



طراحی مسیر بازوی مارگون کابلی صفحه‌ای با مفاصل قفل شونده با استفاده از الگوریتم بهینه‌سازی پرواز پرندگان

علی طاهری فر^۱، حسن سالاریه^{۲*}، آریا الستی^۳، مهرداد بروشکی^۴

- ۱- کارشناسی ارشد، دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف، تهران
 - ۲- استادیار، دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف، تهران
 - ۳- استاد، دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف، تهران
 - ۴- دانشیار، دانشکده مهندسی مکانیک، دانشگاه صنعتی شریف، تهران
- *تهران، صندوق پستی ۱۱۳۶۵۱۱۱۵۵. salarieh@sharif.ir

چکیده- در این مقاله طراحی مسیر بازوی افزونه صفحه‌ای با مفاصل قفل شونده با استفاده از الگوریتم پرواز پرندگان انجام شده است. برای این منظور دو استراتژی متفاوت به کار رفته است. در استراتژی اول، طراحی مسیر به دو بخش تقسیم شده که در بخش اول سینماتیک معکوس بازو با استفاده از الگوریتم پرواز پرندگان پیوسته حل شده و در بخش دوم ترتیب باز و بسته شدن قفل مفاصل با استفاده از الگوریتم پرواز پرندگان گسسته بهینه شده است. نتایج حاصل از بهینه‌سازی توسط این روش، با نتایج حاصل از بهینه‌سازی الگوریتم ژنتیک مقایسه شده است. در یافتن پاسخ سینماتیک معکوس با یک مسئله بهینه‌سازی چند هدفه روبرو شده که برای حل آن از دو روش ترکیب وزنی توابع و روش پرواز پرندگان برداری استفاده شده است. در استراتژی دوم، الگوریتم جدید پرواز پرندگان هیبریدی معرفی شده است. در این الگوریتم اعداد پیوسته و گسسته در یک ذره، کنار هم قرار گرفته تا یک ذره هیبریدی را شکل دهند. در پایان نتایج دو استراتژی با یکدیگر مقایسه شده تا برتری روش پرواز پرندگان هیبریدی مشخص شود.

کلیدواژه‌ها: طراحی مسیر، بهینه‌سازی، بازوهای افزونه، الگوریتم پرواز پرندگان

Path Planning for a Planar Hyper-Redundant Manipulator with Lockable Joints Using Particle Swarm Optimization

Ali Taherifar¹, Aria Alasty², Hassan Salarieh^{3*}, Mehrdad Boroushaki⁴

1- M. Sc., Mechanical Engineering Department, Sharif Univ. of Tech.

2- Prof., Mechanical Engineering Department, Sharif Univ. of Tech.

3- Assist. Prof., Mechanical Engineering Department, Sharif Univ. of Tech.

4- Assoc. Prof., Energy Engineering Department, Sharif Univ. of Tech.

* P.O.B. 1136511155, Tehran, Iran. salarieh@sharif.ir

Abstract- In this paper, the problem of path planning for a special hyper-redundant manipulator with lockable joints is solved using particle swarm optimization. Two strategies have been used. In the first strategy, the path-planning problem is solved in two stages; first, the inverse kinematics of manipulator is solved with continuous PSO and then the sequence of switching is optimized by modified discrete PSO. The results of discrete PSO are compared to discrete GA to show the effectiveness of the discrete PSO. In order to implement multi-objective optimization in inverse kinematics calculation, linear fitness function and Vector Evaluated PSO (VEPSO) are used. In the second strategy, a novel approach in particle swarm optimization is proposed. In this approach, the integer and continuous value numbers are joined together in a single particle to form a hybrid particle. The results of two strategies were compared to show the convergence speed and performance of the second strategy which was proposed here.

Key words: Path planning, Optimization, Redundant manipulators, Particle swarm optimization.

۱- مقدمه

برای ایجاد قابلیت مانوردهی بیشتر و همچنین کارکرد مناسب ربات‌ها در محیط‌های کاملاً مقید، پیچیده و خطرناک نظیر کار در راکتورهای هسته‌ای، اعمال جراحی، کاوشها و کار در ایستگاه‌های فضایی، عملیات اکتشافی در اعماق دریا، امدادسانی در مکانهای زلزله‌زده برای نجات جان مجروحان و فعالیت‌هایی از این قبیل، معمولاً از ربات‌های با درجات آزادی افزونه^۱ استفاده می‌شود [۱]. بازوهای مارگون، بازوهایی با درجات آزادی افزونه هستند که تعداد درجات آزادی آنها از حداقل مورد نیاز برای دسترسی به تمام نقاط فضا بیشتر است. با توجه به شکل ظاهری بازوهای مارگون، از اصطلاحاتی همچون خرطوم فیل^۲ و بازوی هشت پا^۳، توسط محققین استفاده شده است.

