مسیریابی روبات بازویی با استفاده از برنامه‌ریزی غیرخطی عدد صحیح مختلط

اِلیپس مسیحی*[۱] و امیرعباس ابوئی مهریزی[۲]

[۱] استادیار بخش مهندسی صنایع- دانشگاه تربیت مدرس

[۲] دانش‌آموخته کارشناسی ارشد مهندسی صنایع دانشگاه تربیت مدرس و مربی بخش مهندسی صنایع مجتمع آموزش عالی بم

(تاریخ دریافت ۹۱/۲/۹، تاریخ دریافت روایت اصلاح‌شده ۹۱/۷/۱، تاریخ تصویب ۹۱/۱۲/۲۷)

چکیده

در این مقاله، روشی برای برنامه‌ریزی حرکت روبات بازویی مسطح در محیط با مانع با استفاده از مدل‌سازی ریاضی ارائه شده است، به طوری که حین حرکت روبات از پیکربندی اولیه به سمت نقطه هدف، لینک‌های آن با موانع برخورد نداشته باشند. پس از ساختن فضای کاری، کوتاه‌ترین مسیر حرکت از نقطه اولیه عمل‌کننده نهایی به هدف مورد نظر، به کمک گراف دیدنگار تعمیم یافته، ایجاد شده و به نقاط زیرهدف تقسیم می‌شود که باید به ترتیب توسط عمل‌کننده نهایی پیموده شوند. سپس یک مدل ریاضی از نوع برنامه‌ریزی غیرخطی عدد صحیح مختلط (MINLP) با در نظر گرفتن تابع هدف کمینه‌سازی، فاصله عمل‌کننده نهایی تا زیرهدف‌ها به صورت متوالی حل شده و زاویه هر لینک به‌گونه‌ای تعیین می‌شود که لینک‌ها در موانع قرار نگیرند. برای بالا بردن ضریب ایمنی در برخورد با موانع، محیط موانع با افزودن حاشیه‌ای بزرگ‌تر در نظر گرفته می‌شوند. همچنین روش ذکرشده برای کاهش متغیرهای صفر و یک و تعداد محدودیت‌ها و در نتیجه کاهش زمان حل، بهبود داده شده است.

واژه‌های کلیدی: روبات بازویی، برنامه‌ریزی حرکت، برنامه‌ریزی ریاضی، مدل غیرخطی عدد صحیح مختلط، گراف دیدنگار

مقدمه

یکی از مهم‌ترین وظایف روبات متحرک، برنامه‌ریزی حرکت آن است که عبارت است از حل یک مسئله تصمیم‌گیری برای تعیین اینکه روبات چگونه می‌تواند از نقطه شروع تا نقطه هدف حرکت کند، در حالی که با موانع ثابت و متحرک موجود در محیط برخورد نداشته باشد. با استفاده از مدل‌های مختلف تحقیق در عملیات مانند برنامه‌ریزی خطی، برنامه‌ریزی اعداد صحیح، برنامه‌ریزی پویا و غیره، می‌توان محدودیت‌های حرکتی و مسائل مربوط به کنترل روبات را فرمول‌بندی کرد و به کمک آن حل بهینه مسئله را با توجه به آن شرایط به دست آورد.

مسئله برنامه‌ریزی حرکت روبات از نظر پیچیدگی به عنوان یک مسئله PSPACE-hard و NP-complete شناخته می‌شود[۱] که این پیچیدگی، مانع توسعه الگوریتم‌های کاربردی شده است [۲]. یکی از انواع روبات‌ها، روبات‌های بازویی است که برنامه‌ریزی حرکت این نوع روبات‌ها در کاربردهای دنیای واقعی مانند صنایع اتومبیل‌سازی، الکترونیک و هوافضا اهمیت زیادی دارد. یک روبات بازویی، وسیله‌ای مکانیکی است که از لینک‌ها و مفصل‌های متصل به هم ساخته شده است (درجات آزادی) و برای انجام کارهای اتوماتیک با استفاده از عمل‌کننده نهایی[۱] خود حرکت می‌کند. با نصب عمل‌کننده‌های نهایی مختلف می‌توان اعمال جوشکاری، رنگ‌کاری، اتوکشی، مونتاژ، برداشتن و قرار دادن، روی هم چیدن (پالت‌گذاری) و غیره را انجام داد.

کار اصلی در کنترل روبات‌های بازویی، تعیین مسیری از یک موقعیت اولیه عمل‌کننده نهایی به یک موقعیت (یا پیکربندی) هدف مشخص است. بنابراین برنامه‌ریزی حرکت برای روبات بازویی به معنی طراحی مسیر حرکت عمل‌کننده نهایی بین نقاط طراحی شده در سطح کار است؛ در حالی که با موانع برخورد نداشته باشد. تابع هدف این گونه مسائل می‌تواند کوتاه‌ترین زمان، کوتاه‌ترین مسیر و یا مسیری با کمترین انرژی مصرفی برای روبات در نظر گرفته شود.

* نویسنده مسئول: تلفن: ۸۲۸۸۴۹۳۹ فکس: ۸۲۸۸۳۳۸۱ Email: masehian@modares.ac.ir

رویکردهای پایه‌ای برنامه‌ریزی حرکت عبارتند از خط مسیر[۲]، تجزیه سلولی[۳]، میدان پتانسیل[۴] و برنامه‌ریزی ریاضی که در این بخش به مرور مقالاتی که از رویکرد برنامه‌ریزی ریاضی استفاده کرده‌اند، پرداخته می‌شود.

در سال ۱۹۸۶، شین و مکی [۳] راه حلی برای مسئله مینیمم‌کردن هزینه حرکت یک روبات بازویی در میان مسیر معین هندسی با توجه به محدودیت‌های گشتاور/نیرو، در نظر گرفتن اتصالات و دینامیک غیرخطی روبات بازویی ارائه کرده‌اند که در آن از برنامه‌ریزی پویا برای پیدا کردن موقعیت، سرعت، شتاب و گشتاور استفاده شده است.

مقاله چتیبی و همکاران [۴] در سال ۲۰۰۴ در مورد مسئله برنامه‌ریزی خط سیر با کمترین هزینه برای روبات‌های بازویی بحث کرده است، که شامل وصل دو نقطه در فضای عملیاتی با در نظر گرفتن معادلات دینامیکی حرکت و محدود کردن موقعیت‌های مفصل، سرعت، تغییر شتاب و گشتاورها است. مسئله کلی کنترل بهینه توسط یک مدل منحنی مهار شده سه بعدی توسط ارزیابی‌های موقت مفصل به یک مسئله بهینه‌سازی غیرخطی با محدودیت تبدیل شده و سپس به کمک روش برنامه‌ریزی درجه دو متوالی حل می‌شود.

ویلیامز و بلکمور [۵] با مطرح کردن مسئله برنامه‌ریزی حرکت روبات در حضور موانع به عنوان برنامه‌ریزی انفصالی[۵]، روشی را ارائه کرده‌اند که به طور کامل در فضای کاری روبات بازویی برنامه‌ریزی می‌شود، که این کار باعث کاهش محاسبات فرآیند نگاشت موانع از فضای کاری به فضای پیکربندی می‌شود.

در مقاله گوآنگیو و همکاران [۶] یک رویکرد سیستماتیک برای برنامه‌ریزی خط سیر بهینه با وجود تکین‌های[۶] سینماتیکی و ردیابی دقیق مسیر مطلوب عمل‌کننده نهایی ارائه شده است که از برنامه‌ریزی پویا برای حصول به بهینگی استفاده شده است.

یکی از مباحث مهم در برنامه‌ریزی و کنترل بازوهای با تزاید[۷]، جلوگیری از برخورد بهنگام با موانع است[۷]. در این مقاله جنبه‌های الگوریتمی و محاسباتی معیار/فرمولاسیون بر مبنای نامعادله برای جلوگیری از برخورد با موانع روبات بازویی PA10 ارائه شده است. فرمولاسیون به عنوان یک مسئله برنامه‌ریزی درجه دو[۸] (QP) یکپارچه می‌شود.

در سال ۲۰۰۷، چوی و امیر [۸] الگوریتمی مبتنی بر برنامه‌ریزی پویا برای مسیریابی روبات بازویی در فضای دو بعدی با فرض برخورد نکردن روبات با خود ارائه کردند. این روش، توپولوژی بازو و موانع را برای عامل فضای جستجو و کاهش پیچیدگی مسئله برنامه‌ریزی به کار می‌برد.

