



اجتناب از برخورد مانع ربات دو چرخ متحرک به وسیله کنترل پیش‌بین مدل و کنترل بهینه

گزارش پروژه دینامیک پیشرفته

استاد درس:

دکتر حبیب نژاد

اعضای گروه:

فاطمه نامدار

علی صافی

حجت مهدی سوزنی

تابستان ۱۳۹۸

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فهرست مطالب

۱	فصل ۱: مقدمه
۲	۱-۱- مقدمه
۳	۲-۱- شرح موضوع
۵	۳-۱- شرح سرفصلها
۶	فصل ۲: دینامیک ربات
۷	۲-۱- معرفی پارامترها و متغیرهای حالت سیستم
۸	۲-۲- قیود حاکم بر سیستم
۹	۲-۳- معادلات حاکم بر سیستم
۱۳	۲-۴- صحتسنجی معادلات دینامیکی
۱۴	۲-۵- شیوههای اعمال لغزش در طراحی کنترلکنندهها
۱۵	فصل ۳: مسیریابی
۱۶	۳-۱- برنامه‌ریزی حرکت و مسیریابی
۱۸	۳-۱-۲- روش تابع پتانسیل
۲۰	فصل ۴: طراحی کنترل‌کننده فیدبک حالت بهینه
۲۱	۴-۱- کنترل‌پذیری
۲۲	۴-۲- مشاهده‌پذیری
۲۳	۴-۳- طراحی کنترل‌کننده
۲۷	فصل ۵: طراحی کنترل‌کننده پیشبین مدل
۲۸	۵-۱- معرفی کنترل‌کننده پیشبین مدل
۳۱	۵-۲- روش خطی سازی استفاده‌شده
۳۲	۵-۳- پیاده‌سازی کنترل PFC

۴-۵- نکته مهم در رابطه با افق پیشبینی و افق کنترل ۳۳

فصل ۶: شبیه‌سازی و تحلیل نتایج ۳۵

۶-۱- مقدمه ۳۶

۶-۲- نقشه مشخص شده برای ربات ۳۶

۶-۳- نتایج مربوط به کنترل سیستم با در نظر گرفتن لغزش اغتشاشی ۳۸

۶-۴- نتایج مربوط به کنترل سیستم با لغزش به صورت متغیر حالت ۴۱

۶-۵- نتایج کنترل ربات با دید محدود ۴۶

فصل ۷: جمع‌بندی و پیشنهادها ۵۰

۷-۱- مقدمه ۵۱

۷-۲- جمع‌بندی ۵۲

۷-۳- نوآوری ۵۲

مراجع ۵۴

فهرست اشکال

- شکل (۱-۱) نمونه‌ای از ربات متحرک دوچرخ ۳
- شکل (۱-۲) معرفی متغیرهای حالت سیستم ۷
- شکل (۲-۲) مسیر حرکت ربات در سناریو فرضی ۱۳
- شکل (۳-۲) لغزش اعمالی به زاویه ربات برحسب زمان ۱۴
- شکل (۲-۳) الگوریتمهای مسیریابی ۱۷
- شکل (۱-۴) ربات در دستگاه مختصات ۲۳
- شکل (۲-۴) بلوک دیاگرام سیستم کنترلی ۲۵
- شکل (۳-۴) بلوک دیاگرام سیستم کنترلی به همراه اغتشاش ۲۶
- شکل (۱-۵) بلوک دیاگرام سیستم کنترلی به همراه اغتشاش ۳۲
- شکل (۵-۲) نحوه تغییر افقهای پیشبینی و کنترل برحسب زمان ۳۴
- شکل (۱-۶) پردازش مسیر به دست آمده ۳۷
- شکل (۲-۶) پیدا کردن مسیرهای متفاوت ۳۷
- شکل (۳-۶) متغیرهای حالت سیستم برای کنترل کننده LQR ۳۹
- شکل (۴-۶) مسیر کنترل شده با در نظر گرفتن لغزش به صورت اغتشاش ۴۰
- شکل (۵-۶) نمودار مختصات مرکز جرم ربات برحسب زمان ۴۰
- شکل (۶-۶) نمودار مسیر حرکت ربات در سرعت ۱۵ سانتیمتر بر ثانیه ۴۱
- شکل (۷-۶) نمودار مختصات مرکز جرم ربات برحسب زمان در سرعت ۱۵ سانتیمتر بر ثانیه ۴۲
- شکل (۸-۶) نمودار متغیر لغزش طولی ربات برحسب زمان در سرعت ۱۵ سانتیمتر بر ثانیه ۴۲
- شکل (۹-۶) نمودار متغیر لغزش عرضی ربات برحسب زمان در سرعت ۱۵ سانتیمتر بر ثانیه ۴۳
- شکل (۱۰-۶) نمودار مسیر حرکت ربات در سرعت ۳ سانتیمتر بر ثانیه ۴۴
- شکل (۱۱-۶) نمودار مختصات مرکز جرم ربات برحسب زمان در سرعت ۳ سانتیمتر بر ثانیه ۴۴
- شکل (۱۲-۶) نمودار متغیر لغزش طولی ربات برحسب زمان در سرعت ۳ سانتیمتر بر ثانیه ۴۵

- شکل (۶-۱۳) نمودار متغیر لغزش عرضی ربات بر حسب زمان در سرعت ۳ سانتیمتر بر ثانیه ۴۵
- شکل (۶-۱۴) مسیر ربات بدون اطلاع از مانع ۴۷
- شکل (۶-۱۵) کنترل ربات در حضور مانع استاتیکی ۴۸
- شکل (۶-۱۶) خروجیهای کنترل کننده ۴۹
- شکل (۶-۱۷) نرخ ورودیهای کنترل کننده ۴۹
- شکل (۷-۱) روند و مراحل انجام پروژه ۵۱

فهرست جداول

جدول (۱-۵) مقایسه روشهای مختلف کنترل پیشبین ۳۰

فهرست علائم اختصاری

v	سرعت خطی ربات
ω	سرعت دورانی
$A(q)$	ماتریس قیود غیر هلونومیک
S	صفر فضای ماتریس قیود
$q(t)$	مختصات تعمیم یافته
r	شعاع چرخ
M	ماتریس اینرسی
$c(q(t), \dot{q}(t))$	ماتریس کوریولیس و جانب مرکز
$\tau(t)$	گشتاورهای ورودی
E, λ	ضرایب لاگرانژ
m_w	جرم چرخ
m_b	جرم پایه
I_{zzb}	ممان اینرسی پایه
I_{yyw}	ممان اینرسی چرخ
I_{zzw}	ممان اینرسی چرخ حول محور عبوری از قطر آن
ϕ	زاویه ربات
$\dot{\theta}_r$	سرعت زاویه‌ای چرخ سمت راست
$\dot{\theta}_l$	سرعت زاویه‌ای چرخ سمت چپ
$\rho(q)$	فاصله تا مانع
ρ_0	شعاع ایمنی
K_{rep}	ضریب مقیاس مانع
K_{att}	ضریب مقیاس هدف

ρ_{goal}	فاصله اقلیدسی تا هدف
F_{att}	نیرو جاذبه
F_{rep}	نیرو دافعه
x	حالت‌های سیستم
y	خروجی سیستم
u	ورودی کنترلر
A,B,C	ماتریس‌های حالت
J	تابع هزینه
Q	وزن خروجی‌ها
R	وزن ورودی‌ها

فصل ۱:

مقدمه

ربات‌های متحرک^۱ به ربات‌هایی اطلاق می‌شود که قابلیت حرکت در محیط اطراف خود را دارند. این دسته ربات‌ها در مقابل ربات‌های صنعتی^۲ قرار گرفته، که معمولاً ثابت بوده و دارای بازوهای مفصل دار متصل به یک سطح ثابت می‌باشند. ربات‌های متحرک به دلیل نقش مهمی که در صنعت و زندگی انسان داشته‌اند همواره مورد توجه محققان بوده‌اند.

ربات‌های متحرک با توجه به معیارهای زیر دسته‌بندی می‌شوند:

• محیطی که در آن حرکت می‌کنند:

- بر روی زمین، مانند ربات‌های خانگی^۳.
- ربات‌های حمل و نقل که می‌توانند مواد را در محیط‌های کاری جابه‌جا کنند
- ربات‌های هوایی^۴ (پرنده‌های بدون سرنشین)
- ربات‌های زیر آب^۵
- ربات‌های قطبی، که طراحی شده تا در محیط‌های یخ‌زده حرکت کنند

• دستگاهی که برای حرکت خود استفاده می‌کنند:

- ربات‌های دارای پا: پاهای شبیه انسان یا شبیه حیوانات
- ربات‌های چرخ‌دار^۶
- ربات‌های ریل‌دار^۷

¹ Mobile Robots

² Industrial Robots

³ Unmanned Ground Vehicles (UGVs)

⁴ Unmanned Aerial Vehicles (UAVs)

⁵ Autonomous Underwater Vehicles (AUVs)

⁶ Wheeled Robots

⁷ Tracked Robots

در این پروژه، ربات‌های متحرک از نوع چرخ‌دار مورد بررسی قرار گرفته است، که نمونه‌ای از آن در شکل (۱) دیده می‌شود.



شکل (۱-۱) نمونه‌ای از ربات متحرک دوچرخ

۱-۲- شرح موضوع

موضوع این پروژه طراحی کنترل‌کننده برای یک ربات متحرک دوچرخ به منظور عبور ایمن از موانع استاتیکی موجود در نقشه مشخص می‌باشد. این کنترل‌کننده به دو صورت آنلاین و آفلاین طراحی شده، بدین معنی که در دو حالت مشخص بودن موانع و تعیین مسیر ربات قبل از حرکت و همچنین تعیین مسیر به کمک حس‌گر متصل به ربات به صورت آنلاین پیاده‌سازی شده است.

بحث مهم مورد بررسی در این پروژه، تأثیر لغزش بر عملکرد سیستم و کنترل سیستم با وجود لغزش می‌باشد. لغزش به دو صورت در این پروژه در نظر گرفته شده است:

الف) به صورت یک اغتشاش که به زاویه قرارگیری ربات وارد شده و با گذشت زمان به سمت صفر میل می‌کند

ب) به صورت متغیر حالت سیستم؛ که در این مورد، سه پارامتر لغزش، دو پارامتر طولی برای هر چرخ و یک پارامتر عرضی برای کل ربات، در نظر گرفته شده است.

کنترل کننده استفاده شده به منظور کنترل ربات در مسیر مشخص شده، به دو صورت کلی کنترل پیش بین مدل^۱ از نوع PFC و همچنین کنترل فیدبک حالت خطی مرتبه دو^۲ انجام شده و نتایج مربوطه با یکدیگر مقایسه شده اند.

سیستم ربات متحرک دوچرخ یک سیستم چند ورودی چند خروجی^۳ بوده که ورودی های آن، گشتاور ورودی به هر چرخ توسط موتور الکتریکی مجزای هر چرخ بوده و خروجی های آن، مختصات مرکز جرم در صفحه می باشند. هر دو نوع کنترل کننده استفاده شده در این پروژه از نوع کنترل کننده های مدرن می باشند. مزیت عمده تحلیل و طراحی سیستم های کنترل مدرن نسبت به روش های کلاسیک، کاربرد آن در سیستم های چند ورودی و چند خروجی می باشد. همچنین هر دو کنترل کننده استفاده شده، کنترل کننده سیستم های خطی بوده و بر روی این سیستم ها پیاده سازی می شوند. به همین دلیل سیستم دینامیکی ابتدا خطی سازی شده و پس از محاسبه ورودی کنترلی، این ورودی به سیستم غیر خطی اعمال شده تا از صحت عملکرد مناسب سیستم اطمینان حاصل شده و نیازی به طراحی کنترل های پیچیده و پرهزینه غیر خطی نباشد.

مسیر حرکت ربات باید باهدف اجتناب از برخورد با موانع استاتیکی طراحی شود. روش های مسیریابی به منظور اجتناب از برخورد با موانع به دوشاخه کلاسیک و اکتشافی تقسیم می شوند. در این پروژه، الگوریتم مسیریابی از نوع استفاده از تابع پتانسیل بوده، که مؤثرترین روش در حوزه روش های کلاسیک می باشد. توضیحات کامل تر در این زمینه در فصل مسیریابی به وضوح تشریح شده است.

¹ Model Predictive Control (MPC)

² Linear Quadratic Regulator (LQR)

³ Multi-Input-Multi-Output (MIMO)

۱-۳- شرح سرفصل‌ها

این گزارش در ۷ فصل تنظیم شده است. فصل اول شامل مقدمه‌ای بر انواع ربات‌های متحرک و همچنین شرح موضوع پروژه می‌باشد. فصل دوم به توضیح درباره دینامیک سیستم و معادلات دینامیکی حاکم بر آن و همچنین روش خطی‌سازی این معادلات می‌پردازد. در فصل سوم، الگوریتم مسیریابی به کمک تابع پتانسیل تشریح و برنامه کامپیوتری نوشته شده در این بخش توضیح داده شده است. فصل چهارم، به نحوه طراحی و پیاده‌سازی کنترل کننده LQR به سیستم خطی‌سازی شده برای کنترل ربات در مسیری مشخص برای عبور از موانع با اعمال لغزش به صورت یک اغتشاش، پرداخته شده است. فصل پنجم درباره چگونگی طراحی کنترل کننده پیش‌بین مدل برای سیستم خطی شده در دو حالت در نظر گرفتن لغزش به عنوان یک متغیر حالت سیستم و همچنین به صورت اغتشاش را شرح داده است. فصل ششم، نتایج به دست آمده در فصل‌های چهارم و پنجم را با یکدیگر مقایسه کرده است. و در نهایت در فصل هفتم، کنترل کننده پیش‌بین مدل با فرض وجود حس گر بر روی ربات به صورت آنلاین تشریح شده است.

فصل ۲:

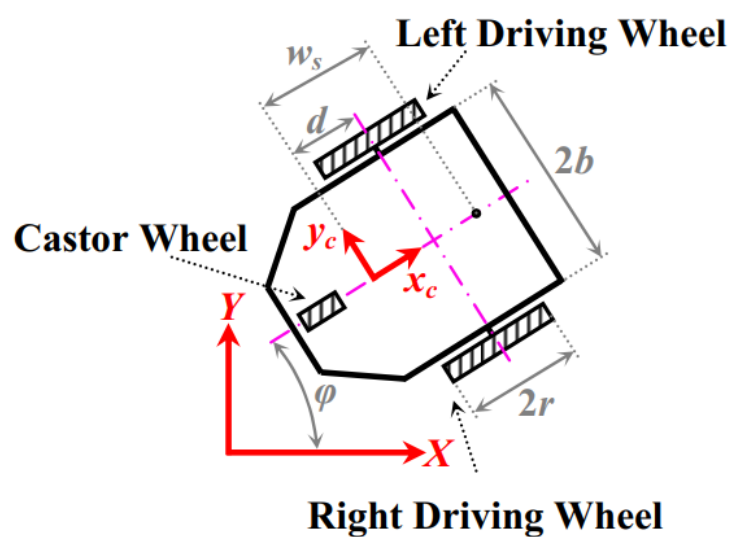
دینامیک ربات

۱-۲- معرفی پارامترها و متغیرهای حالت سیستم

ربات موردبررسی در این پروژه، دارای دو موتور مجزا از یکدیگر، در هر یک از چرخ‌های خود می‌باشد، که گشتاور ورودی به آن‌ها به‌عنوان ورودی سیستم کنترلی در نظر گرفته شده است. همچنین حرکت ربات در صفحه افقی در نظر گرفته شده و نسبت به نقطه‌ای مرجع، با X و Y مشخص شده است. بنابراین در حالت ایده‌آل بدون لغزش، سیستم دارای ۷ متغیر حالت زیر می‌باشد:

$$q = (x_c, y_c, \varphi, \theta_1, \theta_2)^T, \dot{q} = (\dot{\theta}_1, \dot{\theta}_2)^T$$

که در آن‌ها، x_c و y_c موقعیت مرکز جرم ربات، φ زاویه ربات با محور X و θ_1 و θ_2 زاویه چرخ‌های چپ و راست هستند.