با توجه به درجات آزادی بالای بازوهای افزونه و وجود پاسخ‌های زیاد برای سینماتیک معکوس این بازوها [۲]، مسئله طراحی مسیر این گونه بازوها توجه بسیاری از متخصصین را به خود جلب کرده است.

بطور کلی طراحی مسیر بازوها را می‌توان به دو بخش عمده تقسیم نمود. بخش اول، طراحی مسیر به صورت هندسی و بخش دوم، طراحی مسیر وابسته به زمان می‌باشد. منظور از طراحی مسیر وابسته به زمان یافتن نحوه حرکت مفاصل ربات بگونه‌ای است که مجری نهایی ربات بر روی یک مسیر از پیش تعیین شده و یا نقاطی از یک مسیر حرکت کند. در این گونه مسائل، هدف مشخص کردن حرکت مفاصل بر حسب زمان می‌باشد که به این گونه مسائل طراحی تراژکتوری می‌گویند. در این دسته تقسیم بندی جزئی تری را نیز می‌توان انجام داد که شامل رویکرد سینماتیک و رویکرد دینامیکی می‌باشد. با هر کدام از این رویکردها می‌توان توابع هزینه متفاوتی را جهت یافتن مسیر بهینه بکار برد. به عنوان مثال تابع هزینه می‌تواند کمینه کردن مصرف انرژی، کمینه کردن زمان حرکت،

جلوگیری از برخورد با موانع، حرکت یکنواخت بدون ناپیوستگی در سرعت و شتاب و یا حرکت با مینیمم جرک باشد.

در مقاله ای که توسط ساهین کنکوور [۳] در سال ۲۰۰۳ ارائه شد روشی برای تعقیب مسیر برای بازوهای ابر افزونه ارائه شده است. مسیر تعریف شده متشکل از منحنی‌های اسپیرالین به هم متصل شده می‌باشد. بازوی مورد نظر بگونه‌ای برنامه ریزی شده است که لینک‌های همواره مماس بر مسیر حرکت باقی بماند آتا و همکارانش [۴] مسئله ردیابی مسیر بازوهای با درجه آزادی اضافی با استفاده از روش جستجوی الگوی تعمیم یافته مورد بررسی قرار دادند. هدف از حل این مسئله رسیدن به کمترین خطا بین نقاط مسیر مطلوب و نقاط مسیر واقعی می‌باشد. در تعریف تابع هدف، کمینه کردن حرکت مفاصل نیز در نظر گرفته شده است. جیا و چو [۵] به حل مسئله طراحی مسیر برای یک ربات ۹ درجه آزادی، جهت ردیابی مسیرهای مستقیم و مسیرهای دایروی پرداختند و سای و هانگ [۶] روشی برای کنترل بهینه ربات‌های چند درجه آزادی با استفاده از شبکه عصبی ارائه دادند.

در مقالاتی که تاکنون اشاره شد، تمرکز پژوهشگران در مسئله طراحی مسیر، بیشتر بر روی حرکت بدون ناپیوستگی در سرعت و شتاب بوده است اما لیانگ و همکارانش [۷] در سال ۱۹۹۹ طراحی مسیر ربات‌های افزونه، برای عدم برخورد با موانع را انجام دادند و برای این کار از ماتریس شبه معکوس استفاده کردند. آنها با استفاده از فضای پوچ ماتریس شبه معکوس محدوده حرکتی مفاصل و یا عدم برخورد با موانع را در تحقیقات خود لحاظ کردند. سایویکو [۸] روش حلی برای سینماتیک معکوس ربات افزونه مقید ارائه داده است. روابط نهایی بیان شده در آن مقاله، به روشی حلقه بسته و پایدار برای کنترل حرکت ربات منجر شده است. با گسترش سیستماتیک روابط سینماتیک مستقیم، قیود مربوط به عدم برخورد با موانع و عدم رسیدن به اکستریم محدوده حرکت لحاظ شده‌اند.

در میان توابع هزینه مختلفی که برای بازوهای افزونه و غیرافزونه تعریف می‌شود، یافتن مسیر بهینه زمانی، از اهمیت ویژه‌ای برخوردار است. چرا که در بسیاری از ربات‌های صنعتی جهت افزایش بهره وری، باید وظایف مختلفی در کمترین زمان

