



## مسیریابی ربات بستر پویا در محیط ناشناخته برمبنای مفاهیم کنترل کننده‌ی افق پیش‌بین و مخروط تصادم سرعت

مجتبی زارعی<sup>1</sup>، نوید کاشی<sup>2</sup>، مهدی طالع ماسوله<sup>3\*</sup>، احمد کلهر<sup>4</sup>

1- دانشجوی کارشناسی ارشد، مهندسی مکترونیک، دانشگاه تهران، تهران

2- دانشجوی کارشناسی، مهندسی کامپیوتر، دانشگاه شهید بهشتی، تهران

3- استادیار، هوش ماشین و رباتیک، دانشگاه تهران، تهران

4- استادیار، مهندسی کنترل، دانشگاه تهران، تهران

\* تهران، صندوق پستی 143951374، m.t.masouleh@ut.ac.ir

### اطلاعات مقاله

مقاله پژوهشی کامل  
دریافت: 03 شهریور 1396  
پذیرش: 21 مهر 1396  
ارائه در سایت: 27 آبان 1396  
کلید واژگان:

مسیریابی ربات متحرک  
بهینه‌سازی محدب  
مخروط تصادم سرعت  
کنترل افق پیش‌بین  
اجتناب از برخورد

### چکیده

در این پژوهش، الگوریتمی نوین که ترکیبی از برنامه‌ریزی گسسته، بهینه‌سازی محدب، قیود مخروط تصادم سرعت و افق پیش‌بین می‌باشد، برای مسیریابی و کنترل بلادرنگ و عاری از تداخل ربات‌های متحرک ارائه می‌گردد. در این مقاله، محیط مورد نظر محیط ناشناخته می‌باشد که الگوریتم ارائه شده برای دو سناریو مجزا، یکی برای محیط ناشناخته ثابت و دیگری محیط ناشناخته پویا با استفاده از بسته‌ی نرم افزاری سی‌وی‌ایکس در شبیه‌ساز متلب ابتدا شبیه‌سازی می‌شود و سپس در ادامه با در نظر گرفتن ملاحظات لازم برای ربات واقعی، الگوریتم بر روی ربات ای-پاک در محیط رآسپایده‌سازی می‌گردد. به منظور پیاده‌سازی از حل کننده‌ی گورویی با مجوز آموزشی استفاده می‌گردد. به منظور محدب سازی قیود نامحدب، از روش برنامه نویسی مخلوط عدد صحیح خطی استفاده می‌شود. همچنین، تمامی قیود باید حالت خطی خود را در افق‌های بعدی نیز حفظ نمایند. از این رو، برای اجتناب از برخورد، از دو نوع قید استفاده می‌شود. در افق اول، از قید مخروط تصادم به منظور تضمین عدم برخورد و در افق‌های بعدی از قیود مبتنی بر مفهوم روش باگ به منظور عدم برخورد و همچنین مدل کردن قسمتی از نامعینی‌ها در اندازه‌گیری موقعیت و سرعت‌ها استفاده شده است. نتایج نشان می‌دهند که این الگوریتم قابلیت اطمینان بالایی دارد و تمامی موارد ایمنی و عدم برخورد با موانع رعایت می‌شود. همچنین زمان حل در هر مرحله کمتر از 0.004 ثانیه و حاکی از بلادرنگ بودن این روش برای مسیریابی می‌باشد.

## Motion planning of mobile robots in the unknown circumstances based on the receding horizon control and velocity obstacle concepts

Mojtaba Zarei<sup>1</sup>, Navid Kashi<sup>2</sup>, Mehdi Tale Masouleh<sup>3\*</sup>, Ahmad Kalhor<sup>3</sup>

1- Faculty of New Science and Technology, University of Tehran, Tehran, Iran

2- Department of Computer science, Shahid Beheshti University, Tehran, Iran

3- School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

\* P.O.B. 143951374, Tehran, Iran, m.t.masouleh@ut.ac.ir

### ARTICLE INFORMATION

Original Research Paper  
Received 25 August 2017  
Accepted 13 October 2017  
Available Online 18 November 2017

#### Keywords:

Mobile robots motion planning  
Convex optimization  
Velocity obstacle  
Receding horizon control  
Collision avoidance

### ABSTRACT

This paper aims at proposing an algorithm for collision-free motion planning of two wheeled mobile robots. The proposed approach relies on discrete motion planning, convex optimization and receding horizon control (RHC) concepts. The proposed algorithm is employed for motion planning and control of a mobile robot in order to pass through an unknown environment both in simulation and practical implementation. In this regard, CVX package benefited from the Gurobi solver is employed to solve the optimization problem in the simulation. Moreover, in order to perform a collision-free motion planning, corresponding Robot Operating System (ROS) package with the intended mobile robot is employed to cooperate with the provided motion planner package. The provided package utilizes educational license of Gurobi optimizer solely to speed up solving proposed optimization problem and its built in branch and bound for Mixed Linear Integer Programming (MLIP). In order to keep the linear form of the constraints, a combination of the Velocity Obstacle (VO) in the first horizon and Bug method concept for the rest of the horizons is used. Obtained results show the reliability of this algorithm for safe collision avoidance. The reported results reveal this fact that by considering the maximum velocity of the E-puck, obtained computational time is less than 0.004 sec. in each stage which is fast enough for robot motion planning tasks.

### Please cite this article using:

M. Zarei, N. Kashi, M. Tale Masouleh, A. Kalhor, Motion planning of mobile robots in the unknown circumstances based on the receding horizon control and velocity obstacle concepts, *Modares Mechanical Engineering*, Vol. 17, No. 11, pp. 259-268, 2018 (in Persian)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

## 1- مقدمه

ربات‌ها را از لحاظ جابجایی‌پذیری می‌توان به دو دسته‌ی ربات‌های بستر ثابت و بستر پویا تقسیم نمود [1]. ربات‌های بستر ثابت مانند ربات‌های سری و موازی [2-5] به علت فضای کاری محدود در محیط‌های بسته به کار گرفته می‌شوند. اما امروزه ربات‌های متحرک، شامل ربات‌های پرنده، پادار، چرخ‌دار و شناگر، بدون داشتن محدودیت‌های ذکر شده برای ربات‌های بستر ثابت، در بخش‌های مختلفی از زندگی جوامع بشری به کار برده می‌شوند. از این رو مسیریابی عاری از تداخل و ایمن این گونه ربات‌ها از اهمیت بالایی برخوردار است. مسیریابی عبارت است از شروع حرکت یک ربات متحرک از نقطه‌ای مشخص به منظور رسیدن به نقطه‌ی انتهایی و بدون برخورد با موانع موجود در مسیر و در نظر گرفتن قیودی مشخص در طول انجام عملیات. این قیود می‌تواند دربرگیرنده‌ی قیود حاکم بر سینماتیک یا دینامیک ربات مورد مطالعه باشد. همچنین می‌توان شاخص‌هایی از جمله کوتاه‌ترین طول مسیر طی شده، کمترین زمان طی مسیر یا کمترین انرژی مصرفی برای طی کردن مسیر را به عنوان تابع هدف برای انجام مسیریابی در نظر گرفت. روش‌های موجود را می‌توان به دو دسته روش‌های جامع و روش‌های محلی مسیریابی تقسیم نمود [6]. برخلاف مسیریاب‌های جامع، مسیریاب‌های محلی عموماً بر مبنای پردازش داده‌های ارسال شده توسط حسگرها کار می‌کنند که همین مهم منجر به این می‌گردد تا دسته‌ی دوم بیشتر به منظور مسیریاب‌های برخط استفاده شوند. در ادبیات مربوطه، روش‌های زیادی برای محیط‌های ایستا ارائه شده است [7، 8]. اما یکی از چالش برانگیزترین موضوعات در این طبقه‌بندی، موضوع مسیریابی عاری از برخورد ربات‌های موازی در محیط‌های ناشناخته و پویا می‌باشد. علت این امر این است که در مسیریابی در محیط‌های پویا زمانی که ربات با قیود پویا مواجه می‌شود به منظور اجتناب از برخورد دنبال کردن مسیر بسیار سخت‌تر می‌باشد. برای حل این مشکلات، در سال‌های اخیر روش‌هایی ارائه شده است [9-11]. علاوه بر روش‌های یاد شده، روش‌هایی مانند روش گراف دیداری [12]، نمودار ورونی [13]، تجزیه‌ی سلولی [14]، میدان پتانسیل مجازی [15] و مخروط تصادم سرعت [16] نیز به کرات به منظور مسیریابی ربات استفاده شده‌اند. همچنین در برخی روش‌ها به منظور حل مسئله بهینه‌سازی از بهینه‌سازی ازدحام ذرات [17] و یا الگوریتم ژنتیک [18] استفاده شده است. بیشتر روش‌های ذکر شده، از روش‌های هندسی که در فضای پیکربندی اجرا می‌شوند استفاده می‌کنند. بیشتر روش‌های یاد شده از مسائل گوناگونی از قبیل گیرافتادن در مقدار کمینه محلی، عدم قابلیت آن‌ها برای استفاده‌ی برخط در محیط‌های ناشناخته، نیاز به آزادی عمل بالا برای اجتناب از برخورد با مانع و دارا بودن زمان محاسباتی بالا رنج می‌برند. در این راستا و به منظور حل برخی از مشکلات یاد شده، کنترل‌کننده‌های پیش‌بین مبتنی بر مدل و خانواده‌های آن مانند کنترل افق پیش‌بین، یکی از روش‌های پرکاربرد در این حیطه می‌باشند [19-21]. همچنین به منظور مقابله با مشکلات ذکر شده برای روش‌های بیان شده، تلفیقی از روش‌ها در برخی از مراجع استفاده شده است. برای مثال، در [22] به منظور حل بهینه‌سازی در میدان مجازی پتانسیل از روش ژنتیک استفاده گردید تا مشکل گیرافتادن در مقدار کمینه محلی برطرف شود. همچنین به منظور کاهش زمان محاسباتی و در نتیجه بهبود عملکرد به صورت برخط، از تلفیق روش برنامه‌نویسی عدد گسسته [23] و مفهوم کنترل پیش‌بین مبتنی بر مدل در مراجعی مانند [24] و [25] استفاده شده است که مسئله‌ی مسیریابی را در طول یک افق محدود برای داده‌های موجود حل می‌کند. با پیشرفت بسته‌های نرم‌افزاری برای حل مسائل