در مقاله دیگری، ژانگ و همکاران [۹] در سال ۲۰۰۸، یک شبکه عصبی اولیه-دوگان را بر پایه نامعادلات خطی تغییرپذیر[۹] برای برنامه‌ریزی حرکت تکراری بهنگام یک روبات بازویی PA10 ارائه کردند که در آن، محدودیت‌های فیزیکی مربوط به مفاصل و سرعت آنها در فرمول‌بندی مسئله در نظر گرفته شده است و در نهایت به عنوان یک مسئله برنامه‌ریزی درجه دو، بار دیگر فرمول‌بندی می‌شود.

در سال ۲۰۰۹، دینگ و همکاران [۱۰] یک برنامه‌ریز مسیر برای روبات بازویی ارائه کردند، به طوری که عمل‌کننده نهایی برای حرکت بین دو نقطه در فضای کاری، از برخورد با موانع متحرک پرهیز کند. روش ارائه‌شده، مسئله را به یک مدل برنامه‌ریزی عدد صحیح مختلط با لحاظ کردن محدودیت‌های متغیر در طول زمان که ناشی از موانع متحرک می‌شوند، فرمول‌بندی می‌کند.

همان محققان در مقاله دیگری، یک رویکرد بهینه برنامه‌ریزی حرکت با استفاده از برنامه‌ریزی خطی عدد صحیح مختلط (MILP) تعبیه شده در روش کنترل مدل پیشگویانه[۱۰](MPC) پیشنهاد کرده‌اند که برای برنامه‌ریزی حرکت روبات بازویی با موانع متحرک (مانند اپراتور) در فضای $C\times T$ (ترکیب پیکربندی C و زمان T) به کار برده شده است. در این مقاله، هدف مینیمم‌سازی زمان، از شروع تا هدف، در افق پیش‌بینی انتخاب می‌شود[۱۱].

دینگ و همکاران [۱۲] در مقاله‌ای دیگر کار خود را توسعه داده و با به کارگیری خواص هندسی در مدل MILP باعث کاهش چشمگیر متغیرهای تصمیم صفر و یک شدند. این روش باعث کاهش تکرارهای سیمپلکس در طول حل MILP نیز می‌شود. آنها همچنین کاربرد روش برای دو روبات صنعتی را بررسی کردند.

روش ارائه‌شده در این مقاله، یک مدل برنامه‌ریزی غیرخطی عدد صحیح مختلط[۱۱] (MINLP) است. پس از تعیین فضای کاری روبات، ابتدا کوتاه‌ترین مسیر حرکت عمل‌کننده نهایی از نقطه اولیه به هدف مورد نظر، به کمک روش دیدنگار تعمیم‌یافته، توسط نرم‌افزار

MATLAB تولید می‌شود و پس از شکستن مسیر حاصل به زیرمسیرهای مساوی، اطلاعات به صورت مدل ریاضی قابل فهم برای نرم‌افزار LINGO ارسال شده و مدل برای حرکت روبات اجرا می‌شود. نتیجه کار بار دیگر به نرم‌افزار MATLAB برای به روزآوری مدل ریاضی برگردانده شده و این رفت و برگشت تا رسیدن به نقطه هدف ادامه می‌یابد.

مدل ریاضی، با در نظر گرفتن تابع هدف کوتاه‌ترین فاصله تا زیر هدف به صورت متوالی اجرا شده و زوایه هر لینک به کمک مدل به گونه‌ای تعیین می‌شود که لینک‌ها در موانع قرار نگیرند. برای افزایش ضریب ایمنی در برخورد موانع به اندازه حاشیه‌ای ایمنی بزرگ‌تر شده و یک گراف دیدنگار با توجه به موانع بزرگ‌تر شده ساخته می‌شود. همچنین روشی برای کاهش متغیرهای صفر و یک و تعداد محدودیت‌ها ارائه شده است که باعث کاهش زمان حل نیز می‌شود.

مفروضاتی که در این تحقیق در نظر گرفته شده، عبارتند از:

- مفصل‌ها از نوع چرخشی[۱۲] هستند.
- لینک‌ها قطعات مستقیم و صلبی هستند که هیچگونه خمیدگی در آن‌ها نبوده و طول آنها ثابت است.
- محیط و فضای کاری دو بعدی و پیوسته است.
- مختصات نقطه هدف مشخص است.
- در فضای کاری موانعی محدب وجود دارند که ابعاد و مکان آنها مشخص است. البته با تجزیه موانع نامحدب به نواحی محدب، می‌توان مسئله را با روش ارائه شده حل کرد.
- محدودیت‌های دینامیکی مثل سرعت و شتاب در نظر گرفته نمی‌شود.

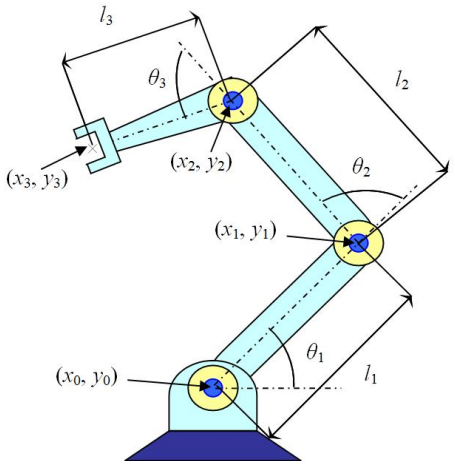
بخش‌های مقاله به صورت زیر سازماندهی شده‌اند: در بخش دوم، مدل‌سازی ارائه‌شده برای حل مسئله برنامه‌ریزی حرکت روبات بازویی بیان می‌شود. این بخش در سه زیربخش به تشریح مدل‌سازی سینماتیک روبات و مدل‌سازی حرکت روبات بازویی در فضای بدون مانع، با یک مانع و با چند مانع می‌پردازد. بخش سوم، حل مدل و بهبود آن در فضای کاری با استفاده از مرتبط‌کردن نرم‌افزارهای MATLAB و LINGO را تشریح می‌کند. در این بخش، چند مثال برای نمایش برنامه‌ریزی حرکت روبات بازویی اجرا و نمایش داده شده‌اند. در دو بخش آخر نیز نتایج محاسباتی، مقایسات و نتیجه‌گیری بیان می‌شود.

مدل‌سازی ریاضی روبات بازویی

در این بخش چگونگی مدل‌سازی ریاضی روبات بازویی بررسی می‌شود. پس از تشریح سینماتیک روبات، مدل‌سازی در دو قسمت اصلی معرفی می‌شود: ابتدا مدل حرکت روبات در فضای بدون مانع ارائه می‌شود، که در آن روبات سعی می‌کند در امتداد مسیری معین به طور متوالی و با هماهنگی منظم بین لینک‌ها، حرکتی عاری از جابه‌جایی‌های ناگهانی داشته باشد. سپس به نحوه مدل‌سازی یک مانع در فضای کاری و ارتباط آن با روبات بازویی پرداخته شده و شرایط قرار نگرفتن یک نقطه و نیز کل لینک داخل موانع بررسی می‌شوند. در انتها نیز مدل کلی ارائه‌شده به صورت یک مدل برنامه‌ریزی غیرخطی عدد صحیح مختلط معرفی و تشریح می‌شود.

مدل‌سازی سینماتیک روبات

همان گونه که بیان شد، روبات بازویی از یک سری لینک‌ها و مفصل‌هایی تشکیل شده است که به صورت متوالی به یکدیگر متصل هستند. فرض می‌شود که مفصل‌ها تنها نقاطی هستند که لینک‌ها حول آن‌ها به چرخش در می‌آیند.



شکل ۱: روبات بازویی دو بعدی با سه لینک

در شکل ۱) یک روبات بازویی مسطح با سه لینک نشان داده شده است. مختصات پایه روبات با (x_0, y_0) و

طول هر لینک با l_i نشان داده شده است. θ_1 زاویه لینک اول با جهت مثبت محور x و مقادیر θ_2 و θ_3 به ترتیب بیانگر زوایای بین لینک دوم و اول و لینک سوم با دوم هستند. با معلوم بودن مختصات پایه روبات، طول لینک‌ها و مقادیر زوایای آنها نسبت به یکدیگر، می‌توان مختصات پایانی هر لینک را با استفاده از روابط قطبی سینماتیک مستقیم به دست آورد. در روابط (۱) نحوه محاسبه مختصات پایانی لینک اول نشان داده شده است:

$$\begin{aligned} x_1 &= x_0 + l_1 \cos(\theta_1) \\ y_1 &= y_0 + l_1 \sin(\theta_1) \end{aligned} \qquad (۱)$$