1. Redundant manipulator
2. Elephant trunk
3. Tentacle manipulator

در سال ۲۰۰۹، هنرور ایده جدیدی را در طراحی بازوهای افزونه کابلی مطرح کرد [۱۵]. نوآوری طرح پیشنهادی وی در استفاده از سیستم قفل در مفاصل بود که به کمک آن، فرد قادر می‌شود تنها با استفاده از سه کابل و سه عملگر، کل بازو را کنترل کند. استفاده از مکانیزم قفل باعث کاهش تعداد عملگرها و سبک‌تر شدن بازو و در نتیجه کاهش هزینه‌ها می‌شود. یک بازوی ۸ درجه آزادی با این شیوه در آزمایشگاه کنترل دانشگاه صنعتی شریف ساخته شده است. اما با اینکه ایده مطرح شده، ساختار بازو را ساده‌تر کرده است اما باعث پیچیدگی طراحی مسیر و کنترل آن شده است. در این مقاله هدف، بررسی طراحی مسیر برای یک مدل ساده شده از بازوی مارگون کابلی با مفاصل قفل شونده معرفی شده در [۱۵] می‌باشد (شکل ۱). در این مدل ساده شده، کلیه لینک‌های بازو، صفحه‌ای فرض می‌شوند، زیرا هدف، نشان دادن صحت ایده و نحوه به‌کارگیری روش ارائه شده است. در طراحی مسیر این بازو، هم با متغیرهای پیوسته و هم با متغیرهای گسسته سر و کار داریم و تا کنون هیچ تحقیقی در خصوص طراحی مسیر این نوع بازوها صورت نگرفته است. در این مقاله سعی شده است تا با استفاده از یک الگوریتم بهینه‌سازی قوی، نحوه تغییر طول کابل‌ها و نحوه باز و بسته شدن قفل‌های مفاصل بگونه‌ای بهینه شوند که مجری نهایی بازو بر روی یک خط راست حرکت کند. جنبه دیگر نوآوری این پژوهش در استفاده از الگوریتم پرواز پرندگان^۲ می‌باشد، چرا که در تحقیقاتی که تاکنون صورت گرفته، غالباً از الگوریتم‌های ژنتیک در طراحی مسیر استفاده شده است.

ادامه این مقاله به شش بخش تقسیم می‌شود. در بخش دوم مفاهیم اولیه راجع به بازوی با مفاصل قفل شونده و مدل‌سازی مسئله ارائه خواهد شد. در بخش سوم توضیح مختصری راجع به روش بهینه‌سازی پرواز پرندگان ارائه خواهد شد و در بخش چهارم طراحی مسیر بازو با دو استراتژی متفاوت صورت می‌گیرد. در استراتژی اول، ابتدا سینماتیک معکوس بازو توسط الگوریتم پرواز پرندگان با متغیرهای پیوسته به‌دست آمده و

توسط ربات صورت پذیرد. در مقالات جدیدتر که در این حوزه مطرح می‌شوند هدف مسئله، پیدا کردن ترتیب بهینه انجام وظایف می‌باشد. زاچاریا و همکارانش [۹] در سال ۲۰۰۵ با برقراری آنالوژی بین مسئله فروشنده دوره‌گرد و مسئله طراحی مسیر، روشی بر پایه الگوریتم ژنتیک ارائه دادند.

زیدپاس و همکارانش [۱۰] در سال ۲۰۱۰ روشی برای پیدا کردن جواب مسئله برنامه ریزی وظایف^۱ در حضور موانع و در کمترین زمان پیشنهاد دادند.

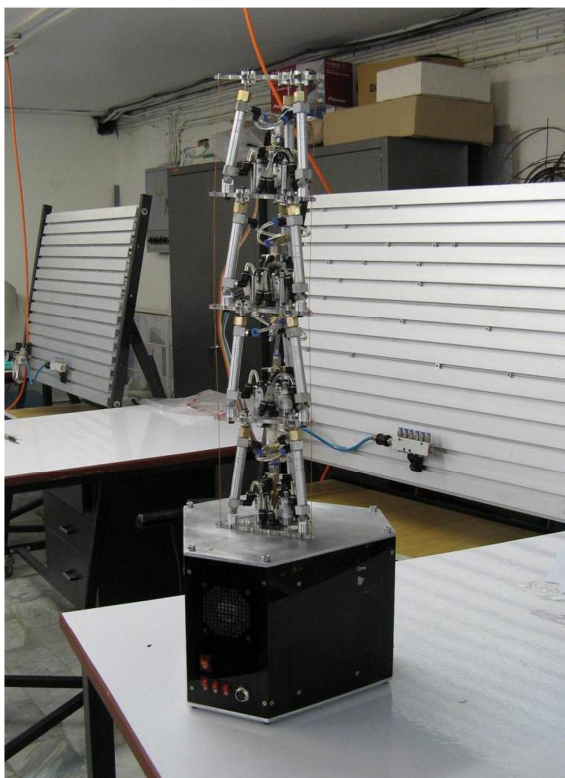
بطور کلی اضافه شدن معادلات دینامیکی در مسئله طراحی مسیر باعث پیچیده تر شدن معادلات و زمانبر شدن حل آنها می‌شود. بنابراین اکثر مقالاتی که با رویکرد دینامیکی به بررسی طراحی مسیر ربات‌ها می‌پردازند، از روش‌های هوشمند برای مدل‌سازی و یا پیش‌بینی مسیر استفاده می‌کنند. ماریا مارکوس [۱۱] با استفاده از الگوریتم ژنتیک روشی جدید برای طراحی مسیر ربات‌های افزونه ارائه داده است. نقطه قوت این تحقیق در بررسی پنج تابع هدف مختلف و ارزیابی دقت روش ارائه شده در هر یک از حالات می‌باشد. ایده اصلی این تئوری، بر پایه استفاده توام از الگوریتم ژنتیک و ماتریس ژاکوبین توسعه یافته قرار گرفته است.

