژی را محاسبه کرده که در این بازه زمانی برابر ۱۱.۹۷۶ مترمربع بر ژول است.

فصل هفتم

نتیجه‌گیری و پیشنهادات

## ۷-۱- نتیجه‌گیری

در این پایان‌نامه به بررسی موارد زیر پرداخته شد.

- ابتدا معادلات دینامیک ربات سیار استخراج شد و معادله کنترلی به دست آمده به عنوان قید مربوط به گشتاورها در بهینه‌سازی انرژی ربات سیار قرار گرفت.
- روش میدان پتانسیل معرفی شد و به بهبود این روش بر مبنای رگرسیون خطی پرداخته شد. به طوری که این روش یک مسیر بهینه برای ربات سیار بدون قرار گرفتن در کمینه محلی و به وجود آمدن مسئله هدف غیرقابل دسترس ایجاد می‌کند. چون مسیری که به وسیله میدان پتانسیل بهبود داده شده به دست می‌آید، در محیط واقعی برای ربات مناسب نیست، با استفاده از روش رگرسیون خطی مسیری با چرخش‌های کمتر برای ربات ایجاد شده است تا در محیط واقعی ربات سیار در زمان کمتر به نقطه هدف برسد.
- مسئله مسیریابی برای یک ربات در یک محیط با مانع دوبعدی با تبدیل به یک مسئله بهینه‌سازی مقید و حل مسئله با استفاده از الگوریتم‌های تکاملی و مقایسه این الگوریتم‌ها با یکدیگر به طراحی مسیر برای ربات سیار پرداخته شد.
- طراحی مسیر بهینه ربات سیار به صورت یک مسئله کنترل بهینه فرمولاسیون شده، که این فرمولاسیون بیانگر معادلات دینامیکی ربات سیار غیرهولونومیک در فضای حالت است، که به عنوان قیود مسئله در نظر گرفته شده است و هدف کمینه کردن حرکت ربات سیار در امتداد خط مستقیم و مسیر منحنی‌الخط، که منجر به حداقل کردن انرژی مصرفی متناسب با منحنی سرعت می‌باشد، است.
- ربات سیار و شبیه‌ساز رومبا را معرفی گردید و به تست‌های مربوط به ناوبری و انرژی با در نظر گرفتن هدف کمینه طول مسیر و کمینه انرژی مصرفی بر روی ربات پرداخته شد.

- همچنین به بررسی بازده انرژی ربات سیار رومبا در مسیرهای مختلف با استفاده از مدل توان به دست آمده از آزمایش تجربی توسط ربات پرداخته شد و دیده شد که بازده انرژی به ناحیه تحت پوشش، سرعت حرکت ربات و نوع مسیرهای پیمایش شده بستگی دارد. برای ناحیه‌های کوچک بازده انرژی حرکت مستطیلی زمانی که ربات در مسیر مستقیم حرکت می‌کند بیشتر از مسیرهای دیگر است. زمانی که ناحیه تحت پوشش بزرگ‌تر می‌شود، بازده انرژی مسیر مارپیچی بیشتر است. زیرا در این حرکت ربات بدون توقف، حرکت چرخشی را انجام می‌دهد.
- در این پژوهش از محصول رومبا شرکت آی‌روبات استفاده شد و برای افرادی که بخواهند از این ربات برای اهداف مختلف استفاده کنند، می‌تواند مورد استفاده قرار گیرد. این اهداف می‌تواند در زمینه پردازش تصویر، دینامیک محیط، بررسی ارتباط در فواصل دور برای عملکردهای مختلف با تعیین اهداف خاص مثل پیدا کردن یک شیء منحصر به فرد در محیط یا پیمایش در محیط با دینامیک خاص به کارگیری گردد.

## ۷-۲- پیشنهادات

- با گسترش دادن روش میدان پتانسیل بهبود داده شده می‌توان از آن در محیط‌های پیچیده‌تر با موانع متحرک و به صورت محلی استفاده کرد. همچنین می‌توان روش‌های ناوبری استفاده شده در این پژوهش را برای سیستم‌های چند ربات ارتقا داد و به طراحی مسیر برای چندین ربات پرداخت.
- می‌توان به کاهش زمان محاسباتی برای الگوریتم میدان پتانسیل بهبود داده شده پرداخت.
- می‌توان با استفاده از الگوریتم کنترل بهینه، معادلات بهینگی ربات را استخراج کرد به گونه‌ای که برای عدم برخورد با موانع مشخص در محیط علاوه بر قيود مربوط به موانع، قيود مربوط به ترم‌های گشتاور و سرعت را به الگوریتم‌های تکاملی و میدان پتانسیل اضافه کرد.
- بکارگیری پردازش تصویر و سنسورهای تعیین کننده موانع و دیگر سنسورهای خارجی و نصب آن‌ها بر روی ربات رومبا برای شناسایی دقیق یک محیط و تعیین موانع به صورت دقیق، برای مسیریابی ربات در محیط‌های پیچیده‌تر می‌تواند کار آمد باشد. البته با توجه به اینکه تجهیزات رومبا کامل‌تر از نمونه‌های مشابه آن می‌باشد، ولی می‌توان با اضافه نمودن سنسورهایی به آن، دقت عملکرد ربات را بیشتر کرد.
- کاملاً روشن است که هدف اصلی این پژوهش طراحی مسیر بهینه ربات سیار با در نظر گرفتن قيود مختلف برای ربات است. جای خالی الگوریتم‌های کنترلی در کارکردهای گوناگون، برای کمک به کارایی ربات سیار، احساس می‌شود.

پیوست الف

معرفی الگوریتم‌های تکاملی

## الف-۱- الگوریتم بهینه سازی ازدحام ذرات

در الگوریتم PSO، براساس حرکت جمعی ذرات است که در فضای جستجو پخش شده‌اند. هر ذره مقدار تابع هدف را در موقعیتی از فضا که در آن قرار گرفته است، محاسبه می‌کند. سپس با استفاده از ترکیب اطلاعات محل فعلی و بهترین محلی که قبلاً در آن بوده است و همچنین اطلاعات یک یا چند ذره از بهترین ذرات موجود در جمع، جهتی را برای حرکت انتخاب می‌کند. پس از انجام حرکت جمعی، یک مرحله از الگوریتم به پایان می‌رسد. این مراحل چندین بار تکرار می‌شوند تا آن که جواب مورد نظر به دست بیاید. مدل اولیه الگوریتم به صورت زیر است.

$x^{i,best}$ : بهترین موقعیت تجربه شده،  $v^i$ : سرعت،  $x^i$ : موقعیت

معادلات توصیف کننده‌ی رفتار ذرات:

$$v^i[t + 1] = wv^i[t] + c_1r_1(x^{i,best}[t] - x^i[t]) + c_2r_2(x^{gbest}[t] - x^i[t]) \quad (\text{الف-۱})$$

$$x^i[t + 1] = x^i[t] + v^i[t + 1] \quad (\text{الف-۲})$$

الگوریتم بهینه سازی ازدحام ذرات بر اساس مفهوم جمعیت پایه‌ریزی شده است و هر ذره در جمعیت یک راه حل در مورد مسئله طراحی مسیر ارائه می‌دهد. راه حل کوتاه‌ترین مسیر از نقطه شروع تا نقطه هدف بدون برخورد با موانع است. بنابراین ما برای هر ذره  $x, y$  داریم. هر ذره به عنوان یک نقطه در فضای کاری در نظر گرفته شده، زمانی که الگوریتم اجرا می‌شود همه‌ی بهترین نقاط<sup>۱</sup> محاسبه شده یک مسیر از نقطه اولیه به نقطه هدف ایجاد می‌کند. بهترین پاسخ در هر تکرار بررسی شده و به عنوان یک نقطه بهینه در مسیر در نظر گرفته می‌شود. با ترکیب کردن همه پاسخ‌های خوب یا بهترین نقاط در هر تکرار مسیر کامل بهینه بدست می‌آید.