با استفاده از مختصات پایانی لینک اول روبات، می‌توان مختصات پایانی لینک دوم را به دست آورد. البته باید توجه کرد که برای محاسبه مختصات قطبی لینک دوم، زاویه مختصات باید نسبت به جهت مثبت محور x در نظر گرفته شود که مقدار آن برابر $\theta_1+\theta_2$ است. در روابط (۲) مختصات پایانی لینک دوم نشان داده شده است:

$$\begin{aligned} x_2 &= x_1 + l_2 \cos(\theta_1+\theta_2) \\ &= x_0 + l_1 \cos(\theta_1) + l_2 \cos(\theta_1+\theta_2) \\ y_2 &= y_1 + l_2 \sin(\theta_1+\theta_2) \\ &= y_0 + l_1 \sin(\theta_1) + l_2 \sin(\theta_1+\theta_2) \end{aligned} \qquad (۲)$$

مختصات پایانی لینک سوم را نیز می‌توان بر اساس مختصات پایانی لینک دوم در مختصات قطبی بیان کرد. باید توجه داشت که مقدار زاویه لینک سوم با جهت مثبت محور x برابر مجموع زوایای سه لینک اول $\theta_1+\theta_2+\theta_3$ است. روابط (۳) بیانگر مختصات لینک سوم روبات بازویی هستند:

$$\begin{aligned} x_3 &= x_0 + \sum_{i=1}^{3} l_i \cos\left(\sum_{j=1}^{i}\theta_j\right) \\ y_3 &= y_0 + \sum_{i=1}^{3} l_i \sin\left(\sum_{j=1}^{i}\theta_j\right) \end{aligned} \qquad (۳)$$

با توجه به روابط (۳) می‌توان مختصات نقاط انتهایی بازوهای یک روبات بازویی را محاسبه کرد. در صورتی که یک روبات بازویی با N لینک باشد، مختصات انتهایی لینک N-ام را می‌توان از روابط (۴) به دست آورد:

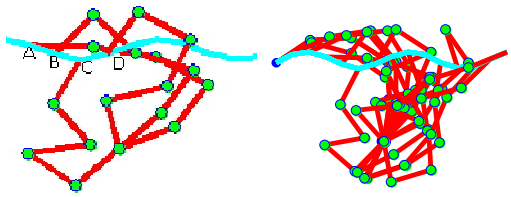
$$\begin{aligned} x_N &= x_0 + \sum_{i=1}^{N} l_i \cos\left(\sum_{j=1}^{i}\theta_j\right) \\ y_N &= y_0 + \sum_{i=1}^{N} l_i \sin\left(\sum_{j=1}^{i}\theta_j\right) \end{aligned} \qquad (۴)$$

حرکت روبات بازویی در فضای بدون مانع

حرکت روبات در فضای بدون مانع، باید به گونه‌ای باشد که با توجه به پیکربندی اولیه روبات و نقطه هدفی که روبات باید به آن برسد، توالی منظمی از حرکات به دست آید، به طوری که روبات بتواند در هر لحظه حرکتی روان و بدون تغییر ناگهانی در پیکربندی داشته باشد. در روابط (۴) مختصات لینک پایانی روبات به صورت تابعی از زوایای تمامی لینک‌ها بیان شده است. از این واقعیت می‌توان استفاده کرد و یک مدل برنامه‌ریزی ریاضی برای حرکت روبات به نقاط دیگر بیان کرد، بدین صورت که اگر نقطه $G(x_g, y_g)$ در همسایگی نقطه انتهایی روبات به عنوان نقطه هدف برای روبات در نظر گرفته شود، زوایای لینک‌های روبات به گونه‌ای محاسبه شوند که فاصله نقطه انتهایی روبات با نقطه G مینیمم شود. در روابط (۵) مدل ریاضی برای حداقل‌کردن فاصله نقطه پایانی روبات با نقطه هدف بیان شده است:

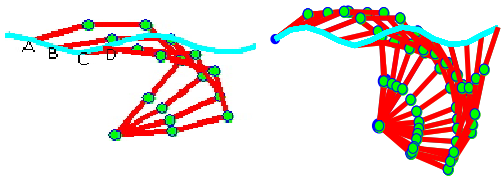
$$\begin{aligned} &Min\ z = \sqrt{(x_N - x_g)^2 + (y_N - y_g)^2} \\ &Subject\ to: \\ &x_N = x_0 + \sum_{i=1}^{N} l_i \cos\left(\sum_{j=1}^{i}\theta_j\right) \\ &y_N = y_0 + \sum_{i=1}^{N} l_i \sin\left(\sum_{j=1}^{i}\theta_j\right) \end{aligned} \qquad (۵)$$

اکنون از روابط بالا می‌توان برای یافتن حرکت روبات روی یک مسیر استفاده کرد؛ بدین صورت که مسیر به قسمت‌های کوچک‌تری تقسیم‌بندی شود و در هر مرحله روبات از حالت کنونی‌اش به نقطه بعدی مسیر تغییر حالت دهد. در شکل (الف) حرکت نقطه انتهایی یک روبات بازویی روی یک مسیر سینوسی نشان داده شده است. البته به دلیل غیرخطی بودن مسئله، عیب بارز این روش پراکندگی جواب‌ها نسبت به یکدیگر هستند. شکل (ب) مثال دیگری با چهار حرکت متوالی روبات را نشان می‌دهد. با وجود این، نقطه پایانی روبات به هدف دست یافته است، اما در توالی حرکت‌ها بین چهار نقطه زوایای روبات نسبت به حالات قبلی خود تغییرات خیلی زیادی داشته‌اند.



(ب) (الف)

شکل ۲: حرکت روبات بازویی روی مسیر سینوسی به طور نامنظم با استفاده از روابط (۵)



(ب) (الف)

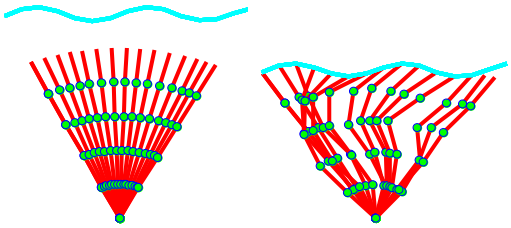
شکل ۳: حرکت روبات بازویی روی مسیر سینوسی به طور منظم

برای حل مشکل پراکندگی زیاد جواب‌ها، می‌توان از یک محدودیت جدید که مانع از پراکنده شدن جواب نسبت به جواب قبلی می‌شود، استفاده کرد. با اضافه کردن رابطه (۶) به دو محدودیت قبلی، مدل ریاضی مربوط به حرکت روبات بازویی در فضای عاری از مانع کامل می‌شود:

$$\left|\theta_j - \theta'_j\right| < \varepsilon, \qquad \forall j \qquad (۶)$$

در این رابطه، θ'_j مقدار اولیه (حالت فعلی) زوایای روبات و مقدار ε که توسط کاربر تعیین می‌شود، نشانگر میزان انحراف مجاز یک لینک از راستای قبلی خود است. شکل (۳)، مدل جدید حل شده مسئله شکل (۲) است. در این شکل‌ها، مقدار $\varepsilon = \pi/10$ در نظر گرفته شده است. همان طور که مشاهده می‌شود، پراکندگی جواب‌ها نسبت به شکل قبلی از بین رفته و روبات حرکت منظمی را روی مسیر طی می‌کند که می‌تواند جواب قابل قبولی به شمار آید.

شکل ۴(الف) و ۴(ب) مثال‌های دیگری از حل مسئله با استفاده از مدل ارائه‌شده را نشان می‌دهند که برای آزمودن درستی مدل در حالت‌های مختلف به کار گرفته شده‌اند.



(ب) (الف)

شکل ۴: حل مثال‌های مختلف به کمک مدل

حرکت روبات بازویی در فضای با یک مانع

تا کنون روبات بازویی در فضایی عاری از موانع، مدل‌سازی شده است. واضح است که در دنیای واقعی برای آنکه یک روبات بتواند درجه‌ای از هوشمندی را داشته باشد، باید به طور خودکار موانعی که سر راه خود دارند را تشخیص داده و با در نظر گرفتن آن‌ها مسیر خود را طوری انتخاب کند که با پرهیز از برخورد با موانع به نقطه هدف مورد نظر برسد. فرض بر آن است که موانع موجود سر راه روبات بازویی از قبل شناسایی شده‌اند و برنامه‌ریزی حرکت روبات در حالت غیر بهنگام انجام می‌پذیرد. برای تعمیم مسئله برنامه‌ریزی حرکت روبات بازویی در حالت قبل به حالتی که موانع نیز در سر راه قرار دارند، باید موانع نیز در مدل برنامه‌ریزی ریاضی گنجانده شوند.

برای مدل‌سازی موانع در فضای کاری از روشی شبیه به کار [۵] و [۱۰] استفاده شده است. ابتدا روابط ریاضی برخورد نکردن یک نقطه واحد روی روبات بیان شده و سپس این روش برای روبات بازویی با مجموعه‌ای از نقاط روی لینک‌ها و مفاصل توسعه داده می‌شود.