شکل (۱-۲) معرفی متغیرهای حالت سیستم

به‌علاوه، در نوشتن معادلات، جرم ربات بدون چرخ‌ها و موتورها با m_c ، جرم چرخ‌ها و موتورها با m_w ، ممان اینرسی ربات بدون چرخ‌ها و موتورها حول محور عمود بر صفحه گذرنده از مرکز جرم با I_c ، ممان اینرسی هر چرخ و موتور حول محور چرخ با I_w و ممان اینرسی هر چرخ و موتور حول قطر چرخ با I_m نشان داده شده است. پارامترهای ابعادی در شکل ۱-۲ نشان داده شده‌اند.

۲-۲- قیود حاکم بر سیستم

به‌طور کلی، سه قید بر سینماتیک حرکت سیستم حاکم است. اولین قید این است که ربات بایستی در جهت محور تقارن حرکت کند، یعنی:

$$\dot{y}_c \cos \varphi - \dot{x}_c \sin \varphi - d\dot{\varphi} = 0 \quad (۱-۲)$$

دو قید دیگر، قیود مربوط به غلتش بوده، یعنی چرخ‌ها سر نخورند:

$$\dot{x}_c \cos \varphi + \dot{y}_c \sin \varphi + b\dot{\varphi} = r\dot{\theta}_1 \quad (۲-۲)$$

$$\dot{x}_c \cos \varphi + \dot{y}_c \sin \varphi - b\dot{\varphi} = r\dot{\theta}_2 \quad (۳-۲)$$

این قیود می‌توانند به شکل ماتریسی زیر نوشته شوند:

$$A^T(q)\dot{q} = 0 \quad (۴-۲)$$

که در آن:

$$A(q) = \begin{bmatrix} -\sin \varphi & \cos \varphi & -d & 0 & 0 \\ -\cos \varphi & -\sin \varphi & -b & r & 0 \\ -\cos \varphi & -\sin \varphi & b & 0 & r \end{bmatrix}$$

با در نظر گرفتن ماتریس S به‌عنوان ماتریس صفر فضای A^T ، خواهیم داشت:

$$A^T(q)S(q) = 0 \quad (۵-۲)$$

با مقایسه روابط (۲-۴) و (۲-۵) نتیجه می‌شود که بردار $\dot{q}$ و ماتریس $S(q)$ در یک فضا قرار داشته و برداری مثل η وجود خواهد داشت که:

$$\dot{q} = S(q)\eta \quad (۶-۲)$$

که با مقایسه این معادله با قیود سینماتیکی سیستم، بردار η به صورت تعریف شده در بخش ۱-۲ درخواهد آمد.

معادله (۶-۲)، مدل فضای حالت سینماتیکی سیستم مورد بررسی در حالت ایده آل می باشد، که در آن:

$$S(q) = \begin{bmatrix} \frac{r}{2b}(b \cos \varphi - d \sin \varphi) & \frac{r}{2b}(b \cos \varphi + d \sin \varphi) \\ \frac{r}{2b}(b \sin \varphi + d \cos \varphi) & \frac{r}{2b}(b \sin \varphi - d \cos \varphi) \\ \frac{r}{2b} & -\frac{r}{2b} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

۳-۲- معادلات حاکم بر سیستم

معادلات لاگرانژ سیستم مکانیکی مورد بررسی دارای شکل کلی زیر می باشد:

$$M(q)\ddot{q} + C(q, \dot{q}) = F + B(q)u \quad (۷-۲)$$

که در آن، M ماتریس اینرسی سیستم، C ماتریس شامل شتابهای کوریولیس و جانب مرکز، u بردار گشتاورهای ورودی سیستم، B ماتریسی رتبه کامل و F نیروهای تعمیم یافته سیستم بوده که در اثر قیود و اطمینان از برآورده شدن آنها به سیستم وارد می گردد.

توان نیروهای تعمیم یافته وارد شده به سیستم در تمام حرکت های تحت قیود سینماتیکی حاکم بر آن برابر با صفر بوده، یعنی:

$$F^T \dot{q} = 0 \quad (۸-۲)$$

این بدین معنی است که برداری به نام ضرایب لاگرانژ وجود دارد، به طوری که:

$$F = A(q)\lambda \quad (۹-۲)$$

با ضرب طرفین معادله (۷-۲) در $S^T(q)$ خواهیم داشت:

$$S^T(q)M(q)\ddot{q} = S^T(q)[C(q, \dot{q}) + B(q)u] \quad (۱۰-۲)$$

معادله (۲-۱۰) در کنار معادله (۲-۶)، معادلات دینامیکی سیستم در حالت ایده‌آل هستند.

اما در شرایط واقعی که بین چرخ‌ها و زمین لغزش وجود دارد و فرض غلتش خالص برقرار نمی‌باشد، دیگر نمی‌توان توان نیروهای تعمیم‌یافته وارد شده به سیستم در اثر قیود سینماتیکی را برابر صفر در نظر گرفت. بنابراین به علت وجود لغزش، دیگر قیود سینماتیکی سیستم به‌طور کامل برقرار نخواهند بود، بنابراین در معادله (۲-۴):

$$A^T(q)\dot{q} \neq 0$$

پس بردار $\dot{q}$ و ماتریس $S(q)$ در یک فضا قرار نداشته و معادله (۲-۶) باید به‌صورت زیر بازنویسی شود:

$$\dot{q} = S(q)\eta + A^T(q)\varepsilon\mu = [S(q) A(q)] \begin{pmatrix} \eta \\ \varepsilon\mu \end{pmatrix} \quad (۱۱-۲)$$

که در آن μ یک بردار ۳ عضوی (به تعداد قیود سینماتیکی) شامل متغیرهای حالت مربوط به لغزش و ε یک فاکتور مقیاس بوده که میزان تخطی از قیود سینماتیکی را نشان می‌دهد. به‌علاوه، تعاریف متغیرهای حالت سیستم نیز به شکل زیر تغییر خواهد یافت:

$$q = (x, y, b\varphi, r\theta_1, r\theta_2)^T, \eta = (\dot{\theta}_1, \dot{\theta}_2)^T, \mu = (v_{s1}, v_{s2}, v_{s3})^T$$

با مشتق‌گیری از رابطه (۲-۱۱) و جایگذاری در (۲-۷) خواهیم داشت:

$$\begin{cases} \dot{q} = S(q)\eta + A^T(q)\varepsilon\mu \\ \begin{pmatrix} \dot{\eta} \\ \varepsilon\dot{\mu} \end{pmatrix} = P(q, \eta, \varepsilon\mu) + R(q)u \end{cases} \quad (۱۲-۲)$$

که معادلات فضای حالت سیستم در حالت واقعی (حضور لغزش) هستند. ماتریس‌های P و R برابرند با:

$$P(q, \eta, \varepsilon\mu) = [S(q) A^T(q)]^{-1} \left(-(\dot{S}(q)\eta + \dot{A}^T(q)\varepsilon\mu) - M^{-1}(q)C(q, S(q)\eta + A^T(q)\varepsilon\mu) \right) \quad (۱۳-۲)$$

$$+ H^{-1}(q) \begin{pmatrix} 0 \\ A(q)F \end{pmatrix}$$

$$R(q) = \begin{pmatrix} R_1(q) \\ R_2(q) \end{pmatrix} = [S(q) A^T(q)]^{-1} M^{-1}(q) B(q) \quad (۱۴-۲)$$

که در آنها:

$$H(q) = \begin{bmatrix} S^T(q)M(q)S(q) & S^T(q)M(q)A^T(q) \\ A(q)M(q)S(q) & A(q)M(q)A^T(q) \end{bmatrix} \quad (۱۵-۲)$$

$$\dot{S}(q) = \frac{\partial S}{\partial q}(S(q)\eta + A^T(q)\varepsilon\mu) \quad (۱۶-۲)$$

$$\dot{A}^T(q) = \frac{\partial A^T}{\partial q}(S(q)\eta + A^T(q)\varepsilon\mu) \quad (۱۷-۲)$$

ماتریس F با فرض مدل "شبه لغزش"، که در آن نیروهای عرضی و طولی وارد شده به چرخ توسط زمین، به ترتیب متناسب با زاویه لغزش و ضریب لغزش می‌باشند. طبق این مدل، نیروی F به صورت زیر محاسبه می‌شود:

$$F = -A^T(q) \sum_{i=1}^2 \left(\frac{1}{v_i} L_i^T \begin{bmatrix} C & 0 \\ 0 & D \end{bmatrix} L_i \right) A(q) \dot{q} \quad (۱۸-۲)$$

که در آن C و D ضرایب ثابت مربوط به جنس و نوع چرخ و:

$$\begin{cases} v_1 = \sqrt{v_{x1}^2 + v_y^2} = \sqrt{(v_{s2} + r\dot{\theta}_1)^2 + v_{s1}^2} \\ v_2 = \sqrt{v_{x2}^2 + v_y^2} = \sqrt{(v_{s3} + r\dot{\theta}_2)^2 + v_{s1}^2} \end{cases} \quad (۱۹-۲)$$

$$L_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \cdot L_2 = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \quad (۲۰-۲)$$

که در آنها v_{s1} مربوط به لغزش عرضی کل ربات، و v_{s2} و v_{s3} مربوط به لغزش طولی هر کدام از چرخ‌ها می‌باشد، که این سه، بردار μ تشکیل می‌دهند.

همچنین ماتریس‌های M و C در معادلات دینامیکی برابر با مقادیر زیر هستند:

$M(q)$

$$= \begin{bmatrix} m & 0 & -m_c d \sin \varphi & 0 & 0 \\ 0 & m & m_c d \cos \varphi & 0 & 0 \\ -m_c d \sin \varphi & m_c d \cos \varphi & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \cdot \begin{cases} m = m_c + 2m_w \\ I = I_c + 2m_w(d^2 + b^2) + 2I_m \end{cases}$$

$$C = \begin{bmatrix} -m_c d \dot{\varphi}^2 \cos \varphi \\ -m_c d \dot{\varphi}^2 \sin \varphi \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -m_c d \left(\left(\frac{1}{r} \right) (S(3,:) \eta + A^T(3,:) \varepsilon \mu) \right)^2 \cos \varphi \\ -m_c d \left(\left(\frac{1}{r} \right) (S(3,:) \eta + A^T(3,:) \varepsilon \mu) \right)^2 \sin \varphi \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

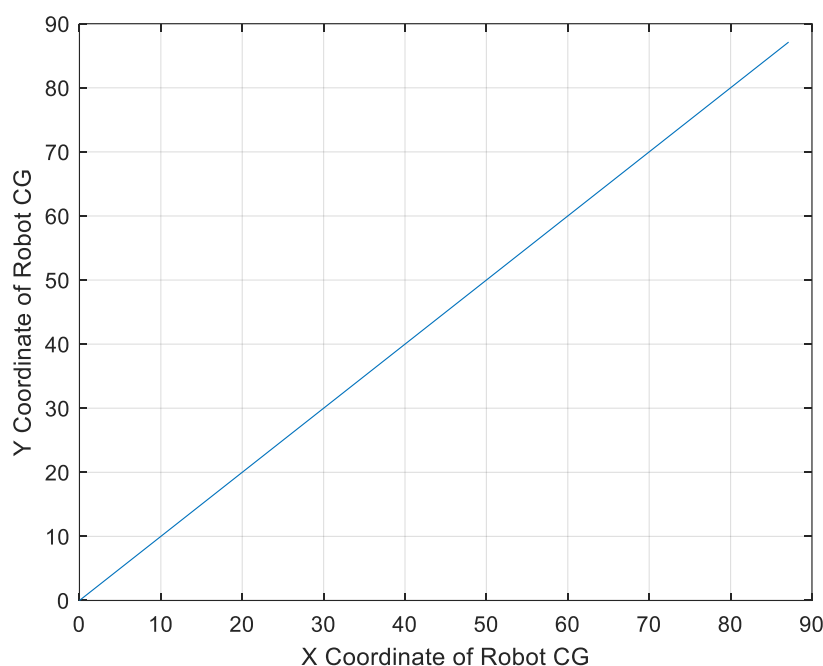
به‌طور خلاصه، در حالت عدم وجود لغزش، معادلات سیستم به‌صورت ۷ درجه آزادی بوده و با معادلات (۱۰-۲) و (۲-۶) مشخص می‌گردد. در حالت وجود لغزش، معادلات دینامیکی سیستم ۱۰ درجه آزادی بوده که با معادلات (۲-۱۲) بیان می‌شود.

بنابراین با داشتن معادلات دینامیکی حامل بر سیستم، برای حل آن بایستی از انتگرال‌گیری عددی استفاده می‌شد. بدین منظور، یک برنامه کامپیوتری در نرم‌افزار MATLAB نوشته شد که برای اعمال شرایط اولیه و ورودی مشخص، معادلات را با روش Runge-Kutta با طول گام متغیر حل می‌کرد و متغیرهای حالت را به‌عنوان خروجی تحویل می‌داد. این برنامه با نام "Nonlinear_system_solver_with_slip" برای حالت با لغزش و "Nonlinear_system_solver" برای حالت بدون لغزش ذخیره‌شده که از فایلی به نام "odefunc" برای حل معادلات استفاده می‌کنند. این کدها در فایل پیوست قرار داده‌شده‌اند.

۴-۲- صحت‌سنجی معادلات دینامیکی

صحت‌سنجی معادلات دینامیکی با طراحی یک سناریو فرضی انجام شد، که در آن ورودی‌های سیستم به نحوی وارد شدند که رفتار حرکت ربات قابل پیش‌بینی باشد، و سپس نتایج حاصل از انتگرال‌گیری عددی معادلات تفسیر شدند.

این سناریو فرضی بدین گونه است: اگر ربات در ابتدای حرکت در مرکز مختصات با زاویه ۴۵ درجه نسبت به محور X ها قرار داشته باشد و دو موتور قرار گرفته در چرخ‌ها، گشتاور دقیقاً مشابهی به چرخ‌ها وارد کند، ربات بایستی در زمان‌های آینده در مسیری مستقیم که با محور X ها قرار دارد حرکت نماید. نمودارهای مربوط به این سناریو در ادامه آمده است.



شکل (۲-۲) مسیر حرکت ربات در سناریو فرضی

همان‌طور که در شکل (۲-۲) دیده می‌شود، ربات مسیری راست و بدون انحراف را طی کرده که با زاویه ۴۵ درجه نسبت به محور X ها قرار دارد. بنابراین انتگرال‌گیر پاسخ‌های منطقی از حل سیستم دینامیکی ارائه می‌کند.

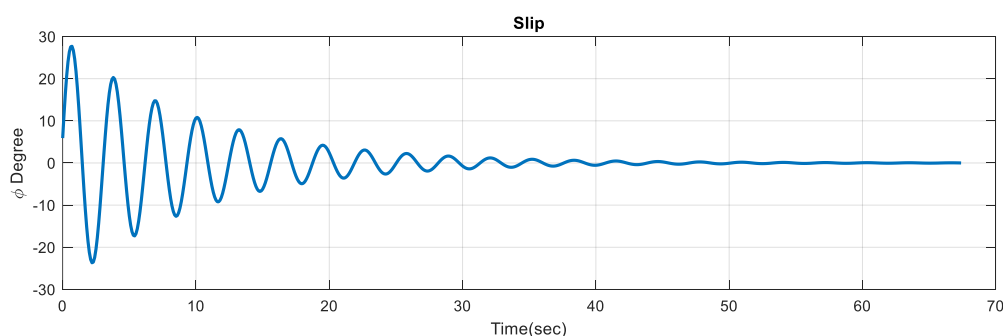
۵-۲- شیوه‌های اعمال لغزش در طراحی کنترل‌کننده‌ها

در طراحی کنترل‌کننده‌هایی که برای قرار دادن ربات در مسیر موردنظر طراحی شده‌اند، نیاز بود که لغزش به‌صورت دینامیکی و متغیر با زمان برای سیستم دینامیکی کنترل شونده در نظر گرفته شود، تا به شرایط واقعی نزدیک‌تر باشد. بدین منظور، لغزش به دو شکل در سیستم دینامیکی و سیستم کنترلی در نظر گرفته شد:

الف) لغزش به‌عنوان متغیر حالت سیستم: در این روش، دو لغزش طولی و یک لغزش عرضی به‌عنوان متغیر حالت سیستم، با فرض مدل "شبه لغزش" در معادلات حرکت سیستم وارد شد که توسط معادلات (۱۲-۲) بیان می‌شود. بنابراین با اعمال سیگنال کنترلی، که گشتاور دو موتور قرار گرفته بر هر چرخ می‌باشد، حل‌گر معادلات دینامیکی به‌طور مستقیم متغیرهای مربوط به لغزش را از انتگرال‌گیری به دست می‌آورد.