## الف-۲- الگوریتم کلونی زنبورها

---

<sup>۱</sup> - Global Best Point



الگوریتم کلونی زنبورهای مصنوعی الهام گرفته از رفتار زنبورها در طبیعت می‌باشد. این الگوریتم از سه گروه زنبورهای کارگر، تماشاگر و پیش‌آهنگ تشکیل شده است. در ابتدا مجموعه‌ای از منابع غذایی به طور تصادفی انتخاب می‌شوند. زنبورهای کارگر به منابع مراجعه کرده و میزان شهد آنها را محاسبه می‌کنند. سپس این زنبورها به کندو بازگشته و اطلاعات خود را با دیگر زنبورها (تماشاگران) به اشتراک می‌گذارند. در مرحله دوم بعد از تبادل اطلاعات، هر زنبور کارگر به سمت منبعی می‌رود که قبلاً دیده است و ممکن است براساس اطلاعات دیداری که از محیط می‌گیرد یک منبع جدید در همسایگی منبع قبلی انتخاب کند. در مرحله سوم تماشاگران با توجه به اطلاعاتی که از زنبورهای کارگر در محل رقص گرفته‌اند یک محدوده منبع غذایی را بر مبنای شهد آن ترجیح می‌دهند. بعد از رسیدن به محل ممکن است با توجه به اطلاعات دیداری یک منبع جدید را که در همان اطراف قرار دارد را انتخاب کنند. زمانی که منبع پایان پذیرد یا ترک شود یک منبع جدید که به طور تصادفی توسط پیش‌آهنگان یافت شده است جایگزین می‌شود. این چرخه تا برآورده شدن نیازها تکرار خواهد شد. در این مدل در هر چرخه حداکثر یک پیش‌آهنگ وجود دارد و تعداد زنبورهای کارگر و تماشاگران برابر است. همان‌طور که گفته شد هر کدام از زنبورهای کارگر و یا تماشاگران ممکن است تغییراتی بر روی موقعیت منبع غذایی (راه حل) در حافظه خود ایجاد کنند و شایستگی آن را محاسبه کرده، در صورتی که میزان شایستگی آن از راه حل قدیمی بیشتر باشد، راه حل جدید انتخاب می‌شود در غیر این صورت همان راه حل قدیمی باقی خواهد ماند. این تغییرات توسط رابطه زیر بدست می‌آید:

$$v_{ij} = x_{ij} + (x_{ij} - x_{kj}) \quad (\text{الف-۳})$$

$$i \neq k, k \in \{1, 2, \dots, BN\}, j \in \{1, 2, \dots, D\}, \emptyset_{ij} \in [-1, 1] \quad (\text{الف-۴})$$

که  $\emptyset_{ij}$  یک عدد تصادفی در بازه  $[-1, 1]$  است. این متغیر تولید موقعیت منابع غذایی همسایه، اطراف  $x_{ij}$  را کنترل می‌کند. در این رابطه،  $BN$  تعداد زنبورهای کارگر می‌باشد و متغیر  $K$  به صورت تصادفی تولید می‌گردد و با  $i$  متفاوت خواهد بود. براساس رابطه الف-۳ هر چه تفاوت بین  $x_{ij}$  و  $x_{ik}$  کاهش یابد، انحراف از موقعیت  $x_{ij}$  نیز کاهش خواهد یافت. در حقیقت در این رابطه سعی می‌کنیم یک بعد از ابعاد

یکی از موقعیت‌ها انتخاب کرده و با توجه به میزان  $\phi_{ij}$  به سمت آن و یا در خلاف جهت آن حرکت کنیم همانند الگوریتم PSO، با این تفاوت که در اینجا با انتخاب تصادفی سعی می‌کنیم تا حدودی ایجاد تنوع نموده و از قرار گرفتن در بهینه محلی جلوگیری کنیم. بعد از اتمام فرآیند جستجو، تماشاگران اطلاعات هر کدام از زنبورهای کارگر را ارزیابی می‌کنند و با یک احتمال که متناسب است با میزان کیفیت شهد منبع، یکی از منابع غذایی را انتخاب می‌کنند. این احتمال از رابطه الف-۵ بدست می‌آید:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (\text{الف-۵})$$

در این رابطه  $fit_i$  میزان شایستگی منبع غذایی متناظر با زنبور عسل  $i$  ام و SN تعداد راه‌حل‌های موجود می‌باشد. در صورتی که منبع پایان پذیرد و یا کیفیت یک منبع غذایی مناسب نباشد، زنبور کارگر آن را رها کرده و به یک پیش‌آهنگ تبدیل می‌شود. این رفتار بدین صورت مدل می‌گردد که اگر شایستگی یک نقطه بعد از چندین تکرار بهبود نیابد، بدین معنی است که در یک بهینه محلی قرار داریم. بنابراین آن نقطه حذف می‌شود و یک نقطه جدید به صورت تصادفی تولید می‌گردد.

### الف-۳- الگوریتم ژنتیک

الگوریتم ژنتیک یک روش بهینه‌سازی الهام گرفته از طبیعت جاندار است که می‌توان در طبقه بندی‌ها، از آن به عنوان یک روش عددی، جستجوی مستقیم و تصادفی معرفی کرد. این الگوریتم، الگوریتمی مبتنی بر تکرار است و اصول اولیه آن از علم ژنتیک اقتباس گردیده است.

به طو خلاصه الگوریتم ژنتیک از عملگرهای زیر تشکیل شده است:

رمزگذاری<sup>۱</sup>: در این مرحله الگوریتم به جای اینکه بر روی پارامترها و متغیرهای مسئله کار کند با شکل کد شده آنها سروکار دارد.

ارزیابی<sup>۲</sup>: تابع برازندگی را از اعمال تبدیل مناسب بر روی تابع هدف یعنی تابعی که قرار است بهینه شود به دست می‌آید. این تابع هر رشته را با یک مقدار عددی ارزیابی می‌کند که کیفیت آن را مشخص

<sup>1</sup> - Encoding

<sup>2</sup> - Evaluation

می‌کند. هر چه کیفیت رشته جواب بالاتر باشد مقدار برازندگی جواب بیشتر است و احتمال مشارکت برای تولید نسل بعدی نیز افزایش خواهد یافت.

ترکیب<sup>۱</sup>: مهمترین عملگر در الگوریتم ژنتیک عملگر ترکیب است. ترکیب، فرآیندی است که در آن نسل قدیمی کروموزوم‌ها با یکدیگر مخلوط و ترکیب می‌شوند تا نسل تازه‌ای از کروموزوم‌ها به وجود بیاید. جفت‌هایی که در قسمت انتخاب، به عنوان والد انتخاب شدند، در این قسمت ژن‌هایشان را با هم مبادله می‌کنند و اعضای جدید بوجود می‌آورند. ترکیب در الگوریتم باعث از بین رفتن پراکندگی یا تنوع ژنتیکی جمعیت می‌شود. زیرا اجازه می‌دهد ژن‌های خوب یکدیگر را بیابند.

جهش<sup>۲</sup>: جهش نیز عملگر دیگری هست که جواب‌های ممکن دیگری را تولید می‌کند. در الگوریتم ژنتیک بعد از اینکه یک عضو در جمعیت جدید به وجود آمد در هر ژن احتمال جهش وجود دارد. در جهش ممکن است ژنی از مجموعه ژن‌های جمعیت حذف شود یا ژنی که تا به حال در جمعیت وجود نداشته است به آن اضافه شود.

رمزگشایی<sup>۳</sup>: عکس عمل رمزگذاری است.

