



مسیریابی بهینه ربات موازی تریپترون با سه درجه آزادی مستقل خطی با استفاده از بهینه‌سازی محدب و مفهوم افق پیش‌بین

رویا صباغ نوین¹، مهدی طالع ماسوله^{2*}، مجتبی یزدانی³، بهزاد دانایی¹

1- دانشجوی کارشناسی ارشد، مهندسی مکاترونیک، آزمایشگاه تعامل انسان و ربات، دانشکده علوم و فنون نوین، دانشگاه تهران، تهران

2- استادیار، مهندسی مکاترونیک، آزمایشگاه تعامل انسان و ربات، دانشکده علوم و فنون نوین، دانشگاه تهران، تهران

3- دانشجوی کارشناسی ارشد، مهندسی مکانیک، دانشگاه صنعتی امیرکبیر، تهران

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

چکیده

اطلاعات مقاله

مقاله پژوهشی کامل

دریافت: 20 اسفند 1393

پذیرش: 20 اردیبهشت 1394

ارائه در سایت: 07 تیر 1394

کلید واژگان:

مسیریابی بهینه

ربات موازی

بهینه‌سازی محدب

برنامه‌ریزی گسسته

مفهوم افق پیش‌بین

با توجه به افزایش چشمگیر تعامل انسان و ربات، به نظر می‌رسد یکی از چالش‌های امروزی، مسأله مسیریابی عاری از تداخل ربات‌ها به صورت بلادرنگ و با تضمین ایمنی می‌باشد. تاکنون روش‌های متعددی برای مسیریابی ربات‌های متحرک و سری ارائه شده است که هر یک مزایا و معایب خود را دارند. با این حال مطالعات کمی بر روی مسیریابی ربات‌های موازی به ویژه در حضور موانع انجام گرفته است. در این مقاله روشی برای مسیریابی و کنترل ربات موازی سه درجه آزادی مستقل خطی، به نام تریپترون، در حضور مانع ثابت معرفی گردیده و نتایج بر روی مدلی از ربات واقعی شبیه‌سازی شده است. روش مورد استفاده ترکیبی از بهینه‌سازی محدب، برنامه‌ریزی گسسته و مفهوم افق پیش‌بین می‌باشد که مزایایی از قبیل پیدا کردن بهینه سراسری و بلادرنگ بودن را دارد. ابتدا مروری بر ساختار فیزیکی و ویژگی‌های این ربات فراهم شده و سپس به ارائه الگوریتم مسیریابی و شبیه‌سازی آن با استفاده از بسته نرم‌افزاری گروبی پرداخته می‌شود. برای شبیه‌سازی هرچه واقعی‌تر نیز بسته نرم‌افزاری کد2مت به کار گرفته شده است که مدل ربات واقعی را از نرم‌افزار سالدورکز به نرم‌افزار متلب انتقال می‌دهد. نتایج حاصل از دو حالت مختلف نشان می‌دهند که ربات به خوبی موانع را دور زده و به نقطه پایان رسیده است که گویای کاربردی و قابل اطمینان بودن الگوریتم ارائه شده می‌باشد. همچنین، بیشینه زمان محاسبات در هر مرحله کمتر از یک ثانیه بوده است که برای این کاربرد بلادرنگ محسوب می‌گردد.

Optimal motion planning of a 3-dof decoupled parallel robot using convex optimization and receding horizon concept

Roya Sabbagh Novin¹, Mehdi Tale Masouleh^{1*}, Mojtaba Yazdani², Behzad Danaei¹

1- Human and Robot Interaction Laboratory, Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

2- Department of Mechanical Engineering, AmirKabir University of Technology, Tehran, Iran

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

ARTICLE INFORMATION

Original Research Paper
Received 11 March 2015
Accepted 10 May 2015
Available Online 28 June 2015

Keywords:

Optimal path planning
parallel robots
convex optimization
disjunctive programming
receding horizon concept

ABSTRACT

Regarding the progress in technology and increase in the capabilities of the robots, one of the main challenges in the field of robotics is the problem of real-time and collision-free path planning of robots. This paper focuses on the problem of path planning of a 3-DOF decoupled parallel robot called Tripteron in the presence of obstacles. The proposed algorithm is a synergy-based algorithm of convex optimization, disjunctive programming and model predictive control. This algorithm has many advantages compared to previous methods reported in the literature including not getting stuck in the local optimums and finding the global optimum and high computational speeds. Finally, the algorithm will be implemented on a model of the real robot. It should be mentioned that this algorithm has been implemented using Gurobi optimization package with C++ programming language in Qt Creator environment and the simulation of the parallel mechanism is performed by the CAD2MAT package for MATLAB. Obtained results reveal that the maximum computational time at each step is less than one second which, for this particular application, could be regarded as a real-time algorithm.

1- مقدمه

ساختار سینماتیکی به دو دسته‌ی سری و موازی تفکیک می‌گردند. ربات موازی سیستمی است که از چند زنجیره بسته سری که همگی به یک مجری نهایی وصل می‌شوند، تشکیل می‌شود [1]. این گونه ربات‌ها به دلیل مزایایی چون استحکام و دقت بالاتر و در عین حال جرم لختی پایین‌تر مورد توجه قرار گرفته‌اند که این امکان را می‌دهد که در گستره بسیاری از کارهای

ربات‌های بستر ثابت، جزیی جدایی‌ناپذیر در اتوماسیون کارخانجات و صنعت بوده و پیشرفت این صنایع در واقع در گرو رشد این دسته از ربات‌ها است. در نتیجه کنترل و مسیریابی آن‌ها به قسمی که امنیت افراد را تضمین کند، همواره یکی از مسائل مورد بحث بوده است. ربات‌های بستر ثابت از نظر

Please cite this article using:

R. Sabbagh Novin, M. Tale Masouleh, M. Yazdani, B. Danaei, Optimal motion planning of a 3-dof decoupled parallel robot using convex optimization and receding horizon concept, *Modares Mechanical Engineering*, Vol. 15, No. 8, pp. 197-206, 2015 (In Persian)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

مبتنی بر مدل¹¹ و برنامه‌نویسی خطی مخلوط با عدد صحیح¹² روشی برای حل مسأله مسیریابی ربات‌های سری در حضور موانع پیشنهاد دادند [7-10]. به این صورت که در هر مرحله، یک رشته کنترلی بهینه برای خط افق پیش‌بینی با استفاده از آن محاسبه می‌شود. اولین سیگنال از این رشته کنترلی به سیستم وارد شده و خط افق یک مرحله جابجا می‌شود و این کار در هر مرحله تکرار می‌گردد.

همان‌طور که گفته شد روش‌های ارائه شده برای ربات‌های پایه پویا و یا سری بوده است [11، 12] و در حوزه مسیریابی ربات‌های موازی مطالعات زیادی انجام نگرفته و بیشتر بر روی دنبال کردن مسیر از پیش تعیین شده کار شده است. از آنجایی که معمولاً از این نوع ربات‌ها در جابجایی نقطه به نقطه استفاده می‌گردد، حل مسأله مسیریابی بهینه برای آن‌ها، به ویژه در حضور موانع، از ارزش شایان توجهی برخوردار است.

در این مقاله سعی بر آن است تا با استفاده از روش بهینه‌سازی محدب که روشی مطمئن محسوب می‌گردد، به اهداف گفته شده دست یافته و بهترین جواب ممکن به دست آورده شود. از معایب روش‌های ارائه شده قبلی، گرفتار شدن در بهینه‌های محلی است. در روش بهینه‌سازی محدب همواره بهینه‌ی سراسری به دست می‌آید و مشکل بهینه‌های محلی برطرف می‌گردد. در [13] از این روش برای مسیریابی ربات‌های سری استفاده شده است.

همچنین زمان محاسبات نسبت به سایر روش‌ها قابل پیش‌بینی‌تر بوده و با به‌کارگیری برنامه‌نویسی گسسته و روش افق پیش‌بین¹³ می‌توان به زمان‌های قابل قبولی رسید. البته واضح است که با افزایش درجات آزادی و قیود مسأله، زمان محاسبات نیز بیشتر می‌گردد. همچنین از کتابخانه‌ای در نرم‌افزار ++C به نام گروبی¹⁴ استفاده می‌گردد که انتظار می‌رود سرعت محاسبات بالایی داشته باشد [14].

برای کنترل ربات نیز روش کنترل پیش‌بین مبتنی بر مدل به کار برده می‌شود که مهمترین مزیت آن امکان بهینه‌سازی برای بازه زمانی جاری با در نظر گرفتن متغیرهای حالت در بازه‌های زمانی آینده، با استفاده از مدل سیستم، می‌باشد. این کار با بهینه‌سازی متغیرها برای یک افق زمانی محدود صورت می‌گیرد، در حالی که دستورات فقط برای بازه زمانی جاری اجرا می‌شوند. در نتیجه امکان پیش‌بینی آینده و تعیین ورودی‌های کنترلی مناسب‌تر را دارد که علاوه بر افزایش سرعت، به دلیل ماهیت پیش‌بین آن، منجر به انعطاف‌پذیری بیشتر و جواب بهینه‌تری نیز خواهد شد.

در این مقاله، ابتدا در بخش 2 توضیح مختصری از مفاهیم استفاده شده ارائه شده و سپس با بررسی ربات تریپترون مورد نظر در بخش 3، قسمت‌های مختلف مسأله، شامل تابع هزینه و انواع قیدها، مسأله بهینه‌سازی کلی در بخش 4 نوشته می‌شود. در نهایت نیز در بخش 5 الگوریتم پیشنهادی برای دو حالت مختلف شبیه‌سازی گشته و نتایج حاصل از آن ارائه می‌گردند.