بهینه‌سازی مانند گورویی [26]، می‌توان از روش‌های بهینه‌سازی نیز در طراحی مسیر بهینه بهره برد. در این راستا، در [1] از مفهوم بهینه‌سازی محدب، برنامه‌نویسی مختلط عدد صحیح خطی و روش باگ [27] در کنترل کننده‌ی افق پیش‌بین به منظور به دست آوردن مسیر بهینه در وجود قیود برخورد استفاده شد. همچنین در [28] از تلفیق بهینه‌سازی محدب، برنامه‌نویسی عدد صحیح مختلط، میدان پتانسیل مجازی و مخروط تصادم سرعت برای ایجاد روشی نوین به منظور مسیریابی ربات‌های پرنده بدون گیرافتادن در مقدار کمینه‌ی محلی استفاده شد.

هدف در این مقاله، ارائه الگوریتمی جامع برای مسیریابی و کنترل بلادرنگ و عاری از تداخل ربات‌های متحرک می‌باشد. از آنجایی که آزمایشگاه تعامل انسان و ربات دانشگاه تهران مجهز به 7 عدد ربات متحرک ای‌پاک می‌باشد، الگوریتم‌های به دست آمده، پس از شبیه‌سازی، بر روی این ربات‌ها پیاده‌سازی می‌گردند. لازم به ذکر است مساله بهینه‌سازی برای شبیه‌سازی با استفاده از بسته نرم‌افزاری سی‌وی‌ایکس که برای حل مسائل بهینه‌سازی محدب در نرم‌افزار متلب فراهم شده است، حل می‌شود و برای پیاده‌سازی از لیسانس آموزشی نرم‌افزار گورویی استفاده خواهد شد. در این پژوهش، به منظور تضمین عدم برخورد از قید مخروط تصادم که یک قید خطی می‌باشد استفاده شده است. همچنین به منظور بهره‌بری از مزایای کنترل پیش‌بین مبتنی بر مدل، از مفهوم کنترل افق پیش‌بین برای بهینه کردن طول پیمایش مسیر و عدم برخورد ربات با موانع استفاده شده است. اثبات پایداری این روش با استفاده از مفهوم پایداری لیاپانوف در [29] ارائه گردیده است. به منظور حل مسئله‌ی بهینه‌سازی که در ذات این نوع کنترل کننده‌ها موجود است، از روش‌های محدب‌سازی قیود و حل مسئله به صورت یک مسئله محدب بهره برده خواهد شد. در این راستا، روش خطی برنامه‌نویسی عدد صحیح مختلط معرفی می‌شود. با توجه به ویژگی‌هایی که برای قید مخروط تصادم سرعت باید در نظر گرفت، روش برنامه‌نویسی مختلط عدد صحیح به منظور شبیه‌سازی و پیاده‌سازی نهایی برای این پژوهش استفاده می‌گردد. در این راستا، در بخش دوم، تعاریف اولیه مربوط به معادلات سینماتیک ربات و قیود عدم برخورد مورد استفاده در این مقاله ارائه می‌شوند. در بخش سوم، مسئله‌ی مسیریابی برای ربات در چارچوب کنترل‌کننده‌ی افق پیش‌بین بیان می‌شود و تابع هدف و قیود مورد استفاده تعریف می‌گردند. بخش چهارم به بیان روش ابداع شده و ارائه‌ی شبیه‌سازی‌ها در محیط‌های ایستا و پویا اختصاص یافته است. همچنین، نتایج شبیه‌سازی به دست آمده در بخش پنجم با نتایج گزارش شده در پژوهش‌های قبلی مقایسه می‌شوند. در بخش ششم، روش ارائه شده بر روی ربات ای‌پاک پیاده‌سازی می‌شود و نتایج آن گزارش می‌گردد. در آخر نیز نتیجه‌گیری این پژوهش ارائه می‌گردد.

## 2- تعاریف اولیه

این بخش، به ارائه‌ی معادلات حاکم بر سینماتیک ربات دوچرخ و تعاریف مربوط به قیود مربوط به عدم برخورد اختصاص یافته است.

## 1-2- معادلات حاکم بر سینماتیک ربات دوچرخ

ربات متحرک ای‌پاک اولین بار در سال 2006 در موسسه صنعتی فدرال سوئیس واقع در لوزان طراحی و ساخته شده است [30]. این ربات که یک ربات آموزشی-پژوهشی محسوب می‌شود، به صورت متن‌باز در اختیار کاربران قرار گرفته تا بتواند برای توسعه نرم‌افزاری و سخت‌افزاری مورد استفاده قرار

[32] و مخروط تصادم سرعت زمان محدود [33] بر مبنای روش مخروط تصادم سرعت توسعه یافته‌اند. به علت ماهیت قید مخروط تصادم و وابستگی مخروط به موقعیت ربات و مانع، این قید برای افق‌های بعدی تبدیل به یک قید غیرخطی شده و عملاً کارایی مد نظر در افق اول را نخواهد داشت. به همین دلیل در افق‌های بعدی ( $h > 1$ ) از قیود باگ استفاده می‌شود. در واقع، علاوه بر استفاده‌ی قابلیت‌های کنترل افق پیش‌بین برای طراحی مسیر عاری از برخورد، با در نظر گرفتن عدم قطعیت‌های موجود در اندازه‌گیری موقعیت و سرعت موانع، می‌توان این نامعینی‌ها را در افق دوم در قالب افزایش شعاع مانع به علت حضور عدم قطعیت‌ها مدل نمود و تا حدی خطای این نامعینی‌ها را به منظور انتخاب مسیری عاری از برخورد در کنترل کننده لحاظ کرد. بدین منظور، شعاع مانع در افق دوم به مقدار 10 درصد بیشتر در نظر گرفته می‌شود و در الگوریتم لحاظ می‌گردد. اما در افق‌های بعدی این شعاع همان شعاع اصلی ربات در نظر گرفته می‌شود. دلیل اصلی، فرض نمودن تخمینی بودن سرعت ربات در افق‌های بعدی می‌باشد. در صورتی که شعاع‌ها افزایش یابد موجب محدود شدن فضای قابل دسترس ربات می‌گردد که بیش از حد محافظه کارانه خواهد بود. روش باگ، یا عدم تقاطع، این مهم را بیان می‌کند که در صورتی که موقعیت بعدی مرکز ربات در موقعیت اضافه شده‌ی شعاع مانع و شعاع ربات قرار نگیرد، برخورد صورت نخواهد گرفت. با در نظر گرفتن ربات و موانع به صورت دایره‌ای، می‌توان این قید بدیهی را به‌عنوان شرط تقاطع به صورت معادله‌ی زیر بیان نمود:

$$IO_{A|B} = \{x | \exists t > 0 : x \in D(x_B, r_A + r_B)\} \quad (9)$$

با توجه به ماهیت "عدمی" قیود مذکور، این قیود قیود غیرمحدب می‌باشند. علت اصلی نامحدب بودن قیود خطی، وجود قیدها به صورت "یا" می‌باشد. به این صورت که سرعت ربات باید خارج از مخروط تصادم باشد که به این معناست که یا باید سرعت آن کوچکتر از یک یال و یا بزرگتر از یال دیگر باشد. قیودی به شکل "یا" ساختاری نامحدب داشته که به منظور استفاده از این قید باید از روش‌های محدب‌سازی استفاده نمود.

### 3- تعریف مسئله‌ی مسیریابی یک ربات متحرک

برای روشن‌تر شدن مساله بهینه‌سازی مسیر، ابتدا بخش‌های مختلف آن، که شامل تابع هزینه و قیدهای مربوط به موانع و ربات می‌باشد، مورد بررسی قرار می‌گیرد و سپس مساله کلی به صورت یک مساله بهینه‌سازی محدب نوشته می‌شود.