مدل سازی برخورد نکردن یک نقطه با مانع

در شرایطی که یک نقطه بیرون از یک مانع چند وجهی واقع شود، می‌توان آن را به کمک محدودیت‌های عدد صحیح مختلط فرموله کرد: با فرض اینکه یک چندوجهی به عنوان مانع می‌تواند به صورت اشتراک N نیم فضا با بردارهای نرمال بیرونی $c_1,\ldots,c_N, c_j \in R^{1\times n}$ و ثابت‌های مکانی $d_1,\ldots,d_N$ $(\forall d_j \in R)$ بیان شود و x یک نقطه در فضای کاری اقلیدسی n بُعدی $X \in \Re^n$ در نظر گرفته شود، فضای X درون مانع می‌تواند به صورت رابطه (۷) بیان شود:

$$p = \left\{x \,\middle|\, C \in R^{N\times n}, d \in R^{N\times 1} : C\cdot x < d\right\} \qquad (۷)$$

برای اینکه یک نقطه خارج از چندوجهی قرار گیرد، باید حداقل یکی از N نامعادله بالا نقض شود. شرط مناسب برای اینکه یک نقطه بیرون از چندوجهی قرار گیرد را می‌توان به کمک روش M بزرگ مانند رابطه (۸) بیان کرد:

$$C \cdot x \geq d + (b - \mathbf{1})M \qquad (۸)$$

که در آن $b = (b_1, ..., b_N)^T$ برداری از متغیرهای صفر و یک $(\forall b_j \in \{0,1\})$، $\mathbf{1}=1^{N\times 1}$ برداری از یک‌ها و M یک ثابت بزرگ (به جای ∞) است. هر یک از معادلات موجود در رابطه (۸) هنگامی غیرفعال می‌شود که b_j متناظر آن مقدار صفر بگیرد و هنگامی فعال می‌شود که b_j متناظر آن مقدار ۱ بگیرد. برای اینکه نقطه‌ای مانند x بیرون از چندوجهی p تعیین شده توسط (۷) قرار گیرد، باید حداقل یکی از N نامعادله طبق شرط زیر ارضا شود:

$$\sum_{j=1}^{N} b_j \geq 1 \qquad (۹)$$

طبق معادلات (۸) و (۹) که مجموعه‌ای از محدودیت‌ها را برای یک مسئله بهینه‌سازی بیان می‌کنند، تضمین می‌شود که هیچ حالتی مانند x (یعنی موقعیت مفصل) هرگز داخل منطقه اشغال شده توسط مانع قرار نمی‌گیرد. در حالتی که بیشتر از یک مانع در نظر گرفته شود، فرمولاسیون مشابهی با ماتریس‌های C و d و متغیرهای صفر و یک (b) متناظر با آن برای هر مانع می‌تواند مورد استفاده قرار گرفته و به محدودیت‌های قبلی اضافه شود.

مدل‌سازی برخورد نکردن روبات بازویی با مانع

برای اینکه برنامه‌ریزی برای روبات بازویی به طور مناسب انجام گیرد، باید بدنه روبات در فضای X بیان شود. بدین منظور نقاط محدودی روی هندسه (بدنه) روبات یعنی روی هر لینک انتخاب می‌شوند. فرض کنید که لینک‌ها به صورت خطوط مستقیم بین دو مفصل مجاور تشکیل شده‌اند. آنگاه نقاط روی لینک‌ها از معادله (۱۰) به دست می‌آیند:

$$x_{sj} = \lambda_s x_j + (1 - \lambda_s) x_{j+1} \qquad (۱۰)$$

که x_{sj}، s-امین نقطه روی لینک jاست. با انتخاب $\lambda_s \in [0,1]$ که $s \in \{1,...,S\}$ و S تعداد نقاط انتخابی روی هر لینک است، موقعیت x_{sj} هر نقطه روی لینک تعریف می‌شود. برای مثال نقطه میانی (میانه‌ی هر لینک) با انتخاب $\lambda = \frac{1}{2}$ به دست می‌آید.

برای اینکه تضمین شود هیچ نقطه تعریف شده روی لینک داخل مانع p قرار نمی‌گیرد، x_{sj} در رابطه (۸) به جای x معادل‌سازی شده و محدودیت (۱۱) برای هر وجه k از مانع p، هر لینک j و هر نقطه s واقع روی لینک j نوشته می‌شود. به علاوه محدودیت (۱۲) برای هر j و s لازم است. در اینجا b_{sjk} متغیر صفر و یک برای معرفی در مدل MILP به عنوان متغیر تصمیم است:

$$C_k \cdot x_{sj} \geq d_k + (b_{sjk} - \mathbf{1})M \qquad (۱۱)$$

$$\sum_{k=1}^{N} b_{sjk} \geq 1 \qquad (۱۲)$$

تعداد نقاط روی لینک باید طوری انتخاب شود که هندسه (بدنه) هر لینک با دقت مناسبی در مقایسه با شکل مانع نمایش داده شود، یعنی اینکه فاصله بین دو نقطه مجاور روی یک لینک باید کوچک‌تر از اندازه هر مانعی که در مسیر آن واقع می‌شود، انتخاب شود.

حرکت روبات بازویی در فضای با چند مانع

حال به معرفی مدل کلی ریاضی برای حرکت روبات بازویی در فضای کاری شامل چندین مانع می‌پردازیم. این مدل، حرکتی عاری از برخورد با موانع یک روبات بازویی را تولید می‌کند؛ البته به طوری که عمل‌کننده نهایی، مسیری از قبل مشخص شده را بپیماید، حرکت روبات هموار بوده و تغییرات ناگهانی در جابه‌جایی لینک‌ها دیده نشود. مدل ارائه‌شده به صورت برنامه‌ریزی غیرخطی عدد صحیح مختلط (MINLP) به صورت زیر است:

$$\textbf{Minimize} \quad Z = \sqrt{\sum_{d=1}^{2} (X_{Nd} - Xg_d)^2} \qquad (۱۳)$$

Subject to:

$$X_{i1} = X0_1 + \sum_{j=1}^{i} l_j \cos\left(\sum_{u=1}^{j} \theta_u\right), \ \forall i = 1,...,N \qquad (۱۴)$$

$$X_{i2} = X0_2 + \sum_{j=1}^{i} l_j \sin\left(\sum_{u=1}^{j} \theta_u\right), \ \forall i = 1,...,N \qquad (۱۵)$$

$$\sum_{d=1}^{2} a_{kvd} X_{id} \geq d_{kv} + (b_{kvi} - \mathbf{1})M \quad \forall k = 1,...,P;\ \forall v = 1,...,Hp;\ \forall i = 1,...,N \tag{۱۶}$$

$$\sum_{v=1}^{Hp} b_{kvi} \geq 1 \quad \forall k = 1,...,P;\ \ \forall i = 1,...,N \tag{۱۷}$$

$$X_{isd}^{mid} = \lambda_s X_{id} + (1-\lambda_s) X_{i+1,d} \quad \forall i = 1,...,N-1;\ \ \forall s = 1,...,S;\ \ \forall d = 1,2 \tag{۱۸}$$

$$\sum_{d=1}^{2} a_{kvd} X_{isd}^{mid} \geq d_{kv} + (b_{kvis}^{mid} - \mathbf{1})M \quad \forall k = 1,...,P;\ \ \forall v = 1,...,Hp;\ \forall i = 1,...,N-1;\ \ \forall s = 1,...,S \tag{۱۹}$$

$$\sum_{v=1}^{Hp} b_{kvis}^{mid} \geq 1 \quad \forall k = 1,...,P;\ \forall i = 1,...,N-1;\ \forall s = 1,...,S \tag{۲۰}$$

$$\theta_{i-1} - \varepsilon \leq \theta_i \leq \theta_{i-1} + \varepsilon;\ \ \forall i = 2,...,N \tag{۲۱}$$

$$b_{kvi} \in \{0,1\} \quad \forall k = 1,...,P;\ \forall v = 1,...,Hp;\ \ \forall i = 1,...,N \tag{۲۲}$$

$$b_{kvis}^{mid} \in \{0,1\} \quad \forall k = 1,...,P;\ \forall v = 1,...,Hp; \forall i = 1,...,N-1; \forall s = 1,...,S \tag{۲۳}$$

$$\forall\ \theta, X, X^{mid} \in \Re \tag{۲۴}$$

رابطه (۱۳) یک تابع هدف مینیمم‌سازی است که فاصله اقلیدسی بین نقطه فعلی عمل‌کننده نهایی و نقطه هدف مورد نظر را کمینه می‌کند. اندیس d ابعاد مسئله است که در اینجا ۲ است و متغیر X موقعیت یا مختصات را در فضای کاری یعنی همان (x, y) را نشان می‌دهد. منظور از X_{Nd}، موقعیت عمل‌کننده نهایی در فضای کاری و Xg_d موقعیت هدفی است که عمل‌کننده نهایی باید به آن برسد. اندیس i مربوط به لینک‌های روبات بازویی و N تعداد لینک‌های آن است. X_{id} نشان‌دهنده مختصات (x, y) مفصل انتهایی متصل به لینک i است که با توجه به روابط مثلثاتی و زوایای هر لینک (θ_i) به گونه‌ای که توضیح داده شد، به دست می‌آید. با روابط (۱۴) و (۱۵) به ترتیب مختصات (x, y) مربوط به هر مفصل به دست می‌آید که $X0 = (X0_1, X0_2)$ مختصات پایه روبات است.