روند الگوریتم ژنتیک:

۱- ایجاد جمعیت تصادفی و ارزیابی آن‌ها

۲- انتخاب والدین و ترکیب آن‌ها برای ایجاد جمعیت فرزندان

۳- انتخاب اعضای جمعیت برای اعمال جهش و ایجاد جمعیت جهش‌یافتگان

۴- ادغام جمعیت اصلی، فرزندان و جهش‌یافتگان و ایجاد جمعیت اصلی جدید

۵- اگر شرایط خاتمه محقق نشده باشد، از مرحله ۲ تکرار می‌کنیم.

۶- پایان

---

<sup>1</sup> - Crossover

<sup>2</sup> - Mutation

<sup>3</sup> - Decoding

## الف-۴- الگوریتم رقابت استعماری

الگوریتم رقابت استعماری روشی در حوزه محاسبات تکاملی است که به یافتن پاسخ بهینه مسائل مختلف بهینه‌سازی می‌پردازد. این الگوریتم با مدلسازی ریاضی فرایند تکامل اجتماعی - سیاسی، الگوریتمی برای حل مسائل ریاضی بهینه‌سازی ارایه می‌دهد. از لحاظ کاربرد، این الگوریتم در دسته الگوریتم‌های بهینه‌سازی تکاملی همچون الگوریتم‌های ژنتیک، بهینه‌سازی انبوه ذرات و ... قرار می‌گیرد. همانند همه الگوریتم‌های قرار گرفته در این دسته، الگوریتم رقابت استعماری نیز مجموعه اولیه‌ای از جواب‌های احتمالی را تشکیل می‌دهد. این جواب‌های اولیه در الگوریتم ژنتیک با عنوان کروموزوم، در الگوریتم ازدحام ذرات با عنوان ذره و در الگوریتم رقابت استعماری نیز با عنوان کشور شناخته می‌شوند. الگوریتم رقابت استعماری با روند خاصی که در ادامه می‌آید، این جواب‌های اولیه (کشور ها) را به تدریج بهبود داده و در نهایت جواب مناسب مسئله بهینه‌سازی (کشور مطلوب) را در اختیار می‌گذارد.

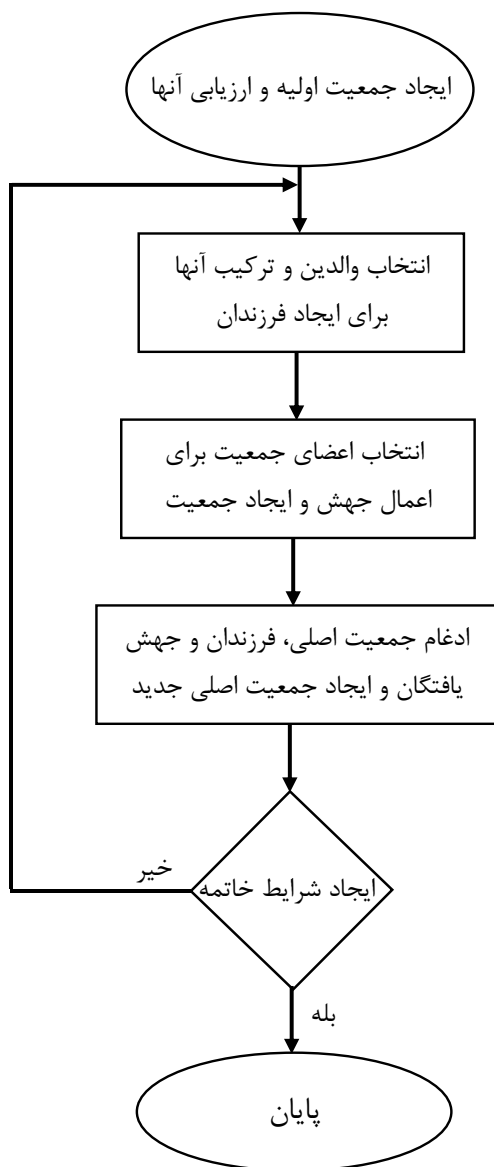
پایه‌های اصلی این الگوریتم را سیاست همسان‌سازی، رقابت استعماری و انقلاب تشکیل می‌دهند. این الگوریتم با تقلید از روند تکامل اجتماعی، اقتصادی و سیاسی کشورها و با مدلسازی ریاضی بخش‌هایی از این فرایند، عملگرهایی را در قالب منظم به‌صورت الگوریتم ارایه می‌دهد که می‌توانند به حل مسائل پیچیده بهینه‌سازی کمک کنند. در واقع این الگوریتم جواب‌های مسئله بهینه‌سازی را در قالب کشورها نگریسته و سعی می‌کند در طی فرایندی تکرار شونده این جواب‌ها را رفته رفته بهبود داده و در نهایت به جواب بهینه مسئله برساند.

## الف-۵- مقایسه الگوریتم‌های تکاملی

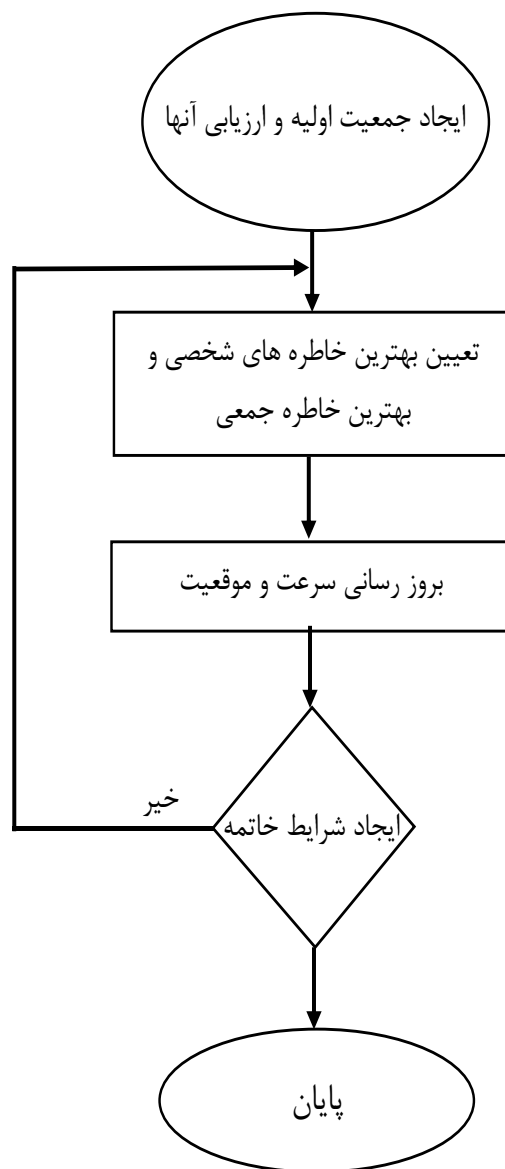
در جدول الف-۱ الگوریتم‌های تکاملی با هم مقایسه شده‌اند [۳۹]. در ادامه مراحل چهار الگوریتم تکاملی که شامل الگوریتم‌های ازدحام جمعیت، ژنتیک، رقابت استعماری و زنبورعسل آمده است.