ب) لغزش به‌عنوان اغتشاش: در این حالت، اثر لغزش به‌صورت اغتشاشی میل‌کننده به صفر به سیستم دینامیکی وارد می‌شود. این لغزش به متغیر φ وارد شده، که تعبیر فیزیکی آن، ایجاد نوسان در این متغیر در اثر لغزش‌های طولی و عرضی می‌باشد، که به واقعیت نزدیک است. معادله تابع اغتشاشی وارد شده به‌صورت زیر می‌باشد:

$$\varphi_{slip} = \frac{\pi}{6} e^{-\frac{t}{10}} \sin 2t$$



شکل (۲-۳) لغزش اعمالی به زاویه ربات برحسب زمان

فصل ۳:

مسیریابی

۳-۱- برنامه‌ریزی حرکت و مسیریابی

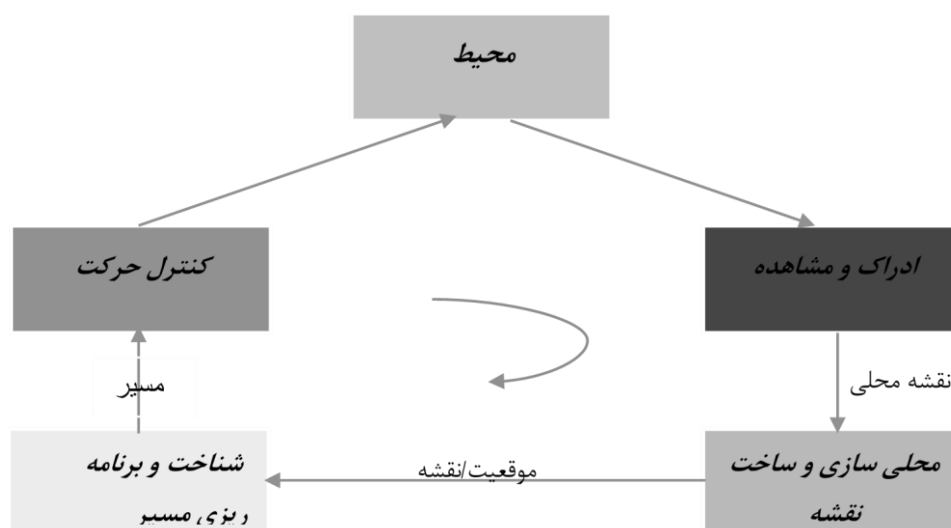
ناوبری مستقل یک روبات به دلیل کاربرد گسترده آن، یک زمینه تحقیقاتی نویدبخش محسوب می‌شود. ناوبری شامل چهار الزام اساسی شناخته‌شده درک، محلی سازی، شناخت و برنامه‌ریزی مسیر و کنترل حرکت است که در آن برنامه‌ریزی مسیر مهم‌ترین و جالب‌ترین بخش است. تکنیک‌های برنامه‌ریزی مسیر به دودسته اصلی تقسیم می‌شوند: روش‌های کلاسیک و روش‌های اکتشافی. روش‌های کلاسیک تجزیه سلول، روش میدان پتانسیل، شبکه زیر هدف و نقشه جاده‌های تشکیل‌شده است که رویکردهای ساده هستند؛ با این حال، آن‌ها معمولاً حجم محاسباتی بالایی مصرف می‌کنند و احتمالاً هنگامی که ربات با عدم اطمینان مواجه می‌شود، شکست می‌خورند. در این میان روش میدان پتانسیل با توجه به نتایج خوبش از بقیه روش‌ها قوی‌تر است. در این بخش به بررسی میدان پتانسیل می‌پردازیم.

هدایت وسیله نقلیه یکی از مهم‌ترین الزامات یک وسیله نقلیه هوشمند است. ناوبری ربات یک فرآیند طراحی شده برای هدایت به سمت یک موقعیت مطلوب است درحالی که از برخورد با موانع اجتناب می‌شود. چهار پارامتر اصلی این فرایند وجود دارد که در شکل ۳-۱ نشان داده شده است [۱]:

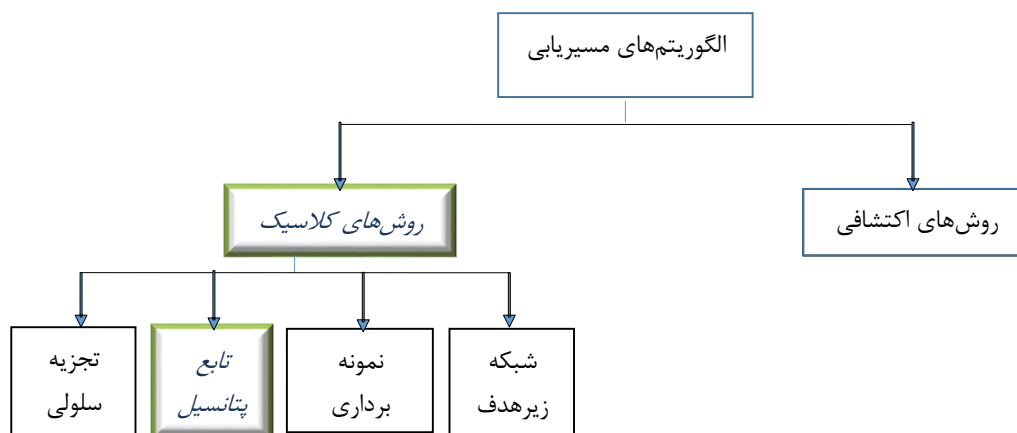
ادراک، ربات از حسگرهای آن برای استخراج اطلاعات استفاده می‌کند؛ **محلی سازی**، ربات محل خود را در فضای کاری تعیین می‌کند؛ **شناخت و برنامه‌ریزی مسیر**، ربات تصمیم می‌گیرد که چگونه برای رسیدن به هدف خود هدایت شود؛ **کنترل حرکت**، ربات حرکت خود را برای انجام مسیر موردنظر تنظیم می‌کند.

برنامه‌ریزی مسیر یک ربات می‌تواند به عنوان مجموعه‌ای از حرکات انتقالی و دورانی از موقعیت شروع به مقصد در نظر گرفته شود که درعین حال از برخورد موانع در محیط کار اجتناب می‌کند. دو تکنیک پیشنهادشده شامل تمام رویکردها در برنامه‌ریزی مسیر ربات است: (i) برنامه‌ریزی مسیر کلی یا برنامه‌ریزی مسیر آفلاین (ii) برنامه‌ریزی مسیر محلی یا برنامه‌ریزی آنلاین [۲، ۳]. یک برنامه‌ریز مسیر کلی معمولاً یک مسیر سطح بالا با رزولوشن پایین بر اساس یک نقشه از محیطی شناخته‌شده یا اطلاعات فعلی و درک خود از محیط اطراف

ایجاد می‌کند. روشی ارزشمند برای ساختن یک مسیر بهینه است؛ با این حال، واکنش مناسبی نسبت به موانع ناشناخته یا پویا ندارد. از سوی دیگر، الگوریتم برنامه‌ریزی مسیر محلی نیاز به اطلاعات پیشینی از محیط اطراف ندارد. معمولاً یک مسیر قطعی با وضوح بالا تنها بر روی یک قسمت از مسیر کلی و بر اساس اطلاعات به دست آمده از حس‌گرهای روی صفحه‌نمایش می‌دهد. این کار به طور مؤثر در محیط‌های پویا کار می‌کند. این روش ناکارآمد است، زمانی که هدف دور است یا محیطی پر از موانع است. به طور معمول، ترکیبی از هر دو روش توصیه می‌شود که مزایای آن‌ها را افزایش دهد و برخی از نقاط ضعف آن‌ها را از بین ببرد [۴-۶]. مسئله برنامه‌ریزی مسیر ربات را می‌توان به روش‌های کلاسیک و روش‌های اکتشافی تقسیم کرد [۷] همان‌طور که در شکل (۲-۳) نشان داده شده است.



شکل (۱-۳) ساختار هدایت ربات



شکل (۲-۳) الگوریتم‌های مسیریابی

۳-۱-۲- روش تابع پتانسیل

میدان پتانسیل یک میدان تولیدشده از سوی توابع پتانسیل مانع و هدف است تا وسیله نقلیه را به سمت هدف هدایت کند و آن را از موانع دورنگه دارد. تابع پتانسیل هدف دارای یک حداقل در هدف است به طوری که هدف وسیله نقلیه را جذب می کند و یک تابع پتانسیل برای مانع دارای حداکثر مقدار در موقعیت مانع است تا از برخورد با مانع جلوگیری کند. در حقیقت تابع پتانسیل موانع را به صورت یک حجم درمیاورد اما هدف را به صورت یک چاله در فضای سه بعدی نمایان می سازد. پس اگر یک گوی را در این محیط رها کنیم گوی مسیر خود را به سمت چاله پیدا می کند بدون این که با موانع برخورد کند. فرمول جدیدی (۳-۱۵) - (۳-۱۸) از پتانسیل بازدارندگی در جهت کاهش نوسانات و اجتناب از تضادهایی که به خاطر نزدیکی بعضی موانع به هدف پیش می آید، ارائه شده است [۸].

همان طور که توضیح داده شد در الگوریتم تابع پتانسیل هر کدام از موانع را به عنوان یک منبع نقطه ای از نیروی دافعه در نظر می گیریم. از این رو مقدار آن وقتی که روبات به موانع نزدیک می شود باید زیاد بوده و وقتی که روبات به قدر کافی از موانع دور است تأثیر چندانی نداشته باشد که با مربع فاصله با نقاط دیگر رابطه معکوس دارد. این نیرو را می توان به صورت زیر نشان داد:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (۳-۱)$$

در این رابطه K_{rep} ضریب مقیاس و $\rho(q)$ حداقل فاصله از q به شیء و ρ_0 فاصله تأثیر شیء است. مقدار میدان دافعه مثبت و یا صفر بوده و با نزدیک شدن به شیء مقدار آن بی نهایت می شود. اگر مرز شیء محدب بوده و به صورت تکه تکه مشتق پذیر باشد می توان از $\Gamma(q)$ مشتق گرفت. در این صورت:

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \frac{q - q_{goal}}{\rho(q)} & \text{if } \rho(q) \geq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (۳-۲)$$

این رابطه یک نقشه ماتریسی ایجاد می‌کند که هر المان آن نشان‌دهنده‌ی مقدار پتانسیل آن در مختصات IJ است. این مقدار در مراکز موانع و مرز نقشه مقدار بزرگ‌تری دارد.

نیروی دیگر توسط هدف ایجاد می‌شود که یک نیروی جاذبه است. این نیرو بافاصله از ربات رابطه مستقیم دارد:

$$U_{att} = \frac{1}{2} k_{att} \rho_{goal}^2(q) \quad (3-3)$$

$$F_{att}(q) = -\nabla U_{att}(q) = -k_{att} \cdot \rho_{goal}(q) \nabla \rho_{goal}(q) = -k_{att}(q - q_{goal}) \quad (4-3)$$

در این رابطه K_{att} ضریب مقیاس و ρ_{goal} فاصله اقلیدسی $\|q - q_{goal}\|$ تا هدف را مشخص می‌کند. با مشتق‌گیری از این رابطه مقدار نیروی جاذب به دست خواهد آمد. با تغییر ضریب مقیاس و افزایش یا کاهش آن، امکان دستیابی به مسیرهای متفاوتی به وجود می‌آید.

فصل ۴:

طراحی کنترل کننده فیدبک حالت بهینه

در این فصل ابتدا به بررسی کنترل پذیری و رؤیت پذیری سیستم پرداخته می شود و سپس با تعریف تابع هزینه مناسب، کنترل کننده بهینه طراحی می شود. برای به دست آوردن این کنترل کننده از حل معادله شناخته شده ریگاتی^۱ استفاده می شود. برای طراحی کنترل کننده فیدبک حالت نیاز به دانستن ماتریس های فضای حالت می باشد ولی سیستم مورد بررسی دارای معادلات غیر خطی می باشد که خطی سازی آن ها حول تنها یک نقطه کاری امکان پذیر نیست. به همین دلیل برای کنترل این سیستم توسط کنترل کننده فیدبک حالت، دو راه وجود دارد. راه اول خطی سازی مدام سیستم در طول فرایند کنترل می باشد که این کار باعث کاهش سرعت کنترل کننده می شود. راه دوم به وسیله دوران خطای اندازه گیری شده می باشد. در این قسمت از راه دوم برای کنترل سیستم مذکور استفاده شده است. در این روش علاوه بر افزایش سرعت کنترل کننده، می توان سیستم را حول یک نقطه کاری خطی سازی کرد و از آن مدل برای به دست آوردن کنترل کننده استفاده کرد. برای خطی سازی سیستم از روش تیلور استفاده شده است.

۴-۱- کنترل پذیری

برای بررسی کنترل پذیری سیستم، نیاز است که ابتدا معادله خطی سازی شده ی سیستم استخراج شود. همچنین باید ورودی های سیستم به درستی تعریف شود و متغیرهای کنترلی مشخص شود. در معادله (۴-۱) سیستم خطی سازی شده به فرم فضای حالت می باشد.

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{۴-۱}$$

که در آن ماتریس های A ، B و C ماتریس های فضای حالت می باشند. در ابتدا کنترلی پذیری برای کنترل همه ی متغیرهای حالت سیستم که تعداد آن ها γ عدد می باشد بررسی شد. اما ورودی سیستم قادر به کنترل

¹ Reccati

تمامی متغیرهای حالت نمی‌باشد. در نتیجه به دلیل عدم نیاز به کنترل زاویه چرخ‌ها، این دو متغیر از معادلات خطی سازی شده حذف شدند. در نتیجه با فرض وجود حس‌گر برای هر متغیر حالت، ماتریس C یک ماتریس همانی 5×5 می‌باشد و ماتریس‌های A و B به صورت زیر تعریف شده است.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0.041 & 0.041 \\ 0 & 0 & 8.25 \times 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0.253 & -0.253 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 18.609 & 1.590 \\ 1.590 & 18.609 \end{bmatrix} \quad (2-4)$$

حال با تشکیل ماتریس کنترل‌پذیری به صورت زیر می‌توان و محاسبه رنک ماتریس، می‌توان به کنترل‌پذیری سیستم پی برد.

$$Cm = [B \quad AB \quad A^2B \quad A^3B \quad A^4B] \quad (3-4)$$

در نتیجه رنک ماتریس محاسبه شده برابر 5 می‌باشد که نشان‌دهنده کنترل‌پذیری سیستم می‌باشد.

۴-۲- مشاهده پذیری

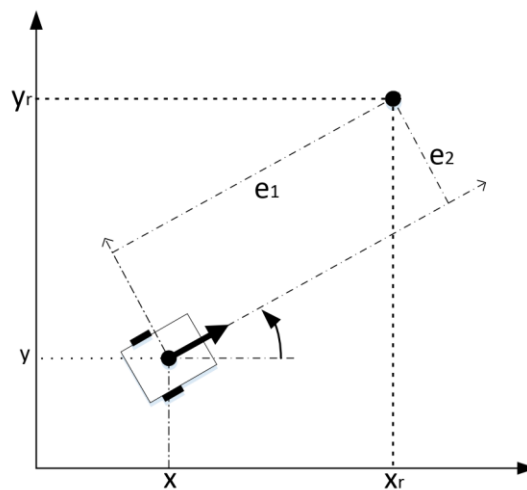
از آنجایی که فرض شده است همه‌ی متغیرهای حالت را می‌توان با استفاده از حسگرهای تعبیه شده مشاهده کرد، در نتیجه نیازی بررسی مشاهده‌پذیری نمی‌باشد. ولی با تشکیل ماتریس رؤیت پذیری می‌توان این امر را مورد بررسی قرار داد.