زوایای هر لینک (θ_iها) به عنوان متغیر تصمیم اصلی مسئله هستند و مقدار آن‌ها در مدل با توجه به محدودیت‌ها به گونه‌ای تعیین می‌شود که نقاط مفصلی و نقاط روی لینک‌های روبات در موانع قرار نگیرند و همچنین تغییرات زاویه‌ای به گونه‌ای باشند که روبات حرکت منظم (همواری) را داشته و تغییرات ناگهانی نداشته باشد.

محدودیت‌های (۱۶) و (۱۷) تضمین می‌کنند که نقاط مفصلی در موانع قرار نگیرند. اندیس k مربوط به موانع و P تعداد موانع موجود در فضای کاری است. اندیس v مربوط به اضلاع هر مانع و Hp تعداد اضلاع مانع p است. a پارامتر مربوط به ضرایب x و y خط گذرنده از هر ضلع مانع و d پارامتر سمت راست مربوط به آن خط است. متغیرهای b متغیرهای صفر و یکی هستند که برای آنکه نقطه‌ای خارج از مانع قرار گیرد، باید در نظر گرفته شوند. M مقدار ثابت بزرگی است که نسبت به دیگر مقادیر داخل مدل، بی‌نهایت در نظر گرفته می‌شود.

محدودیت (۱۸) مختصات نقاط میانی بین دو مفصل متوالی (X^{mid}) را تولید می‌کند تا لینک‌ها نیز با موانع برخورد نداشته باشند. λ پارامتری در بازه (۰, ۱) است و s اندیس اعضای این مجموعه است، به طوری که اگر S عضو داشته باشد، S نقطه روی لینک انتخاب شده و لینک به $S+1$ قسمت، تقسیم می‌شود. قابل ذکر است که هر چه تعداد نقاط میانی بیشتر باشد، دقت مدل از نظر برخورد نکردن با موانع افزایش می‌یابد.

محدودیت‌های (۱۹) و (۲۰) همانند محدودیت‌های (۱۶) و (۱۷) برای اینکه نقاط میانی X^{mid} داخل مانع واقع نشوند، به کار گرفته می‌شوند. b^{mid} نیز مانند b متغیر صفر و یک است. محدودیت (۲۱) مانند محدودیت (۶) برای آن در نظر گرفته شده است که روبات حرکتی منظم و بدون تغییرات ناگهانی داشته باشد. محدودیت‌های (۲۲) و (۲۳) مربوط به صفر و یک بودن متغیرهای b و b^{mid} هستند و محدودیت (۲۴) نشان می‌دهد که متغیرهای θ، X و X^{mid} اعداد حقیقی آزاد در علامت هستند.

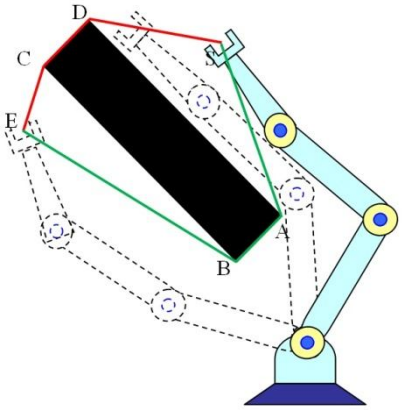
حل و اجرای مدل

در این بخش به نحوه پیاده‌سازی و اجرای مدل پیشنهادی و نمایش نتایج آن به کمک اشکال مختلف پرداخته می‌شود. این مهم با استفاده از ارتباط دادن نرم‌افزار LINGO به عنوان نرم‌افزار بهینه‌سازی مسائل برنامه‌ریزی ریاضی و نرم‌افزار MATLAB به عنوان آماده‌ساز پارامترهای مسئله، مسیریاب و نمایشگر حرکت روبات محقق می‌شود.

یافتن کوتاه‌ترین مسیر عمل‌کننده نهایی

با حل مدل برنامه‌ریزی حرکت ارائه‌شده، روبات روی مسیری که از قبل انتخاب شده است، طوری حرکت می‌کند که کمترین فاصله را از زیراهداف داشته و در موانع قرار نگیرد. در این مقاله برای یافتن کوتاه‌ترین مسیر از گراف دیدنگار[۱۳] استفاده شده است که با اصلاحاتی که در آن انجام می‌گیرد، برای حرکت عمل‌کننده نهایی مناسب می‌شود. گراف دیدنگار، گرافی است که گره‌های آن شامل همه رئوس موانع به اضافه دو نقطه شروع و هدف بوده و یال‌های آن همه خطوط راستی هستند که دو رأس قابل رؤیت از یکدیگر را به هم متصل می‌کنند. سپس کوتاه‌ترین فاصله میان نقاط شروع و هدف روی این گراف از طریق روش‌های جستجو مانند دایکسترا محاسبه می‌شود. قابل ذکر است که روش‌های دیگری مانند نمودار ورونویی[۱۴] -که مسیری است با بیشترین فاصله از موانع نیز برای مسیریابی بین دو نقطه در میان موانع وجود دارند، اما چون گراف دیدنگار، کوتاه‌ترین مسیر را در بین همه روش‌ها تولید می‌کند، در اینجا استفاده شده است.

به کمک گراف دیدنگار، استاندارد مسیر اولیه‌ای بین دو نقطه شروع و پایان به دست می‌آید، اما برای کاربردی بودن آن نیاز است که تغییراتی در آن اعمال شود. در شکل ، هدف، حرکت بازوی نهایی روبات از نقطه شروع S به نقطه پایان E است. کوتاه‌ترین مسیر بین دو نقطه S و E، مسیر SDCE است که بر اساس روش دیدنگار محاسبه می‌شود. با این حال، مسیر SDCE با وجود کوتاه‌ترین بودن، مسیر ممکنی برای روبات بازویی نیست، چرا که رأس D در این مسیر، نقطه‌ای غیر قابل دسترس برای روبات است. بنابراین مسیر SDCE حذف می‌شود و دومین مسیر کوتاه که مسیر SABE است، مورد بررسی قرار می‌گیرد. پس در اینجا الگوریتم دیدنگار، علاوه بر یافتن کوتاه‌ترین مسیر، باید قادر باشد مسیر ممکنی را برای روبات بازویی بیابد. برای این منظور در ابتدا همه رئوسی که برای روبات غیرقابل دسترس هستند، از مجموعه رئوسِ واقع در گراف دیدنگار حذف شده و سپس گراف دیدنگار بار دیگر برای یافتن کوتاه‌ترین مسیر بعدی جستجو می‌شود.



شکل ۵: گراف دیدنگار تعمیم یافته

تقسیم مسیر به زیر اهداف

کوتاه‌ترین مسیر شدنی برای روبات به قسمت‌های مساوی، به عنوان زیرهدف‌ها تقسیم می‌شود که اندازه هر قسمت از طریق پارامتری توسط کاربر تعیین می‌شود. سپس کل مسیر به فواصل مورد نظر تقسیم شده و تعداد تقسیمات را به نزدیک‌ترین عدد صحیح گِرد می‌کند. مختصات همه نقاط به دست آمده، ذخیره شده و هر یک به عنوان یک زیرهدف در نظر گرفته می‌شود.

ساخت مدل ریاضی

پس از تعیین زیر هدف‌ها باید مدل ریاضی به فرم قابل فهم برای نرم‌افزار LINGO ساخته شود. برای این کار، ابتدا اطلاعات مربوط به تعداد موانع و موقعیت آن‌ها اعم از نقاط رئوسی و شیب خطوط سازنده اضلاع در فایل متنی ذخیره می‌شود. سپس همچنین اطلاعات مربوط به تعداد لینک‌ها، نقطه پایه روبات، نقطه زیرهدف، طول لینک‌ها، محدوه حرکتی مجاز برای هر لینک (به کمک تعیین پارامتر ε) و زوایه‌ای که هر لینک نسبت به لینک قبلی خود می‌سازد، توسط فایل‌های متنی به نرم‌افزار LINGO منتقل می‌شود.