جدول الف-۱ مقایسه الگوریتم‌های تکاملی

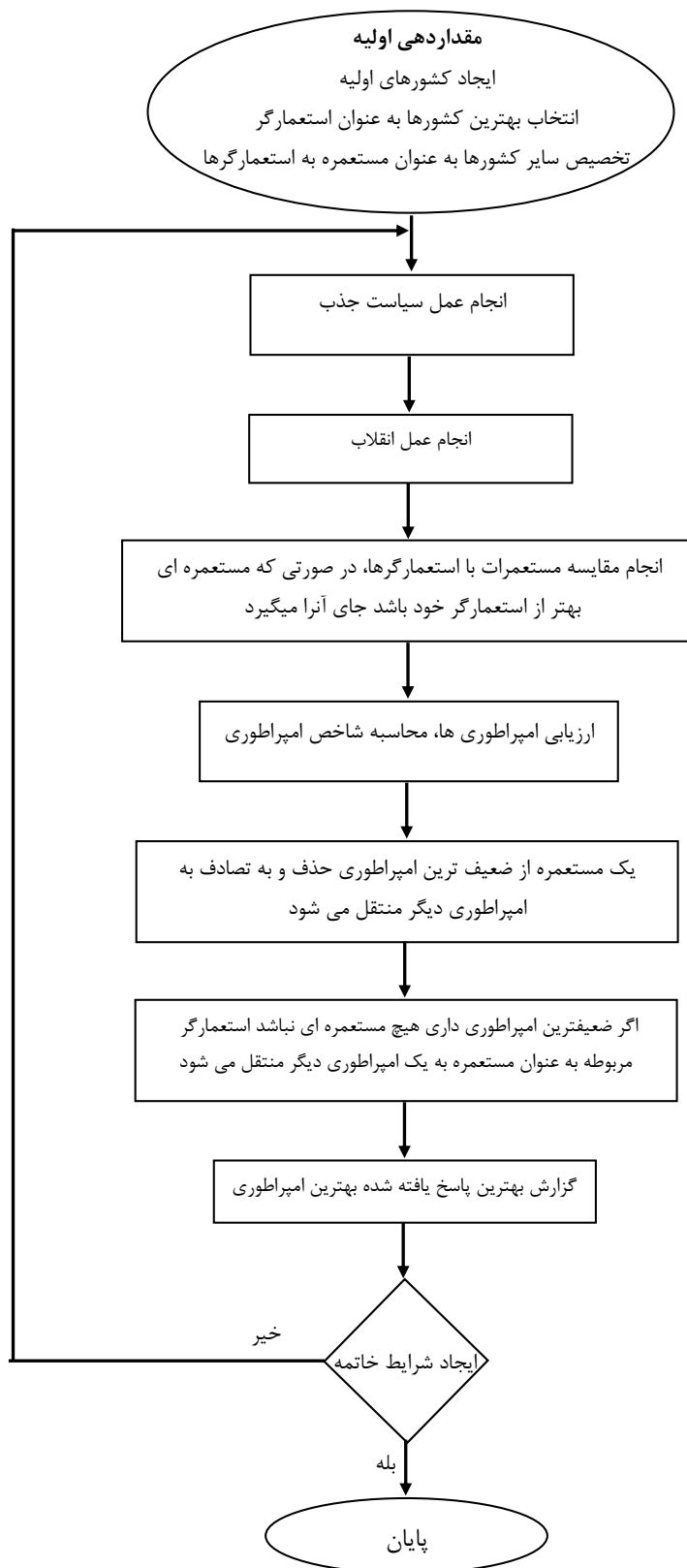
معایب	مزایا	نوع الگوریتم
۱- استفاده از معادلات پیچیده و بسیار ۲- دارای مدت زمان طولانی محاسبات ۳- تنظیم کردن پارامترهای بسیار	۱- سازگاری با مسایل مختلف ۲- قدرت کارایی بالا و قابلیت انعطاف در محیط‌های دینامیک	رقابت استعماری
۱- دارای تجزیه و تحلیل تئوری پیچیده ۲- الگوریتم اصلی برای جستجو در فضای گسسته ایجاد شده	۱- قدرت کارایی بالا و قابلیت انعطاف در محیط‌های دینامیک ۲- مناسب برای مسایلی که دارای گراف	زنبور عسل
۱- دارای جستجو محلی ضعیف ۲- دارای نرخ همگرایی کم و در مسائل پیچیده به راحتی در کمینه محلی قرار می‌گیرد	۱- دارای ساختار ساده ۲- داری پیاده سازی ساده می‌باشد ۳- سریع و ارزان ۴- داشتن پارامترهای اندک برای تنظیم	ازدحام ذرات
۱- دارای نرخ همگرایی کم ۲- نتایج غیرقابل پیش‌بینی ۳- وابستگی زیاد به ترکیب و جهش	۱- در انواع مختلف مسائل بهینه‌سازی بکار می‌رود. ۲- به راحتی با الگوریتم‌های دیگر ترکیب می‌شود. ۳- در فضای جستجو به صورت باینری و پیوسته وجود دارد.	ژنتیک



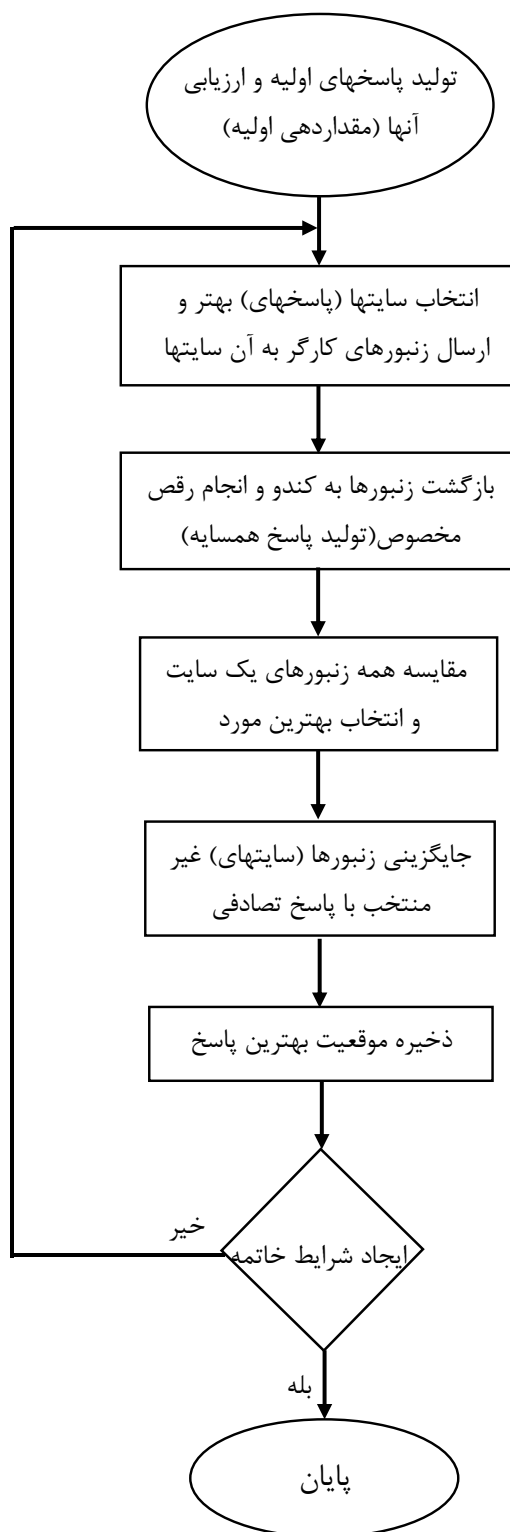
شکل الف-۲ مراحل الگوریتم ژنتیک



شکل الف-۱ مراحل الگوریتم ازدحام ذرات



شکل الف-۳ مراحل الگوریتم رقابت استعماری



شکل الف-۴ مراحل الگوریتم زنبورعسل



پیوست ب

معرفی ربات سیار رومبا و

نحوه ارتباط آن با سی شارپ

## ب-۱ - مقدمه

در این بخش به معرفی ربات رومبا و سنسورهای موجود در آن پرداخته می‌شود. یکی از خصوصیات رومبا رابط ROI آن می‌باشد یعنی واسط کاربری باز و آزاد دارد ROI که در اینجا بررسی می‌شود فرماندهی از طریق پورت سریال می‌باشد.