۴-۳- طراحی کنترل کننده

به منظور طراحی کنترل کننده حالت بهینه، ابتدا نیاز است که متغیرهای حالت مشخص شوند. همان گونه در قسمت کنترل پذیری یاد شد، ۵ متغیر حالت برای سیستم خطی سازی شده در نظر گرفته شده است که این ۵ متغیر عبارت اند از:

$$\tilde{x} = [x \quad y \quad \varphi \quad \dot{\theta}_1 \quad \dot{\theta}_2] \quad (4-4)$$

همان گونه که در مقدمه اشاره شد، سیستم دارای معادلات غیرخطی می باشد که عمل پیاده سازی کنترل کننده خطی را غیرممکن می سازد به همین دلیل برای می توان به وسیله چرخاندن دستگاه مختصات از دوران متغیرهای حالت، از سیستم خطی سازی شده برای به دست آوردن کنترل کننده حالت بهینه استفاده کرد. به منظور انجام این کار، باید سیستم شکل (۴-۱) را در نظر گرفت:



شکل (۴-۱) ربات در دستگاه مختصات

که در آن x_r و y_r نقطه مقصد کنترلی می باشند. در نتیجه با دوران خطاهای مربوط به دستگاه $x-y$ و محاسبه خطا در دستگاه e_1 و e_2 می توان بردار خطا را به صورت (۴-۵) تعریف کرد:

$$\tilde{e} = \begin{bmatrix} e_1 & e_2 & e_\varphi & e_{\dot{\theta}_1} & e_{\dot{\theta}_2} \end{bmatrix} \quad (5-4)$$

که در آن مقادیر خطا به صورت (۶-۴) محاسبه می شود.

$$e_\varphi = \varphi - \varphi_r \quad e_{\dot{\theta}_2} = \dot{\theta}_2 - \dot{\theta}_{2d} \quad e_{\dot{\theta}_1} = \dot{\theta}_1 - \dot{\theta}_{1d}$$

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \cos(-\varphi) & -\sin(-\varphi) \\ \sin(-\varphi) & \cos(-\varphi) \end{bmatrix} \times \begin{bmatrix} x - x_r \\ y - y_r \end{bmatrix} \quad (6-4)$$

در نتیجه با دوران خطای موقعیت x و y ، می توان به بردار خطا برای سیستم خطی سازی شده دست یافت. در قدم بعد با تعریف تابع هزینه درجه دو به نحوه زیر و تلاش برای کمینه کردن آن می توان مقادیر ورودی را محاسبه نمود.

$$J = \int e^T Q e + u^T R u \quad (7-4)$$

که در آن ماتریس های R و Q ، ماتریس های مثبت معین می باشند. در نتیجه با کمک گرفتن از معادله ریکاتی که همانند معادله (۸-۴) تعریف می شود:

$$AP + PA + Q - PBR^{-1}B^*P = 0 \quad (8-4)$$

می توان مقدار ضرایب کنترل کننده بهینه را همانند معادله (۹-۴) محاسبه نمود.

$$K = -(R^{-1}B^*P) \quad (9-4)$$

در نتیجه با تعریف بردار ورودی سیستم خواهیم داشت:

$$\tilde{u} = -K \times \tilde{e} \quad (10-4)$$

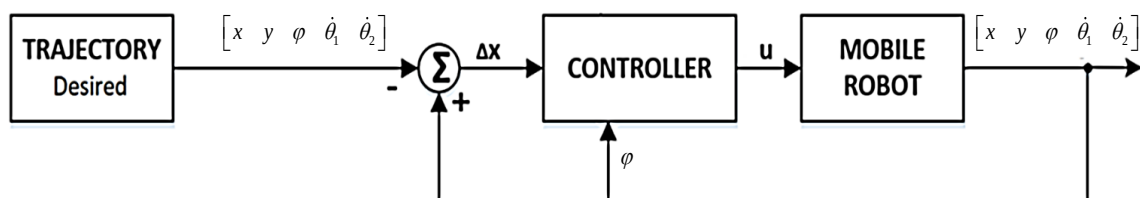
پارامترهای کنترلی که در اختیار طراح قرار دارند، عبارت‌اند از ماتریس‌های R و Q . که انتخاب مقادیر آن‌ها در به دست آوردن کنترل‌کننده بهینه اهمیت زیادی دارد. نکته مهم در انتخاب این ضرایب، نسبت مقادیر روی قطر اصلی این ماتریس‌ها می‌باشد. به دلیل اهمیت بالای کنترل موقعیت ربات، مقدار ضرایب مربوط به متغیرهای خطای موقعیت، بزرگ‌تر از خطای سرعت زاویه چرخ‌ها و زاویه ربات در نظر گرفته شده است. همچنین به دلیل کم‌اهمیت بودن کنترل زاویه ربات، مقدار ضریب کوچکی به خطای آن اختصاص داده شده است. ضرایب به دست آمده از سعی و خطا حاصل شده‌اند. در نتیجه ماتریس‌های R و Q به صورت (۱۱-۴) تعریف شده‌اند.

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & \text{eps} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix} \quad (11-4)$$

با حل معادله ریکاتی، ماتریس K که شامل بهره‌های کنترلی می‌باشد، مانند (۱۲-۴) محاسبه می‌شود.

$$K = \begin{bmatrix} 0.0224 & 0.0224 & 1.6e-10 & 0.0134 & 0.0034 \\ 0.0224 & -0.0224 & 1.6e-10 & 0.0036 & 0.0134 \end{bmatrix} \quad (12-4)$$

بدین گونه می‌توان بلوک دیاگرام این سیستم کنترلی را مانند شکل (۲-۴) نمایش داد:

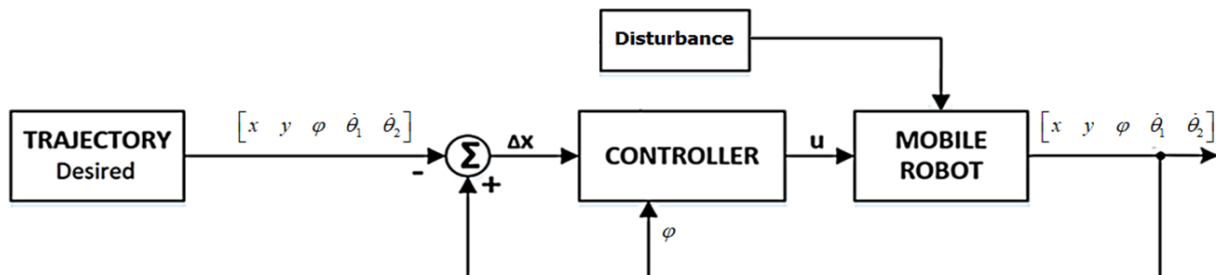


شکل (۲-۴) بلوک دیاگرام سیستم کنترلی

که در آن Δx بردار خطای سیستم قبل از دوران می‌باشد. در نتیجه کنترل‌گر دو وظیفه اصلی دارد، وظیفه اول دوران خطا و وظیفه دوم محاسبه بردار ورودی به وسیله رابطه زیر است.

برای ایستادن چرخ‌ها و جلوگیری از حرکات کوچک در زمانی که ربات به نقطه‌ی انتهایی می‌رسد، مقدار مرجع صفر برای آن در نظر گرفته شده است.

با اعمال لغزش به صورت اغتشاش، بلوک دیاگرام سیستم به شکل (۳-۴) تبدیل می‌شود.



شکل (۳-۴) بلوک دیاگرام سیستم کنترلی به همراه اغتشاش

فصل ۵:

طراحی کنترل کننده پیش بین مدل

۵-۱- معرفی کنترل کننده پیش بین مدل

کنترل پیش بین مدل، یک روش پیشرفته برای کنترل سیستم‌هاست که برای کنترل یک سیستم در حضور مجموعه‌ای از قیود به کار می‌رود. این نوع کنترل، معمولاً بر پایه مدل دینامیکی سیستم عمل می‌کند، که این مدل می‌تواند ریاضی یا بر اساس روش‌های شناسایی باشد. مهم‌ترین مزیت کنترل پیش بین مدل، بهینه کردن وضعیت زمانی کنونی سیستم با در نظر گرفتن وضعیت زمانی آینده آن می‌باشد. همچنین، این روش کنترلی قابلیت در نظر گرفتن رخداد‌های آینده را داشته و می‌تواند برای آن‌ها اقدام کنترلی انجام دهد. این روش کنترلی برای محاسبه حرکات کنترلی بهینه به موارد زیر نیازمند است:

- یک مدل دینامیکی از سیستم
 - تاریخچه حرکات کنترلی گذشته
 - تابع هزینه مورد نظر به منظور بهینه‌سازی در افق پیش‌بینی
- کنترل پیش بین مدل به دو دسته کلی خطی و غیرخطی تقسیم می‌شود. روش‌های خطی کنترل پیش بین مدل، خود به چهار دسته تقسیم شده که در جدول شماره ۵-۱ معرفی شده‌اند.
- از میان روش‌های مختلف کنترل پیش بین مدل، روش PFC برای کنترل سیستم دینامیکی ربات در این پروژه استفاده شده است. دلیل این امر ابتدا تصمیم به استفاده از روش کنترل خطی و در نتیجه خطی سازی سیستم، به دلیل سرعت محاسباتی بیشتر و پیچیدگی کمتر، و ثانیاً در دست بودن مدل فضای حالت سیستم دینامیکی ربات بوده که با ورودی‌های الگوریتم PFC همخوانی دارد.
- همچنین، الگوریتم PFC خود دارای یک انتگرال گیر ذاتی بوده، که سبب از بین رفتن خطای حالت پایدار سیستم می‌گردد. این الگوریتم بر روی مدل گسسته فضای حالت سیستم پیاده‌سازی می‌شود:

$$\begin{cases} x_m(k+1) = A_m x_m(k) + B_m u(k) \\ y(k+1) = C_m x_m(k+1) + D_m u(k+1) \end{cases}$$

با تغییر k به $k-1$ و کم کردن آن‌ها از رابطه (۵-۱) خواهیم داشت:

$$\begin{pmatrix} \Delta x_m(k+1) \\ y(k+1) \end{pmatrix} = \begin{pmatrix} A_m & 0 \\ C_m A_m & 1 \end{pmatrix} \begin{pmatrix} \Delta x_m(k) \\ y(k) \end{pmatrix} + \begin{pmatrix} B_m \\ C_m B_m \end{pmatrix} \Delta u(k) \quad (۱-۵)$$

که با توجه به آن خواهیم داشت:

$$X(k+1) = AX(k) + B\Delta U(k) \quad (۲-۵)$$

$$y(k) = (0_m \ 1) \begin{pmatrix} \Delta x_m(k) \\ y(k) \end{pmatrix} \quad (۳-۵)$$

همچنین با انجام همین پروسه تا n پله جلوتر، به رابطه زیر خواهیم رسید:

$$Y = FX(k_i) + \Phi \Delta U \quad (۴-۵)$$

$$Y = \begin{pmatrix} y(k_i + 1|k_i) \\ \vdots \\ y(k_i + N_p|k_i) \end{pmatrix} \quad (۵-۵)$$

$$\Delta U = \begin{pmatrix} \Delta u(k_i) \\ \vdots \\ \Delta u(k_i + N_c - 1) \end{pmatrix} \quad (۶-۵)$$

$$F = \begin{pmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{pmatrix} \quad (۷-۵)$$

$$\Phi = \begin{pmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{(N_p-N_c)}B \end{pmatrix} \quad (۸-۵)$$

همچنین به منظور تعیین سیگنال کنترلی، بایستی ابتدا یک تابع هزینه تعریف نماییم، که با حداقل کردن آن،

مقدار سیگنال کنترلی محاسبه گردد. معمولاً توابع هزینه شامل دو بخش می باشند: بخش اول به منظور قرار

دادن خروجی های سیستم بر روی مرجع موردنظر بوده و بخش دوم برای کاهش تلاش کنترلی:

$$J = (R_s - Y)^T Q (R_s - Y) + \Delta U^T R \Delta U \quad (۹-۵)$$

$$\frac{\partial J}{\partial \Delta U} = 0 \rightarrow \Delta U = (\Phi^T \Phi + R)^{-1} \Phi^T (R_s - Fx(k)) \quad (۱۰-۵)$$

که در آن R_s مرجع موردنظر، و Q و R وزن‌های در نظر گرفته شده برای هر ترم می‌باشد.

بدین ترتیب، با داشتن معادلات دینامیکی سیستم و خطی سازی آن‌ها، سیگنال کنترلی توسط معادله (۹-۵) قابل محاسبه بوده که بایستی در هر مرحله از حل به سیستم وارد شود.

جدول (۵-۱) مقایسه روش‌های مختلف کنترل پیش‌بین

کاربرد	معایب	مزایا		
صنعتی	نامناسب برای سیستم‌های چند ورودی چند خروجی	عدم نیاز به اطلاعات اولیه فرایند شناسایی ساده و NMP توصیف ساده سیستم‌های تأخیر دار	MAC	کنترل پیش‌بین خطی
صنعتی	سیستم باید پایدار باشد برای سیستم‌های چند ورودی چند خروجی زیاد مناسب نیست	عدم نیاز به اطلاعات اولیه فرایند شناسایی ساده توصیف ساده NMP و سیستم‌های تأخیر دار	DMC	
مناسب برای سیستم‌های ناپایدار		در نظر گرفتن نویز و اغتشاش در طراحی قابل اعمال به سیستم‌های ناپایدار	GPC	

مناسب برای سیستم‌های MIMO		مزایای فیدبک حالت استفاده از نقاط تطابق	PFC	
سیستم غیر خطی با سرعت پایین	حجم محاسباتی بالا، مسئله بهینه‌سازی غیر محدب، مشکلات شدید در پایداری و مقاومت	قابلیت استفاده از مدل غیر خطی	کنترل پیش‌بین غیر خطی	

۵-۲- روش خطی سازی استفاده‌شده

همان‌طور که گفته شد، الگوریتم کنترلی استفاده‌شده (PFC) نیازمند سیستم خطی فضای حالت می‌باشد تا سیگنال‌های کنترلی را در هر مرحله محاسبه نماید. بدین منظور باید سیستمی خطی برای تقریب سیستم دینامیکی ربات، که غیرخطی می‌باشد، در نظر گرفته شود. روش خطی‌سازی مورد استفاده در این پروژه، خطی‌سازی بر اساس بسط تیلور می‌باشد. در این روش، دو جمله اول بسط تیلور تقریب زننده از سیستم به‌عنوان تقریبی از سیستم در نظر گرفته‌شده، و ماتریس‌های A و B فضای حالت سیستم محاسبه می‌گردد. نکته مهم در این روش، تعیین یک نقطه کاری بوده که بسط تیلور حول آن نوشته‌شده و تقریبی از سیستم غیرخطی حول آن نقطه است. اگر سیستم غیرخطی به‌صورت معادله (۵-۱۰) باشد داریم:

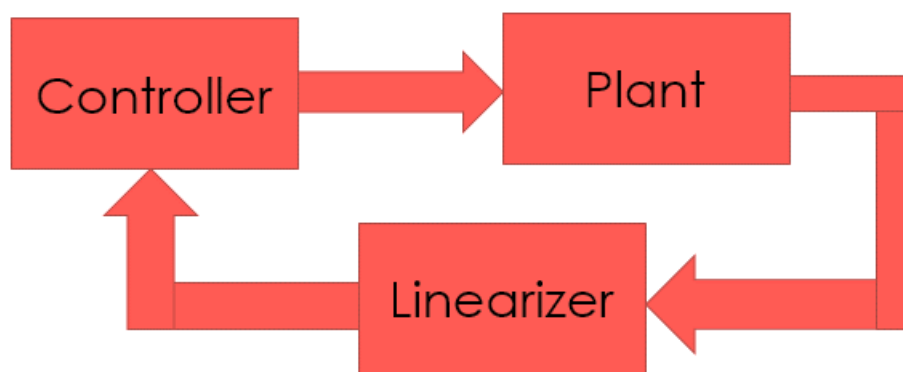
$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m) \quad (۵-۱۱)$$

$$\dot{x}_i = \frac{\partial f_i}{\partial x_j} \bigg|_{(x_j^*, u_j^*)} x_i + \frac{\partial f_i}{\partial u_j} \bigg|_{(x_j^*, u_j^*)} u_i = Ax + Bu \quad (۵-۱۲)$$

$$\dot{x} = A(x_j^*, u_j^*)x + B(x_j^*, u_j^*)u \quad (۵-۱۳)$$

که در آن‌ها x_j^* و u_j^* نقاط کاری بوده که ماتریس‌های A و B به صورت توابعی از آن‌ها می‌باشند.