برای آنکه یک نقطه خارج از مانع قرار گیرد، باید اضلاع موانع نیز برای نرم‌افزار LINGO معرفی شوند.

بدین منظور ابتدا مرکز ثقل هر مانع در MATLAB محاسبه شده و با توجه به آن نقطه، برای هر ضلع از هر مانع به طور جداگانه تعیین علامت صورت می‌گیرد. سپس با توجه به شیب خط هر ضلع و علامت مربوط به آن، محدودیت‌های متناسب با آن، به صورت خودکار در نرم‌افزار LINGO تولید می‌شوند.

حل مدل

با مشخص‌شدن پارامترهای مدل، مختصات نقاط زیرهدف‌ها به صورت متوالی توسط MATLAB به LINGO ارسال می‌شود که آن نیز با حل بهینه مدل، زاویه هر لینک روبات را با توجه به تابع هدفِ مینیمم‌کردنِ فاصله عمل‌کننده نهایی از زیرهدف، به گونه‌ای تعیین می‌کند که هیچ لینکی داخل مانع قرار نگیرد. اطلاعات خروجی LINGO به MATLAB ارسال و ذخیره می‌شود. این عملیات به طور متوالی برای زیرهدف‌های دیگر تا آخرین آنها که نقطه هدف نهایی است، ادامه می‌یابد.

پس از اتمام اجراهای نرم‌افزار LINGO با توجه به خروجی‌های ذخیره‌شده در نرم‌افزار MATLAB نحوه حرکت روبات در بین موانع برای رسیدن به هدف به صورت گرافیکی قابل نمایش است. به عنوان مثال، در شکل ۶ مراحل حل مسئله یک روبات دو لینکی با یک مانع چهار ضلعی و در شکل ۷ مراحل حل یک روبات سه لینکی با دو مانع نشان داده شده است.

افزایش ایمنی مسیر

از آنجایی که برای تضمین برخورد نکردن لینک‌ها با موانع، تعداد محدودی از نقاط میانی روی هر لینک تعریف می‌شود، ممکن است گاهی حد فاصل بین دو نقطه میانی متوالی روی لینک، وارد گوشه‌های موانع شود. برای حل این مشکل، می‌توان موانع را با لحاظ کردن یک حاشیه ایمنی با اندازه مشخص و دلخواه بزرگ‌تر در نظر گرفت، بدین صورت که زمانی که گراف دیدنگار برای پیدا کردن کوتاه‌ترین مسیر، مورد استفاده قرار می‌گیرد، برای بررسی کردن رأس‌ها از مانع بزرگ‌تر شده استفاده کند، ولی برای اینکه فضای حل مسئله محدود نشود، هنگامی که پارامترهای مانع به برنامه LINGO داده می‌شود، از مانع اصلی (در اندازه واقعی) استفاده شود. شکل ، حل مسئله را با بزرگ کردن مانع نمایش می‌دهد.

کاهش تعداد متغیرها و محدودیت‌ها

همان طور که از مدل ریاضی مسئله، قابل تشخیص است، با افزایش تعداد موانع یا لینک‌ها تعداد متغیرهای صفر و یک به سرعت افزایش می‌یابند. برای حل این مشکل و کاهش متغیرهای صفر و یک، بهبودی در روش پیشنهاد می‌شود: ابتدا کوتاه‌ترین فاصله نقطه پایه روبات تا نزدیک‌ترین مانع محاسبه شده (به نام D_{min}) و با حداکثر فاصله‌ای که هر لینک $i-$ ام از نقطه پایه روبات می‌تواند داشته باشد(که برابر است با $R_i = l_1 + l_2 + \ldots + l_i$)، مقایسه می‌شود. پس از محاسبه R_i-ها برای همه لینک‌ها، بزرگ‌ترین شماره لینکی را انتخاب می‌کنیم که R آن کوچک‌تر از D_{min} باشد. به عبارت دیگر:

$$t = \left\{ i \mid R_i \le D_{min} \le R_{i+1}, i = 1,\ldots,N \right\} \qquad (۲۵)$$

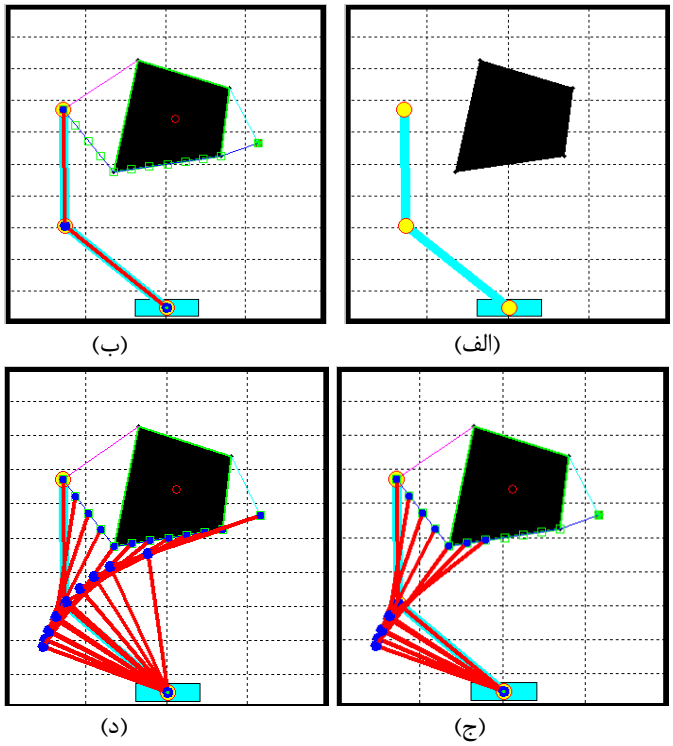
در این صورت، t شماره لینکی است که بازوی روبات تا انتهای آن در هر موقعیتی هم که باشد، به هیچ مانعی برخورد نخواهد کرد و بنابراین نیازی به در نظر گرفتن نقاط میانی از اولین لینک تا لینک t-ام برای چک کردن برخورد آنها با موانع طبق روابط (۱۸)-(۲۰) نیست. با استفاده از این استدلال، تعداد متغیرهای صفر و یک و محدودیت‌های مدل به طرز قابل توجهی کاهش می‌یابد.

نتایج محاسباتی

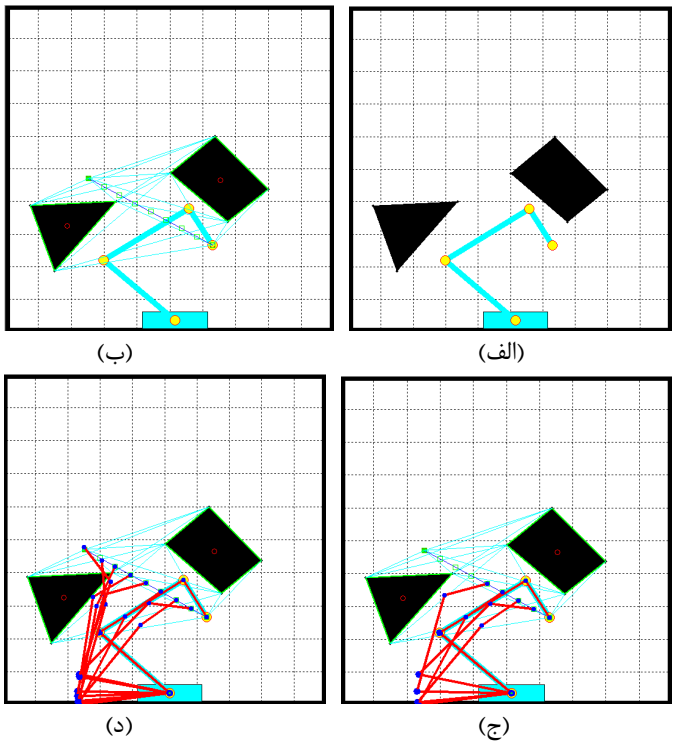
با توجه با اینکه مسیر عمل‌کننده نهایی از طریق گراف دیدنگار تعمیم‌یافته حاصل می‌شود و مدل مسئله با رویکرد برنامه‌ریزی ریاضی و به صورت دقیق (و نه هیوریستیک) حل می‌شود، مسیر حرکت به دست آمده از این دو نظر، بهینه است و در مقایسه با روش‌های ابتکاری توسعه داده شده در ادبیات، حد پائین را ارائه می‌کند. با این حال سؤالی که می‌تواند مطرح شود، این است که میزان بهبود مدل از نظر کاهش تعداد متغیرها و محدودیت‌ها و زمان حل چقدر است؟ در ادامه به مقایسه دو روش اولیه و بهبود یافته از نظر تعداد متغیرهای صفر و یک، تعداد محدودیت‌ها و زمان حل می‌پردازیم.