## ب-۲ - منبع تغذیه

برای هر سیستم رباتیک منبع تغذیه مهم است. رومبا دارای منبعی با قابلیت بالای شارژ مناسبی می‌باشد. این پکیج قدرتی برای اجرای بیش از ۱۰۰ دقیقه از یک عملکرد عادی را فراهم می‌سازد و می‌تواند تا مدت زمان ۳ ساعت شارژ شود.

## ب-۲-۱- جزئیات منبع تغذیه

در داخل باتری ۱۲ سلول نیکل متال هیدرید جای دارد. هر سلول ۱.۲ ولت خروجی می‌دهد. جدیدترین پک زرد باتری حداقل از ۳۰۰۰ تا ۳۲۰۰ میلی‌آمپر استفاده می‌شود.

## ب-۲-۲- توان در دسترس

در زمان روشن شدن دستگاه در حالتی که بدون حرکت است رومبا ۱۵۰ تا ۲۵۰ میلی‌آمپر برق می‌کشد. البته این موضوع بستگی به مدل هم دارد در حین عملیات رومبا ۱۵۰۰ تا ۲۰۰۰ میلی‌آمپر در حالت جاری می‌کشد. این تنوع در مصرف جاری بستگی به تنوع نوع زمین دارد فرش ضخیم بیشتر از زمین سخت جریان می‌کشد. پک باتری می‌تواند با بیش‌ترین مقدار تخلیه در نرخ ۴ آمپر باشد، که می‌توان آن را با یک سوئیچ داخلی محدود نمود.

## ب-۳- موتورها

رومبا دارای پنج کنترلر موتور الکتریکی می‌باشد. دو عدد از این درایوهای موتور از طریق مدولاسیون عرض پالس PWM سرعت را تغییر می‌دهد و در هر دو حالت رو به جلو و عقب حرکت می‌کند. سه عدد از موتورها برای برس اصلی برس طرفین و مکش و دارای کنترلر روشن و خاموش می‌باشد.

### ب-۳-۱- درایورهای موتور

دو چرخ در خطا المکزین آن قرار دارد و درست پشت مرکز گرانشی است داشتن چرخ در پشت مرکز ثقل حرکت رومبا را به جلو در هر جهت بدون چرخش چرخکها فراهم می‌سازد. موتورها می‌تواند رومبا را با سرعت ۵۰۰ میلی‌متر بر ثانیه حرکت دهد، وقتی که در سرعت نرمال کار می‌کند و همچنین در آهسته‌ترین حالت ۳۰۰ میلی‌آمپر می‌کشد.

### ب-۴- حسگرها

رومبا عمدتاً توسط سنسورهای ضربه‌گیر و سنسور مادون قرمز دیوار و سنسور خاک هدایت می‌شود. برای تشخیص شرایط خطرناک دارای صخره یا دست‌انداز تشخیص‌دهنده مادون قرمز و حسگر چرخ به خوبی عمل می‌کنند.

#### ب-۴-۱- حسگر ضربه

رومبا دو سنسور ضربه در موقعیت ساعت ۱۱ و ۱ دارد. سنسور ضربه‌گیر که در قسمت جلو ربات قرار دارد زمانی که به مانع برخورد کند، عمل می‌کند. دو سنسور ضربه‌گیر یکی در سمت چپ و دیگری در سمت راست وجود دارد اما زمانی که در قسمت جلو ضربه‌ای وارد شود هر دو سنسور ضربه‌گیر چپ و راست عمل می‌کند.

#### ب-۴-۲- حسگر مادون قرمز

شش سنسور مادون قرمز روی رومبا وجود دارد همه آنها روی ضربه‌گیر جلو می‌باشند ۴ عدد از آنها به صورت پایین و سنسورهای صخره‌ای هستند (برای دست‌انداز) و دیگری سمت راست است و سنسور دیوار است. آخرین سنسور مادون قرمز شامل کنترل از راه دور یا دیوار مجازی می‌باشد که دکمه تمیزکننده محیط در موقعیت ساعت ۱۲ روی ضربه‌گیر را نشان می‌دهد. این سنسور با لنز ۳۶۰ درجه قابلیت دیدن هر جهتی را دارد.

#### ب-۴-۳- حسگر داخلی

شایع‌ترین سنسوری که استفاده می‌شود سنسور چرخ رها شده است. هر سه چرخ میکرو سویچ دارد که از کار افتادن چرخ را تشخیص می‌دهد. این چرخ‌های معلق تشخیص‌دهنده‌های صخره هستند. آن‌ها نشان می‌دهند که رومبا در وضعیت وخیم است و باید الگوریتم جاری لغو شود. سنسور خاک یک یا دو دیسک کوچک فلزی است که در زیر برس اصلی است و یک سنسور لمسی خازنی به نظر می‌رسد. سنسور خازنی در لامپ لمسی استفاده می‌شوند که با قرار دادن انگشت روی سطح فلزی لامپ کنترل می‌شود اگرچه سنسور لامپ لمسی تنها یک نتیجه خاموش و روشن را فراهم می‌کند. سنسورهای خاک هم یک مقدار آنالوگ را فراهم می‌کند. آخرین مجموعه از سنسورهای داخلی سنسور اندازه‌گیری قدرت‌های مختلف است، زیرا قدرت بسیار مهم است در یک سیستم رباتیک تعداد زیادی باتری و سنسورهای شارژی وجود دارد. یک تخمین زنده ظرفیت و درصد ظرفیت شارژ وجود دارد هر دو آن‌ها مقادیر آنالوگ و با واحدهای میلی‌آمپر ساعت دارند

#### ب-۴-۴- حسگر دیوار

در شکل ب-۲ سنسور دیوار به نمایش درآمده است که در سمت راست بر روی ضربه‌گیر رومبا قرار گرفته است.

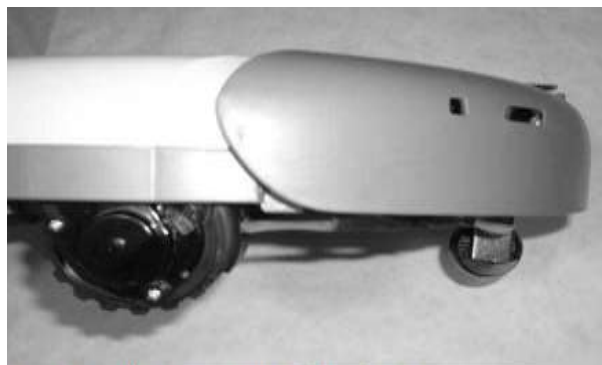
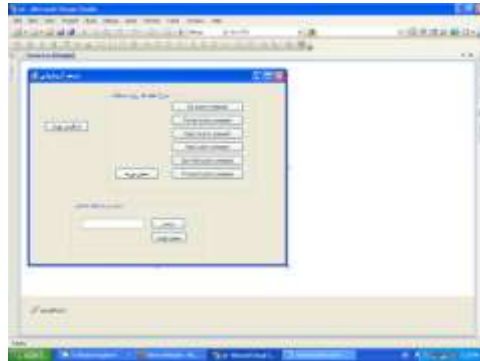


FIGURE 6-5: Wall sensors, an optical object detector

شکل ب-۱ نمایش سنسور دیوار [۳۶]

#### ب-۵- ارتباط با سی‌شارپ:

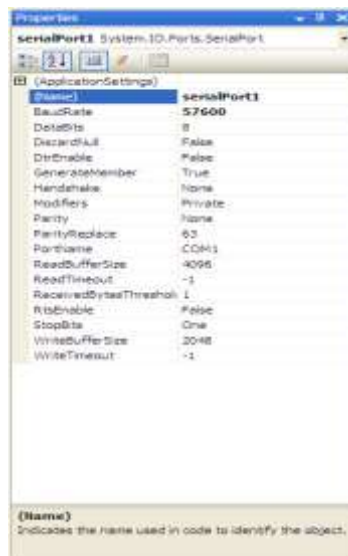
در این بخش با استفاده از زبان برنامه‌نویسی سی‌شارپ با رومبا از طریق پورت سریال ارتباط برقرار می‌شود. برای این منظور ابتدا یک application windows ایجاد شده و اشیائی را که در شکل ب-۲ نشان داده شده را در آن قرار می‌دهیم.