در مقاله خوخی [۱۲] مسئله بهینه‌سازی چند هدفه تولید مسیر بازوهای با درجه آزادی اضافی مورد بررسی قرار گرفته است. هدف این مسئله شروع از یک نقطه از پیش تعیین شده و رسیدن به نقطه نهایی معلوم با حداکثر چالاکی ربات و حداقل انرژی مصرفی و ارضاء قیوداتی به شکل تساوی و ناتساوی می‌باشد. از جمله مقالاتی که به طراحی مسیر برای مینیمم گشتاور یا نیرو پرداخته‌اند، می‌توان به [۱۳] اشاره کرد. در مقاله بعدی خوخی و بارون [۱۴] توابع هدف چندگانه برای طراحی مسیر ربات افزونه ارائه شده است. برای این کار از یک شبکه عصبی-فازی آفلاین برای تولید مسیرهای چند هدفه در فضای کاری ربات استفاده شده است. در مرحله بعد، از یک شبکه دیگر برای آموزش دینامیک ربات استفاده شده است.

2. Particle Swarm Optimization

1. Task scheduling problem

می‌شود. هر لینک دارای دو جک هیدرولیکی کوچک می‌باشد. در پیچه های جک های هیدرولیکی توسط لوله های انعطاف پذیر به یکدیگر وصل شده‌اند. در مسیر لوله‌ها از یک عدد شیر سلونوئیدی استفاده شده است که در صورت باز بودن شیر، جک‌ها آزادانه حرکت می‌کنند و کاهش طول یک جک منجر به افزایش طول برابر در جک دیگر خواهد شد و زمانی که شیر سلونوئیدی بسته باشد، کل لینک قفل خواهد شد.



شکل ۱ بازوی مارگون کابلی با مفاصل قفل شونده [۱۵]

با متصل کردن لینک‌ها به یکدیگر و تشکیل بازو، در هر لحظه، تمامی لینک‌ها بجز یکی از آنها قفل شده و سپس با تغییر طول کابل، تغییرات لازم در اندازه پایه آن لینک بوجود آمده و بدین ترتیب می‌توان با هر ترتیب دلخواهی مفصلها را آزاد و یا قفل کرد و با کشش دو کابل، هر شکل دلخواهی را به بازو داد.

سپس ترتیب بهینه سوئیچینگ‌ها با بکارگیری الگوریتم پرواز پرندگان با متغیرهای گسسته یافت خواهد شد. در این بخش همچنین، روش جدیدی بنام پرواز پرندگان هیبریدی، به‌عنوان استراتژی دوم معرفی خواهد شد. نتایج بهینه‌سازی در بخش پنجم ارائه می‌شود. در این بخش نتایج بهینه‌سازی دو استراتژی با هم مقایسه می‌شوند و تفاوت میان آنها بحث خواهد شد تا بدین ترتیب سرعت همگرایی و عملکرد روش جدید ارائه شده به روشنی مشخص شود. بعلاوه مقایسه‌ای بین نتایج پرواز پرندگان گسسته و الگوریتم ژنتیک گسسته در این بخش صورت خواهد گرفت. در نهایت نتیجه گیری پایانی در بخش ششم آورده خواهد شد.

۲- مفاهیم اولیه

۲-۱- تعریف مسئله

در این مقاله، هدف طراحی مسیر نوع خاصی از بازوهای با درجات آزادی افزونه می‌باشد. در بازوهای کابلی مرسوم از سه کابل برای کنترل موقعیت هر لینک استفاده می‌شود. اما در این بازو تنها از سه کابل برای کنترل موقعیت کل بازو استفاده می‌شود و در کنار آن از سیستم قفل در روی هر لینک بهره می‌برد. این بازو دارای لینک‌های مشابهی می‌باشد که به‌صورت سری به یکدیگر متصل شده‌اند در حالی که هر کدام از این لینک‌ها دارای یک مکانیزم موازی مطابق شکل ۱ هستند. عملگرهای این بازوها به‌صورت کابلی می‌باشند به‌طوری که در حالت صفحه‌ای دو کابل از لبه های لینک‌ها عبور کرده و انتهای آن به بالاترین لینک بازو متصل می‌شود. انتهای دیگر کابلها توسط موتور و قرقره متصل به آن کشیده می‌شود. شماتیک لینک‌های بازو در حالت صفحه ای در شکل ۲ نشان داده شده است. برای جلوگیری از پیچیدگی‌های لینک سه بعدی مورد نظر، ایده اصلی عملکرد هر لینک در حالت دو بعدی در شکل ۲ آورده شده است. در این مقاله جهت پرهیز از پیچیدگی‌های سینماتیک ربات اصلی، و بررسی ایده اصلی به کارگیری الگوریتم پرواز پرندگان گسسته و هیبریدی در طراحی مسیر بازوی ربات مارگون کابلی با مفاصل قفل شونده، صرفاً به طراحی مسیر برای بازوی مارگون صفحه‌ای پرداخته