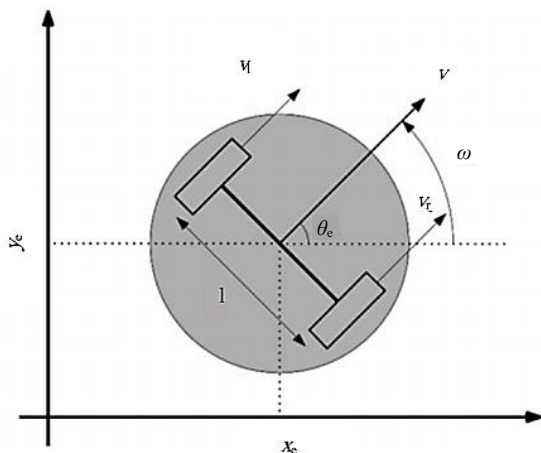


Fig. 1 Corresponding parameters with the kinematic of robot under study

شکل 1 پارامترهای مربوط به سینماتیک ربات مورد مطالعه

گیرد. از آنجایی که ربات متحرک ای‌پاک فقط می‌تواند دستور سرعت بگیرد و قابلیت کنترل گشتاور موتورها را ندارد، در این قسمت، سینماتیک ربات مورد بررسی قرار می‌گیرد که در قسمت‌های بعدی از آن استفاده می‌شود. همان‌طور که در "شکل 1" نشان داده شده است، موقعیت ربات در مختصات کلی به صورت زیر بیان می‌شود:

$$x = [x_e \ y_e \ \theta_e]^T \quad (1)$$

که در آن،  $\theta_e$  زاویه بین محور اصلی ربات (موازی با چرخ‌ها) و محور X می‌باشد. همچنین تغییرات موقعیت ربات را نیز می‌توان به صورت زیر بیان کرد:

$$\dot{x} = [\dot{x}_e \ \dot{y}_e \ \dot{\theta}_e]^T \quad (2)$$

اگر سرعت‌های خطی و زاویه‌ای ربات به عنوان ورودی کنترلی  $u$  و سرعت ربات در راستای محورهای X و Y به‌عنوان خروجی در نظر گرفته شوند، معادله کنترلی به فرم زیر نوشته می‌شود:

$$\dot{x} = Bu \quad (3)$$

که در معادله فوق ماتریس ضرایب ورودی B به صورت زیر است:

$$B = \begin{bmatrix} \cos \theta_e & 0 \\ \sin \theta_e & 0 \\ 0 & 1 \end{bmatrix} \quad (4)$$

و ورودی‌های سیستم که سرعت خطی و سرعت زاویه‌ای ربات هستند، به

شکل زیر تعریف می‌گردند:

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta_e & \sin \theta_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} \quad (5)$$

همچنین سرعت خطی و زاویه‌ای ربات را می‌توان برحسب سرعت چرخ

چپ و راست و زاویه  $\theta_e$  به صورت زیر نوشت:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} (v_r + v_l) \\ 2 \\ (v_r - v_l) \\ l \end{bmatrix} \quad (6)$$

که در رابطه فوق  $v_r$  و  $v_l$  به ترتیب، سرعت چرخ راست و چرخ چپ ربات و  $l$  فاصله بین دوچرخ می‌باشند. بنابراین، تغییرات سرعت برحسب سرعت چرخ‌های چپ و راست با استفاده از معادلات زیر به دست می‌آیند:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} (v_r + v_l) \cos \theta_e \\ 2 \\ (v_r - v_l) \\ l \end{bmatrix} \quad (7)$$

### 2-2- قیود مخروط تصادم سرعت و مفهوم روش باگ

قید مخروط تصادم سرعت در واقع یک قید هندسی خطی می‌باشد که خانواده‌های آن به صورت مکرر در مسیریابی مورد استفاده قرار گرفته‌اند. این قید را می‌توان به صورت زیر بیان نمود:

$$VO_{A|B} = \{v | \exists t > 0 : (v - v_B)t \in D(x_B - x_A, r_A + r_B)\} \quad (8)$$

که در آن  $D(x, r)$  دیسکی با شعاع  $r$  به مرکز  $x$  می‌باشد. این قید بیان می‌کند که در صورتی که سرعت نسبی ربات و مانع در مخروطی که رأس آن روی مرکز ربات و یال‌های آن مماس بر دایره‌ای به مرکز مانع و شعاع حاصل جمع شعاع مانع و ربات می‌باشد، قرار نگیرد، ربات در موقعیت بعدی خود حتماً با مانع برخورد نخواهد داشت. در واقع، قید بیان شده قیدی محافظه کارانه برای برخورد در نظر گرفته است. براساس این روش، روش‌های جدیدتری نیز همچون مخروط تصادم سرعت دوسویه [31] برای مسیریابی گروهی ربات‌ها و یا روش‌های همچون مخروط تصادم سرعت توسعه یافته

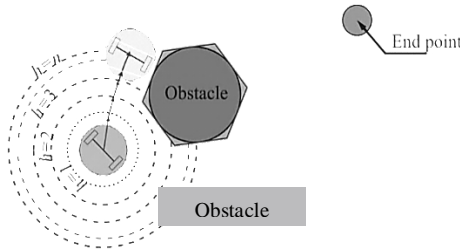


Fig. 2 General overview of the proposed algorithm based on RHC  
شکل 2 شمای کلی از روش ارائه شده بر مبنای کنترل افق پیش‌بین

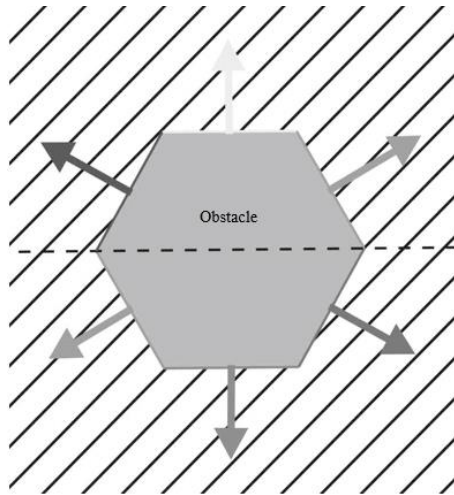


Fig. 3 Feasible perimeter represented by hatched area.  
شکل 3 محدوده جواب قابل قبول اطراف مانع که با هاشور مشخص شده است.

از خطوط 6 گانه‌ی مشخص شده در "شکل 3" این فضا قابل دسترس خواهد بود. به عبارت دیگر:

$$A\xi > b + (v - 1)M \quad (13)$$

$$\sum_{i=1}^s v_i(1) \geq 1 \quad (14)$$

که در آن  $v = (v_1, \dots, v_s)$  برداری از متغیرهای دودویی  $v_i \in \{0, 1\}$  و  $M$  یک عدد ثابت بزرگ است که در روش عدد-بزرگ استفاده می‌شود [34]. معادله (14) تضمین می‌کند که حداقل یکی از المان‌های بردار  $v$  یک بوده و در نتیجه نقطه  $\xi$  خارج از چندوجهی قرار گیرد. این روش، یکی از روش‌های متداول برای حل مسائل نامحدب است که به برنامه‌نویسی عدد صحیح مختلط شهرت دارد و در این پژوهش نیز استفاده شده است.

### 3-3- قیود سرعت بیشینه

تنها قید سینماتیکی در نظر گرفته شده در این پژوهش، محدودیت در سرعت‌ها می‌باشد که با  $v_{\max}$  نشان داده می‌شود. بنابراین می‌توان نوشت:

$$\|z_e(k+1) - z_e(k)\| \leq v_{\max} \Delta t, \quad k = 1, \dots, (h-1) \quad (15)$$

که در آن  $\Delta t$  بازه زمانی بین دو نقطه بوده که مقداری ثابت است.

### 3-4- مسئله بهینه کلی

با در نظر گرفتن توضیحات ارائه شده برای هر قسمت، مساله مسیریابی و کنترل یک ربات متحرک به صورت زیر خواهد بود:

$$\min \sum_{k=1}^h \|z_e(k+1) - z_e(k)\|_2^2 + \omega \|z_e(h) - z_g\|_2^2 \quad (16\text{-الف})$$

$$\text{s.t. } A_{v0}(1)z_e(1) > b_{v0}(1) + (v(1) - 1)M, \quad (16\text{-ب})$$

### 3-1- تابع هزینه

تابع هزینه می‌تواند شامل قسمت‌های متفاوتی باشد و با در نظر گرفتن هدف مساله تعیین می‌گردد. هدف مساله می‌تواند رسیدن به نقطه پایانی در کمترین زمان، طی کوتاه‌ترین مسیر و یا صرف کمترین انرژی در طول مسیر رسیدن به نقطه پایانی باشد. در این پژوهش، قید رسیدن به مقصد پایانی و طی کوتاه‌ترین مسیر قیود با اختصاص یک درصد وزنی مشخص برای الویت‌دهی قیود در نظر گرفته برای حل مسئله می‌باشد. اما با توجه به ساختار حل بهینه‌سازی محدب و محدب سازی قیود برای مسئله، نتایج آن می‌توان برای دیگر هدف‌ها نیز استفاده گردد. از آنجایی که از ساختار برنامه‌ریزی گسسته استفاده می‌شود، طول مسیر طی شده را می‌توان با جمع فواصل بین نقاط روی مسیر محاسبه نمود. به عبارت دیگر، برای یک ربات متحرک (که به صورت یک نقطه در نظر گرفته می‌شود) برای یک افق پیش‌بینی به طول  $h$  می‌توان تابع هزینه کوتاه‌ترین مسیر را به صورت زیر نوشت:

$$\min \sum_{k=1}^h \|z_e(k+1) - z_e(k)\|_2^2 \quad (10)$$

که در آن  $z_e(k) \in \mathbb{R}^2$  مختصات ربات متحرک در مرحله  $k$  ام می‌باشد. علاوه بر این، از آنجایی که از مفهوم افق پیش‌بین استفاده می‌گردد، یک تابع هزینه کمکی برای تضمین حرکت به سوی نقطه پایانی لازم است. این قسمت به صورت فاصله نقطه آخر افق پیش‌بین با نقطه پایانی در نظر گرفته می‌شود، به عبارت دیگر:

$$\min \|z_e(h) - z_g\|_2^2 \quad (11)$$

که در آن  $z_g$  مختصات نقطه پایانی را نشان می‌دهد. شمای کلی طراحی مسیر برای ربات را می‌توانید در "شکل 2" مشاهده نمایید.