2- مفاهیم به کار برده شده

در ادامه مفاهیم به کار گرفته شده در الگوریتم پیشنهادی مقاله به طور مختصر توضیح داده می‌شود. برای جزئیات بیشتر مطالعه مراجع ذکر شده، پیشنهاد می‌گردد.

2-1- بهینه‌سازی محدب

همان‌طور که پیش‌تر ذکر شد، در روش‌های بهینه‌سازی که در قسمت مقدمه به

تکراری مانند خطوط مونتاژ، شبیه‌سازهای پرواز، دستگاه‌های ماشین‌کاری، ابزارهای فوق دقیق و قطعات پزشکی به کار گرفته شوند. هرچند که پیچیده بودن روابط سینماتیکی و دینامیکی آن‌ها و همچنین وجود نقاط تکین زیاد، باعث سخت شدن کنترل آن‌ها می‌گردد.

یک دسته از ربات‌های موازی معروف، خانواده مالتیپترون¹ هستند که توسط پروفیسور گاسلن ارائه شده است [2]. در این مقاله مسیریابی ربات تریپترون² از این خانواده مورد بحث قرار می‌گیرد. این ربات دارای فضای کاری کارترین بوده و در این فضا هیچ نقطه تکینگی ندارد. این امر و همچنین تحریک مستقل بودن آن سبب شده تا کنترل آن نسبت به سایر ربات‌های موازی بسیار آسان گردد.

در مسأله مسیریابی ربات‌ها به طور کلی نکات زیر مورد توجه و حائز اهمیت می‌باشند:

- پیدا کردن مسیر بهینه از نقطه شروع تا نقطه پایان، این بهینگی می‌تواند برای طول مسیر طی شده، زمان طی مسیر و یا انرژی مصرفی تعریف گردد؛
- اجتناب از موانع موجود در مسیر که می‌توانند ثابت یا متحرک باشند و تضمین ایمنی؛
- رعایت قیود سینماتیکی و دینامیکی ربات که در واقع بخش عمده‌ای از پیچیدگی مسأله از این قیود ناشی می‌شود.

برای حل این‌گونه مسائل روش‌های مختلفی تاکنون پیشنهاد شده‌اند که هر یک مزایا و معایب خود را دارند. یکی از اولین و پرکاربردترین روش‌ها، استفاده از میدان پتانسیل مصنوعی³ [3] است که در آن، نقطه‌ی پایانی به عنوان چاه و موانع به صورت چشمه مدل‌سازی می‌شوند و مسیر بهینه به دست می‌آید. از مزایای این مدل بلادرنگ⁴ بودن و توانایی در نظر گرفتن موانع متحرک است. ولی مشکل اصلی آن گرفتار شدن در کمینه‌های محلی است. همچنین امکان بهینه کردن انرژی در این روش وجود ندارد [4].

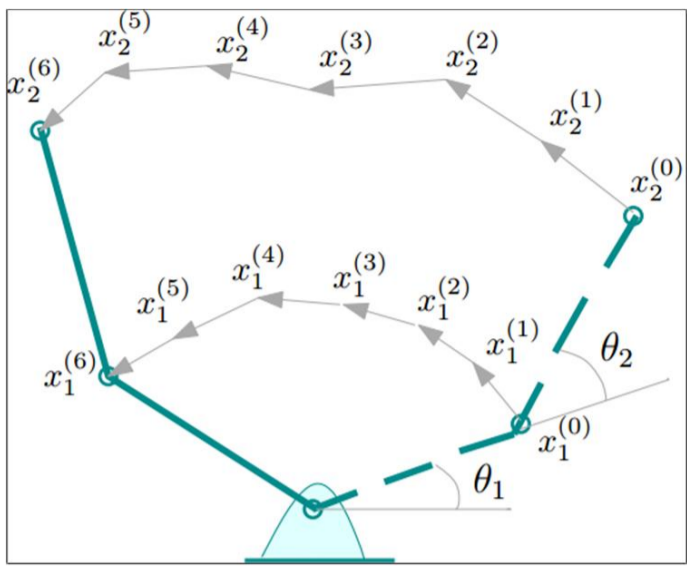
روش‌های دیگر شامل گراف دیداری⁵ و تجزیه سلولی⁶ نیز به کار گرفته شده‌اند که به دلیل محدودیت در تعریف موانع مناسب به نظر نمی‌رسند [5]. برای غالب شدن بر این مشکل روش‌های نقشه راه احتمالی⁷ و درخت تصادفی به سرعت در حال کاوش⁸ پیشنهاد شدند که در آن یک نمونه تصادفی از فضای پیکربندی انتخاب شده و با استفاده از آن، یک مسیر بهینه به دست می‌آید و برخورد نکردن آن با موانع نیز با مقایسه آن با فضای کاری ربات بررسی می‌شود. در این روش‌ها نیز امکان بهینه کردن انرژی و زمان وجود ندارد. علاوه بر این، در این روش‌ها، میزان نمونه‌ها در زمان حل و اساساً امکان آن بسیار تاثیرگذار و در نتیجه محدود کننده است.

در سال 2006 استفاده از برنامه‌ریزی گسسته⁹ برای پیدا کردن مسیر بهینه ربات‌های سری پیشنهاد شد که در آن مسیری که هر مفصل طی می‌کند به صورت نقاط گسسته‌ای با فواصل زمانی یکسان در نظر گرفته می‌شود. با افزودن قیدهای سینماتیکی، دینامیکی و قیدهای ناشی از وجود موانع مسأله حرکت بهینه از یک نقطه به نقطه هدف حل می‌گردد [6].

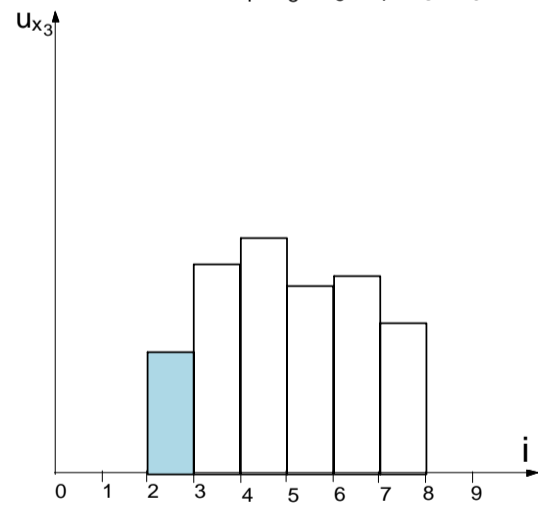
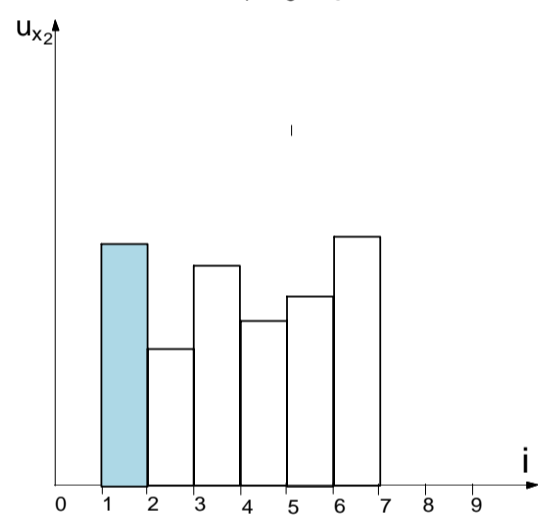
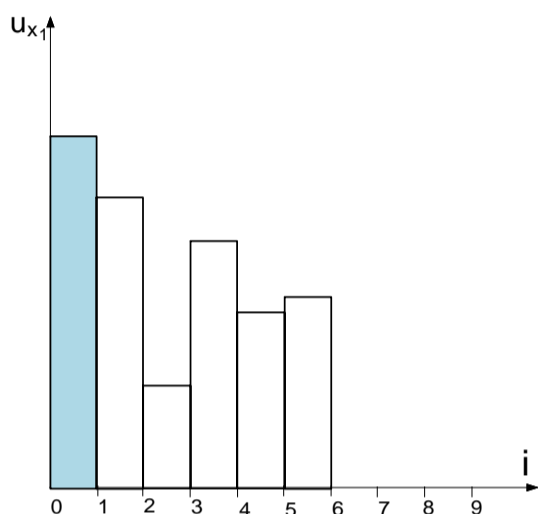
در سال‌های 2009 تا 2011، گروهی از محققان دانشگاه کسل¹⁰ واقع در آلمان، با به‌کارگیری این پیشنهاد و با استفاده از روش کنترل پیش‌بین

1. Multipteron family
2. Tripteron
3. Artificial Potential Fields (APF)
4. Real-time
5. Visibility graph
6. Cell decomposition
7. Probabilistic Road Map (RPM)
8. Rapidly-exploring Random Trees (RRT)
9. Disjunctive Programming
10. Kassel University

11. Model Predictive Control (MPC)
12. Mixed Integer Linear Programming (MILP)
13. Receding horizon
14. Gurobi optimization