به دلیل اینکه نقاط کاری سیستم، که همان متغیرهای حالت و ورودی‌های سیستم می‌باشند، در طیف گسترده‌ای قرار دارند، انتخاب یک نقطه کاری، تقریب خوبی از سیستم ارائه نمی‌دهد. به منظور حل این مشکل، نقطه کاری در نظر گرفته شده برای تعیین ماتریس‌های فضای حالت سیستم خطی تقریبی در هر مرحله، برابر با متغیرهای حالت و ورودی‌های مرحله قبلی بود. به عبارت دیگر، در هر مرحله ابتدا ماتریس‌های A و B با استفاده از مرحله قبلی، تعیین شده و به الگوریتم PFC وارد شده، که در آنجا پس از محاسبه سیگنال کنترلی و اعمال آن به سیستم اصلی غیرخطی، متغیرهای حالت محاسبه گشته و این روند تا مرحله آخر ادامه پیدا می‌کند. فلوچارت نشان‌دهنده این فرآیند در شکل (۵-۱) نمایش داده شده است.



شکل (۵-۱) بلوک دیاگرام سیستم کنترلی به همراه اغتشاش

۵-۳- پیاده‌سازی کنترل PFC

همان‌طور که در فصل دوم گفته شد، حضور لغزش در سیستم به دو صورت اغتشاش به زاویه قرارگیری ربات، و به صورت متغیر حالت سیستم مدل‌سازی شد. در حالت اول، که لغزش چرخ‌ها با زمین به صورت اغتشاشی میل‌کننده به صفر و وارد شونده به زاویه قرارگیری ربات با محور X ها به سیستم وارد شد، سیستم دینامیکی اصلی بدون لغزش بوده و تنها دارای γ متغیر حالت می‌باشد. در این حالت ابعاد ماتریس‌های A و B به ترتیب 7×7 و 7×2 بوده که در بین آن‌ها، سطرها و ستون‌های مربوط به زوایای چرخ را می‌توان حذف نمود، زیرا این

متغیرها به صورت صریح در معادلات حرکت سیستم حضور ندارند، بنابراین تمامی ترم‌های مربوط به مشتق‌های آنان برابر با صفر می‌باشد. بنابراین ابعاد ماتریس‌های A و B به 5×5 و 5×2 تقلیل می‌یابند. در حالت دوم، که لغزش به صورت ۲ متغیر حالت طولی برای دو چرخ و یک متغیر حالت عرضی برای کل ربات در معادلات دینامیکی سیستم مدل‌سازی شده است، ابعاد ماتریس‌های A و B به ترتیب 10×10 و 10×2 بوده که پس از حذف سطرها و ستون‌های مربوط به زوایای چرخ‌ها، به 7×7 و 7×2 تقلیل یافتند. تمامی محاسبات مربوط به نحوه خطی‌سازی و تعیین سیگنال‌های کنترلی با استفاده از الگوریتم PFC توسط نرم‌افزار MATLAB انجام شده است.

برنامه‌های نوشته شده برای دو حالت اعمال لغزش شامل بخش‌های کلی زیر می‌باشند:

۱. حل سیمبولیک معادلات دینامیکی سیستم و تعیین روابط ماتریس‌های A و B تخمین زننده برحسب

ورودی‌ها و متغیرهای حالت و جایگذاری این متغیرها در این روابط

۲. تعیین پارامترهای مربوط به مدل‌سازی، نظیر زمان کل مدل‌سازی، زمان نمونه‌برداری، مسیر مرجع،

مقادیر وزنی R و Q تابع هزینه، افق‌های پیش‌بینی و کنترل و شرایط اولیه سیستم

۳. الگوریتم PFC؛ تشکیل ماتریس‌های مربوط به روابط (۵-۵) تا (۵-۸)، تشکیل تابع هزینه و محاسبه

سیگنال ورودی

۴. اعمال سیگنال ورودی به سیستم غیرخطی، حل عددی معادلات حرکت و محاسبه متغیرهای حالت

۵-۴- نکته مهم در رابطه با افق پیش‌بینی و افق کنترل

از لحاظ ریاضی، طبق معادلات (۵-۵) تا (۵-۹)، با افزایش مقادیر N_p و N_c ، ابعاد ماتریس‌های F و Φ ، و در نتیجه

ماتریس‌های Y و ΔU افزایش می‌یابد. این امر سبب افزایش درجات آزادی کنترل‌کننده و همچنین در نظر

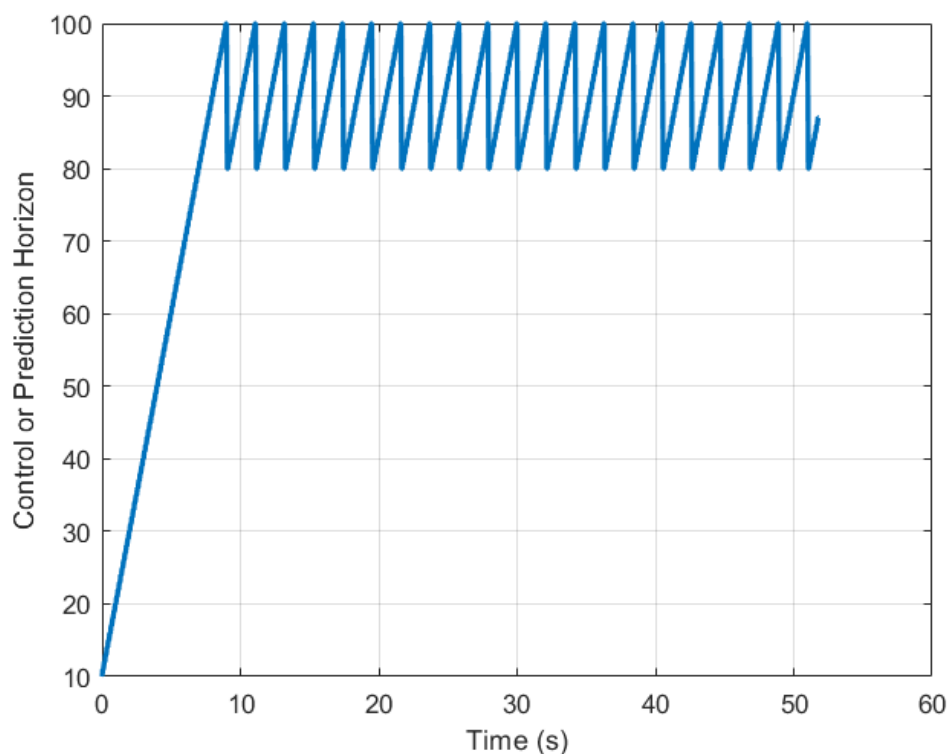
گرفتن نقاط جلوتری از تابع مرجع در تعیین سیگنال کنترلی شده، و در نتیجه سبب بهبود عملکرد کنترل‌کننده

می‌گردد. از سوی دیگر، همان‌طور که در معادلات (۵-۷) و (۵-۸) دیده می‌شود، مقادیر N_p و N_c

به صورت توان در ماتریس A ظاهر شدند، بنابراین افزایش آن‌ها، بسته به اینکه مقادیر درایه‌های ماتریس A

کوچک‌تر یا بزرگ‌تر از یک باشند، به ترتیب سبب کاهش یا افزایش این درایه‌ها شده که می‌تواند سبب ایجاد خطاهای عددی و محاسباتی شود. برای مثال، اگر ماتریس A تعیین‌شده از خطی‌سازی، دارای یک درایه با مرتبه 10^5 باشد، معمولاً در شروع مدل‌سازی اعدادی در این مرتبه و حتی ظاهر می‌شوند و بیشتر به علت خطی‌سازی و ریاضیات مربوط به آن ایجاد شده‌اند، و افق پیش‌بینی انتخاب‌شده برابر با ۱۰۰ انتخاب شود، اعدادی با مرتبه 10^7 در ماتریس‌های F و Φ ظاهر می‌شود که سبب ایجاد سیگنال‌های کنترلی ضعیف و خروج از مسیر مرجع می‌گردند.

به‌منظور حل این مشکل، در حالتی که لغزش به‌عنوان متغیر حالت در سیستم در نظر گرفته‌شده است، افق‌های پیش‌بینی و کنترل، متغیر در نظر گرفته‌شده‌اند. بدین شکل که در ابتدای مدل‌سازی که در ماتریس A ممکن است درایه‌هایی با اعداد خیلی مثبت یا خیلی منفی وجود داشته باشد، مقادیر افق پیش‌بینی و کنترل اعداد کوچکی لحاظ شده، که با افزایش زمان، افزایش می‌یابند تا سبب بهبود عملکرد کنترلی شوند. این تغییر به‌صورت خطی در نظر گرفته‌شده، یعنی به‌طور خطی با زمان افزایش پیدا کنند، همچنین یک حد بالا برای آن‌ها تعیین گشته تا از پدید آمدن اعداد بسیار بزرگ یا کوچک در حین مدل‌سازی جلوگیری شود.



شکل (۵-۲) نحوه تغییر افق‌های پیش‌بینی و کنترل برحسب زمان

فصل ۶:

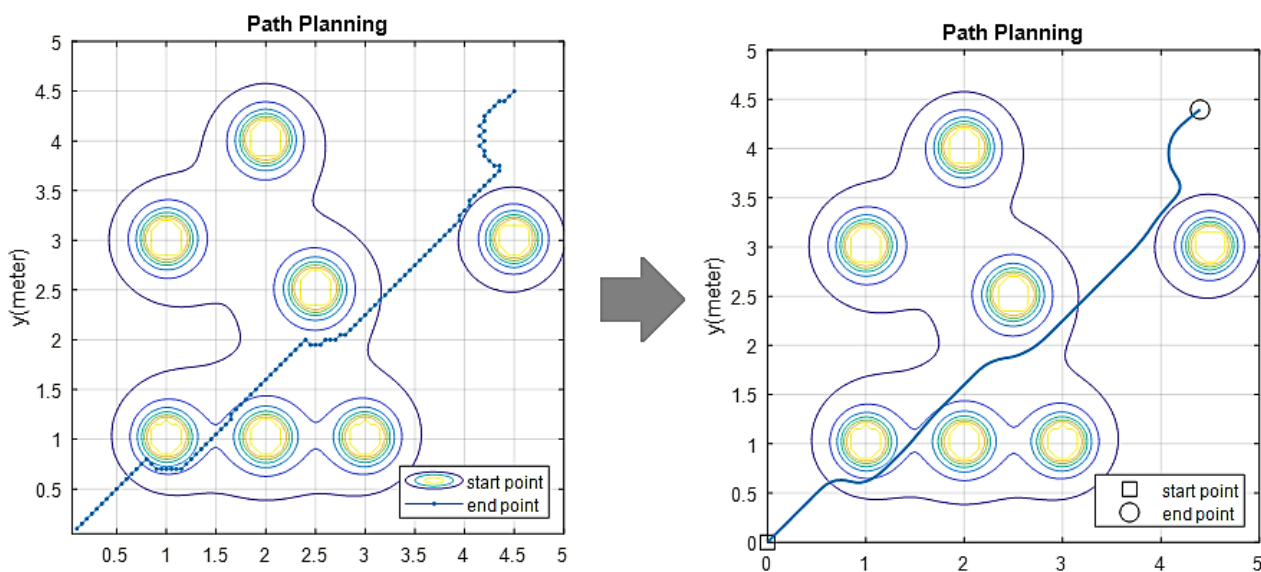
شبیه‌سازی و تحلیل نتایج

۶-۱- مقدمه

در این بخش، نتایج مربوط به کنترل ربات برای حرکت از بین موانع ارائه شده است. نتایج در دو گروه کلی در نظر گرفتن لغزش به صورت اغتشاش به زاویه قرارگیری ربات و همچنین به صورت متغیر حالت سیستم تقسیم می شوند. در حالت لغزش به صورت اغتشاش، از دو روش کنترل LQR و PFC برای قراردادن ربات در مسیر تعیین شده استفاده شده که نتایج آن ها با یکدیگر مقایسه شده اند. همچنین در حالت مدل سازی لغزش به صورت متغیر حالت سیستم، تنها روش کنترلی PFC مورد استفاده واقع شده است.

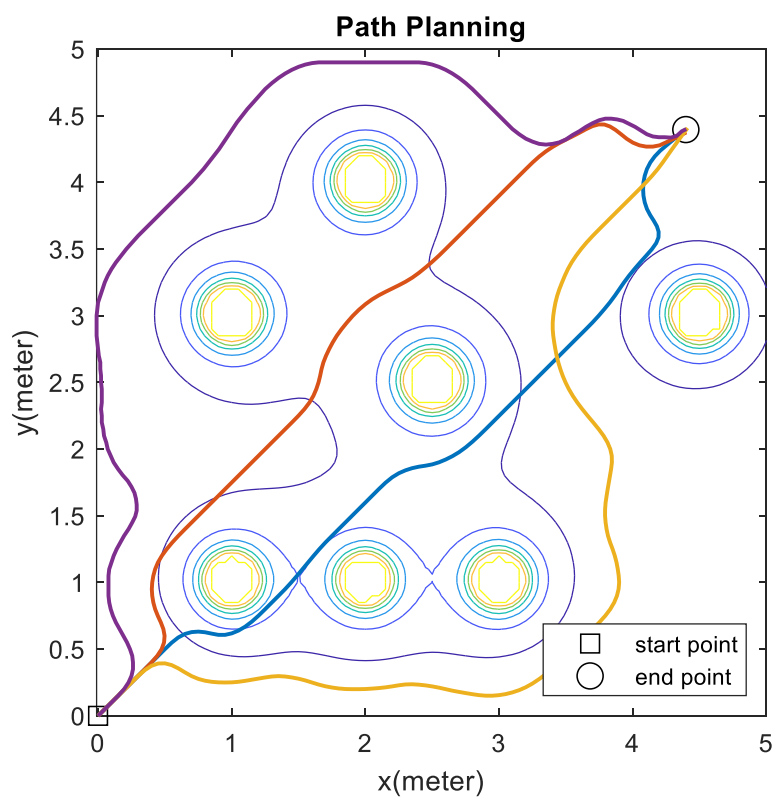
۶-۲- نقشه مشخص شده برای ربات

با در نظر گرفتن یک نقشه با ابعاد پنج در پنج مترمربع و انتخاب نقطه شروع در روی نقطه $\{0,0\}$ و جایگذاری موانع همانند شکل (۶-۱) و پیدا کردن مسیر به مقصد نهایی توسط الگوریتم تابع پتانسیل، می توان به شکل سمت چپ دست یافت. از آنجایی که نقاط همه گسسته هستند و به هیچ نقطه ای زمان اختصاص داده نشده است، ابتدا باید این نقاط را به یکدیگر متصل نمود. با توجه به اینکه نقاطی که از الگوریتم مسیریابی محاسبه شده اند، به ترتیب قرار ندارند، با استفاده از الگوریتم فروشنده دوره گرد، مرتب شده اند و به یکدیگر متصل شده اند. در نهایت با در نظر گرفتن سرعت متوسط ربات، و طول مسیر به دست آمده، به هر نقطه از مسیر زمان اختصاص داده شده است. سپس با پردازش مسیر و کاهش نوسانات آن، شکل سمت چپ در شکل (۶-۱) حاصل شده است.



شکل (۶-۱) پردازش مسیر به دست آمده

حال با استفاده از ضریب مقیاس متفاوت در تابع پتانسیل می توان مسیرهای متفاوتی را به دست آورد. که در شکل (۶-۲) مشاهده می شود.