برای مقایسه، دو مسئله متفاوت با تعداد لینک‌ها و تعداد موانع مختلف هر یک با هر دو روش اولیه و بهبودیافته دو تکرار حل شده و زمان حل به صورت

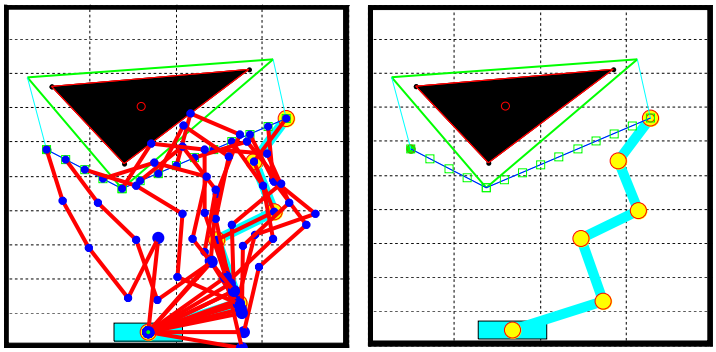
میانگین، زمان حل دو تکرار به ثانیه گزارش شده است. در این مثال‌ها، پارامتر ε برای تعیین محدوده حرکتی هر لینک $\pi/10$ در نظر گرفته شده است. همچنین برای افرایش ایمنی مسیر، از روش بزرگ‌کردن موانع استفاده شده است.



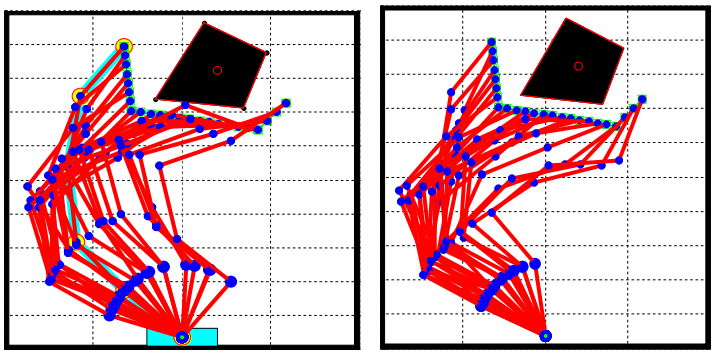
شکل ۶: مسیریابی روبات بازویی ۲ لینکی با یک مانع: (الف) پیکربندی اولیه، (ب) یافتن کوتاه‌ترین مسیر به کمک گراف دیدنگار، (ج) حرکت به سمت هدف، (د) پیکربندی نهایی و رسیدن به هدف



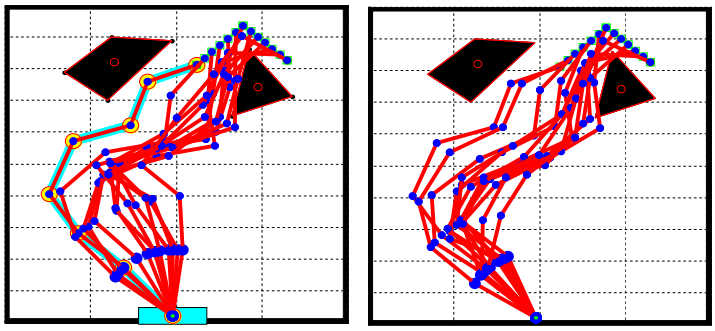
شکل ۷: مسیریابی روبات بازویی ۳ لینکی با دو مانع (توضیحات مشابه شکل ۶)



شکل ۸: حل مسئله با بزرگ کردن موانع



شکل ۹: حل مسئله با روش اولیه (بالا)، و بهبود یافته (پایین)



شکل ۱۰: حل مسئله با روش اولیه (بالا) و بهبود یافته (پایین)

مثال اول، فضایی است با یک روبات بازویی ۵ لینکی که یک مانع چهار ضلعی در جلوی آن به گونه‌ای قرار دارد که از لینک چهارم به بعد با مانع برخورد می‌کند (یعنی در $t = 3$ (۲۵)). کل مسیر به ۲۳ قسمت (زیرهدف) تقسیم شده است، بنابراین برای رسیدن به نقطه هدف، برنامه LINGO باید ۲۳ بار اجرا شود. شکل ، نحوه حرکت روبات را برای رسیدن به هدف مورد نظر در یکی از تکرارهای حل شده در دو روش اولیه و بهبودیافته نشان می‌دهد.

جدول (۱)، مقایسه دو روش را نشان می‌دهد که در آن روش بهبود داده شده، تعداد متغیرهای صفر و یک و تعداد محدودیت‌های کمتر و در نتیجه زمان حل کوتاه‌تری دارد. قابل ذکر است که مقدار تابع هدف صفر به معنی فاصله تا نقطه نهایی است که در صورت رسیدن به آن، مقدار آن صفر خواهد بود.

مثال دوم، فضایی با یک روبات بازویی ۷ لینکی و دو مانع سه و چهار ضلعی است. این موانع به گونه‌ای در صفحه قرار گرفته‌اند که در بدترین شرایط روبات بازویی از لینک چهارم به بعد با مانع برخورد می‌کند ($t = 3$). برای طی کردن طول مسیر عمل‌کننده نهایی به ۱۲ قسمت تقسیم شده است. به عبارت دیگر برای رسیدن به نقطه هدف، مدل ریاضی باید ۱۲ بار اجرا شود. شکل ۱۰ نحوه

حرکت روبات را برای رسیدن به هدف مورد نظر در یکی از تکرارهای حل شده در دو روش اولیه و بهبود یافته نشان می‌دهد.

جدول ۱: مقایسه دو روش اولیه و پیشنهادی مربوط به مثال اول

	روش اولیه	روش بهبود یافته
تعداد متغیرهای صفر و یک	۶۸	۴۴
تعداد محدودیت‌ها	۱۲۱	۷۹
متوسط زمان حل مسئله	۲۱۲/۵	۱۳۳/۵
متوسط زمان کل اجرای LINGO	۱۹۹	۱۲۰/۵
متوسط زمان اجرای هر تکرار LINGO	۸/۶۵	۵/۲۴
متوسط مقدار تابع هدف	۰	۰

جدول (۲)، مقایسه دو روش را نشان می‌دهد. همان طور که انتظار می‌رود، روش بهبود داده شده، تعداد متغیرهای صفر و یک و همچنین تعداد محدودیت‌های کمتری دارد و زمان حل مسئله نیز بهبود یافته است.

برای کاهش هر چه بیشتر زمان محاسباتی، تلاش شد تا با افزایش فاصله بین زیرهدف‌ها روی مسیر به دست آمده از گراف دیدنگار (که به صورت پیش‌فرض ۰/۵ واحد طول در نظر گرفته شده بود) تعداد تکرارهای مسئله و در نتیجه تعداد حل‌های مدل ریاضی با Lingo کاهش یافته و زمان حل مسئله به طور کامل کمتر شود.

جدول ۲: مقایسه دو روش اولیه و پیشنهادی مربوط به مثال دوم

	روش اولیه	روش بهبود یافته
تعداد متغیرهای صفر و یک	۱۹۶	۱۳۳
تعداد محدودیت‌ها	۳۰۴	۲۱۱
متوسط زمان حل مسئله	۲۷۳۸/۵	۱۸۹۹
متوسط زمان کل اجرای LINGO	۲۷۳۱	۱۸۹۱
متوسط زمان اجرای هر تکرار LINGO	۲۲۷/۶	۱۵۷/۶
متوسط مقدار تابع هدف	۰	۰

بنابراین دو مقدار ۱/۰ و ۱/۲۵ برای فواصل زیرهدف‌ها در نظر گرفته شده و دو مسئله نمونه دیگر برای روبات ۵-لینکی با آن مقادیر، بار دیگر حل شد. مشاهده شد که بدون اینکه کیفیت جواب کاهش چندانی کند، زمان‌های پردازش به ازای دو مقدار ۱/۰ و ۱/۲۵ به ترتیب ۳۶٪ و ۷۲٪ برای مسئله اول در شکل (۱۱) (الف)-(ج) و ۵۰٪ و ۶۸٪ برای مسئله دوم در شکل (۱۱) (د)-(و) کاهش یافتند.