شکل 1 شماتیک برنامه‌ریزی گسسته اعمال شده بر روی یک بازوی ماهر



شکل 2 نمایش شماتیک مفهوم افق کاهنده

هر سه شاخه به صورت متعامد به مجری نهایی متصل گشته‌اند و در هر کدام از آنها محور کشویی و همه مفاصل دورانی هم‌جهت می‌باشند [17]. شایان ذکر است که ساختار سینماتیکی این ربات با استفاده از روش سنتز نوعی ربات‌های موازی به دست آورده شده است [18]. در ادامه مشخصات ساخت‌افزاری و نرم‌افزاری و همچنین معادلات سینماتیکی مربوط به ربات تریپترون ساخته شده در آزمایشگاه تعامل انسان و ربات دانشگاه

آن‌ها اشاره گردید، یکی از مشکلات اساسی به دام افتادن در بهینه‌های محلی است. با استفاده از بهینه‌سازی محدب می‌توان از این مسأله دوری جست، چرا که در مسأله‌ی بهینه‌سازی محدب، نقطه بهینه محلی همواره همان نقطه بهینه سراسری است [15]. یک مسأله بهینه‌سازی به صورت کلی به فرم زیر تعریف می‌گردد:

$$\min f_0(x) \tag{1.1}$$

$$\text{s.t. } f_i(x) \leq 0, i = 1, \dots, p \tag{1.2}$$

$$h_j(x) = 0, j = p + 1, \dots, m \tag{1.3}$$

این مسأله در صورتی بهینه‌سازی محدب خواهد بود که:

- تابع هدف $f_0(x)$ تابعی محدب باشد.

- قیود نامساوی $f_i(x)$ توابعی محدب باشند.

- قیود مساوی $h_j(x)$ توابعی مستوی¹ باشند.

تابعی محدب نامیده می‌شود که اولاً دامنه‌ی آن ناحیه‌ای محدب باشد و ثانیاً به ازای هر $(x, y) \in \text{dom}(f)$ و $0 \leq \alpha \leq 1$ بتوان نوشت:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \tag{2}$$

لازم به ذکر است اگر تابع f دوبار مشتق‌پذیر باشد، شرط بالا به صورت رابطه (3) خلاصه می‌گردد [15]:

$$\nabla^2 f(x) \geq 0 \tag{3}$$

به عبارت دیگر کافی است تا ماتریس هسیان² آن مثبت معین باشد. همچنین در صورتی که:

$$h(x) = Ax + b \tag{4}$$

که در آن A ماتریس و b یک بردار است، تابع $h(x)$ یک تابع مستوی می‌باشد.

2-2- برنامه‌ریزی گسسته و افق پیش‌بین

یکی از پارامترهای مهم در امر کنترل ربات، زمان محاسبات می‌باشد که استفاده از مفهوم برنامه‌ریزی گسسته و افق پیش‌بین می‌تواند تأثیر زیادی در کاهش آن داشته و منجر به مناسب شدن الگوریتم برای کاربردهای بلادرنگ گردد [16].

همان‌طور که پیش‌تر اشاره شد و در شکل 1 نیز مشاهده می‌گردد، برنامه‌ریزی گسسته به معنی در نظر گرفتن کل مسیر (که مجهول است) به صورت یک سری نقاط گسسته با فواصل زمانی یکسان (ولی همچنان مجهول) است. در این روش تعداد مجهولات بسیار زیاد شده و در واقع مسأله در یک مرحله برای کل مسیر حل می‌گردد. این کار باعث افزایش زمان محاسبات گشته و از بهره‌وری آن می‌کاهد [6].

برای حل این مشکل از مفهوم افق کاهنده استفاده می‌شود. به این صورت که در هر مرحله مسیر بهینه برای یک افق زمانی (با مکانی) تعیین می‌گردد، اما فقط مقادیر به دست آمده برای آن گام حفظ شده و بقیه حذف می‌شوند. این کار تا جایی ادامه می‌یابد که مجری نهایی به نقطه هدف برسد. به همین دلیل به این روش افق کاهنده می‌گویند. این کار علاوه بر افزایش سرعت محاسبات، به دلیل ماهیت پیش‌بین آن، منجر به جواب بهینه‌تری نیز خواهد شد. همچنین امکان در نظر گرفتن تغییرات محیط را نیز در هنگام انجام عملیات دارد. به طور مثال حرکت موانع مشکلی در کار ربات ایجاد نکرده و ربات به راحتی می‌تواند مسیر خود را تغییر دهد. شکل 2 مفهوم افق کاهنده را بهتر نشان می‌دهد [16].

3- ربات موازی تریپترون

ربات تریپترون یک ربات موازی با سه درجه آزادی مستقل خطی می‌باشد که هر یک از شاخه‌های آن دارای ساختار PRRR بوده و توسط یک عملگر کشویی کنترل می‌گردد.

1. Affine function
2. Hessian

جدول 1 پارامترهای طراحی ربات تریپترون، ساخته شده در آزمایشگاه تعامل انسان و

پارامتر	مقدار برحسب متر
L_{x1}, L_{y1}, L_{z1}	0/440
L_{x2}, L_{y2}, L_{z2}	0/392
b_{xy}	0/092
b_{xz}	0/019
b_{yx}	-0/092
b_{yz}	0/019
b_{zx}	0
b_{zy}	-0/080

3-2- معادلات سینماتیک ربات

از آنجایی که این ربات، یک ربات مستقل می‌باشد، معادلات سینماتیک بین ورودی، که همان تغییر طول راستای کشویی می‌باشد، و خروجی، که موقعیت مجری نهایی را تعیین می‌نماید، بسیار ساده بوده و به صورت روابط (5) تا (7) نوشته می‌شود:

$$\rho_x = X_{e.e} \quad (5)$$

$$\rho_y = Y_{e.e} \quad (6)$$

$$\rho_z = Z_{e.e} \quad (7)$$

که در آن، ρ_x, ρ_y, ρ_z تغییرات طول راستای کشویی و $X_{e.e}, Y_{e.e}, Z_{e.e}$ موقعیت مجری نهایی را نشان می‌دهند که در شکل 4 نیز قابل مشاهده می‌باشند. در نتیجه برای ژاکوبین ربات می‌توان نوشت:

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

که نشان می‌دهد این ربات در داخل فضای کاری خود عاری از تکینگی⁴ می‌باشد. با این وجود، به منظور در نظر گرفتن برخوردها با موانع لازم است که معادلات سینماتیک مربوط به بازوها نیز در نظر گرفته شود. در این راستا دو معادله سینماتیکی برای هر شاخه، در صفحه تشکیل شده از دو پایه دیگر به دست می‌آید. بنابراین برای شاخه اول که متصل به راستای کشویی محور X است، می‌توان نوشت:

$$Y_{e.e} = L_{x1} \cos \theta_{x1} + L_{x2} \cos(\theta_{x1} + \theta_{x2}) + b_{xy} \quad (9)$$

$$Z_{e.e} = L_{x1} \sin \theta_{x1} + L_{x2} \sin(\theta_{x1} + \theta_{x2}) + b_{xz} \quad (10)$$

به همین ترتیب برای پایه‌های Y و Z معادلات به صورت روابط (11) تا (12) نوشته می‌شوند:

$$Z_{e.e} = L_{y1} \cos \theta_{y1} + L_{y2} \cos(\theta_{y1} + \theta_{y2}) + b_{yz} \quad (11)$$

$$X_{e.e} = L_{y1} \sin \theta_{y1} + L_{y2} \sin(\theta_{y1} + \theta_{y2}) + b_{yx} \quad (12)$$

$$Y_{e.e} = L_{z1} \cos \theta_{z1} + L_{z2} \cos(\theta_{z1} + \theta_{z2}) + b_{zy} \quad (13)$$

$$X_{e.e} = L_{z1} \sin \theta_{z1} + L_{z2} \sin(\theta_{z1} + \theta_{z2}) + b_{zx} \quad (14)$$

در معادلات بالا L_{x1}, L_{y1}, L_{z1} و L_{x2}, L_{y2}, L_{z2} به ترتیب، طول بازوهای اول و دوم هر شاخه و $b_{xy}, b_{xz}, b_{yx}, b_{yz}, b_{zy}, b_{zx}$ بردار وصل‌کننده مرکز مجری نهایی به مفاصل اتصال‌دهنده هر شاخه با مجری نهایی را نشان می‌دهند. با حل این معادلات زوایای بازوها و در نتیجه موقعیت تمام مفاصل به دست می‌آیند که برای قیود اجتناب از موانع به کار گرفته می‌شوند.

4- تعریف مسأله مسیریابی

برای روشن‌تر شدن مسأله بهینه‌سازی مسیر، ابتدا بخش‌های مختلف آن، که

4. Singularity-free

تهران که شبیه‌سازی‌های این مقاله بر روی مدلی از این ربات واقعی انجام گرفته است، ارائه می‌گردد. شکل 3 نیز این ربات را نشان می‌دهد.