$$l_1 \sin \theta_1 + l_b \sin \theta_m = l_2 \sin \theta_2 \quad (3)$$

باید توجه داشت که طول سکوی ثابت و متحرک با یکدیگر برابر و با l_b نشان داده شده‌اند. اگر θ_m بین (۲) و (۳) حذف شود، داریم:

$$l_1^2 + l_2^2 - 2l_1 l_2 \cos(\theta_2 - \theta_1) + 2l_b(l_2 \cos \theta_2 - l_1 \cos \theta_1) = 0 \quad (4)$$

با جایگزاری $\theta_1 = \theta_2$ و $l_2 = L - l_1$ ، زاویه θ_1 را می‌توان براحتی توسط رابطه زیر محاسبه کرد.

$$\theta_1 = \cos^{-1} \left(\frac{2l_1 - L}{2l_b} \right) \quad (5)$$

با جایگزاری در فرمول (۲):

$$\theta_m = \cos^{-1} \left(1 - \frac{(L - 2l_1)^2}{2l_b^2} \right) \quad (6)$$

از آنجایی که l_2 را براحتی می‌توان توسط رابطه (۱) برحسب l_1 نوشت، تنها پارامتری که پیکربندی لینک را تغییر می‌دهد، l_1 خواهد بود. در ادامه این مقاله، هر جا صحبت از طول پایه‌ها به میان آمد، منظور l_1 می‌باشد.

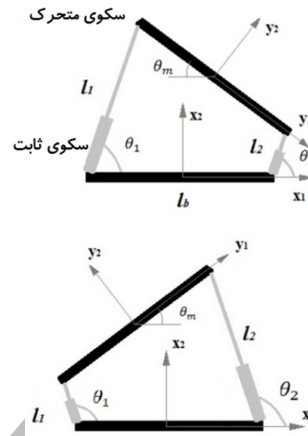
یک سیستم مختصات کارترین به نام A در مرکز سکوی ثابت طوری قرار می‌دهیم که محور x آن موازی با لبه‌ی سکوی ثابت باشد. همچنین سیستم مختصات دیگری به نام B بر روی مرکز سکوی متحرک قرار می‌دهیم (شکل ۲). ماتریس تبدیل بین این سیستم‌های مختصات، ${}^A_B T$ به این صورت تعریف می‌شود.

$$P_A^* = {}^A_B T P_B^* \quad (7)$$

$${}^A_B T = [{}^A_B R \quad P_{BORG}] \quad (8)$$

$${}^A_B R = \begin{bmatrix} \cos \theta_m & -\sin \theta_m \\ \sin \theta_m & \cos \theta_m \end{bmatrix} \quad (9)$$

$$P_{BORG} = \begin{bmatrix} \frac{l_b}{2} (\cos \theta_m - 1) + l_1 \cos \theta_1 \\ l_1 \sin \theta_1 + \frac{l_b}{2} \sin \theta_m \end{bmatrix} \quad (10)$$



شکل ۲ شماتیک لینک یک درجه آزادی بازو در دو وضعیت

پایه‌های لینک را از چپ به راست به ترتیب با l_1 و l_2 نشان می‌دهیم. با توجه به رابطه قیدی بین حرکت جک‌های هیدرولیکی می‌توان نوشت:

$$l_1 + l_2 = L \quad (1)$$

برای اینکه درجه آزادی هر لینک را به یک کاهش دهیم، فرض دیگری به صورت $\theta_1 = \theta_2$ نیز انجام شده است. در عمل می‌توان این فرض را با اضافه کردن یک لینک براحتی ایجاد نمود، برای انجام این کار، می‌توان میله‌ای را به سیلندرهای دو جک هیدرولیکی لولا کرد و یک مکانیزم متوازی الضلاع تشکیل داد تا همواره قید $\theta_1 = \theta_2$ برقرار بماند. اساساً اگر این فرض را انجام ندهیم، هر لینک از بازو دارای دو درجه آزادی خواهد شد که حرکت دادن آن با دو کابل امکانپذیر نمی‌باشد زیرا کابل‌ها فقط نیروی کششی را تحمل می‌کنند. با توجه به توضیحات ارائه شده در این بخش، با مشخص کردن طول یکی از جک‌ها، مثلاً l_1 ، می‌توان پیکربندی کل لینک را تعیین کرد.