### 3-2- قیود اجتناب از مانع

همان‌گونه که بیان گردید، قیود مخروط تصادم سرعت و عدم تقاطع جزو قیود نامحدب می‌باشند. از این رو با کمک روش برنامه نویسی مخلوط عدد صحیح خطی برای حل مسئله استفاده مبادرت خواهد شد. به این منظور، در اولین قدم برای محدب‌سازی مسئله بهینه‌سازی برای قیود عدم تقاطع، تمام موانع، حتی آنهایی که محدب نیستند، به صورت شکل محدبی که آن‌ها را در بر می‌گیرد، در نظر گرفته می‌شوند. در این راستا، هر مانع با یک چندوجهی محیطی (در حالت صفحه‌ای، چندضلعی محیطی) تخمین زده می‌شود. حال تمامی قیود (هم قیود مرتبط با مخروط تصادم سرعت و هم قیود عدم تقاطع) به صورت قیود خطی تبدیل می‌گردند. کفایت حداقل یکی از این قیود خطی در هر افق ارضا شود تا ربات یک حرکت عاری از برخورد داشته باشد. لازم به ذکر است که این کار باعث محافظه کارانه‌تر شدن جواب شده و ایمنی بالاتری را برقرار می‌کند. با توجه به این موضوع که مخروط تصادم سرعت را می‌توان نوعی از چندوجهی دانست، تنها روابط مربوط به چندوجهی برای رعایت اختصار توضیح داده می‌شود. در این راستا، یک چندوجهی در فضا به صورت تقاطع یک دسته نیم فضا تعریف می‌شود:

$$\sigma = \{\xi | A\xi < b\} \quad (12)$$

در معادلات بالا،  $A$  ماتریسی با ردیف‌های  $A_k$ ،  $k \in \{1, \dots, s\}$  و همچنین  $b = (b_1, \dots, b_s)$  بردار مربوط به نیم‌فضاها می‌باشند، که  $s$  تعداد وجه‌های چندوجهی را نشان می‌دهد. نقطه  $\xi$  در صورتی خارج از چندوجهی  $\sigma$  قرار می‌گیرد که حداقل یکی از نامعادلات  $A\xi < b$  ارضا گردد. با توجه به شکل فضای عدم برخورد، فضای هاشور خورده می‌باشد که با ارضای هر یک

که همان سرعت‌های خطی و زاویه‌ای ربات هستند با استفاده از معادلات (29)-(31) به‌دست می‌آیند که در آن  $\Delta t$  بازه‌ی زمانی بوده و  $dy_e(1)$  و  $dx_e(1)$  مقادیر تغییر مختصات از مرحله کنونی تا مرحله بعد و  $\theta_e$  زاویه حرکت را نشان می‌دهند. همچنین  $v$  و  $\omega$  به ترتیب بیان‌گر سرعت‌های خطی و زاویه‌ای می‌باشند. با اعمال این مقادیر کنترلی به مدل سینماتیکی ربات که در قسمت‌های قبل به آن اشاره گردید، مقادیر جدید متغیرهای حالت سیستم محاسبه می‌شوند که همان ورودی‌های مساله بهینه‌سازی در مرحله بعد می‌باشند. این مراحل تا جایی تکرار می‌شوند که ربات به اندازه کافی به نقطه پایانی نزدیک شود. در انتها نیز مقادیر کنترلی برای رسیدن به نقطه پایانی محاسبه شده و ربات مستقیماً به آن جهت حرکت می‌کند.

$$\phi(i) = \frac{2\pi i}{s}, \quad \forall i = 1, \dots, s, \quad (19)$$

$$m_{\sigma}(i) = \tan(\phi(i) + \frac{\pi}{2} - \frac{\pi}{s}), \quad \forall i = 1, \dots, s, \quad (20)$$

$$x_{aux}(i) = \alpha \times r_{\sigma} \times \cos(\phi(i)) + x_{\sigma}, \quad \forall i = 1, \dots, s, \quad (21)$$

$$y_{aux}(i) = \alpha \times r_{\sigma} \times \sin(\phi(i)) + y_{\sigma}, \quad \forall i = 1, \dots, s, \quad (22)$$

$$b_{\sigma} = y_{aux}(i) - m_{\sigma}(i) \times x_{aux}(i), \quad \forall i = 1, \dots, s, \quad (23)$$

$$R_e = \alpha(r_e + r_{\sigma}), \quad l = x_m - x_{\sigma}, \quad (24)$$

$$q_0 = x_{\sigma} + \frac{R_e^2}{||l||}, \quad t_d = \frac{R_e \sqrt{||l|| - R_e}}{||l||} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} l, \quad (25)$$

$$q_1 = q_0 + t_d, \quad q_2 = q_0 - t_d \quad (26)$$

$$\alpha_1 = \tan^{-1}(q_1 - x_m), \quad \alpha_2 = \tan^{-1}(q_2 - x_m), \quad (27)$$

$$A_{vo} = \begin{bmatrix} \alpha_1 & 1 \\ \alpha_2 & 1 \end{bmatrix}, \quad b_{vo} = \begin{bmatrix} \alpha_1 v_{relx} - v_{rely} \\ \alpha_2 v_{relx} - v_{rely} \end{bmatrix} \quad (28)$$

$$\theta_e = \tan^{-1}\left(\frac{dy_e(1)}{dx_e(1)}\right), \quad (29)$$

$$v = \frac{dy_e(1)}{\Delta t \times \sin \theta_e}, \quad (30)$$

$$\omega = \frac{\theta_e - \theta_{e0}}{\Delta t \times \sin \theta_e}. \quad (31)$$

### 5- شبیه‌سازی روش ارائه شده

در این بخش، روش ارائه شد برای دو داستان محیط ناشناخته‌ی ایستا و محیط ناشناخته‌ی پویا در محیط شبیه‌ساز به کار گرفته می‌شود و نتایج شبیه‌سازی ارائه می‌گردد. تعداد افق‌های در نظر گرفته شده 5 و بازه‌های زمانی 0.2 ثانیه و نقطه‌ی پایانی  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$  می‌باشد.

#### 5-1- نتایج شبیه‌سازی مسیریابی ربات متحرک دوچرخ با مانع ثابت

مساله مسیریابی با مقادیر ورودی و محدودیت‌های بیان شده، برای تابع هزینه کوتاه‌ترین مسیر حل می‌گردد که نتیجه آن در "شکل 4" مشاهده می‌شوند. نتایج به دست آمده نشان می‌دهند که طول مسیر طی شده در این حالت برابر 1.4764 متر و زمان طی آن 28.05 ثانیه بوده است. همچنین زمان حل مساله بهینه برای این حالت، حدود 0.45 ثانیه بوده است. علت این زمان بالا هر بار اجرای نرم افزار سی‌وی‌ایکس برای حل مسئله هست که با توجه به پیاده‌سازی این الگوریتم با گورویی در محیط رآس انتظار می‌رود که این زمان در پیاده‌سازی به کمتر از 0.01 ثانیه کاهش یابد که برای این کاربرد بلادرنگ می‌باشد. موقعیت‌های در نظر گرفته شده برای موانع و اندازه‌ی آن‌ها در جدول 1 ارائه شده است. لازم به ذکر است که از نتایج به‌دست آمده در این شبیه‌سازی برای مقایسه روش ارائه شده با روش‌های مرسوم معرفی شده در مراجع استفاده خواهد شد.

#### 5-2- نتایج شبیه‌سازی مسیریابی ربات متحرک دوچرخ با مانع متحرک

مانند حالت قبل، بار دیگر مساله مسیریابی برای تابع هزینه کوتاه‌ترین مسیر

$$\sum_{i=1}^2 v_i(1) \geq 1, \quad (ج-16)$$

$$A_{IO}(l)z_e(l) > b_{IO}(l) + (j(l) - 1)M, \quad (د-16)$$

$$\sum_{i=1}^s j_i(1) \geq 1, \quad (ه-16)$$

$$z_e(1) = z_s, \quad (و-16)$$

$$|z_e(l+1) - z_e(l)| \leq v_{max} \Delta t \quad \forall l \in \{2, \dots, h\}. \quad (ز-16)$$

همان‌طور که پیش‌تر اشاره شد، در معادلات بالا،  $z_e(l) \in \mathbb{R}^2$  مختصات ربات متحرک در افق  $l$  و  $z_g$  مختصات نقطه پایانی را نشان می‌دهد. لازم به ذکر است برای این مساله کوتاه‌ترین مسیر و شرط حرکت به سمت مقصد نهایی به‌عنوان تابع هزینه نوشته شده و برای هر یک ضریب وزنی در نظر گرفته شده است. در نظر گرفتن این ضرایب به‌عنوان پارامتر بهینه‌سازی باعث تغییر فرم از معادلات مربع کامل به معادلات درجه سه می‌گردد. به‌منظور جلوگیری از این اتفاق، این ضرایب به‌صورت ضریب ثابت در نظر گرفته می‌شوند و در بهینه‌سازی مورد استفاده قرار می‌گیرند. به‌دلیل تفاوت زیاد مقادیر  $\|z_e(h) - z_g\|_2^2$  با  $\sum_{k=1}^h \|z_e(k+1) - z_e(k)\|_2^2$  در نظر گرفته شده باید در وهله اول این دو مقدار را بی‌بعدسازی نمود. همچنین در شرایطی که ربات به موانع نزدیک می‌گردد وزن بیشتر باید به مقادیر مربوط به اجتناب از مانع داده شود و حرکت به سمت هدف از وزن کمتری برخوردار باشد.

$$\omega = \omega_0 \times \omega_1 \quad (17)$$

$$\omega_0 = \frac{v_{max} \Delta t}{\|z_e(h) - z_g\|_2}, \quad \omega_1 = \xi \sqrt{\|z_e(1) - z_{NO}\|_2^2} \quad (18)$$

مقادیر  $\omega_0$  و  $\omega_1$  از معادله (18) محاسبه می‌شوند. در این معادلات  $\xi$  یک عدد مثبت می‌باشد و مقدار نسبت وزن مسیریابی بدون برخورد به حرکت به سمت هدف را مشخص می‌نماید. همچنین  $Z_{NO}$  موقعیت نزدیکترین مانع به ربات است. با در نظر گرفتن  $\xi = 10$  در این پژوهش شبیه‌سازی و پیاده‌سازی‌ها ادامه خواهد یافت. علاوه بر این، معادلات (16-ب) تا (16-ه) عدم برخورد با مانع را تضمین می‌کنند. قید (16-و) نقطه‌ی ابتدایی را مشخص و (16-ز) سرعت بیشینه را محدود می‌کند. در ادامه از این مسئله بهینه‌سازی به منظور مسیریابی و کنترل ربات متحرک استفاده شده و برای یک ربات دوچرخ شبیه‌سازی و پیاده‌سازی می‌گردند.