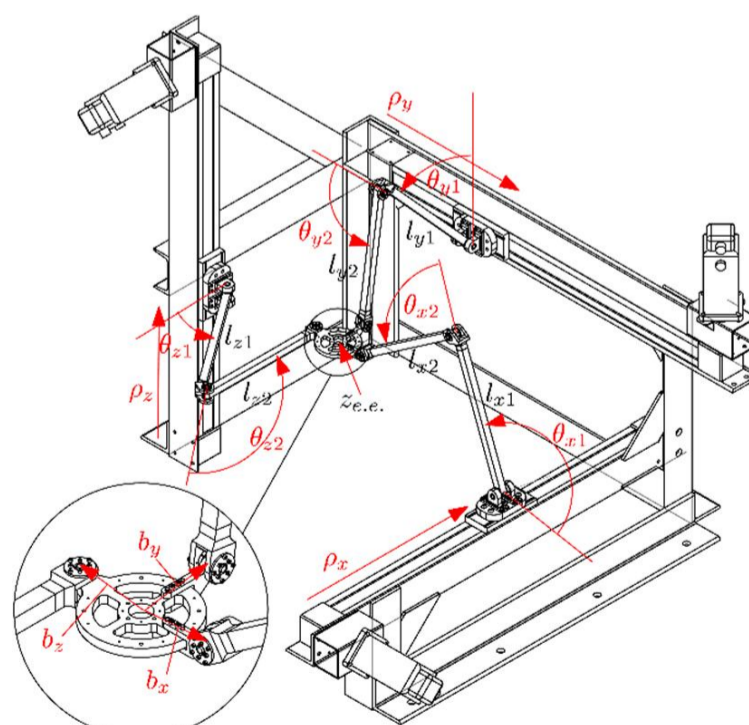
3-1- ساختار ربات

ربات تریپترون سه درجه آزادی انتقالی دارد که مجری نهایی آن توسط سه شاخه به پایه‌های ثابت متصل شده است. هر یک از این شاخه‌ها که هر کدام شامل دو بازو می‌باشند، با مفصل دورانی، از یک طرف به مجری نهایی و از طرف دیگر به لغزنده راستای کشویی وصل شده‌اند. مکانیزم ربات به گونه‌ای است که هر یک از موتورها تنها یکی از درجات آزادی انتقالی ربات را کنترل می‌کنند و در نتیجه این مکانیزم جزو مکانیزم‌های مستقل محسوب می‌گردد. به منظور ایجاد حرکت انتقالی، از ماژول‌های خطی استفاده گردیده که این ماژول‌ها حرکت دورانی موتور را به حرکت انتقالی تبدیل می‌کنند. موتورهای استفاده شده در این ربات از نوع سرو¹ می‌باشند.

به منظور دسترسی بهتر به منابع ارتباطی کامپیوتر و ایجاد ارتباط بلادرنگ، تمام برنامه‌های راه‌اندازی و واسط گرافیکی در سیستم عامل لینوکس² و در چارچوب برنامه‌نویسی کیوتی-کریتور³ نوشته شده است. جدول 1 پارامترهای طراحی ربات را مشخص می‌کند.

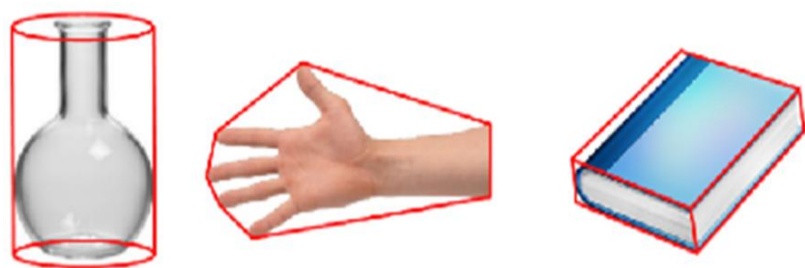


شکل 3 ربات واقعی تریپترون ساخته شده در آزمایشگاه تعامل انسان و ربات دانشگاه تهران

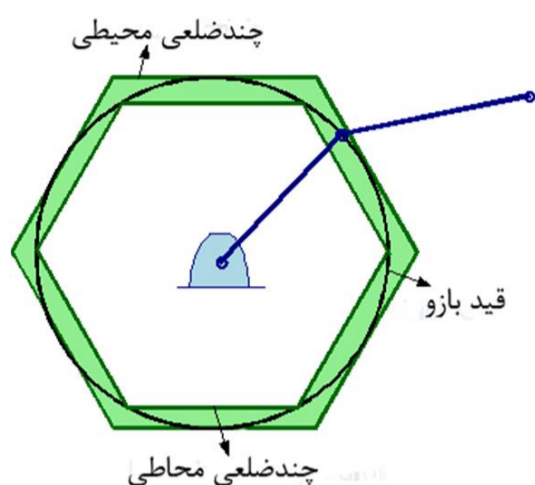


شکل 4 شماتیک ربات موازی تریپترون به همراه پارامترهای به کار برده شده

1. AC-Servo
2. Linux-Ubuntu
3. Creator Qt



شکل 5 نمونه‌هایی از محدب‌سازی اشکال نامحدب



شکل 6 نمایش چند ضلعی‌های محیطی و محاطی تقریب زده شده برای قیدهای سینماتیکی [13]

$$A\xi \geq b + (v - 1)M \quad (19)$$

$$\sum_{i=1}^s v_i \geq 1 \quad (20)$$

که در آن $v = (v_1, \dots, v_s)^T$ برداری از متغیرهای دودویی² بوده $(v_i \in \{0,1\})$ و M یک عدد ثابت بزرگ است که برای استفاده در روش عدد-بزرگ³ تعیین می‌گردد [10]. معادله (20) تضمین می‌کند که حداقل یکی از المان‌های بردار v ، یک بوده و در نتیجه نقطه ξ خارج از چندوجهی مانع قرار گیرد.

4-3- قیود سینماتیک ربات

همچنین لازم است که سینماتیک ربات نیز، که از هندسه آن ناشی می‌شود، در مساله وارد شود و در واقع بازوهای ربات تعریف شوند. از آنجایی که هر مفصل روی دایره‌ای به مرکز مفصل قبل و شعاع بازوی بین آنها حرکت می‌کند و قید مساوی درجه دوم، قید محدب نمی‌باشد، نیاز است این قید با یک سری قیود محدب تخمین زده شود. در این راستا، از چندوجهی‌های محدب استفاده می‌گردد. به این صورت که هر دایره با دو چندوجهی محیطی⁴ و محاطی⁵ خود تخمین زده می‌شود. شکل 6 این مفهوم را واضح‌تر نشان می‌دهد. قید هندسی بازوهای هر شاخه از این ربات به صورت رابطه (21) نوشته می‌شود:

$$(z_{j+1}(t) - z_j(t))^T (z_{j+1}(t) - z_j(t)) = l_{j,j+1}^2 \quad (21)$$

که در آن، z_j و z_{j+1} مختصات دو مفصل مجاور و $l_{j,j+1}$ طول بازوی بین آنها را نشان می‌دهند. قیود چندوجهی‌های محیطی و محاطی را نیز می‌توان به شکل روابط (22) تا (24) نوشت:

$$A_{cs} (z_{j+1}(t) - z_j(t)) \leq b_{cs} \quad (22)$$

$$A_{is} (z_{j+1}(t) - z_j(t)) \geq b_{is} + (u(t) - 1)M, \quad (23)$$

$$\sum_{k=1}^{n_{is}} u_k(t) \geq 1. \quad (24)$$

معادلات (17) و (18) به ترتیب چندوجهی‌های محیطی و محاطی دایره مربوط به قید سینماتیک را تعریف کرده و $u(t) = (u_1(t), \dots, u_{n_{is}}(t))^T$ نیز

شامل تابع هزینه و قیدهای مربوط به موانع و ربات می‌باشند، مورد بررسی قرار می‌گیرد و سپس مسأله کلی به صورت یک مسأله بهینه‌سازی محدب نوشته می‌شود.

4-1- تابع هزینه

تابع هزینه می‌تواند شامل قسمت‌های متفاوتی باشد و با در نظر گرفتن هدف مسأله تعیین می‌گردد. هدف مسأله می‌تواند رسیدن به نقطه پایانی در کمترین زمان، طی کوتاه‌ترین مسیر و یا صرف کمترین انرژی در طول مسیر رسیدن به نقطه پایانی باشد. تنها شرط محدود کننده در تعیین تابع هدف در این مقاله، محدب بودن آن است.

همچنین می‌توان از ترکیبی از توابع هزینه ذکر شده، به عنوان تابع هزینه چند-منظوره¹ استفاده نمود که در آن صورت لازم است برای هر یک از آنها یک ضریب وزنی تعیین شود. این ضرایب وزنی نیز با در نظر گرفتن اولویت‌های مسأله و نسبت به کاربرد آن در محیط‌های مختلف و شرایط گوناگون مشخص می‌شوند. در این مقاله، تابع هزینه کوتاه‌ترین مسیر در نظر گرفته شده است، اما همان‌طور که گفته شد، توابع هزینه دیگری نیز می‌توان در نظر گرفت.

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 مجری نهایی ربات، برای یک افق پیش‌بینی به طول h می‌توان تابع هزینه کوتاه‌ترین مسیر را به صورت رابطه (15) نوشت:

$$\min \sum_{k=1}^h \|z_{e,e}(k+1) - z_{e,e}(k)\|_2^2 \quad (15)$$

که در آن $z_{e,e}(k) \in \mathbb{R}^2$ مختصات مرکز مجری نهایی ربات در مرحله k می‌باشد. همچنین زمان طی مسیر نیز با جمع فواصل زمانی مربوط به هر مرحله به صورت رابطه (16) به دست می‌آید:

$$\min \sum_{k=1}^h dt(k) \quad (16)$$

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:

$$\min \sum \|z_{e,e}(h) - z_g\|_2^2 \quad (17)$$

که در آن z_g مختصات نقطه پایانی را نشان می‌دهد.