شکل ب-۲ محیط سی‌شارپ

در ابتدا برای استفاده از هر پورته می‌بایست آن را تنظیم کرد که در شکل ب-۳ اجزا پورت سریال مطابق با سخت‌افزار رومبا تنظیم شده است. برای این منظور قطعه کد زیر در آن نوشته می‌شود.

`serialPort1.Open()`



شکل ب-۳ تنظیمات مربوط به ارتباط سریال

با توجه به کدهای موجود در شکل ب-۴ که نحوه فرماندهی به رومبا را مشخص می‌کند، این کدها را به روی پورت ارسال تا از دستگاه پاسخ‌های مناسب دریافت گردد.

Table 2-4 Remote Control Codes		
Value	Hexadecimal values	Description
255	0xff	No button pressed
138	0x8a	Power button pressed
137	0x89	Pause button pressed
136	0x88	Clean button pressed
133	0x85	Max button pressed
132	0x84	Spot button pressed
131	0x83	Spin left button pressed
130	0x82	Forward button pressed
129	0x81	Spin left button pressed

شکل ب-۴ کدهای اعمالی به ربات

کدهای زیر این عملیات را انجام می‌دهند:

```
private void button2_Click(object sender, EventArgs e)
{
s[0]=0xff;serialPort1.Write(s, 0, 1);
}

private void button3_Click(object sender, EventArgs e)
{
s[0]=0x8a;serialPort1.Write(s, 0, 1);
}

private void button4_Click(object sender, EventArgs e)
{
s[0]=0x85;serialPort1.Write(s, 0, 1);
}

private void button5_Click(object sender, EventArgs e)
{
s[0]=0x81;serialPort1.Write(s, 0, 1);
}

private void button9_Click(object sender, EventArgs e)
{
s[0]=0x83;serialPort1.Write(s, 0, 1);
}

private void button10_Click(object sender, EventArgs e)
{
s[0]=0x82;serialPort1.Write(s, 0, 1);
}
```

در انتها بایستی پورت را ببندیم که نشانگر اتمام کار و فرماندهی به رومبا است.

```
serialPort1.Close();
```

بعد از برقراری ارتباط سریال با استفاده از رابط کاربری رومبا می‌توان دستورات مربوطه را برای اجرا به

رومبا داد. این دستورات به صورت زیر مورد استفاده قرار می‌گیرد. برای حرکت رو به جلو رومبا از طریق

سی‌شارپ مجموعه دستورات زیر فرستاده می‌شود.

```
byte[] s = new byte[6];  
s[0] = 0x82;  
s[1] = 0x89;  
s[2] = 0x00;  
s[3] = 0x10;  
s[4] = 0x80;  
s[5] = 0x00;  
serialPort1.Write(s, 0, 6);
```

مجموعه دستورات بالا برای حرکت رو به جلو رومبا در قالب کد هگز نوشته شده است. در مجموع شش بایت برای رومبا فرستاده شده است. که بایت اول حالت کنترل، بایت دوم برای راهاندازی و چهار بایت بعدی دستورات سرعت و شعاع حرکت رومبا است. ابتدا کد ۱۳۰ که رومبا را در حالت کنترلی قرار می‌دهد و کاربر می‌تواند بر روی رومبا کنترل داشته باشد. سپس کد ۱۳۷ یا ۸۹ هگز بر روی رومبا فرستاده شده است که دستور درایو یا راهانداز رومبا است. دستور درایو دارای پنج بایت است که دو بایت اول برای سرعت و دو بایت بعدی برای شعاع حرکت رومبا است.

**سرعت:** مقدار سرعت برحسب میلی‌متر بر ثانیه است. رومبا در سرعت با علامت مثبت، به صورت مستقیم و در سرعت با علامت منفی رومبا به سمت عقب حرکت می‌کند.

**شعاع:** پارامتر دوم مقدار شعاع گردش است که برحسب میلی‌متر است. این شعاع با یک دایره در پیرامون رومبا توصیف می‌شود، که رومبا روی آن حرکت می‌کند. حرکت بر روی این کمان رومبا را قادر می‌سازد که به راحتی در اطراف بسیاری از چیزها مانند گوشه‌های دیوار، پایه‌های میز و غیره حرکت کند. اگر مقدار شعاع مثبت باشد، رومبا به سمت چپ حرکت می‌کند و اگر منفی باشد، رومبا به سمت راست حرکت می‌کند. در سرعت با علامت مثبت رومبا به صورت مستقیم و در سرعت با علامت منفی رومبا به سمت عقب حرکت می‌کند. با توجه به علامت سرعت و شعاع، رومبا می‌تواند در چهار جهت مختلف حرکت کند. داشتن شعاع باعث می‌شود که رومبا بتواند در امتداد یک دایره حرکت کند. تنها رومبا در مورد این سه مقدار شعاع خاص حرکت دایره‌ای را انجام نمی‌دهد:

حرکت مستقیم (۳۲۷۶۸ که در مبنای هشت 0X8000)، چرخش به چپ (۱ که در مبنای هشت 0X0001)، چرخش به راست (۱- که در مبنای هشت 0XFFFF) می باشد.

با انجام عملیات فوق رومبا به راحتی می تواند مسافت های مستقیم رو به جلو و عقب را پیمایش کند. حرکت رومبا در مسافت های خاص: با توجه به طرز کار رابط پورت سریال رومبا همه دستورات را شروع به اجرا می کند. اما نمی توان به رومبا گفت؛ چه زمانی متوقف شود. پس از صدور دستور spinRightAt در بخش بعدی رومبا شروع به چرخش در اطراف می کند بدون اینکه متوقف شود، حتی اگر از برنامه خارج شود. اگر شما می خواهید یک مسافت خاصی را به صورت مستقیم حرکت کنید یا چرخشی با زاویه خاص داشته باشید، شما باید فرمان حرکت در یک سرعت خاص را به رومبا بدهید و سپس برای یک زمان مشخص صبر کنید. برای مثال برای حرکت به صورت مستقیم به اندازه ۳۰۰ میلی متر با سرعت ۱۰۰ میلی متر بر ثانیه باید به اندازه ۳ ثانیه صبر کنید و سپس متوقف شوید. برای این منظور برنامه مورد نظر در سی شارپ به صورت زیر نوشته شده است:

```
private void distance(float dis)
{
    byte[] s = new byte[6];
    s[0] = 0x82;
    s[1] = 0x89;
    s[2] = speed[0];
    s[3] = speed[1];
    s[4] = angle[0];
    s[5] = angle[1];

    timer18.Interval = (int)(Math.Abs((dis) / 145) * 1000);
    serialPort1.Write(s, 0, 6);
    timer18.Enabled = true;
}
```

تابع distance محاسبه می کند که رومبا چه مدت صبر کند و با چه سرعتی و چه مسافتی را طی کند. با اجرا این تابع رومبا در مدت زمان مشخص شروع به حرکت می کند و سپس متوقف می شود. حال برای چرخش ها و زوایای خاص رومبا به طریق زیر عمل می شود:



برای تعیین مقدار چرخش رومبا تابع spin تعریف می‌شود. در اینجا سرعت بر حسب میلی متر بر ثانیه است و شما به درجه بر ثانیه احتیاج دارید. برای تبدیل آن، شما احتیاج دارید بفهمید که رومبا چگونه در محیط دایره بر حسب میلی متر در هر درجه در حال حرکت است.