#### 4- معادلات حاکم بر قیود

در الگوریتم مورد استفاده در این مقاله ابتدا مقادیر ثابت موردنیاز، شامل طول افق پیش‌بینی  $h$ ، شعاع دایره محیطی ربات  $r_e$ ، سرعت بیشینه مجاز  $v_{max}$ ، تعداد وجه‌های مورد استفاده در محدب‌سازی موانع  $s$ ، پارامتر مناسب برای روش عدد-بزرگ  $M$ ، دقت موردنظر در رسیدن به نقطه پایانی  $\epsilon$ ، و پارامترهای مربوط به موانع  $(x_{\sigma}, y_{\sigma}, r_{\sigma})$  و سرعت‌های نسبی موانع نسب به ربات  $v_{rel}$  تعیین می‌گردند. لازم به ذکر است پارامترهای مربوط به مانع به صورت بر خط به آن منتقل می‌گردند که با توجه به این موضوع، ماهیت ناشناخته بودن محیط بر مسئله باقی می‌ماند. سپس با استفاده از معادلات (19)-(23) پارامترهای لازم برای نیم‌فضاهای تشکیل‌دهنده چندوجهی، که در مساله بهینه‌سازی به کار می‌روند، مشخص می‌گردند. همچنین با معادلات (28)-(24) پارامترهای مربوط به فضای عدم برخورد در روش مخروط تصادم سرعت به‌دست می‌آید. با مشخص شدن تمام پارامترها، مساله بهینه‌سازی در هر مرحله حل شده و از خروجی‌های آن برای تعیین مسیر استفاده می‌گردد. در این قسمت اولین درایه از مقادیر به دست آمده برای افق پیش‌بینی به عنوان مقادیر مرجع در نظر گرفته شده و زاویه حرکت و متغیرهای کنترلی

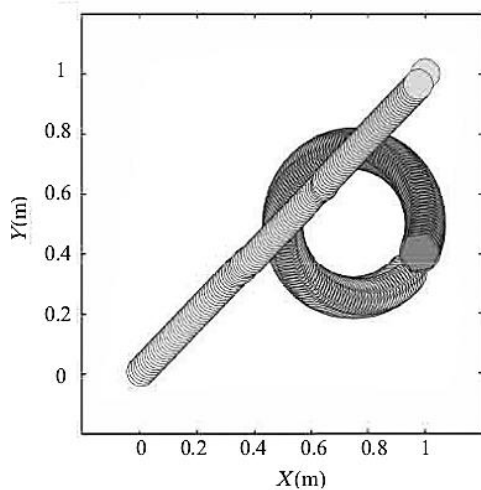


Fig. 7 Obtained result in the dynamic environment, 3<sup>rd</sup> scenario

شکل 7 نتیجه به دست آمده برای محیط پویا، حالت سوم

در حالت اول، که مانع متحرک بوده ولی نسبت به حالت بعدی حرکت خطی کندی دارد ربات متحرک با رسیدن به آن از مقابل مانع متحرک عبور کرده تا مسیر کوتاه‌تری را طی کند. در این حالت، طول مسیر طی شده برابر 1.441 متر و زمان طی آن 22.6 ثانیه بوده است و نتیجه آن در "شکل 5" مشاهده می‌گردد.

در حالت دوم، همان حرکت خطی با سرعت بیشتر تکرار می‌گردد. در این حالت ربات به سمت خلاف جهت حرکت مانع رفته و از پشت مانع عبور می‌کند. در این حالت، طول مسیر طی شده برابر 1.432 متر و زمان طی آن 21.4 ثانیه بوده است و نتیجه آن در "شکل 6" ارائه شده است.

در حالت سوم، مانع به صورت یک حرکت دایره‌ای حرکت می‌کند که در مجموع دو بار در مقابل ربات قرار می‌گیرد. در این حالت ربات در دو حالت برخورد با مانع با توجه به حرکت مانع ابتدا برای عبور مانع سرعت خود را کاهش می‌دهد و سپس در برخورد دوم برای انتخاب حرکت خود از مقابل یا پشت مانع، حرکت از پشت مانع را انتخاب کرده و در نتیجه به مقصد می‌رسد. در این حالت، طول مسیر طی شده برابر 1.415 متر و زمان طی آن 20 ثانیه بوده است و نتیجه آن در "شکل 7" مشاهده می‌گردد. همچنین موقعیت و پارامترهای مربوط به موانع در این حالات در جدول 2 ارائه شده است. لازم به ذکر است که از نتایج حالت سوم به منظور مقایسه روش با روش‌های دیگر استفاده خواهد شد.

جدول 1 موقعیت موانع و اندازه‌ی شعاع آن‌ها در محیط ایستا

Table 1 Position and radius of the obstacles in the static environment

شعاع	موقعیت	شماره مانع
0.1	[0.15 0.25]	1
0.15	[0.6 0.5]	2

جدول 2 موقعیت موانع و اندازه‌ی شعاع آن‌ها در محیط پویا

Table 2 Positions and radiuses of the obstacles in the dynamic environments

شعاع	موقعیت	شماره حالت
0.06	$\begin{bmatrix} 0.4 + 0.25t \\ 0.6 - 0.25t \end{bmatrix}$	1
0.06	$\begin{bmatrix} 0.1 + 0.45t \\ 1 - 0.45t \end{bmatrix}$	2
0.05	$\begin{bmatrix} -0.25 \cos(0.15t - 2) + 0.7 \\ -0.25 \sin(0.15t - 2) + 0.4 \end{bmatrix}$	3

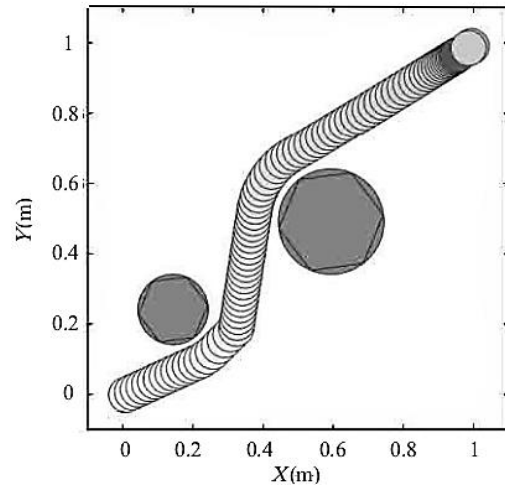


Fig. 4 Obtained result in the static environment

شکل 4 نتیجه بدست آمده برای محیط ایستا

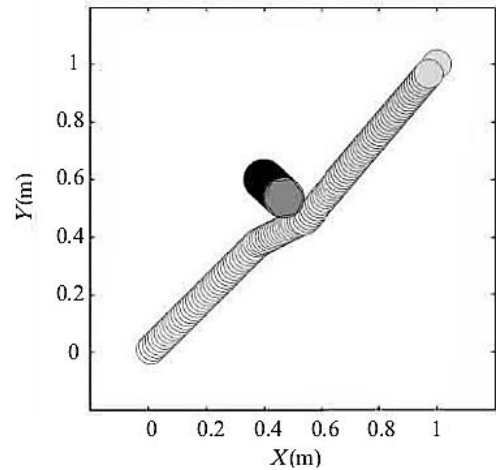


Fig. 5 Obtained result in the dynamic environment, 1<sup>st</sup> scenario

شکل 5 نتیجه به دست آمده برای محیط پویا، حالت اول

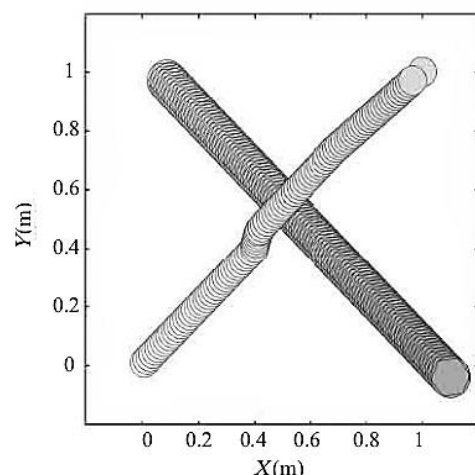


Fig. 6 Obtained result in the dynamic environment, 2<sup>nd</sup> scenario

شکل 6 نتیجه بدست آمده برای محیط پویا، حالت دوم

با همان مقادیر ورودی و محدودیت‌ها حل می‌شود. به منظور نمایش کارایی الگوریتم ارائه شده، در سه حالت موانع پویا معرفی می‌گردند. در این حالات پویا، زمان حل مساله بهینه‌سازی در شبیه‌سازی حدود 0.43 ثانیه می‌باشد.