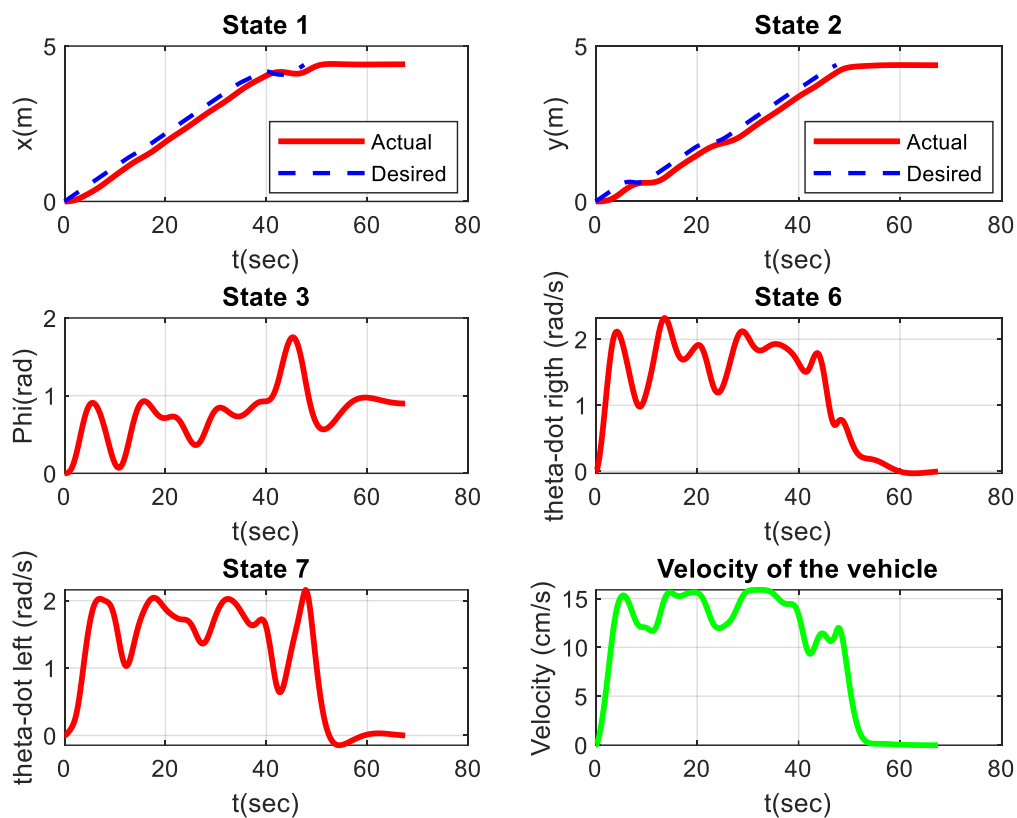
شکل (۶-۲) پیدا کردن مسیرهای متفاوت

با توجه به این که کوتاه‌ترین مسیر، مسیر آبی‌رنگ به ازای بزرگ‌ترین ضریب تابع پتانسیل می‌باشد، ربات بر روی این مسیر کنترل می‌شود. البته این امکان وجود دارد که تابع هزینه آفلاین تعریف شود و پس از اعمال عملیات کنترلی بر روی همه‌ی مسیرها، مسیری که هزینه کمتری دارد، انتخاب شود و نتایج مربوط به آن بررسی شود. تابع هزینه آفلاین کنترلی مانند معادله (۶-۱) طراحی شده است. به‌منظور هم‌طراز سازی پارامترهای تابع هزینه تعریف‌شده، این پارامترها با استفاده مقدار بیشینه خود بی بعدسازی شده‌اند.

$$Cost = \int \left(\frac{u_1}{u_{1max}} \right)^2 + \left(\frac{u_2}{u_{2max}} \right)^2 + \left(\frac{(x - x_{destination})}{(x - x_{destination})_{max}} \right)^2 + \left(\frac{(y - y_{destination})}{(y - y_{destination})_{max}} \right)^2 dt \quad (۶-۱)$$

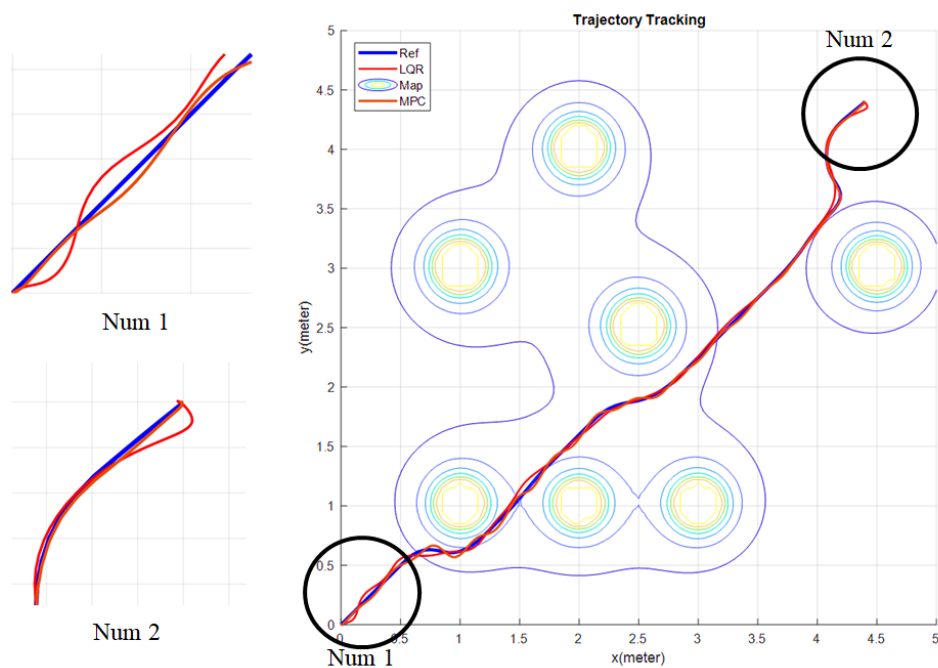
۶-۳- نتایج مربوط به کنترل سیستم با در نظر گرفتن لغزش اغتشاشی

در ابتدا با استفاده از کنترل‌کننده LQR بدون در نظر گرفتن لغزش به کنترل ربات پرداخته می‌شود. متغیرهای حالت مربوط به سیستم را می‌توان در شکل (۶-۳) مشاهده کرد. باید توجه کرد سرعت متوسط مسیر بر روی ۱۵ سانتی‌متر بر ثانیه تنظیم شده است.



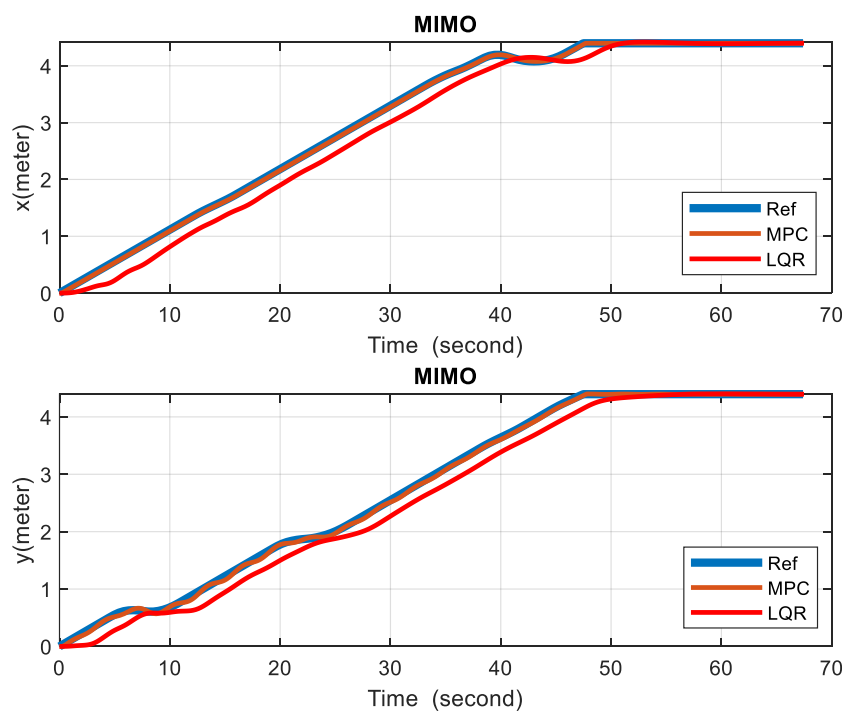
شکل (۳-۶) متغیرهای حالت سیستم برای کنترل کننده LQR

در ادامه با اضافه کردن لغزش به صورت اغتشاش به زاویه حرکت ربات که دیاگرام آن مانند شکل (۳-۴) می شود، شبیه سازی کنترلی برای کنترل کننده های LQR و MPC صورت گرفته است. مسیر طی شده توسط ربات برای هر دو کنترل کننده در شکل (۴-۶) نمایش داده شده است.



شکل (۴-۶) مسیر کنترل شده با در نظر گرفتن لغزش به صورت اغتشاش

همان گونه که مشاهده می شود، کنترل کننده MPC رفتار بهتری را از خود نشان داده است و انطباق بیشتری بر مسیر دارد. همچنین متغیرهای x و y در شکل (۵-۶) نمایش داده شده است.

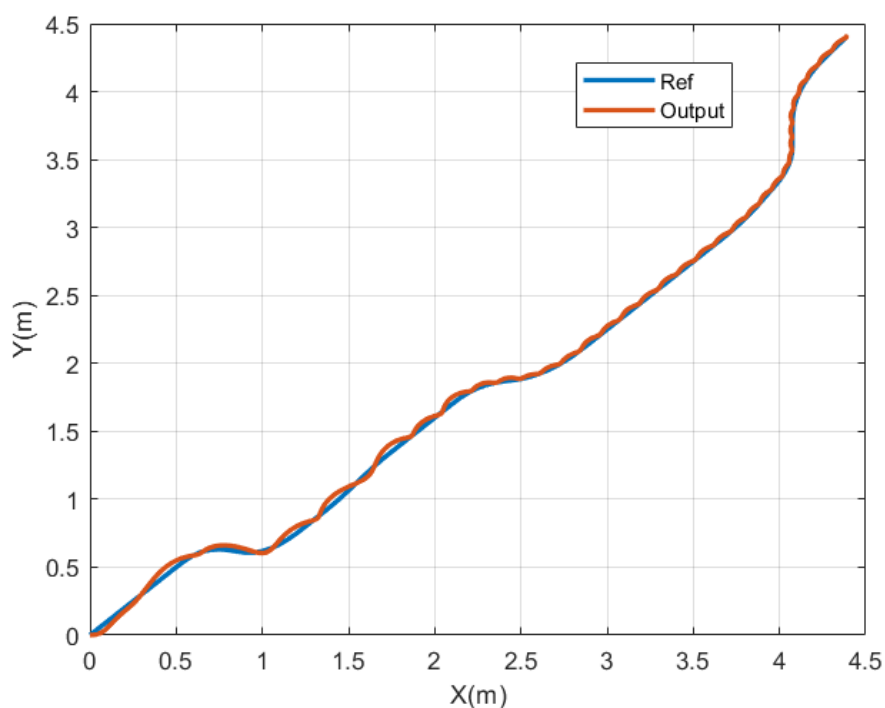


شکل (۵-۶) نمودار مختصات مرکز جرم ربات بر حسب زمان

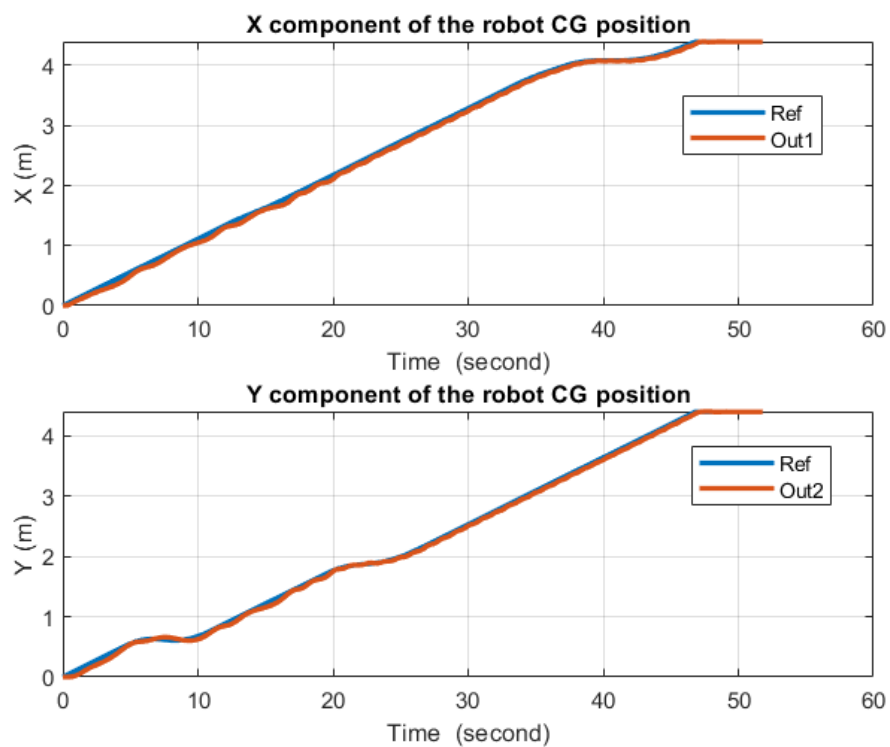
مشاهده می‌شود که کنترل‌کننده MPC تطابق بیشتری بر روی مسیر دارد. همچنین تأخیری که در کنترل‌کننده LQR وجود دارد را ندارد چراکه با استفاده از قابلیت پیش‌بینی، از این اتفاق جلوگیری می‌کند.

۴-۶- نتایج مربوط به کنترل سیستم با لغزش به صورت متغیر حالت

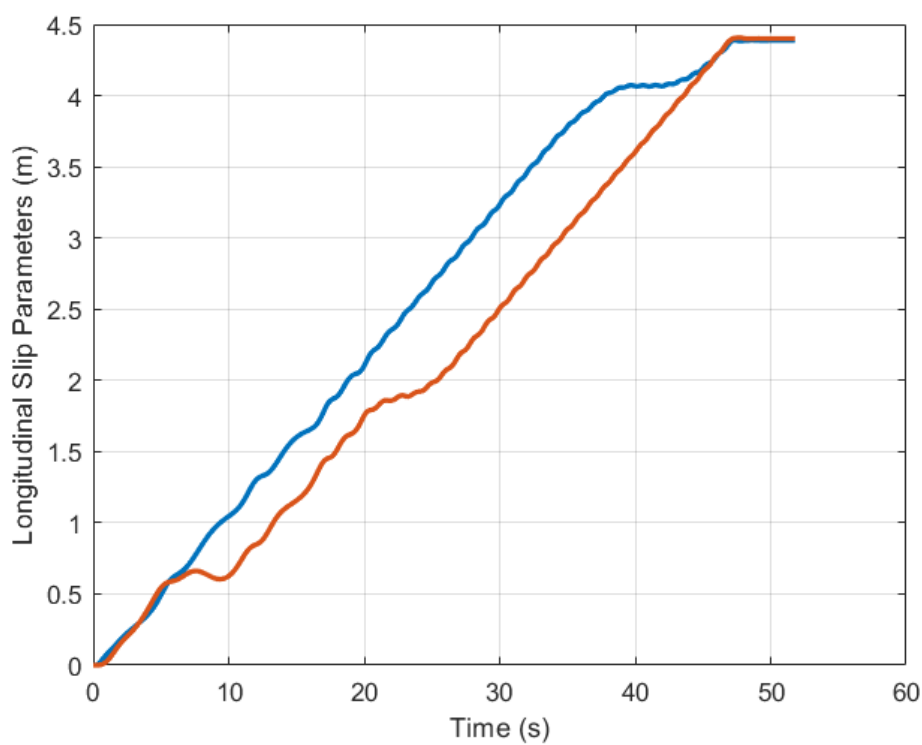
نتایج در این بخش شامل نمودار مسیر حرکت ربات، پارامترهای لغزش و مختصات مرکز جرم ربات برحسب زمان می‌باشد که برای دو سرعت ۱۵ سانتی‌متر بر ثانیه و ۳ سانتی‌متر بر ثانیه ترسیم شده است.