4-2- اجتناب از موانع

به منظور محدب ماندن مسأله بهینه‌سازی، تمام موانع، حتی آن‌هایی که محدب نیستند، به صورت شکل محدبی که آن‌ها را در بر می‌گیرد، در نظر گرفته می‌شوند. بنابراین، همان‌طور که در شکل 5 نشان داده شده است، هر مانع با یک چندوجهی محیطی (در حالت صفحه‌ای، چندضلعی محیطی) تخمین زده می‌شود. لازم به ذکر است که این کار باعث محافظه‌کارانه‌تر شدن جواب شده و ایمنی مطمئن‌تری را برقرار می‌کند. یک چندوجهی در فضا به صورت تقاطع یک سری نیم فضا تعریف می‌شود:

$$O = \{\xi \mid A\xi < b\} \quad (18)$$

در معادله بالا، A ماتریسی با ردیف‌های A_k به طوری که $k \in \{1, \dots, s\}$ و $b = (b_1, \dots, b_s)^T$ بردار مربوط به نیم‌فضاها می‌باشند، که در آن‌ها s تعداد وجه‌های چندوجهی را نشان می‌دهد. نقطه ξ در صورتی خارج از چندوجهی O قرار می‌گیرد که حداقل یکی از نامعادلات $A\xi < b$ ارضا گردد. به عبارت دیگر:

1. Multi-objective function

2. Binary variables
3. Big-M method
4. Circumscribing polyhedron
5. Inscribing polyhedron

هزینه نوشته شده و برای هر یک ضریب وزنی در نظر گرفته شده است. علاوه بر این، معادلات (26.و) و (26.ز) عدم برخورد با مانع را تضمین می‌کنند ($\emptyset \cap \mathcal{Z} = \emptyset$). قیدهای (26.ط) و (26.ی) نیز به ترتیب، نقطه ابتدایی و نقطه پایانی را مشخص کرده و قید (26.ک) سرعت بیشینه را محدود می‌گرداند. همچنین معادله (26.ح) اتصال شاخه‌ها با مجری نهایی را تضمین می‌نماید.

با استفاده از این مساله الگوریتم مسیریابی این ربات نوشته می‌شود که در الگوریتم (1) آمده است. در این الگوریتم ابتدا مقادیر ثابت مورد نیاز، شامل طول افق پیش‌بینی h ، تعداد شاخه‌ها و بازوها (n_{limb}, n_{link}) ، سرعت بیشینه مجاز v_{max} ، تعداد وجه‌های مورد استفاده در محدب‌سازی موانع s ، پارامتر مناسب برای روش عدد-بزرگ M ، دقت مورد نظر در رسیدن به نقطه پایانی ϵ ، و پارامترهای مربوط به موانع (y_0, x_0, r_0) ، تعیین می‌گردند.

با استفاده از معادلات زیر، پارامترهای لازم برای نیم فضاهای تشکیل‌دهنده چندوجهی موانع، که در مساله بهینه‌سازی به کار می‌روند، مشخص می‌گردند:

$$\phi(i) = \frac{2\pi i}{s}, \quad (27.ا)$$

$$m_o(i) = \tan\left(\frac{2\pi i}{s} + \frac{2\pi i}{s} - \frac{2\pi i}{s}\right), \quad (27.ب)$$

$$x_{aux}(i) = r_o \times \cos(\phi(i)) + x_o, \quad (27.ج)$$

$$y_{aux}(i) = r_o \times \sin(\phi(i)) + y_o, \quad (27.د)$$

$$b_o(i) = y_{aux}(i) - m_o(i) \times x_{aux}(i), \quad (27.ه)$$

$$\forall i = 1, \dots, s,$$

با مشخص شدن تمام پارامترهای بالا، مساله بهینه‌سازی در هر مرحله حل شده و از خروجی‌های آن برای تعیین مسیر استفاده می‌گردد. در این قسمت دومین درایه از مقادیر به دست آمده برای افق پیش‌بینی به عنوان مقادیر مرجع در نظر گرفته شده و متغیرهای کنترلی که همان سرعت‌های خطی راستاهای کشویی ربات هستند با استفاده از معادلات (28) تا (30) به دست می‌آیند:

$$\dot{\rho}_x = \frac{dx_{e,e}(2)}{dt}, \quad (28)$$

$$\dot{\rho}_y = \frac{dy_{e,e}(2)}{dt}, \quad (29)$$

$$\dot{\rho}_z = \frac{dz_{e,e}(2)}{dt}, \quad (30)$$

که در آن dt بازه زمانی بوده و $dx_{e,e}(2)$ ، $dy_{e,e}(2)$ و $dz_{e,e}(2)$ مقادیر تغییر مختصات مجری نهایی از مرحله کنونی تا مرحله بعد را نشان می‌دهند.

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.

```

1: Initialize:  $n_{limb}, n_{link}, h, v_{max}, s, M, r_o, x_o, y_o, \epsilon$ ;
   % Find the circumscribing polygon parameters for obstacles
2: Eqs. (22)  $\rightarrow m_o, b_o$ ;
3: while  $dz_{final} \geq \epsilon$  do
   % Solve the optimization problem by Gurobi
4:  $(z_{e,e}, dt, dx_{e,e}, dy_{e,e}, dz_{e,e}) = \text{Path\_Optimizer}(h, z_s, z_g, v_{max}, m_o, b_o)$ ;
   % Solve the Inverse Kinematic Problem
5: Eqs. (7-9)  $\rightarrow \theta_{x1}, \theta_{x2}, \theta_{y1}, \theta_{y2}, \theta_{z1}, \theta_{z2}$ ;
   % Update the linear velocities
6:  $\dot{\rho}_x = dx_{e,e}(2)/dt$ ;
7:  $\dot{\rho}_y = dy_{e,e}(2)/dt$ ;
8:  $\dot{\rho}_z = dz_{e,e}(2)/dt$ ;
   % Update the position of the robot using kinematic model
9: Eqs. (5)  $\rightarrow x_{new}, y_{new}, z_{new}$ ;
10:  $z_s = [x_{new}, y_{new}, z_{new}]$ ;
11:  $dz_{final} = \|z_s - z_g\|_2$ ; % Find the distance to the goal point
12:  $T_{total} = T_{total} + dt$ ; % Update the total time
13: end while
14: Go to the goal position and stop;

```

الگوریتم 1 مسیریابی ربات موازی تریپترون

نشان‌دهنده برداری از متغیرهای دودویی بوده که n_{is} بیانگر تعداد وجه‌های چندوجهی می‌باشد. معادله (24) تضمین می‌کند که حداقل یکی از نامعادلات $A_{is}(z_{j+1}(t) - z_j(t)) \geq b_{is}$ ارضا می‌شود و در نتیجه مفص \mathcal{Z}_{j+1} خارج از چندوجهی محاطی قرار می‌گیرد.

شایان توجه است که با افزایش تعداد وجه‌های این چندوجهی‌ها، تقریب زده شده به دایره نزدیک‌تر شده و از مقدار خطا می‌کاهد. از طرفی دیگر تعداد قیود افزایش پیدا کرده و زمان محاسبات بیشتر می‌شود. بنابراین مقدار متوسطی برای آن می‌تواند شرایط مطلوبی را فراهم نماید. به طور مثال در مسائل مطرح شده در این مقاله هشت ضلعی در نظر گرفته شده است.

4-4- سرعت بیشینه

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

$$\mathcal{O} = \{\xi \mid A\xi < b\} \quad \left| \frac{z_{e,e}(k+1) - z_{e,e}(k)}{\delta t} \right| \leq v_{max}, \quad k = 1, \dots, (h-1), \quad (25)$$

که در آن δt بازه زمانی بین دو نقطه بوده که می‌تواند ثابت (در حالت تابع هزینه کوتاه‌ترین مسیر) و یا متغیر (در حالت تابع هزینه کمترین زمان) باشد.

4-5- مساله کلی مسیریابی

با در نظر گرفتن توضیحات ارائه شده برای هر قسمت، مساله مسیریابی ربات موازی تریپترون با در نظر گرفتن m نقطه روی هر بازو به صورت زیر نوشته می‌شود:

$$\min w_1 T + w_1 \sum_{i=1}^h \|z_{e,e}(i+1) - z_{e,e}(i)\|_2^2 + w_3 \|z_{e,e}(h) - z_g\|_2^2 \quad (26.ا)$$

$$s.t. A_{cs,j,k}(i) (z_{j,k,m}(i) - z_{j,(k-1),m}(i)) \leq b_{cs,j,k}(i), \quad (26.ب)$$

$$A_{is,j,k}(i) (z_{j,k,m}(i) - z_{j,(k-1),m}(i)) \geq b_{is,j,k}(i) + (u_j(i) - 1)M, \quad (26.ج)$$

$$\sum_{q=1}^{n_{is}} u_{j,k,q}(i) \geq 1, \quad (26.د)$$

$$z_{j,k+1,p} = \frac{p}{m} (z_{j,k+1,m} + z_{j,k,m}), \quad (26.ه)$$

$$A_{\mathcal{O}}(i) z_{j,k,p}(i) \geq b_{\mathcal{O}}(i) + (v(i) - 1)M \quad (26.و)$$

$$\sum_{q=1}^n v_q(i) \geq 1, \quad (26.ز)$$

$$z_{j,2,m}(i) = z_{e,e} + b_j \quad (26.ح)$$

$$z_{e,e}(t_0 = 0) = z_s \quad (26.ط)$$

$$z_{e,e}(t_f = T) = z_g \quad (26.ی)$$

$$\left| \frac{z_{j,k,m}(i+1) - z_{j,k,m}(i)}{\delta t} \right| \leq v_{j,k,max} \quad \forall i \in \{1, \dots, h\}, \forall j \in \{1,2,3\}, \forall k \in \{1,2\}, \forall p \in \{1, \dots, m\}, \quad (26.ک)$$

که در آن $z_{j,k,p}(i)$ نشان‌دهنده مکان نقطه p -ام از بازوی k -ام شاخه z در لحظه i و b_j و بیان‌گر بردار وصل‌کننده مرکز مجری نهایی به نقطه انتهایی شاخه z می‌باشد.