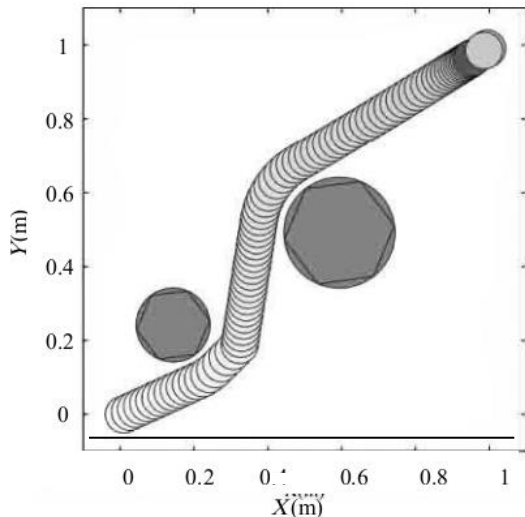


Fig. 8 Proposed method

شکل 8 روش ارائه شده در این مقاله

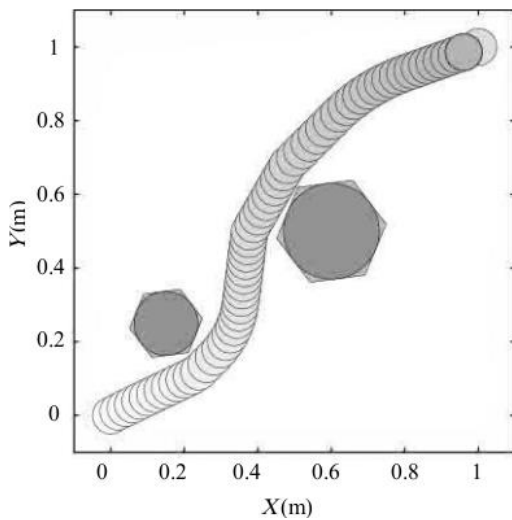


Fig. 9 Reported result in [1]

شکل 9 روش گزارش شده در [1]

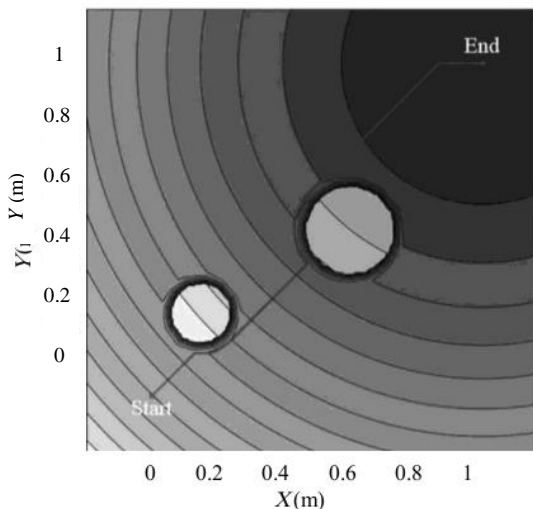


Fig. 10 Potential field algorithm

شکل 10 روش میدان پتانسیل مجازی

### 3-5- مقایسه عملکرد روش ارائه شده

به منظور مقایسه عملکرد روش ارائه شده، نتایج به دست آمده برای شبیه‌سازی با دو مانع ثابت و یک مانع متحرک با حرکت دایره‌ای با نتایج گزارش شده در [1] که یک الگوریتم برای مسیریابی در محیط ناشناخته پویا می‌باشد، مقایسه شده است. همچنین در محیط ایستا نتایج با نتایج به دست آمده با روش میدان پتانسیل مجازی و در محیط پویا با الگوریتم ژنتیک ارائه شده در مقاله [35] که به صورت برون خط اعمال شده اند مقایسه می‌گردد. به منظور مقایسه با نتایج ارائه شده در [1] در حالت ایستا بیشینه سرعت ربات 0.05 و در حالت پویا 0.01 متر بر ثانیه در نظر گرفته شده‌اند. نتایج در "شکل‌های 8-13" مشاهده می‌گردند. همچنین معادلات حرکت و موقعیت موانع در جدول‌های 1 و 2 به ترتیب برای محیط ایستا و پویا ارائه شده‌اند. با توجه به شبیه‌سازی‌های صورت گرفته، نتایج به دست آمده نشان می‌دهد که روش ارائه شده از قابلیت‌های بهتری در مقایسه با روش‌های مطرح شده به منظور مقایسه بهره می‌برد. جدول 3 دربرگیرنده‌ی اطلاعات مربوط به طول مسیر طی شده و زمان پیموده شدن مسیر برای مقایسه روش‌ها است.

### 6- پیاده‌سازی روش مسیریابی و کنترل بر روی ربات ای‌پاک

با توجه به نتایج مطلوب به دست آمده از شبیه‌سازی، روش به دست آمده بر روی ربات متحرک واقعی ای‌پاک پیاده‌سازی می‌گردد و نتایج آن گزارش می‌شود. فریم‌هایی از پیاده‌سازی در انتهای مقاله در "شکل‌های 16 و 17" آورده شده‌اند. لازم به ذکر است که به علت عدم ارسال همزمان درست داده‌ها توسط ربات‌های موجود در آزمایشگاه، امکان راه‌اندازی همزمان دو ربات متحرک (یکی به عنوان ربات و یکی به عنوان مانع متحرک) میسر نشد و تست‌ها صرفاً در محیط ایستا انجام گرفته است. همچنین نقطه پایان در پیاده‌سازی‌ها نقطه  $[0.9, 0.9]$  می‌باشد. پارامترهای مربوط به موانع در جدول 4 ارائه شده است.

### 6-1- حالت اول، نتایج پیاده‌سازی الگوریتم مسیریابی بر روی ربات ای‌پاک با یک مانع ثابت

در پیاده‌سازی نیز پارامترهای الگوریتم طراحی مسیر شبیه به شبیه‌سازی در نظر گرفته شده. کل الگوریتم در داخل محیط گوروبی به زبان سی‌پلاس پلاس نوشته شده و زمان حل مسئله در این حالت در بیشترین زمان به 0.0039 ثانیه رسید که برای ربات مورد استفاده بلادرنگ بود. همین سرعت بالا با سرعت ارسال و دریافت داده‌های ربات متحرک ایجاد مشکل می‌نمود. در واقع سرعت مناسب به منظور ارسال و دریافت برای این ربات چیزی حدود

جدول 3 مقایسه‌ی مسیر طی شده و زمان مورد نیاز مربوط به روش‌های گزارش شده.

Table 3 Corresponding traversed path and required time with the reported algorithms

روش	محیط	مسیر طی شده (متر)	زمان مورد نیاز (ثانیه)
ارائه شده	ایستا	1.4764	28.05
	پویا	1.4059	154.49
ارائه شده	ایستا	1.49	28
در [1]	پویا	1.46	153.68
میدان	ایستا	1.53	34.7
پتانسیل	پویا	-	-
ژنتیک	ایستا	-	-
	پویا	1.83	188.65



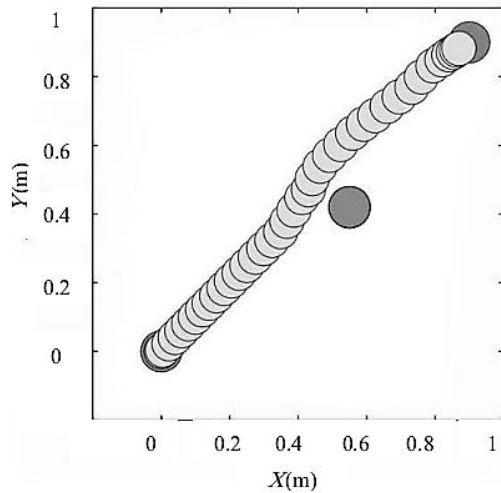


Fig. 14 Obtained result in the implementation, 2<sup>nd</sup> scenario

شکل 14 نتایج به دست آمده برای پیاده‌سازی، حالت دوم

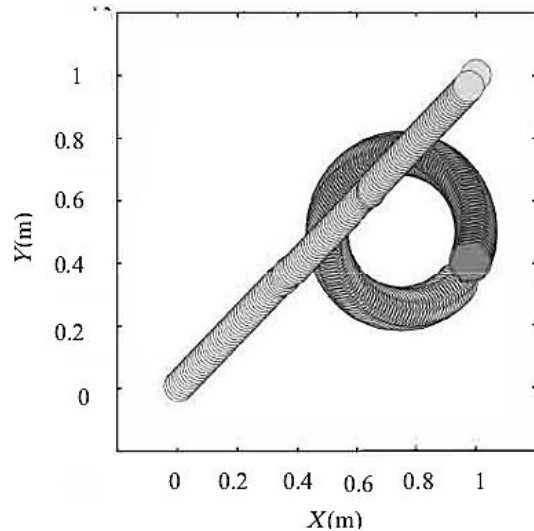


Fig. 11 Proposed method

شکل 11 روش ارائه شد در این مقاله

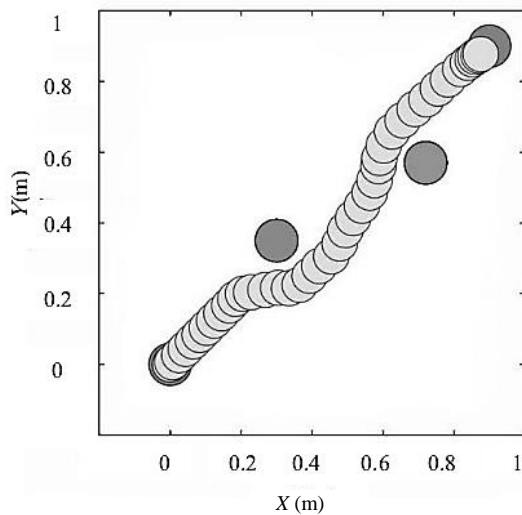


Fig. 15 Obtained result in the implementation, 2<sup>nd</sup> scenario

شکل 15 نتایج به دست آمده برای پیاده‌سازی، حالت دوم

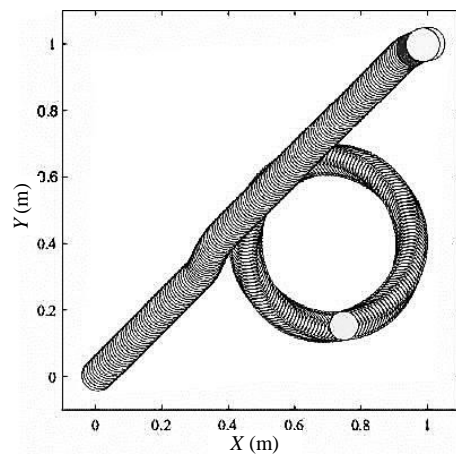


Fig. 12 Reported in [1]