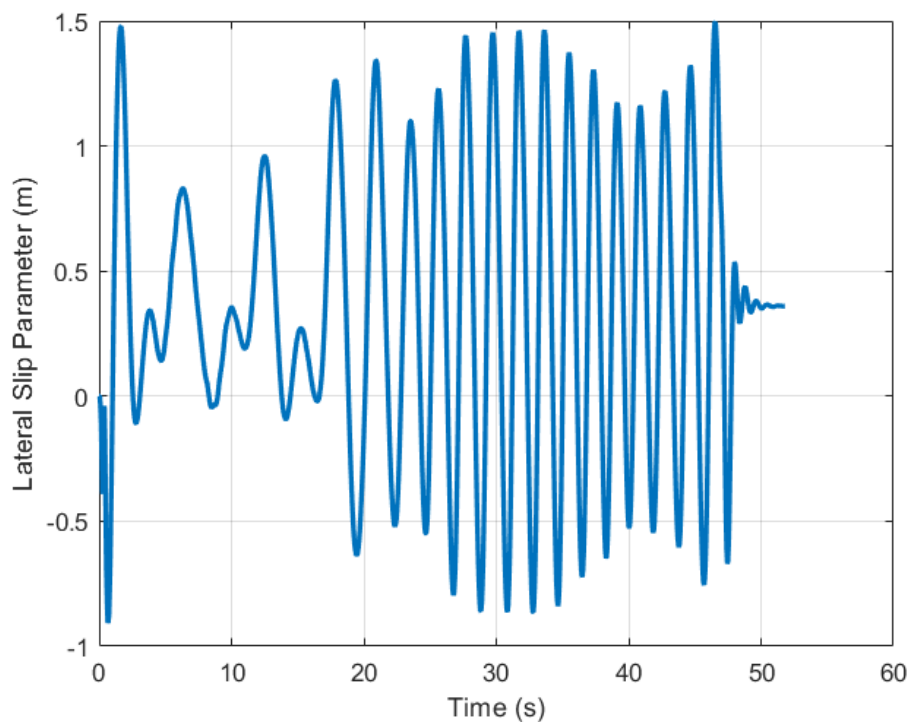
شکل (۴-۶) نمودار مسیر حرکت ربات در سرعت ۱۵ سانتی‌متر بر ثانیه



شکل (۶-۷) نمودار مختصات مرکز جرم ربات برحسب زمان در سرعت ۱۵ سانتی‌متر بر ثانیه

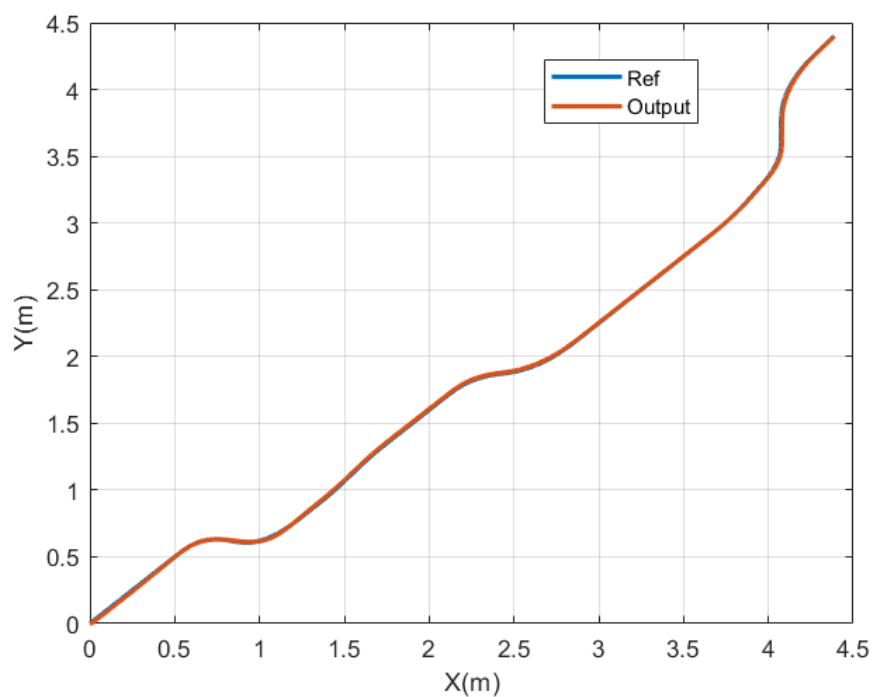


شکل (۶-۸) نمودار متغیر لغزش طولی ربات برحسب زمان در سرعت ۱۵ سانتی‌متر بر ثانیه

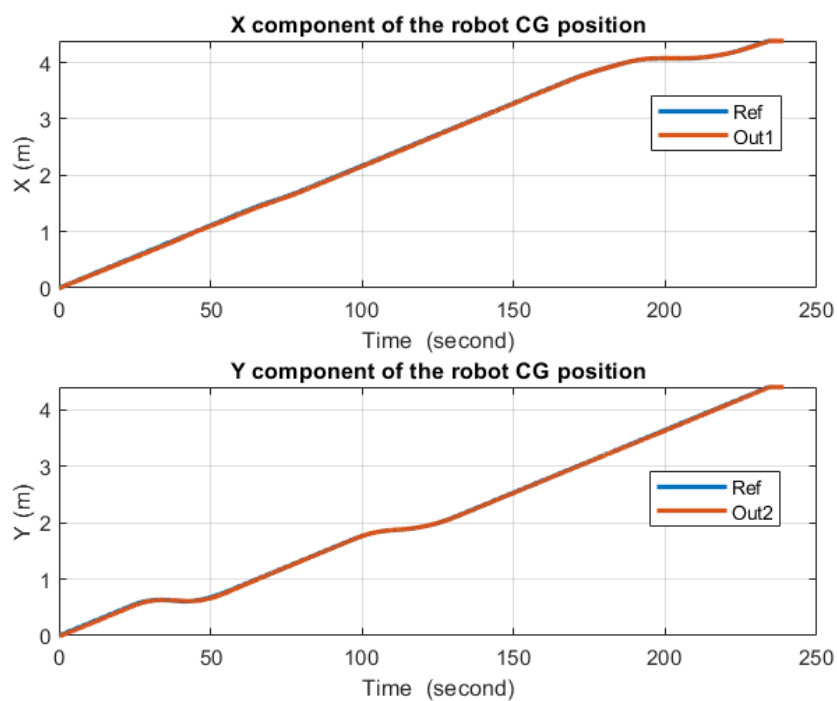


شکل (۶-۹) نمودار متغیر لغزش عرضی ربات برحسب زمان در سرعت ۱۵ سانتی‌متر بر ثانیه

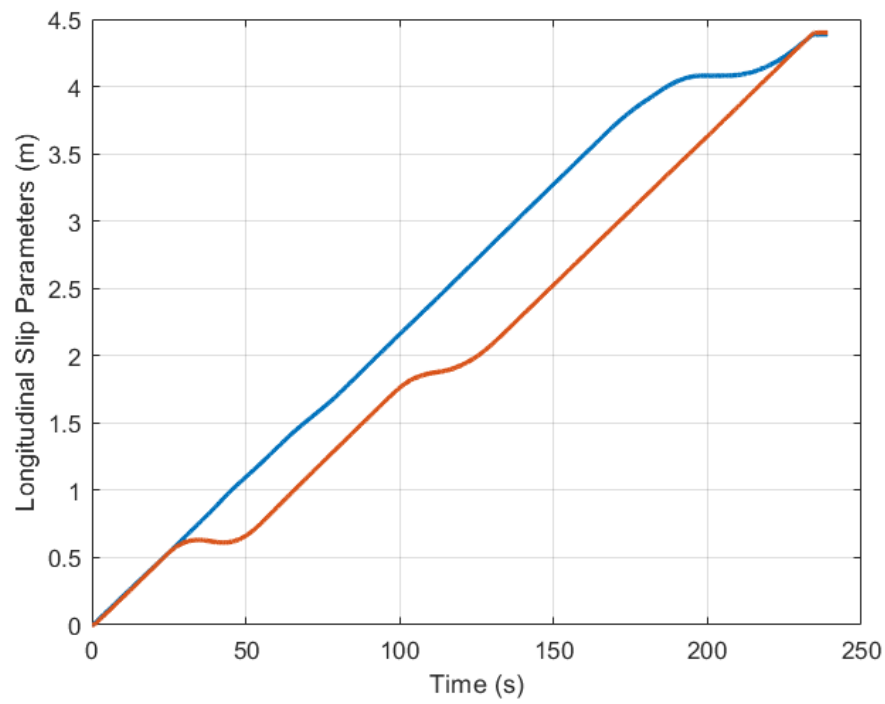
همان‌طور که در شکل (۶-۶) پیداست، نمودار قرمز رنگ که نشان‌دهنده مسیر طی شده توسط ربات می‌باشد، توانسته به‌خوبی مسیر مشخص شده را دنبال کند، اما به علت وجود لغزش، در اطراف این مسیر حرکتی نوسانی شکل را انجام داده که ناشی از سر خوردن ربات می‌باشد. با توجه به شکل (۶-۷)، این سر خوردن هم در مؤلفه X و هم Y مرکز جرم ربات رخ داده و در اثر غیر صفر بودن متغیرهای لغزش طولی و عرضی می‌باشد که در شکل (۶-۸) و شکل (۶-۹) مشاهده می‌شوند.



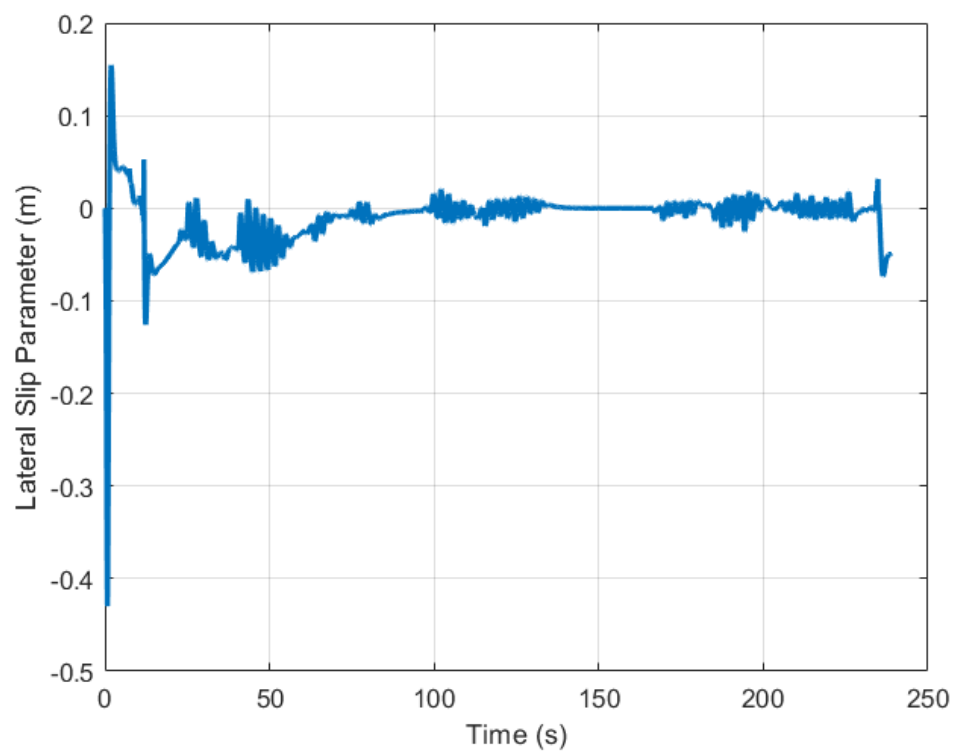
شکل (۶-۱۰) نمودار مسیر حرکت ربات در سرعت ۳ سانتی‌متر بر ثانیه



شکل (۶-۱۱) نمودار مختصات مرکز جرم ربات بر حسب زمان در سرعت ۳ سانتی‌متر بر ثانیه



شکل (۶-۱۲) نمودار متغیر لغزش طولی ربات برحسب زمان در سرعت ۳ سانتی متر بر ثانیه

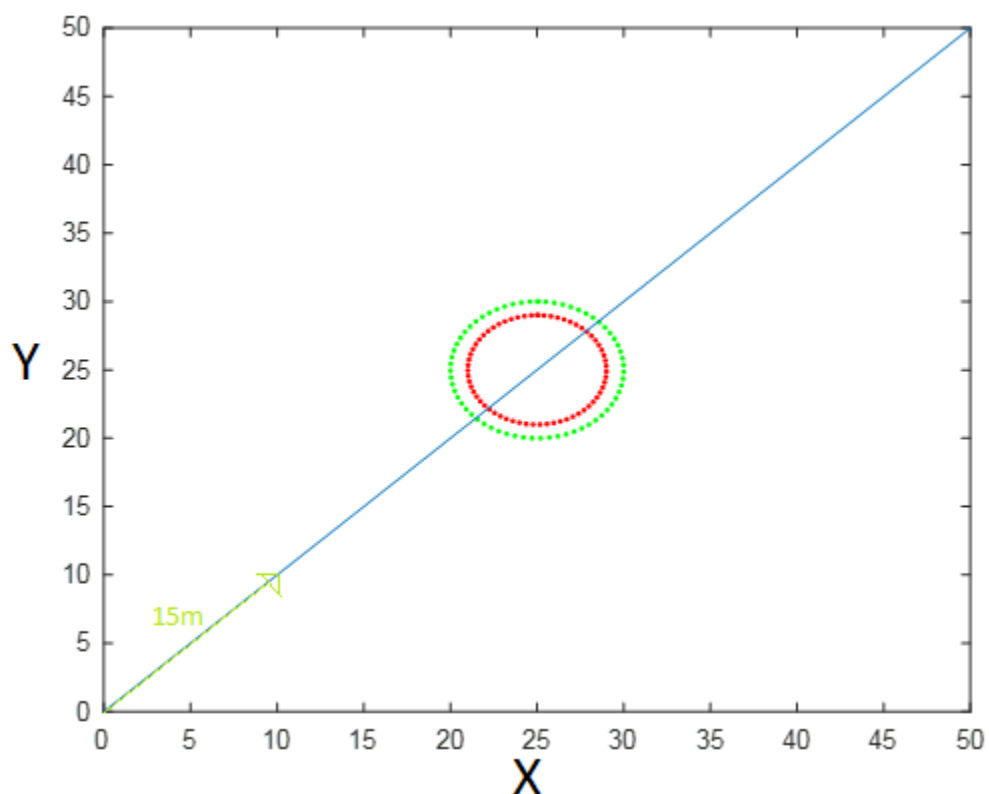


شکل (۶-۱۳) نمودار متغیر لغزش عرضی ربات برحسب زمان در سرعت ۳ سانتی متر بر ثانیه

کاهش سرعت ربات سبب ایجاد کنترل بیشتر توسط کنترل کننده بر دینامیک حرکت ربات در حضور لغزش می شود. با کاهش سرعت، میزان سرخوردن و در نتیجه خروج ربات از مسیر تعیین شده نیز کاهش می یابد. این موضوع در شکل (۶-۱۰) و شکل (۶-۱۱) به خوبی دیده می شود، که در آن ها نمودارهای قرمز رنگ و آبی رنگ بر روی یکدیگر قرار گرفته اند. همچنین با توجه به شکل (۶-۱۲) و شکل (۶-۱۳)، میزان متغیرهای لغزش به طور چشمگیری کاهش یافته است.