۲-۲- سینماتیک مستقیم

سینماتیک مستقیم بازو را می‌توان با در نظر گرفتن حلقه‌های سینماتیکی هر لینک برحسب $l_1, l_b, L, \theta_1, \theta_m$ نوشت:

$$l_1 \cos \theta_1 + l_b \cos \theta_m = l_b + l_2 \cos \theta_2 \quad (2)$$

$$\frac{L - l_b}{2} \leq l_1 \leq \frac{L + l_b}{2} \quad (12)$$

که در آن l_b طول سکو، L مجموع l_1 و l_2 و مقدارش عددی ثابت می‌باشد. با بکارگیری این معادله، مقدار θ_1 همواره بین 60° و 120° درجه تغییر خواهد کرد. البته قید فوق قابل تغییر است، و برای پیاده سازی شرایط واقعی ربات، اعمال می‌شود.

۳- بهینه‌سازی پرواز پرندگان

۱-۳- مقدمه

روش پرواز پرندگان یک الگوریتم بهینه‌سازی بر پایه جمعیت می‌باشد [۱۶]. این روش در ابتدا توسط ابرهات و کندی ابداع شد. بهینه‌سازی یک مسئله در این روش، با تعدادی از پاسخ‌های اتفاقی آغاز شده و سپس با تکرارهای پیاپی، پاسخ‌های بهینه در فضای جستجو کشف می‌شوند. در برخی سیستم‌ها، الگوریتم پرواز پرندگان بسیار هوشمند تر و سریع تر از الگوریتم‌های ژنتیک عمل می‌کند و پیاده‌سازی آن نیز ساده‌تر می‌باشد. توصیف جامع‌تر در مورد این الگوریتم را می‌توان در [۱۶] و [۱۷] یافت. در ادامه توصیف مختصری از الگوریتم پرواز پرندگان پیوسته داده خواهد شد.

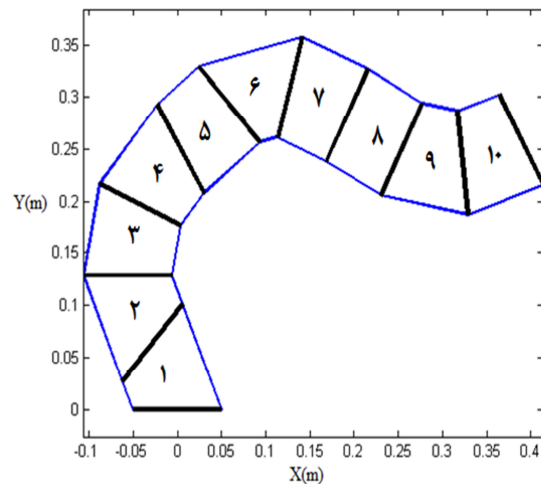
۲-۳- الگوریتم پرواز پرندگان پیوسته

فرض کنید یک فضای جستجوی D بعدی و یک دسته N تایی از ذرات در این فضا داشته باشیم (این ذرات همان پرندگان متعلق به گروه می‌باشند). بردار D بعدی به نام موقعیت ذره می‌باشد. هر ذره از بردار $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ، $i = 1, 2, \dots, N$ نشان دهنده موقعیت ذره می‌باشد و هر ذره می‌تواند یک پاسخ احتمالی برای مسئله بهینه‌سازی باشد. ارزندگی هر ذره را می‌توان با قرار دادن موقعیت آن ذره در تابع هزینه به‌دست آورد. تمایل ذره نام برای تغییر موقعیت در یک جهت بخصوص را با $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ، $i = 1, 2, \dots, N$ نشان می‌دهیم و آن را سرعت ذره می‌نامیم (در اینجا سرعت ذره مفهوم تغییرات موقعیت Δx را دارد). بهترین موقعیت تجربه شده توسط ذره i ام تا تکرار t را با $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ، نشان می‌دهیم و

که در آن، P_B^* موقعیت نقطه P در سیستم متحرک است و P_A^* موقعیت این نقطه در سیستم مختصات ثابت می‌باشد. ماتریس دوران با ${}^A_B R$ نشان داده شده است و P_{BORG} موقعیت مرکز سیستم مختصات متحرک در سیستم ثابت می‌باشد. در حالتی که n لینک به‌صورت سری به یکدیگر متصل شوند، موقعیت مجری نهایی را می‌توان با ضرب کردن ماتریس‌های تبدیل لینک‌های مختلف به‌دست آورد. اگر P_1^* موقعیت مجری نهایی بازو باشد، داریم:

$$P_1^* = {}^2_1 T {}^3_2 T \dots {}^n_{n-1} T P_n^* \quad (11)$$

در شکل ۳ شماتیکی از یک بازوی ده درجه آزادی نشان داده شده است. لینک‌ها به‌ترتیب از پایین به بالا نام‌گذاری شده‌اند. در این شکل فرض شده که $l_b = 10$ و $l_1 = 14$ سانتی‌متر و همچنین $l_1 = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$ سانتی‌متر باشد (طول بازوی اول در ۱۰ لینک). در اینجا l_1 برداری است که نشان دهنده طول پایه‌های سمت چپ می‌باشد.