همان‌طور که پیش‌تر اشاره شد، در معادلات بالا، عبارت $z_{e,e}(k) \in \mathbb{R}^2$ مختصات مجری نهایی ربات در مرحله k و z_g مختصات نقطه پایانی را نشان می‌دهد. همچنین T کل زمان طی مسیر در طول افق پیش‌بینی می‌باشد و برابر است با:

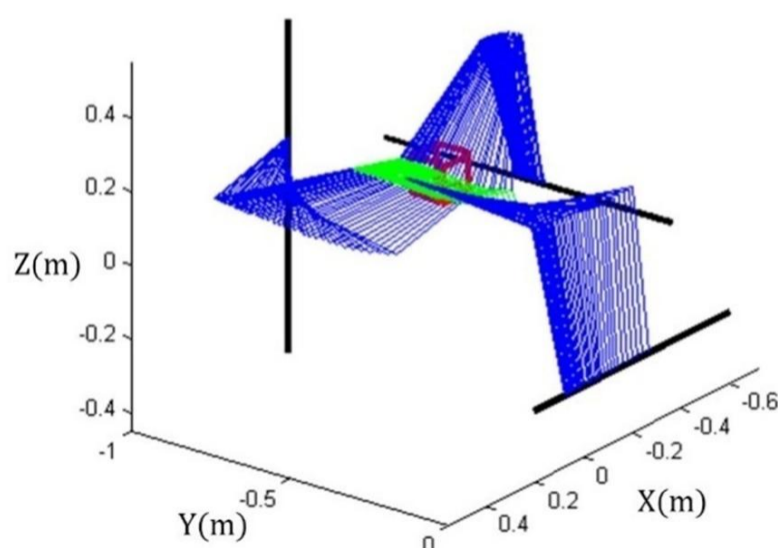
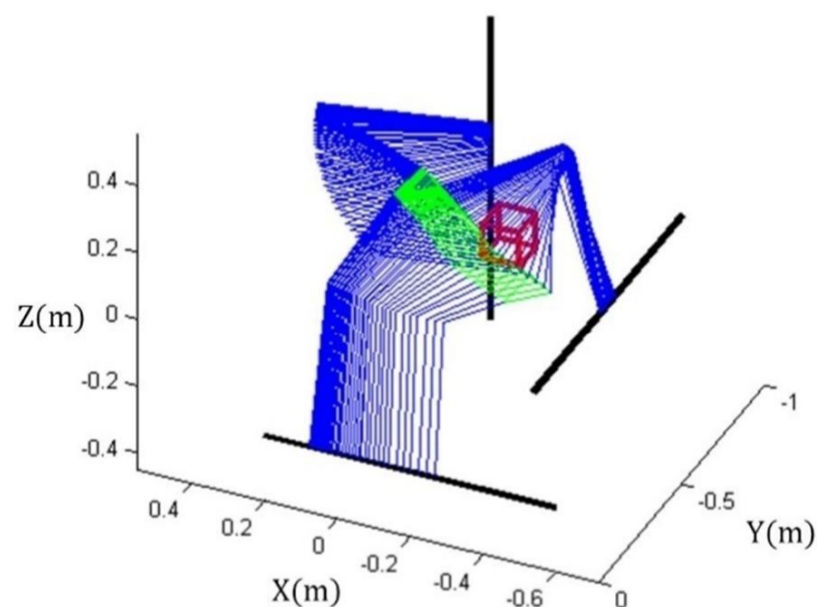
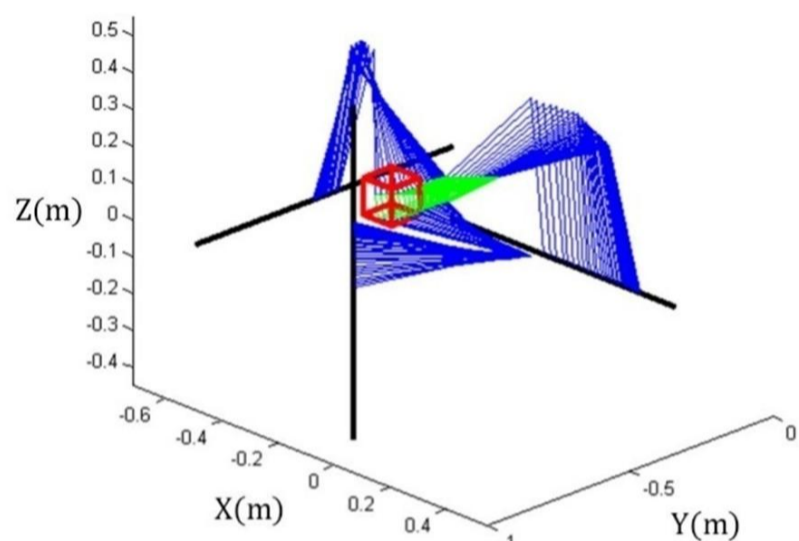
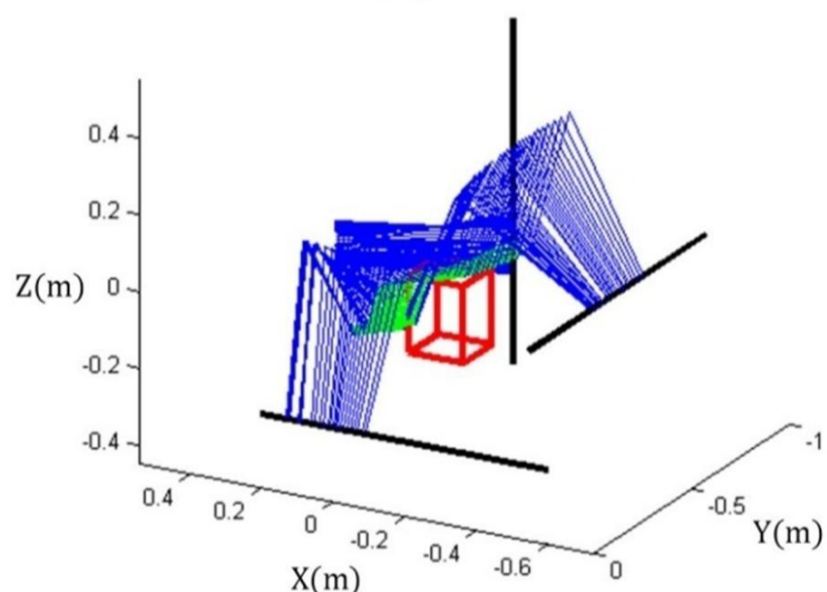
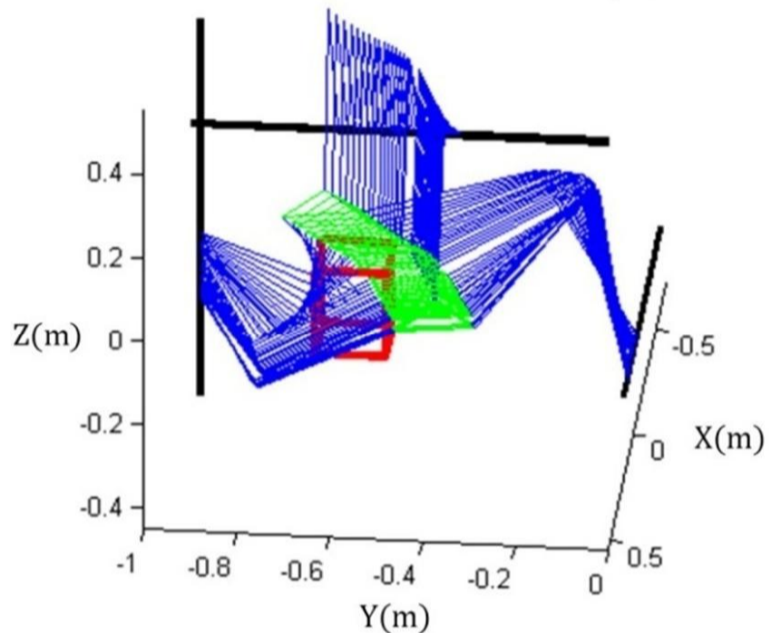
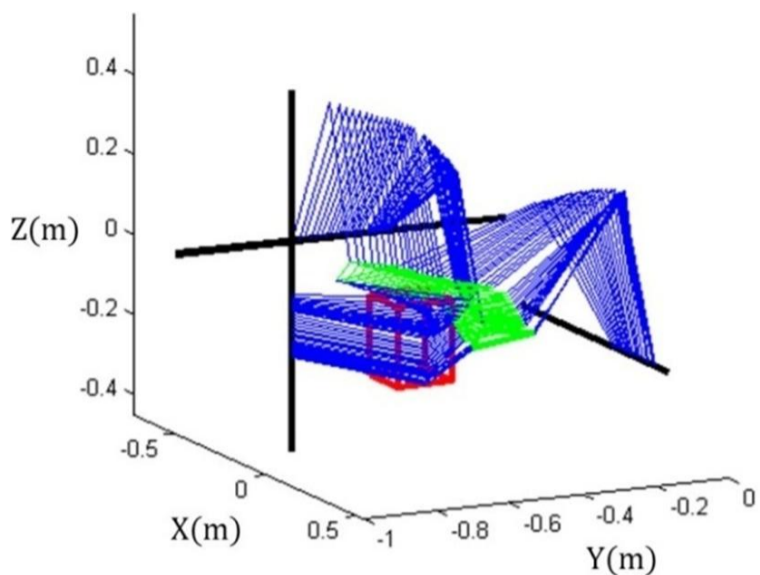
$$T = \delta t \times h \quad (27)$$

که در آن h طول افق پیش‌بینی را تعیین می‌کند. لازم به ذکر است برای این مساله ترکیبی از کمترین زمان طی مسیر و کوتاه‌ترین مسیر به عنوان تابع

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.