شکل 12 روش گزارش شده در [1]

بسته‌ی نرم افزاری معرفی شده توسط شرکت سازنده ربات ای‌پاک در محیط رأس استفاده می‌شود که با ارسال داده‌های موردنیاز واحد طراحی مسیر و دریافت دستورات از این واحد عملیات تمام خودکار انجام می‌شود. نتایج پیاده‌سازی در "شکل 14" مشاهده می‌شوند.

با در نظر گرفتن مقادیر جدول به عنوان ورودی، طول افق پیش‌بینی 5 و سرعت بیشینه مجاز 0.05 متر بر ثانیه در این پیاده‌سازی با در نظر گرفتن بازه زمانی بین نقاط 0.02 طول مسیر به دست آمده برابر 1.219 متر و زمان طی آن 21.92 ثانیه می‌باشد.

#### 6-2- حالت دوم، نتایج پیاده‌سازی الگوریتم مسیریابی بر روی ربات

##### ای‌پاک با دو مانع ثابت

برای این حالت، با در نظر گرفتن ورودی‌های جدول 4 و همچنین طول افق پیش‌بینی 5 و سرعت بیشینه مجاز 0.05 متر بر ثانیه، طول مسیر طی شده برابر با 1.312 متر و زمان طی مسیر برابر با 23.22 ثانیه می‌باشد. همچنین، زمان مورد نیاز برای هر تکرار نسبت به پیاده‌سازی قبلی تغییر محسوسی نداشت. نتایج پیاده‌سازی در "شکل 15" قابل ملاحظه می‌باشد.

به‌طور خلاصه می‌توان روند اجرایی پی‌گرفته شده در پیاده‌سازی روش

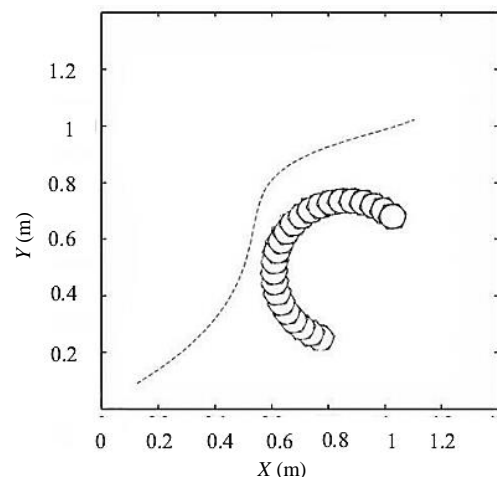


Fig. 13 Genetic algorithm

شکل 13 روش ژنتیک

0.01 ثانیه می‌باشد. به همین علت در کد مذکور به‌صورت دستی ایجاد تاخیر کرده تا ربات بتواند داده‌های موردنظر را ارسال و دستورات تولید شده در واحد طراحی مسیر را دریافت نماید. داده‌های برای پیاده‌سازی از طریق



گروبی به زبان سی‌پلاس‌پلاس و بهره بردن از بسته‌ی نرم‌افزاری موجود ربات ای‌پاک در محیط رأس سرعت انجام هر قسمت به حدود 0.004 ثانیه

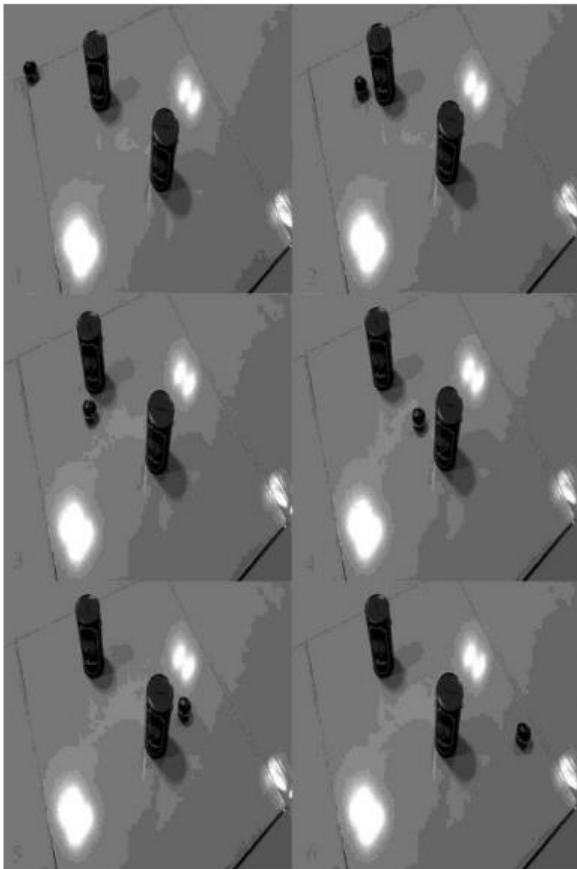


Fig. 17 Practical implementation of the proposed algorithm, 2<sup>nd</sup> scenario  
شکل 17 پیادسازی عملی روش ارائه شده، حالت دوم

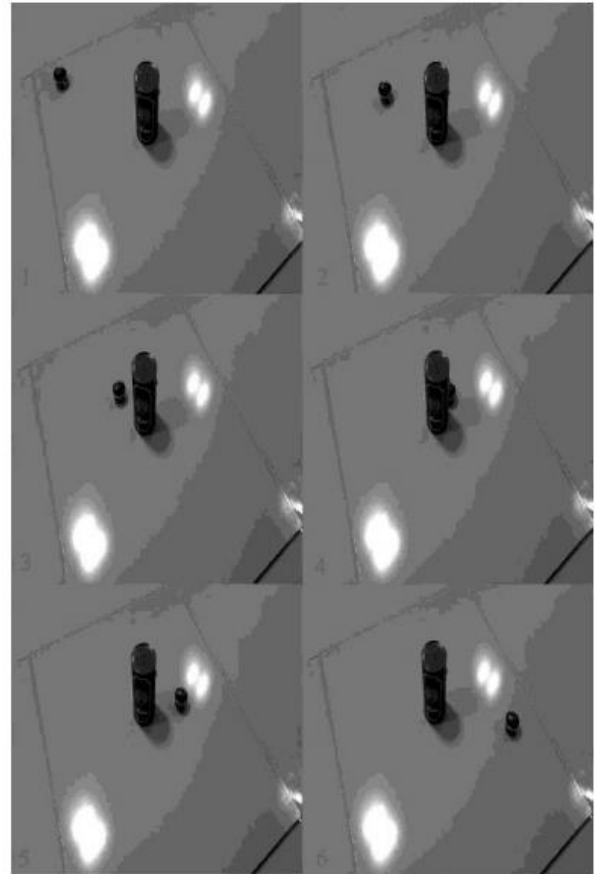


Fig. 16 Practical implementation of the proposed algorithm, 1<sup>st</sup> scenario  
شکل 16 پیادسازی عملی روش ارائه شده، حالت اول

جدول 4 موقعیت موانع و اندازه‌ی شعاع آن‌ها برای پیاده‌سازی

**Table 4** Positions and radiuses of the obstacles in the implementation

شماره حالت	موقعیت مانع اول	موقعیت مانع دوم	شعاع
1	$[0.55 \pm 0.01]$ $[0.42 \pm 0.01]$	-	0.06
2	$[0.33 \pm 0.01]$ $[0.37 \pm 0.01]$	$[0.75 \pm 0.01]$ $[0.58 \pm 0.01]$	0.06

ارائه شده را در "شکل 18" مشاهده کرد.