شکل ۳ شماتیک بازوی کابلی ۱۰ درجه آزادی

برای تکمیل مدل‌سازی، قید فیزیکی به‌صورت زیر برای طول l_1 در نظر گرفته شده است:

که در آن c_1 ، c_2 ضرایب یادگیری ادراکی و اجتماعی هستند. اگر بخش‌های داخلی معادله را برای راحتی به صورت زیر نام گذاری کنیم: $a_i^t = w \otimes F_1(\cdot)$ ، $\beta_i^t = c_1 \otimes F_2(\cdot)$. اولین بخش از معادله (۱۵)، $a_i^t = w \otimes F_1(\cdot)$ ، نقشی مشابه عبارت اینرسی در PSO استاندارد را دارد. اپراتور \otimes ، جایگشتی را با احتمال w بر روی عبارت $F_1(\cdot)$ اعمال می‌کند. عبارت دوم این معادله، $\beta_i^t = c_1 \otimes F_2(\cdot)$ ، جایگشتی را با احتمال c_1 بر روی ذره اعمال می‌کند که آن را شبیه بهترین موقعیت تجربه شده توسط آن ذره (P_i^{t-1}) می‌کند و به طور مشابه عبارت $c_2 \otimes F_3(\cdot)$ نقش ترم یادگیری اجتماعی را بازی می‌کند. توضیح جامع این روش را می‌توان در [۱۹] یافت.

۴- تعریف و مدل‌سازی مسئله بهینه‌سازی

در طراحی مسیر تابع زمان، موقعیت مجری نهایی بازو در فضای کاری کارترین داده می‌شود و میزان تغییرات لازم در فضای مفصلی برای ایجاد این حرکت، خروجی مطلوب می‌باشد. برای رسیدن به این هدف، ناچار به حل سینماتیک معکوس بازو می‌باشیم و از آنجا که ماتریس ژاکوبین بازوهای افزونه به فرم مربعی نمی‌باشد، حل سینماتیک معکوس آنها دشوار است. اخیراً از رویکردهای بهینه‌سازی در طراحی مسیر در بازوهای افزونه استفاده شده است. در این روش‌ها ابتدا سینماتیک مستقیم بازو را به دست می‌آورند و سپس با بکار بردن آن در الگوریتم بهینه‌سازی و در تکرارهای متوالی، مقادیر بهینه متغیرهای مفصلی را به دست می‌آورند.

با توجه به ساختار ویژه ربات صفحه‌ای مورد نظر، حرکت نقطه نهایی بازو روی یک خط راست، یک مسیر نسبتاً پیچیده برای این ربات محسوب می‌شود. لذا برای نشان دادن نحوه عملکرد الگوریتم، مسیر مطلوب را برای نقطه انتهایی بازو، یک خط راست در نظر می‌گیریم. بدیهی است که الگوریتم پیشنهادی را برای سایر مسیرهای دیگر نقطه انتهایی نیز می‌توان عیناً به کار برد. لذا در این مقاله هدف این است که نقطه نهایی بازو بر روی یک خط راست حرکت کند. نقاط ابتدایی و انتهایی مسیر نیز داده شده‌اند. با استفاده از روش بهینه‌سازی باید مقدار تغییر طول پایه هر لینک و ترتیب باز و بسته

بهترین موقعیت تمامی ذرات تا تکرار t را $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ می‌نامیم. موقعیت ذره i ام در هر تکرار را می‌توان توسط رابطه زیر بروز کرد:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (13)$$

$$V_i(t+1) = wV_i(t) + b_1r_1(P_i - X_i(t)) + b_2r_2(P_g - X_i(t)) \quad (14)$$

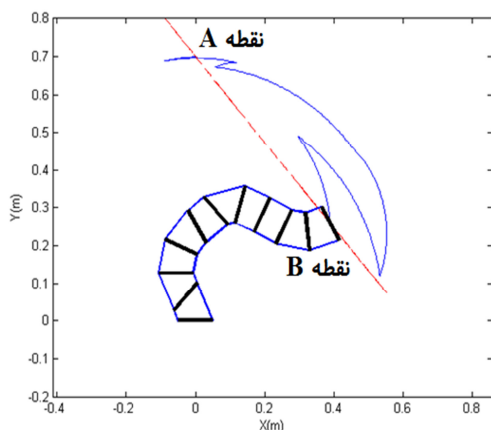
که در آن $i=1,2,\dots,m$ و w ضریب اینرسی است و مقدار ثابتی در بازه $[0,1]$ می‌گیرد، b_1 و b_2 به ترتیب ضریب یادگیری ادراکی و اجتماعی هستند و r_1 و r_2 اعداد اتفاقی تولید شده در بازه $[0,1]$ می‌باشند. معیار پایان پذیری الگوریتم می‌تواند حداکثر تعداد تکرارها و یا رسیدن به مقدار مشخصی از ارزندگی برای P_g باشد [۱۸].