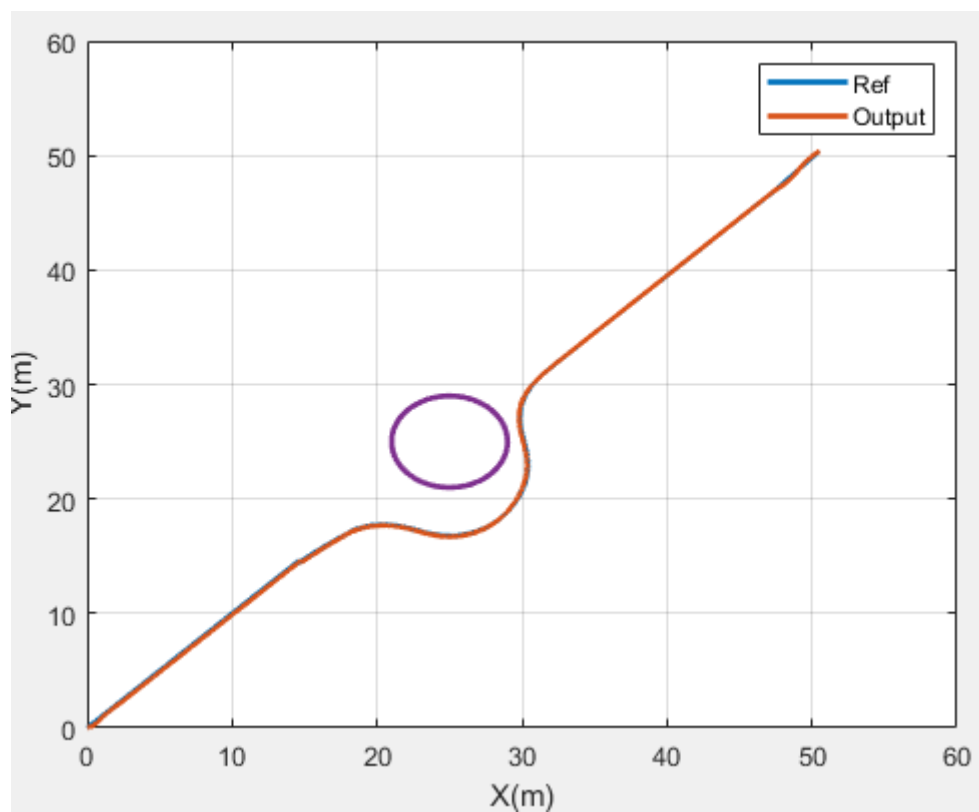
۶-۵- نتایج کنترل ربات با دید محدود

در این قسمت فرض بر آن است که ربات در ابتدا از موقعیت موانع مطلع نیست و به کمک یک حس گر از محیط اطراف آگاه می شود. در نتیجه در ابتدا مستقیم به سمت هدف مطلوب شروع به حرکت می کند تا زمانی که حس گر ها اطلاعات جدیدی از موقعیت موانع به سیستم ارائه می کنند سپس مسیر جدید به کنترل کننده ارائه می شود. ربات از موقعیت اولیه (۰،۰) به سمت موقعیت هدف (۵۰،۵۰) حرکت می کند. در حالی که در موقعیت (۲۵،۲۵) مانع با شعاع ۴ متر و حاشیه ایمن ۱ متر قرار دارند. برد حس گر ۱۵ متر در نظر گرفته شد. مسیری که ربات در ابتدا بدون اطلاع از مانع در پیش می گیرد در شکل (۶-۱۴) قابل مشاهده است:



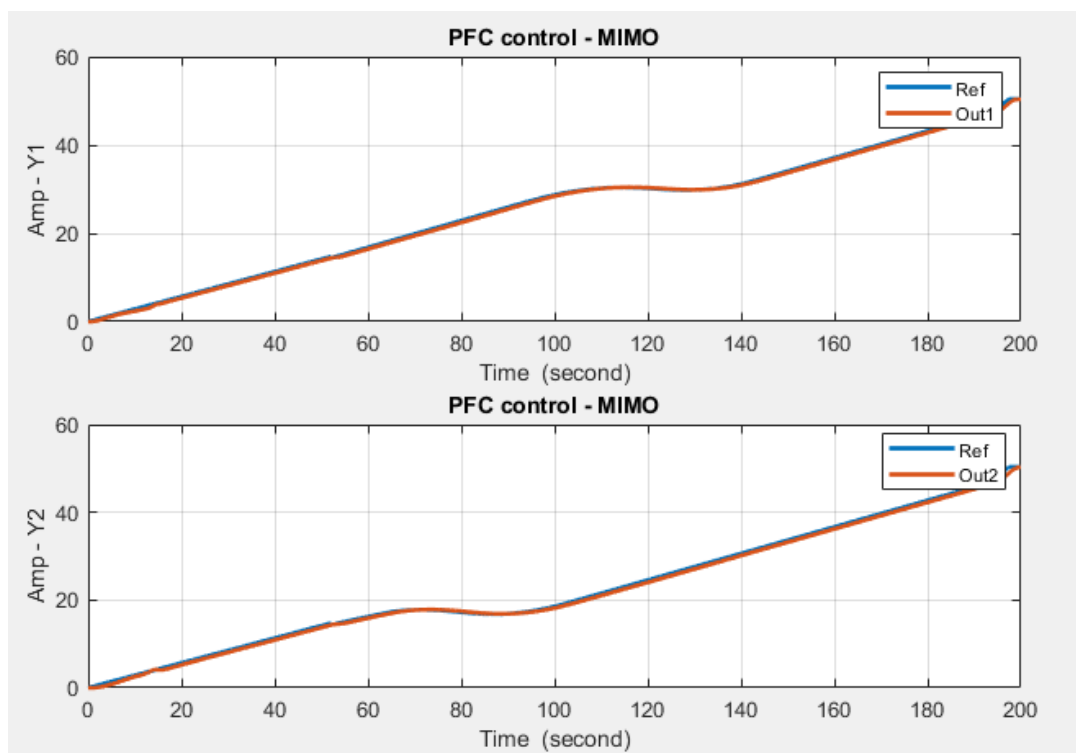
شکل (۶-۱۴) مسیر ربات بدون اطلاع از مانع

انتظار داریم که ربات در ۱۵ متری مانع از آن مطلع شود همان طور که مشاهده می شود ربات در مختصات $(x,y)=(14.412,14.329)$ از وجود مانع مطلع می شود. فاصله ربات در این نقطه تا مانع ۱۴٫۹۹ بود. شکل (۶-۱۵) کنترل ربات در حضور مانع را نشان می دهد.

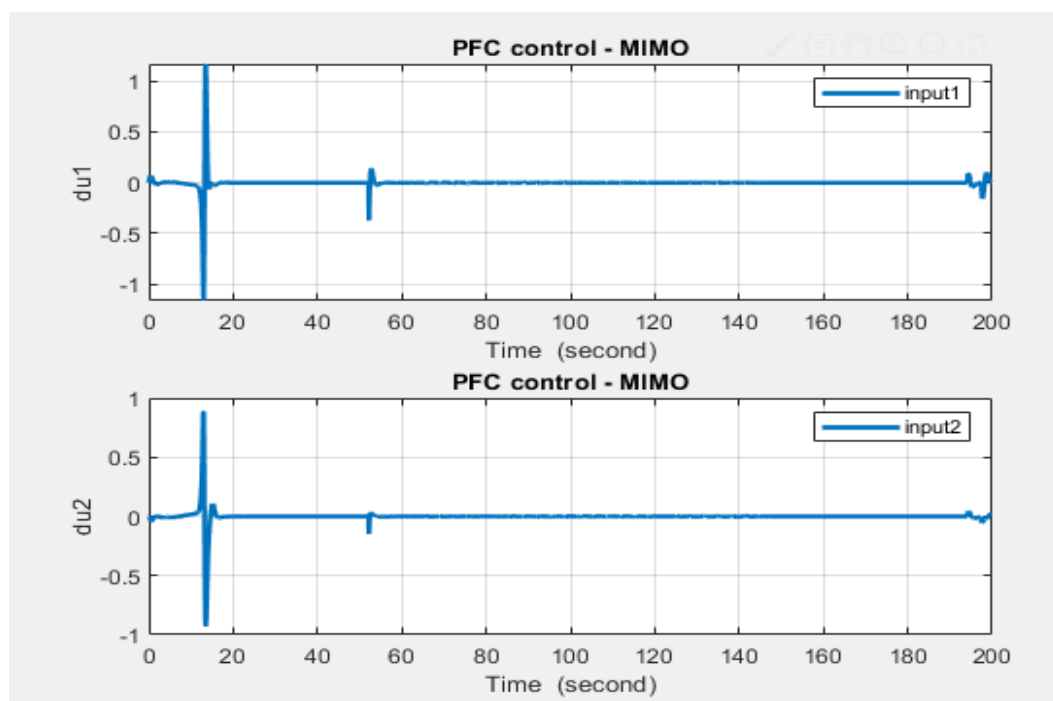


شکل (۶-۱۵) کنترل ربات در حضور مانع استاتیکی

در شکل (۶-۱۶) مقایسه خروجی‌های کنترل‌کننده (x,y) و مسیر را می‌توان دید. در شکل (۶-۱۷) ورودی‌های کنترل‌کننده را می‌توان دید.



شکل (۶-۱۶) خروجی‌های کنترل‌کننده



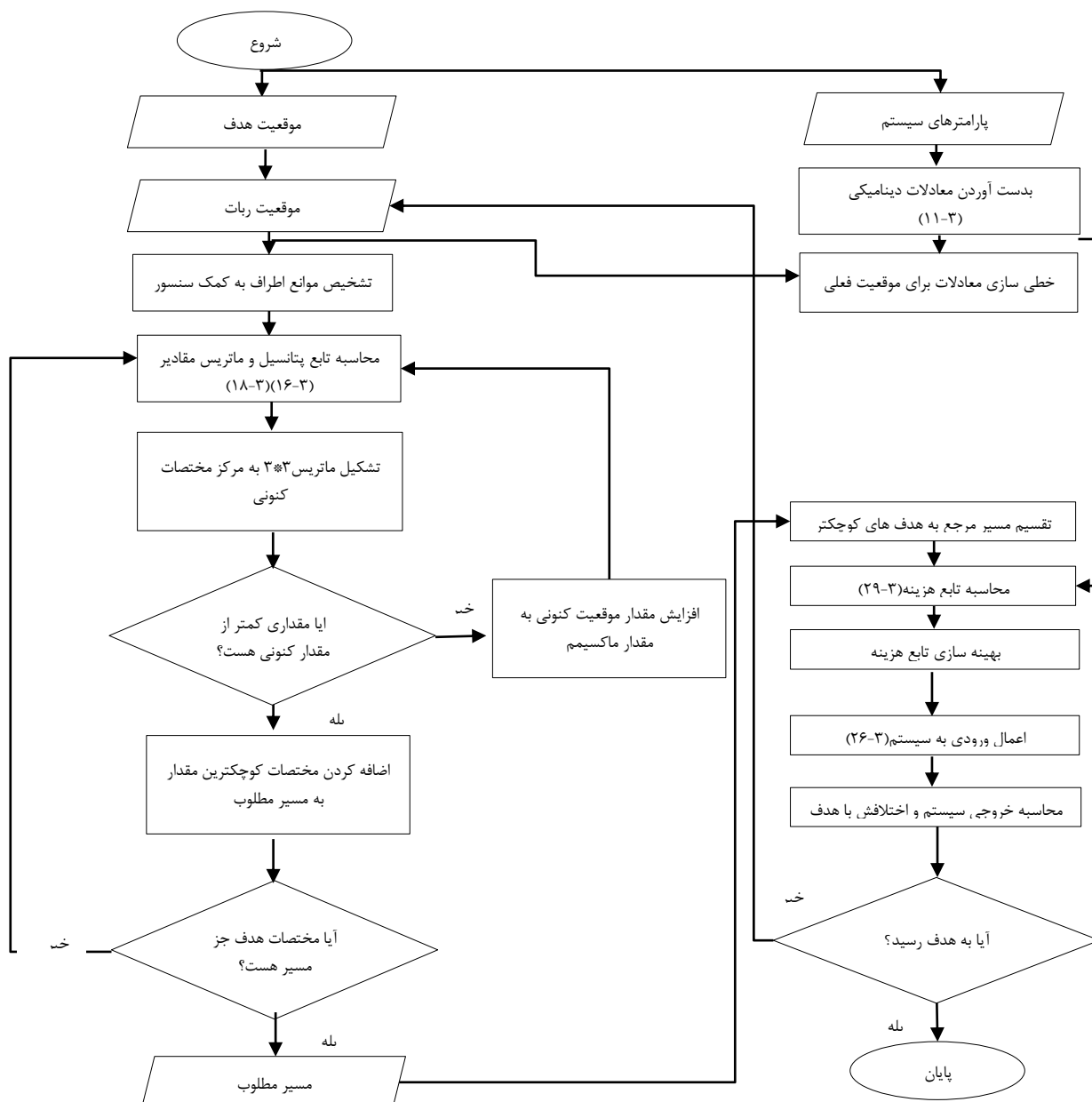
شکل (۶-۱۷) نرخ ورودی‌های کنترل‌کننده

فصل ۷:

جمع‌بندی و پیشنهادها

۱-۷- مقدمه

کنترل ربات‌های متحرک در حضور موانع ناشناخته در محیط بسیار مهم است. در این تحقیق برای پایه متحرک ربات اسکات کنترل پیش‌بین مدلی طراحی شد که با کمک روش مسیریابی تابع پتانسیل توانست



به خوبی ربات را در حضور موانع استاتیکی هدایت کند. روند طراحی را در شکل (۱-۷) مشاهده می کنید:

شکل (۱-۷) روند و مراحل انجام پروژه

۷-۲- جمع بندی

همان طور که در فصل قبل مشاهده شد، با استفاده از کنترل کننده پیش بین مدل توانستیم ربات را در حضور موانع استاتیکی کنترل کنیم. برای این منظور ابتدا مدل دینامیکی سیستم مورد نیاز است. دینامیک سیستم را می توان به روش لاگرانژ استخراج کرد. معادله لاگرانژ برای سیستمی با قیود غیرهولونومیک دارای ماتریس-هایی چون اینرسی، کوریولیس و قیود می باشد. ماتریس قیود سیستم از اهمیت ویژه ای برخوردار است زیرا هرگونه تغییر در سیستم که منجر به تغییر در قیود سیستم می شود، در ماتریس قیود قابل مشاهده است. گام بعدی خطی سازی معادلات دینامیکی است. برای این نوع سیستم ها چندین نقطه پایداری وجود دارد. در حقیقت نقطه پایداری حرکت می کند. به همین منظور خطی سازی به طور پیوسته انجام شد. ماتریس های سیستم خطی شده به کنترل کننده پیش بین مدل در فضای حالت وارد شدند. مزیتی که این نوع کنترل کننده نسبت به انواع دیگر کنترل کننده های پیش بین خطی دارد، قابلیت کنترل سیستم های چند ورودی چند خروجی با دقت بالا است. بعلاوه لزومی ندارد سیستم پایدار باشد درحالی که به منظور کنترل به کمک روش DMC برای به دست آوردن ضرایب پله، پاسخ پله سیستم می بایست پایدار باشد به همین دلیل در مورد سیستم مورد بررسی نمی توان از پاسخ پله سیستم استفاده کرد. زیرا که در این روش ارتباط ورودی ها با یکدیگر در سیستم های چند ورودی در نظر گرفته نمی شود و باید به طور جداگانه ورودی پله را به سیستم اعمال کنیم تا ضرایب پله را به دست آوریم. اگر تنها به یکی از موتور ها ورودی اعمال کنیم ربات به دور خود می چرخد و پاسخ پله ناپایدار خواهد بود. کنترل کننده PFC به خوبی می تواند سیستم های پیچیده را با در نظر گرفتن روابط بین ورودی ها و خروجی ها کنترل کند.

۷-۳- نوآوری

کنترل ربات ها جهت اجتناب از برخورد با موانع محیطی موضوعی است که همیشه مورد بحث بوده است اما انتخاب کنترل کننده پیش بین مدل با توجه به قابلیت های منحصربه فرد این کنترل کننده ، موضوع جدیدی است که کمتر به آن توجه شده است. با بهره گیری از مزایای این کنترل کننده می توان این مشکل را حل

کرد. در این تحقیق به کمک کنترل کننده پیش بین PFC، پایه متحرک اسکات بعد از خطی سازی معادلات دینامیکی در حضور موانع استاتیکی کنترل شد. به این منظور از روش تابع پتانسیل به منظور مسیریابی در حضور موانع استاتیکی استفاده شد. یکی از مشکلات این روش مسیریابی، وجود مینیمم های محلی است که به کمک الگوریتم ارائه شده در این تحقیق این مشکل حل شد. با استفاده از قابلیت پیش بینی و گسسته بودن کنترل کننده پیش بین، حسگری طراحی شد تا بتوان در هر گام اطلاعات محیط را بررسی کرد و پایه را در حضور موانع استاتیکی که به صورت ناگهانی در مسیر پدیدار می شوند کنترل کرد. همچنین روش های کنترلی در حضور لغزش به صورت متغیر در معادلات و به صورت اغتشاش نیز پیاده سازی شدند.

مراجع

مراجع

١. Siegwart, R. and I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robot*, 1a edição, 2004, MIT Press, Cambridge, Massachusetts, EUA.
٢. Qin, Y.-Q., et al. *Path planning for mobile robot using the particle swarm optimization with mutation operator*. in *Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826)*. 2004. IEEE.
٣. Raja, P. and S. Pugazhenth, *Optimal path planning of mobile robots: A review*. International journal of physical sciences, 2012. 7(9): p. 1314-1320.
٤. Zhang, H., J. Butzke, and M. Likhachev. *Combining global and local planning with guarantees on completeness*. in *2012 IEEE International Conference on Robotics and Automation*. 2012. IEEE.
٥. Wang, L.C., L.S. Yong, and M.H. Ang. *Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment*. in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*. 2002. IEEE.
٦. Bi, Z., Y. Yimin, and Y. Wei. *Hierarchical path planning approach for mobile robot navigation under the dynamic environment*. in *2008 6th IEEE International Conference on Industrial Informatics*. 2008. IEEE.
٧. Masehian, E. and D. Sedighizadeh, *Classic and heuristic approaches in robot motion planning-a chronological review*. World Academy of Science, Engineering and Technology, 2007. 23(5): p. 101-106.
٨. Sfeir, J., M. Saad, and H. Saliyah-Hassane. *An improved artificial potential field approach to real-time mobile robot path planning in an unknown environment*. in *2011 IEEE international symposium on robotic and sensors environments (ROSE)*. 2011. IEEE.