نتیجه‌گیری

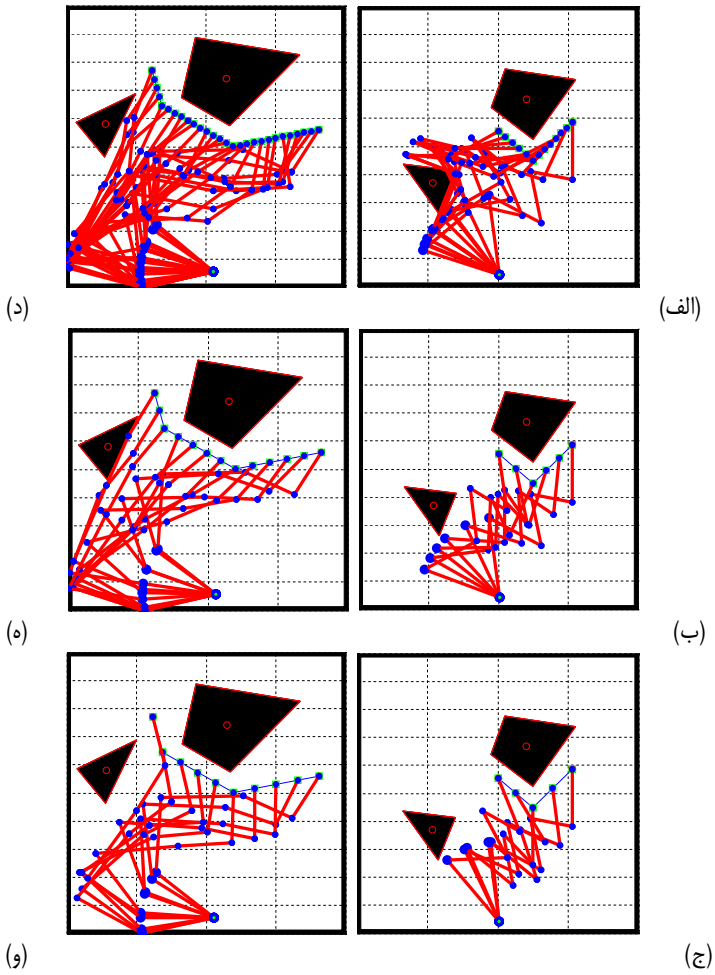
در این مقاله، روشی جدید مبتنی بر بهینه‌سازی ریاضی برای برنامه‌ریزی حرکت روبات بازویی در محیط با مانع ارائه شده است، به طوری که هنگام حرکت بازوی روبات از نقطه اولیه‌اش به یک نقطه هدف مشخص، هیچ لینکی با موانع برخورد نداشته باشد و عمل‌کننده نهایی در نقطه هدف مورد نظر قرار گیرد. برای این منظور، پس از محاسبه کوتاه‌ترین مسیر شدنی از نقطه آغازین عمل‌کننده نهایی تا نقطه هدف، به کمک گراف دیدنگار تعمیم‌یافته و تقسیم آن به چند زیرهدف محلی، یک مدل برنامه‌ریزی غیرخطی عدد صحیح مختلط (MINLP) توسعه داده شد که تابع هدف آن، کمینه‌سازی فاصله عمل‌کننده نهایی از یک زیرهدف به زیرهدف دیگر بوده و محدودیت‌های آن، نبود مفاصل و نقاط میانی لینک‌ها در داخل موانع است.

برای بهبود مدل، روشی پیشنهاد شد که متغیرهای میانی لینک‌هایی که امکان برخورد آنها با نزدیک‌ترین مانع وجود ندارد، از مدل حذف شده و به این ترتیب از تعداد متغیرهای صفر و یک و نیز محدودیت‌ها کاسته شد و در نتیجه، زمان حل، کاهش یافت.

مزیت روش ارائه‌شده، قابلیت تعمیم آن به هر نوع فضای کاری و روبات بازویی با هر تعداد درجه آزادی قابل حل است. همچنین هر مسئله دلخواهی با تعداد لینک‌ها و موانع در حالت‌های مختلف را می‌توان با این روش حل کرد و نیازی به مدلسازی و استخراج محدودیت‌ها به صورت دستی برای آن نیست و نرم‌افزار نوشته شده به طور اتوماتیک، مدل ریاضی را می‌سازد. برای موانع نامحدب باید آنها را به نواحی محدب تقسیم کرده و هر ناحیه را جداگانه وارد نرم‌افزار کرد. همچنین امکان در نظر گرفتن چند روبات نیز وجود دارد که در این صورت، هر یک از روبات‌ها برای دیگری حکم مانع خواهد بود و باید در مدل مربوط به آن وارد شود. از نوآوری‌های دیگر مقاله، می‌توان به ترکیب و ادغام دو رویکرد مجزا، یعنی روش گراف دیدنگار (که در اصل در هندسه محاسباتی مطرح می‌شود) و روش برنامه‌ریزی غیرخطی عدد صحیح مختلط (که در بهینه‌سازی ریاضی مطرح می شود) نام برد. همچنین،

کاهش تعداد متغیرهای صفر و یک به کمک روش بهبودیافته، از نوآوری‌های فرعی کار محسوب می‌شود.

در تحقیقات بعدی، مدل به فضای سه بعدی تعمیم داده خواهد شد. همچنین می‌توان با وارد کردن اطلاعات حسگرها، وضعیت موانع متحرک در محیط را در هر لحظه شناسایی و محدودیت‌های مربوطه را فرمول‌بندی کرد.



شکل ۱۱: (الف) فواصل زیرهدف‌ها ۰/۵ با زمان ۱۸۵/۴ (ب) فواصل زیرهدف‌ها ۱/۰ با زمان ۱۱۸/۵ (ج) فواصل زیرهدف‌ها ۱/۲۵ با زمان ۵۱/۴ (د) فواصل زیرهدف‌ها ۰/۵ با زمان ۵۳۹/۸ (ه) فواصل زیرهدف‌ها ۱/۰ با زمان ۲۷۰/۷ (و) فواصل زیرهدف‌ها ۱/۲۵ با زمان ۱۷۲/۸ ثانیه.

مراجع

1- Canny, J. F. (1988). *The Complexity of Robot Motion Planning.* The MIT press, Cambridge, Mass.

2- Hwang, Y. K. and Ahuja, N. (1992). "Gross Motion Planning—A Survey." *ACM Computing Surveys*, Vol. 24, No. 3, PP. 219-291.

3- Shin, K. G. and Mckay, N. D. (1986). "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators." *IEEE Transactions on Automatic Control,* Vol. 31, No. 6, PP. 491-500.

4- Chettibi, T., Lehtihet, H. E., Haddad, M. and Hanchi, S. (2004). "Minimum cost trajectory planning for industrial robots." *European Journal of Mechanics A/Solids,* Vol. 23, PP. 703–715.

5- Blackmore, L. and Williams, B. (2006). "Optimal Manipulator Path Planning with Obstacles using Disjunctive Programming." *American Control Conference,* Minneapolis, Minnesota, USA, PP. 3200-3202.

6- Guangyu, L., Zengqi, S. and Chundi, M. (2006). "Optimal Motion Planning Passing through Kinematic

Singularities for Robot Arms." *International Conference on Intelligent Robots and Systems,* Beijing, China, PP. 4349–4354.

7- Zhang, Y., Li, Z. and Tan, H. Z. (2006). "Inequality-based Manipulator-Obstacle Avoidance Using the LVI-based Primal-dual Neural Network." *International Conference on Robotics and Biomimetics.* Kunming, China, PP. 1459–1464.

8- Choi, J. and Amir, E. (2007). "Factor-Guided Motion Planning for a Robot Arm." *IEEE/RSJ International Conference on Intelligent Robots and Systems,* San Diego, CA, USA, PP. 27–32.

9- Zhang, Y., Lv, X., Li, Z., Yang, Z. and Chen, K. (2008). "Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal–dual neural network." *Mechatronics,* Vol. 18, No. 9, PP. 475–485.

10- Ding, H., Zhou, M. and Stursberg, O. (2009). "Optimal Path Planning in the Workspace for Articulated Robots using Mixed Integer Programming." *IEEE/RSJ International Conference on Intelligent Robots and Systems,* Louis, USA, PP. 5770–5775.

11- Ding, H., Zhou, M. and Stursberg, O. (2009). "Optimal Motion Planning for Robotic Manipulators with Dynamic Obstacles using Mixed-Integer Linear Programming." *Mediterranean Conference on Control & Automation,* Makedonia Palace, Thessaloniki, Greece, p. 934-939.

12- Ding, H., Reissig, G., Gross, D. and Stursberg, O. (2011). "Mixed-Integer Programming for Optimal Path Planning of Robotic Manipulators." *IEEE Internationl Conference on Automation Science and Engineering,* Trieste, Italy, PP. 133-138.

واژه‌های انگلیسی به ترتیب استفاده در متن

1- End-effector
2- Roadmap
3- Cell Decomposition
4- Potential Field
5- Disjunctive Programming
6- Singularities
7- Redundancy
8- Quadratic Programming
9- Linear Variational Inequalities
10- Model Predictive Control
11- Mixed Integer Non-Linear Programming
12- Revolute
13- Visibility Graph
14- Voronoi Diagram