5- شبیه‌سازی الگوریتم بر روی مدل ربات واقعی

با استفاده از الگوریتم ارائه شده، مسیریابی این ربات برای دو حالت مختلف شبیه‌سازی می‌گردد که نتایج اولیه آن در شکل 7 و 8 نشان داده شده‌اند. با در نظر گرفتن مقادیر ورودی به شرح جدول 2، با تابع هزینه کوتاه‌ترین مسیر، ربات مسیری را به صورت شکل 7 طی می‌کند. در صورتی که طول افق پیش‌بینی برابر $h = 7$ و سرعت بیشینه هر مفصل نسبت به مفصل قبل $v_{max} = 0/05$ متر بر ثانیه باشد، برای این مسیر، کل زمان 37 ثانیه گزارش شده است.



شکل 8 نتیجه شبیه‌سازی الگوریتم مسیریابی ربات موازی تریپترون با شرایط جدول 2 در حضور مانع ثابت و با تابع هزینه کوتاه‌ترین مسیر از نماهای مختلف. طول مسیر طی شده توسط مجری نهایی برابر 0/44 متر و زمان طی آن 36 ثانیه می‌باشد.

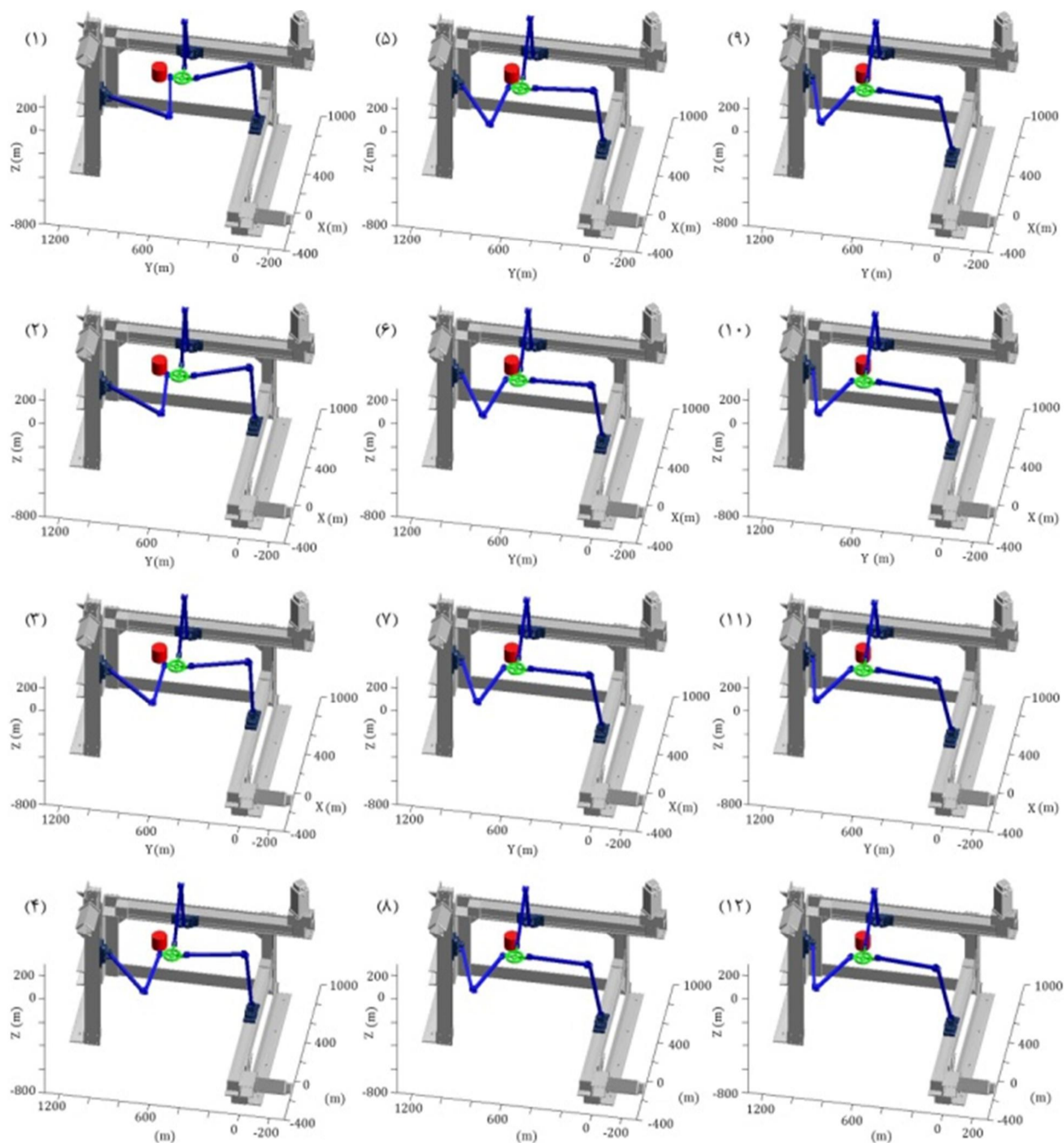
جدول 2 مقادیر ورودی حالت اول برای نقاط ابتدایی و انتهایی مجری نهایی و محل و

ابعاد مانع					
h (m)	R (m)	z (m)	y (m)	x (m)	
		0	-0/5	-0/3	نقطه ابتدایی
		0/1	-0/6	0/1	نقطه انتهایی
0/1	0/05	0/1	-0/6	-0/2	مانع ثابت

جدول 3 مقادیر ورودی حالت دوم برای نقاط ابتدایی و انتهایی مجری نهایی و محل و

ابعاد مانع					
h (m)	R (m)	z (m)	y (m)	x (m)	
		0	-0/7	-0/1	نقطه ابتدایی
		-0/2	-0/4	0/1	نقطه انتهایی
0/35	0/07	-0/15	-0/6	0	مانع ثابت

شکل 7 نتیجه شبیه‌سازی الگوریتم مسیریابی ربات موازی تریپترون با شرایط جدول 1 در حضور مانع ثابت و با تابع هزینه کوتاه‌ترین مسیر از نماهای مختلف. برای طی این مسیر، کل زمان 37 ثانیه گزارش شده است. همچنین طول مسیر طی شده توسط مجری نهایی برابر 0/45 متر بوده است.



شکل 9 فریم‌هایی از فیلم شبیه‌سازی الگوریتم مسیریابی بر روی مدل واقعی ربات موازی تریپترون با شرایط جدول 2 و با تابع هزینه کوتاه‌ترین مسیر در حضور مانع ثابت

به این منظور از بسته نرم‌افزاری کد 2مت استفاده شده است که برای انتقال مدل طراحی شده با استفاده از نرم‌افزار سالیدورکز به نرم‌افزار متلب به کار گرفته می‌شود.