### 7- نتیجه گیری

در این مقاله، مسیریابی و کنترل ربات متحرک با استفاده از مفاهیم بهینه‌سازی محدب، ساختار برنامه‌ریزی گسسته و برنامه‌نویسی مختلط عدد صحیح، مخروط تصادم سرعت، مفهوم روش باگ و کنترل پیش‌بین مبتنی بر مدل مورد بحث و بررسی قرار گرفت. ابتدا معرفی کوتاهی از ربات متحرک مورد مطالعه به منظور استخراج معادلات سینماتیکی حاکم بر آن صورت گرفت. پس از آن مساله مسیریابی یک ربات متحرک به صورت ریاضی بیان شده و شبه‌کد الگوریتم آن ارائه شد. با استفاده از الگوریتم پیشنهادی و به کارگیری بسته نرم‌افزاری سی‌وی‌ایکس در نرم‌افزار متلب، ابتدا شبیه‌سازی برای یک ربات متحرک دوچرخ انجام شده و با به‌دست آمدن نتایج مطلوب، الگوریتم بر روی ربات واقعی پیاده‌سازی گردید. نتایج به دست آمده در شبیه‌سازی با نتایج گزارش شده بر مبنای روش‌های مشابه یا شناخته شده مقایسه شد و نتایج به‌دست آمده نشان دهنده‌ی این بود که این روش توانایی بهتری نسبت به روش‌های مقایسه شده برای انجام عملیات موردنظر را دارد. به منظور پیاده‌سازی با استفاده از نرم‌افزار حل مسائله بهینه‌سازی محدب

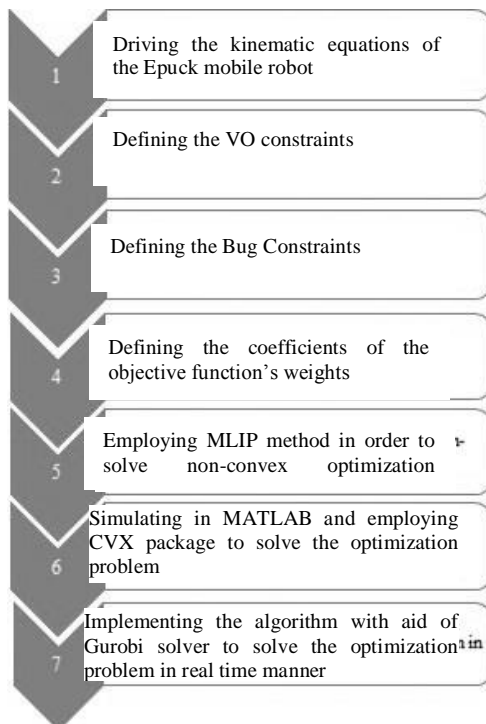


Fig. 18 The flowchart of the proposed method  
شکل 18 فلوجارت مربوط به روش ارائه شده

رسید.

با توجه به محدودیت‌های ربات موجود، با اعمال تأخیر دستی پیاپی - سازی‌ها برای حالت ایستا انجام گرفت. در ادامه، پژوهش‌های در حال انجام شامل تلفیق الگوریتم معرفی شده برای انجام کنترل مشترک با استفاده از یک واسط لامسه‌ای، پیاده‌سازی روش برای ربات‌های موازی و سری و محدب سازی قیود با استفاده از روش مک‌کورمیک می‌باشند.

## 8- مراجع

- [16] P. Fiorini, Z. Shiller, Motion planning in dynamic environments using velocity obstacles, *The International Journal of Robotics Research*, Vol. 17, No. 7, pp. 760-772, 1998.
- [17] G. S. Chyan, S. Ponnambalam, Obstacle avoidance control of redundant robots using variants of particle swarm optimization, *Robotics and Computer-Integrated Manufacturing*, Vol. 28, No. 2, pp. 147-153, 2012.
- [18] Y. Hu, S. X. Yang, A knowledge based genetic algorithm for path planning of a mobile robot, *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, IEEE*, pp. 4350-4355, 2004.
- [19] M. Shahbazzadeh, S. J. Sadati Rostami, S. Minagar, Designing model predictive control for trajectory tracking of a wheeled mobile robot using lager functions and exponential data weighting, *Modares Mechanical Engineering*, Vol. 17, No. 10, pp. 21-28, 2017. (in Persian فارسی)
- [20] T. Howard, C. Green, A. Kelly, Receding horizon model-predictive control for mobile robot navigation of intricate paths, *Field and Service Robotics, Springer Berlin/Heidelberg*, pp. 69-78, 2010.
- [21] C. H. Hsieh, J. S. Liu, Nonlinear model predictive control for wheeled mobile robot in dynamic environment, *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pp. 363-368, 2012.
- [22] P. Vadakkepat, K. C. Tan, W. Ming-Liang, Evolutionary artificial potential fields and their application in real time robot path planning, *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, IEEE*, Vol:1, pp. 256-263, 2000.
- [23] L. Blackmore, B. Williams, Optimal manipulator path planning with obstacles using disjunctive programming, *American Control Conference, IEEE*, pp. 1-3, 2006.
- [24] H. Ding, M. Zhou, O. Stursberg, Optimal path planning in the workspace for articulated robots using mixed integer programming, *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE*, pp. 5770-5775, 2009.
- [25] H. Ding, G. Reißig, O. Stursberg, Increasing efficiency of optimization-based path planning for robotic manipulators, *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, IEEE*, pp. 1399-1404, 2011.
- [26] G. Optimization, Inc., Gurobi optimizer reference manual, 2014, URL: <http://www.gurobi.com>, 2014.
- [27] I. Kamon, E. Rivlin, Sensory-based motion planning with global proofs, *IEEE transactions on Robotics and Automation*, Vol. 13, No. 6, pp. 814-822, 1997.
- [28] J. Alonso-Mora, T. Naegeli, R. Siegwart, P. Beardsley, Collision avoidance for aerial vehicles in multi-agent scenarios, *Autonomous Robots*, Vol. 39, No. 1, pp. 101-121, 2015.
- [29] J. B. Rawlings, K. R. Muske, The stability of constrained receding horizon control, *IEEE Transactions On Automatic Control*, Vol. 38, No. 10, pp. 1512-1516, 1993.
- [30] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli, The e-puck, a robot designed for education in engineering, *Proceedings of the 9th conference on autonomous robot systems and competitions, IPCB: Instituto Politécnico de Castelo Branco*, pp. 59-65, 2009.
- [31] J. Van Den Berg, S. Guy, M. Lin, D. Manocha, Reciprocal n-body collision avoidance, *International symposium on Robotics Rresearch*, pp. 3-19, 2009.
- [32] D. Wilkie, J. Van Den Berg, D. Manocha, Generalized velocity obstacles, *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on IEEE*, pp. 5573-5578, 2009.
- [33] M. Kim, J. H. Oh, Study on optimal velocity selection using velocity obstacle (OVVO) in dynamic and crowded environment, *Autonomous Robots*, Vol. 40, No. 8, pp. 1459-1470, 2016.
- [34] D. Bertsimas, J. N. Tsitsiklis, *Introduction To Linear Optimization*, pp. 13-21. Belmont, MA: Athena Scientific, 1997.
- [35] R. Sadeghian, M. T. Masouleh, Controller tuning based on optimization algorithms of a novel spherical rolling robot, *Journal of Mechanical Science and Technology*, Vol. 30, No. 11, pp. 5207-5216, 2016.
- [1] M. Zarei, R. S. Novin, M. T. Masouleh, Experimental study on optimal motion planning of wheeled mobile robot using convex optimization and receding horizon concept, *Robotics and Mechatronics (ICROM), 2016 4th International Conference on, IEEE*, pp. 386-391, 2016.
- [2] A. Safaryazdi, M. Zarei, O. Abolghasemi, M. Tale Masouleh, Experimental study on the model-based control of a 2-degree-of-freedom spherical parallel robot camera stabilizer based on multi-thread programming concept, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, doi:10.1177/0954406217711469, 2017.
- [3] M. Zamani Fekri, M. Zarei, M. Tale Masouleh, M. Yazdani, Optimal design and fabrication of a 4-dof quattuoraar parallel robot with singularity-free workspace by abc and pso algorithms, *Modares Mechanical Engineering*, Vol. 16, No. 6, pp. 149-158, 2016. (in Persian فارسی)
- [4] A. S. Lafmejani, M. T. Masouleh, A. Kalhor, Position control of a 6-DoF pneumatic gough-stewart parallel robot using backstepping-sliding mode controller, *Modares Mechanical Engineering*, Vol. 17, No. 10, pp. 101-111, 2017. (in Persian فارسی)
- [5] A. Nikoobin, A. Kamel, Time optimal trajectory planning of four bar mechanism using indirect approach, *Modares Mechanical Engineering*, Vol. 16, No. 3, pp. 353-363, 2016. (in Persian فارسی)
- [6] J. C. Latombe Business, *Robot Motion Planning*, pp. 43-51, USA, Springer Science & Media, 2012.
- [7] S. M. LaValle, *Planning Algorithms*, pp. 103-111, UK, Cambridge university press, 2006.
- [8] A. Ismail, A. Sheta, M. Al-Weshah, A mobile robot path planning using genetic algorithm in static environment, *Journal of Computer Science*, Vol. 4, No. 4, pp. 341-344, 2008.
- [9] G. Chesi, Visual servoing path planning via homogeneous forms and LMI optimizations, *IEEE Transactions on Robotics*, Vol. 25, No. 2, pp. 281-291, 2009.
- [10] A. V. Savkin, C. Wang, Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation, *Robotics and Autonomous Systems*, Vol. 62, No. 10, pp. 1568-1580, 2014.
- [11] A. Richards, J. P. How, Aircraft trajectory planning with collision avoidance using mixed integer linear programming, *American Control Conference, 2002. Proceedings of the 2002, IEEE*, vol. 3, pp. 1936-1941, 2002.
- [12] T. Lozano-Pérez, M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, Vol. 22, No. 10, pp. 560-570, 1979.
- [13] O. Takahashi, R. J. Schilling, Motion planning in a plane using generalized Voronoi diagrams, *IEEE Transactions On Robotics And Automation*, Vol. 5, No. 2, pp. 143-150, 1989.
- [14] Y. K. Hwang, N. Ahuja, Gross motion planning—a survey, *ACM Computing Surveys (CSUR)*, Vol. 24, No. 3, pp. 219-291, 1992.
- [15] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal Of Robotics Research*, Vol. 5, No. 1, pp. 90-98, 1986.