$$\text{mm/degree} = (258 \text{ mm} \times \pi) / 360 = 2.2515 \text{ mm/degree} \quad (\text{ب-۱})$$

برای به دست آوردن سرعت زاویه‌ای رومبا بر حسب میلی‌متر بر ثانیه از فرمول ب-۲ استفاده می‌شود.

$$\text{angular\_speed in degrees/s} = \text{speed in mm/s} / (\text{mm/degree}) \quad (\text{ب-۲})$$

حال برای اینکه مشخص شود رومبا چقدر باید صبر کند، تا به اندازه مورد نظر شما چرخش را انجام دهد، داریم:

$$\text{wait time} = (\text{mm/degree}) \times (\text{angle/speed}) \quad (\text{ب-۳})$$

برای مثال برای چرخش ۱۸۰ درجه با سرعت ۳۰۰ میلی متر بر ثانیه به اندازه ۱,۳۵ ثانیه زمان لازم

است. یعنی رومبا به اندازه ۱,۳۵ ثانیه باید صبر کند تا چرخش ۱۸۰ درجه را انجام دهد. برای این

منظور برنامه مورد نظر در سی‌شارپ به صورت زیر نوشته شده‌است:

```
private void spin (float angle)
{
    byte[] s = new byte[6];
    s[0] = 0x82;
    s[1] = 0x89;
    s[2] = 0x00;
    s[3] = 0x80;
    if (angle > 0)
    {
        s[4] = 0xff;
        s[5] = 0xff;
        timer17.Interval = (int)(Math.Abs(2.2515 * angle / 154) * 1000);
    }
    else
    {
        s[4] = 0x00;
        s[5] = 0x08;
        timer17.Interval = (int)(Math.Abs(2.2515 * angle / 140) * 1000);
    }
    serialPort1.Write(s, 0, 6);
    timer17.Enabled = true;
}
```

گام بعدی خواندن سنسورهای رومبا است. ربات رومبا در هر لحظه اطلاعات سنسورهای خود را از طریق پورت سریال در اختیار کاربر قرار می‌دهد. گروه‌های مختلفی از اطلاعات سنسور وجود دارد. سنسورهای فیزیکی: این سنسورها محیط اطراف را شناسایی می‌کنند که شامل سنسورهای ضربه‌گیر، صخره و دیوار می‌باشد (شامل ۱۰ بایت). کلیدهای موجود روی پنل رومبا و سنسورهای داخلی مسافت و زاویه (شامل ۶ بایت) است و سنسورهای توان رومبا شامل سیستم‌های باتری و شارژر آن می‌باشد. (شامل ۱۰ بایت) در شکل ب-۶ نحوه ارسال این اطلاعات نشان داده شده است.

byte 0	7	6	5	4	3	2	1	0
bump and wheelstop	n/a	n/a	n/a	wheel drop center	wheel drop left	wheel drop right	bump left	bump right
byte 1	7	6	5	4	3	2	1	0
wall	n/a	n/a	n/a	n/a	n/a	n/a	n/a	wall
byte 2	7	6	5	4	3	2	1	0
cliff left	n/a	n/a	n/a	n/a	n/a	n/a	n/a	cliff left
byte 3	7	6	5	4	3	2	1	0
cliff front left	n/a	n/a	n/a	n/a	n/a	n/a	n/a	cliff front left
byte 4	7	6	5	4	3	2	1	0
cliff front right	n/a	n/a	n/a	n/a	n/a	n/a	n/a	cliff front right
byte 5	7	6	5	4	3	2	1	0
cliff right	n/a	n/a	n/a	n/a	n/a	n/a	n/a	cliff right
byte 6	7	6	5	4	3	2	1	0
virtual wall	n/a	n/a	n/a	n/a	n/a	n/a	n/a	virtual wall
byte 7	7	6	5	4	3	2	1	0
motor drive currents	n/a	n/a	n/a	drive left	drive right	main brush	vacuum	side brush
byte 8	7	6	5	4	3	2	1	0
girt detector left	range 0-255							
byte 9	7	6	5	4	3	2	1	0
girt detector right	range 0-255							

Figure 2-5: ROI sensor packet group 1, physical sensors

byte 16	7	6	5	4	3	2	1	0
charging state	values: 0: not charging, 1: charging recovery, 2: charging, 3: idle charging, 4: no flag, 5: charging error							
byte 17	7	6	5	4	3	2	1	0
battery voltage (high byte)	range 0 to 65535 mV							
byte 18	7	6	5	4	3	2	1	0
battery voltage (low byte)	range 0 to 65535 mV							
byte 19	7	6	5	4	3	2	1	0
battery current (high byte)	range -32768 to 32767 mA							
byte 20	7	6	5	4	3	2	1	0
battery current (low byte)	range -32768 to 32767 mA							
byte 21	7	6	5	4	3	2	1	0
battery temp-sens	range -128 to 127 degrees Celsius							
byte 22	7	6	5	4	3	2	1	0
battery charge (high byte)	range 0 to 65535 mAh							
byte 23	7	6	5	4	3	2	1	0
battery charge (low byte)	range 0 to 65535 mAh							
byte 24	7	6	5	4	3	2	1	0
battery capacity (high byte)	range 0 to 65535 mAh							
byte 25	7	6	5	4	3	2	1	0
battery capacity (low byte)	range 0 to 65535 mAh							

Figure 2-7: ROI sensor packet group 3, power and charging sensors

byte 10	7	6	5	4	3	2	1	0
remote control	range 0-255							
byte 11	7	6	5	4	3	2	1	0
buttons	n/a	n/a	n/a	n/a	Power	Spot	Clean	Max
byte 12	7	6	5	4	3	2	1	0
distance (high byte)	range -32768 to 32767 mm							
byte 13	7	6	5	4	3	2	1	0
distance (low byte)	range -32768 to 32767 mm							
byte 14	7	6	5	4	3	2	1	0
angle (high byte)	range -32768 to 32767 mm							
byte 15	7	6	5	4	3	2	1	0
angle (low byte)	range -32768 to 32767 mm							

Figure 2-6: ROI sensor packet group 2, buttons and internal sensors

شکل ب-۵: بسته‌های اطلاعاتی سنسور رومبا

در برنامه سی شارپ با ارسال کد ۱۴۲ از رومبا درخواست می‌شود که بسته اطلاعات سنسور را در اختیار کاربر قرار دهد. با داشتن اطلاعات لازم برای کنترل رومبا از طریق نرم افزار سی شارپ می‌توان رومبا را در سطوح شیب‌دار و صاف به حرکت وا داشت و از طریق خواندن اطلاعات سنسوری که شامل مقدار جریان تلفاتی باتری و مقدار شارژ باتری می‌باشد به بررسی مسیر پیمایش شده پرداخت.

- [1]Schwartz J.T. and Sharir M. (1988). "A survey of motion planning and related geometric algorithms", *Artificial Intelligence*, pp 157-169.
- [2]Chang C.C. and Song K.-T. (1996). "Dynamic motion planning based on real-time obstacle prediction", *Robotics and Automation, Proceedings, IEEE International Conference on*, pp 2402-2407.
- [3]Chang Y.C. and Yamamoto Y., (2008). "On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments", *Industrial Robot: An International Journal*, 35, 5, pp 421-434.
- [4]Xie L.-J., Xie G.-R., Chen H.-W. and Li X.-L., (2008). "Solution to reinforcement learning problems with artificial potential field", *Journal of Central South University of Technology*, pp 552-557.
- [5]Barraquand J. and Ferbach P. (1994). "Path planning through variational dynamic programming", *Robotics and Automation, Proceedings., IEEE International Conference on*, pp 1839-184.
- [6]Lin H.-S., Xiao J. and Michalewicz Z. (1994). "Evolutionary algorithm for path planning in mobile robot environment", *Evolutionary Computation, IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp 211-216.
- [7]Borenstein J. and Koren Y. (1989). "Real-time obstacle avoidance for fast mobile robots", *Systems, Man and Cybernetics, IEEE Transactions on*, pp 1179-1187.
- [8]Sánchez-Hermosilla J., Rodríguez F., González R., Guzmán J.L. and Berenguel M. (2010). "A mechatronic description of an autonomous mobile robot for agricultural tasks in greenhouses", *Mobile Robots Navigation*, pp 583-608.
- [9]Wehn H.W. and Belanger P.R. (1997). "Ultrasound-based robot position estimation", *Robotics and Automation, IEEE Transactions*, pp 682-692.
- [10]Trzynadlowski A. (1988). "Energy optimization of a certain class of incremental motion DC drives", *Industrial Electronics, IEEE Transactions*, pp 60-66.