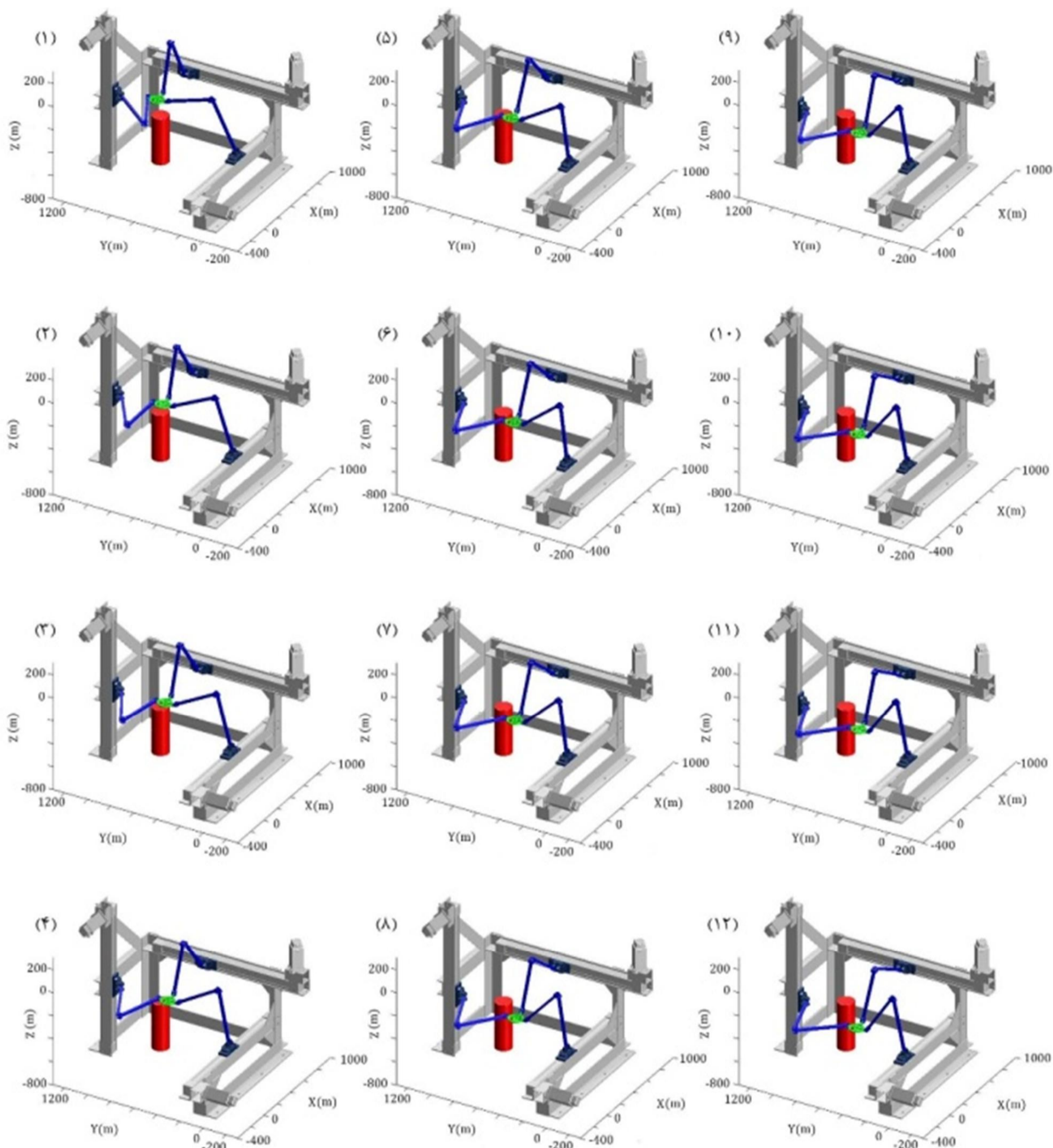
6- نتیجه‌گیری

در این مقاله مساله مسیریابی ربات موازی تریپترون مورد بحث قرار گرفت. این ربات دارای سه درجه آزادی مستقل خطی می‌باشد و هیچ نقطه تکنیکی در فضای کاری خود ندارد، که باعث کارایی بالای آن شده و برای بررسی الگوریتم ارائه شده مناسب است. در ابتدا به معرفی کوتاهی از ربات و مشخصات نرم‌افزاری و سخت‌افزاری آن پرداخته و سپس با ارائه الگوریتم مسیریابی مربوط به آن، نتایج حاصل از شبیه‌سازی گزارش شده‌اند.

همچنین طول مسیر طی شده توسط مجری نهایی برابر $0/45$ متر بوده است. بیشینه زمان حل برای هر مرحله نیز $0/9$ ثانیه بوده است.

مساله بار دیگر با شرایط جدید ذکر شده در جدول 3 برای تابع هزینه کوتاه‌ترین مسیر و با همان مقادیر طول افق پیش‌بینی و سرعت بیشینه، حل می‌گردد. طبق گزارش طول مسیر طی شده توسط مجری نهایی برابر $0/44$ متر و زمان طی آن 36 ثانیه می‌باشد. شکل 8 نتیجه حاصل از این شبیه‌سازی را نشان می‌دهد. بیشترین زمان محاسبات نیز برای هر مرحله $0/9$ ثانیه می‌باشد.

با استفاده از نتایج به دست آمده در قسمت قبل، شبیه‌سازی با استفاده از مدل ربات واقعی نیز انجام گرفته است که در شکل‌های 9 و 10 فریم‌هایی از آن مشاهده می‌گردد.



شکل 10 فریم‌هایی از فیلم شبیه‌سازی الگوریتم مسیریابی بر روی مدل واقعی ربات موازی تریپترون با شرایط جدول 3 و با تابع هزینه کوتاه‌ترین مسیر در حضور مانع ثابت

محاسبات می‌شود که معمولاً مطلوب نیست، از این رو این پارامتر با توجه به خواسته کاربر و اولویت دقت یا سرعت تعیین می‌گردد. برای حل مسأله بهینه‌سازی نیز از بسته نرم‌افزاری گروبی استفاده شده است که یک بسته نرم‌افزاری تجاری بوده و از سرعت محاسباتی بالایی برخوردار است. همچنین لازم به ذکر است، برای هرچه دقیق‌تر بودن شبیه‌سازی از مدل ربات واقعی که با استفاده از بسته نرم‌افزاری کد2مت از نرم‌افزار سالیدورکز به نرم‌افزار متلب انتقال یافته است، استفاده گردیده است. نتایج حاکی از این است که ربات به خوبی مسیر ایمن و عاری از تداخل را پیموده و همچنین زمان محاسبات نیز کمتر 1 ثانیه بوده که برای این کاربرد بلادرنگ محسوب می‌گردد.

الگوریتم پیشنهاد شده برای مسیریابی این ربات ترکیبی از مفاهیم بهینه‌سازی محدب، برنامه‌ریزی گسسته و کنترل افق پیش‌بین می‌باشد که منجر به مزایای ارزشمندی همچون سراسری بودن جواب بهینه به دست آمده و کاهش زمان محاسبات گردیده است. با توجه به وجود قیود نامحدب در مسأله، از جمله قیود سینماتیکی و همچنین قیود مربوط به موانع موجود در فضای کاری، از روش تقریب زدن برای محدب‌سازی مسأله بهینه‌سازی بهره برده شده است. لازم به ذکر است که این روش باعث ایجاد خطا می‌گردد که با استفاده از لوپ کنترلی با فیدبک از خطای جمع‌شونده جلوگیری می‌گردد. همچنین با افزایش تعداد اضلاع چندضلعی‌ها می‌توان از مقدار خطا کاسته و جواب‌های دقیق‌تری به دست آورد. از طرف دیگر این امر باعث افزایش زمان

7- مراجع

- [10] H. Ding, G. Reibig, D. Grob, O. Sturberg, Mixed-Integer Programming for Optimal Path Planning of Robotic Manipulators, *International Conference on Automation Science and Engineering*, 2011.
- [11] M. H. Korayem, M. Nazemizade, and H. Ghaffarpour. Optimal Path Planning of Nonholonomic Mobile Robots using Optimal Control Method and Verification of the Method via Experimental Tests of the Scout Mobile Robot, *Journal of Modares Mechanical Engineering*, Vol. 12, No. 2, pp. 87-94, 2012. (In Persian)
- [12] M. H. Korayem, M. Irani. New Optimization Method to Solve Motion Planning of Dynamic Systems: Application on Mechanical Manipulators, *Journal of Multibody System Dynamics*, Vol. 31, No. 2, pp. 169-189, 2014.
- [13] R. Sabbagh Novin, M. Tale Masouleh, and A. Karimi. Path Planning of 2 and 3 Link Planar Serial Manipulators via Convex Optimization and Receding Horizon Concept, *22th International Conference on Mechanical Engineering (ISME)*, 2014. (In Persian)
- [14] *Gurobi Optimizer Reference Manual*, Accessed 20 July 2014, URL: <http://www.gurobi.com>.
- [15] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2009.
- [16] T. Schouwenaars, J. How, E. Feron, Receding Horizon Path Planning with Implicit Safety Guarantees, *American Control Conference*, 2004.
- [17] X. Kong, and C. M. Gosselin, Kinematics and singularity analysis of a novel type of 3-CRR 3-DOF translational parallel manipulator, *The International Journal of Robotics Research*, 2002.
- [18] X. Kong, and C. M. Gosselin, *Type synthesis of parallel mechanisms*, Springer Publishing Company, Incorporated, 2007.
- [1] J-P. Merlet, *Parallel robots*. Vol. 74. Springer Science & Business Media, 2001.
- [2] C. M. Gosselin, et al. Parallel mechanisms of the multipteron family: kinematic architectures and benchmarking, *IEEE International Conference on Robotics and Automation*, 2007.
- [3] B. Krogh. A Generalized Potential Field Approach to Abstacle Avoidance Control, *ASME Conference of Robotic Research: The Next Five Years and Beyond*, Bethlehem, Pennsylvania, 1984.
- [4] O. Khatib, Real-time Obstacle Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1986.
- [5] A. Gasparetto, P. Bpsariol, A. Lanzutti, R. Vidoni, Trajectory Planning in Robotics, *Mathematics in Computer Science*, pp. 269-279, 2012.
- [6] L. Blackmore, B. Williams, Optimal Manipulator Path Planning With Obstacles Using Disjunctive Programming, *Proceedings of the American Control Conference*, 2006.
- [7] H. Ding, M. Zhou, O. Stursberg, Optimal Motion Planning for Robotic Manipulators with Dynamic Obstacles Using Mixed-integer Linear Programming, *17th Mediterranean Conference on Control and Automation*, 2009.
- [8] H. Ding, M. Zhou, O. Stursberg, Optimal Path Planning in the Workspace for Articulated Robots Using Mixed Integer Programming, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.
- [9] H. Ding, S. Reibig, O. Stursberg, Increasing Efficiency of Optimization-based Path Planning for Robotic Manipulators, *50th IEEE Conference on Decision and Control and European Control Conference*, 2011.