- [11]Barili A., Ceresa M. and Parisi C. (1995). "Energy-saving motion control for an autonomous mobile robot", *Industrial Electronics, Proceedings of the IEEE International Symposium on*, pp 674-676.
- [12]Bamdad M. (2013), "Time-energy optimal trajectory planning of cable-suspended manipulators", pp. 41-51, In: *Cable-Driven Parallel Robots*", Springer.
- [13]Dulger L. and Uyan S. (1997). "Modelling simulation and control of a four-bar mechanism with a brushless servo motor", *Mechatronics*, pp 369-383.
- [14]Sergaki E.S., Stavrakakis G.S. and Pouliezios A.D. (2002). "Optimal robot speed trajectory by minimization of the actuator motor electromechanical losses", *Journal of Intelligent and Robotic Systems*, pp 187-207.
- [15]Tal J., (1973). "The optimal design of incremental motion servo systems", *Proceedings of the 2nd Annual Symposium on Incremental Motion Control Systems and Devices*, pp 72-76 .
- [16]Dubins L.E. (1957). "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents", *American Journal of mathematics*, pp 497-516.
- [17]Reeds J. and Shepp L. (1990). "Optimal paths for a car that goes both forwards and backwards", *Pacific Journal of Mathematics*, pp 367-393.
- [18]Mohareri O., (2009). "Mobile robot trajectory tracking using neural networks a thesis in mechatronics", *American University of Sharjah*.
- [19]Park D.-H., Hoffmann H., Pastor P. and Schaal S. (2008). "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields", *Humanoid Robots, International Conference on*, pp 91-98.
- [20]Huang W.H., Fajen B.R., Fink J.R. and Warren W.H. (2006). "Visual navigation and obstacle avoidance using a steering potential function", *Robotics and Autonomous Systems*, pp 288-299.
- [21]Khatib O., (1986). "Real-time obstacle avoidance for manipulators and mobile robots", *The international journal of robotics research*, pp 90-98.

- [22]Krogh B.H. and Thorpe C.E. (1986). "Integrated path planning and dynamic steering control for autonomous vehicles", Robotics and Automation, IEEE International Conference on, pp 1664-1669.
- [23]Suh S.-H. and Shin K.G. (1988). "A variational dynamic programming approach to robot-path planning with a distance-safety criterion", Robotics and Automation, IEEE Journal of, pp 334-349.
- [24]Amato N., (2004). "Potential field methods Randomized Motion Planning", University of Padova, pp 23-28.
- [25]Hong Z., Liu Y., Zhongguo G. and Yi C. (2011). "The dynamic path planning research for mobile robot based on artificial potential field", Consumer Electronics, Communications and Networks (CECNet), International Conference on, pp 2736-2739.
- [26]Li G., Tamura Y., Yamashita A. and Asama H. (2013). "Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning", International Journal of Mechatronics and Automation, pp 141-170.
- [27]Saska M., Macas M., Preucil L. and Lhotská L. (2006). "Robot path planning using particle swarm optimization of Ferguson splines", Emerging Technologies and Factory Automation, IEEE Conference on, pp 833-839.
- [28]Singh V., Singh D., Tiwari R. and Shukla A. (2012). "Path Planning Using Particle Swarm Optimization with Linear Crossover Operator", Advances in Mathematical and Computational Methods, pp 533-544.
- [29]Andersson F. and Kvernes B. (2003). "Bezier and B-spline technology", Umea university Sweden .
- [30]Iglesias A., Echevarría G .and Gálvez A. (2004). "Functional networks for B-spline surface reconstruction", Future Generation Computer Systems, pp 1337-1353.
- [31]Engelbrecht A.P. (2007). "Computational intelligence: an introduction", Reprinted from: John Wiley & Sons.

- [32]Yu S., Zhu K. and Zhang X. (2012). "Energy demand projection of China using a path-coefficient analysis and PSO-GA approach", *Energy Conversion and Management*, pp 142-153.
- [33]Makimoto T. and Sakai Y. (2003). "Evolution of low power electronics and its future applications", *Proceedings of the international symposium on Low power electronics and design*, pp 2-5.
- [34]Gracia L. and Tornero J. (2008). "Optimal trajectory planning for wheeled mobile robots based on kinematics singularity", *Journal of Intelligent and Robotic Systems*, pp 145-168.
- [35]Kim H. and Kim B.-K. (2008). "Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots", *Control Automation, Robotics and Vision, 10th International Conference on*, pp 1730-1735.
- [36]Kurt T.E. (2006). "Hacking Roomba: Extreme Tech", Vol. 48 ,Reprinted from: John Wiley & Sons.
- [37]Salzberger C., (2011). "MATLAB Simulator for the iRobot create", Cornell University.
- [38]Mei Y., Lu Y.-H., Hu Y.C. and Lee C.G. (2005). "A case study of mobile robot's energy consumption and conservation techniques", *Advanced Robotics, ICAR'05. Proceedings, 12th International Conference on*, pp 492-497.
- [39]Beheshti Z. and Shamsudding S. (2013). "A review of population-based meta-heuristic algorithms", *Int. J. Adv. Soft Comput. Appl*, pp 1-35.





## **Abstract**

For any mobile device, the ability to navigate in its environment is important. Avoiding dangerous situations such as collisions and unsafe conditions provide a trajectory optimization problem which is the process of designing a trajectory that minimizes or maximizes some measure of performance within prescribed constraint boundaries. In particular, a procedure for computing optimal trajectories in presence of obstacles is proposed.

This thesis addresses the problem of planning collision free paths for mobile robots and different evolutionary techniques are used to solve optimization problems. Moreover including obstacle avoidance, a gradient of a potential field centered on the obstacle is considered based on potential field algorithm.

This optimization problem can accommodate such effects as terrain, robot dynamics. In order to investigate energy efficiency, the new problem is developed. An energy object function is minimized in order to extend run-time of battery-powered on trajectory. The energy is dissipated mostly in the motors, which strongly depends on the velocity profile. The algorithms are presented for wheeled mobile robot trajectory generation that achieves a high degree of generality and efficiency. Experimental tests have been also carried out in order to validate the effectiveness of the proposed procedure. The reporting results are obtained by gathering movement data from an empirical study of iRobot's Roomba™. The efforts are focused on improvements to a research robotic system by the sensors available on the Roomba. Graphical simulator is designed for the detail necessary in the algorithms.

Simulation and experiments are performed, and compared to the results presented. They show good performance and ability to avoid the local minimum problem in most of the cases.

**Keywords: Navigation, Optimization, Potential field algorithm, Robot dynamics, Energy, Wheeled mobile robot, Velocity profile, Roomba robot, Graphical simulator.**



**Shahrood University of Technology**  
**Faculty of mechanic, mechatronic Engineering**

**M.S Thesis**

**Dynamic trajectory planning and navigation implementation for a  
mobile robot in an unknown environment with the presence of  
obstacles**

**Mahdi Nazari Sarcheshmeh**

**Supervisor:**

**Dr.Mahdi bamdad**

**January 2015**