۳-۳- الگوریتم PSO گسسته

با توجه به توضیحات بخش قبلی می‌توان متوجه شد که الگوریتم پرواز پرندگان معمولی را نمی‌توان در مسائلی که دارای متغیرهای گسسته هستند، بکار برد. زیرا تفاضل موقعیت دو ذره ممکن است موقعیتی منفی را ایجاد کند و یا ضرب موقعیت یک ذره، عددی غیر صحیح تولید کند.

روشهای متعددی برای حل مسائل بهینه‌سازی گسسته توسط الگوریتم پرواز پرندگان ارائه شده است. برخی از آنها جایگشت‌های جدیدی را برای بروز رسانی موقعیت ذرات ارائه می‌دهند [۱۹]، درحالی که دیگر مقالات، از ذرات دودویی به شکل ماتریسی بهره می‌برند [۱۷]. در این تحقیق برای مدل‌سازی صحیح مسئله بهینه‌سازی گسسته، صورت مشخصی از این الگوریتم بکار رفته است [۱۹] که در ادامه این بخش مختصراً توضیح داده می‌شود. اگر موقعیت ذره i ام در تکرار t ام با X_i^t نشان داده شود، می‌توان آن را توسط رابطه زیر بروز رسانی کرد.

$$X_i^t = c_2 \otimes F_3(c_1 \otimes F_2(w \otimes F_1(X_i^{t-1}), P_i^{t-1}), G_i^{t-1}) \quad (15)$$



شکل ۵ شماتیک بازوی ۱۰ درجه آزادی با ترتیب سوئیچینگ

$$S=[۱۰, ۵, ۸, ۳, ۹, ۲, ۷, ۶, ۱, ۴]$$

۴-۱- استراتژی اول

استراتژی اولی که به ذهن می‌رسد، تقسیم راه حل مسئله به دو بخش اصلی می‌باشد. بخش اول محاسبه سینماتیک معکوس بازو و بخش دوم بهینه‌سازی سوئیچینگ می‌باشد. یعنی در قدم اول با حل سینماتیک معکوس در نقاط ابتدایی و انتهایی، پیکربندی و شکل کلی بازو تعیین می‌شود و در قدم دوم ترتیب اعمال تغییر طول پایه‌ها به دست آورده می‌شود. به عبارت دیگر در قدم دوم، از نتایج سینماتیک معکوس استفاده می‌کنیم و میزان تغییرات لازم در طول پایه هر لینک را به دست می‌آوریم و سپس ترتیب اعمال آنها را طوری بهینه می‌کنیم که مسیر حرکت بازو یک خط راست شود. در ادامه راجع به تابع هدف و قیود بهینه‌سازی توضیح داده خواهد شد.

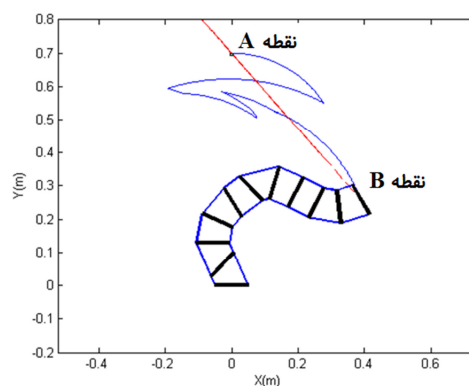
تابع هزینه در بخش دوم، مقدار مساحت بین مسیر حرکت واقعی و مسیر حرکت مطلوب، در نظر گرفته می‌شود. باید توجه داشت که در این بخش فقط ترتیب سوئیچینگ‌ها بهینه می‌شود و مقدار تغییر طول پایه‌ها از قبل در بخش اول محاسبه شده‌اند. شکل ۶ نشان دهنده مساحت بین یک مسیر دلخواه و مسیر مطلوب می‌باشد. بنابراین می‌توان تابع هزینه این بخش را اینگونه تعریف کرد:

$$Area\ cost = S_{Area} \quad (۱۶)$$

شدن قفل‌ها طوری به دست آیند که مسیر حرکت نقطه انتهایی بازو تا حد ممکن بر روی خط راست مفروض قرار گیرد. همانطوری که پیش‌تر نیز توضیح داده شد، دو عامل باعث تغییر شکل مسیر حرکت مجری نهایی بازو می‌شود. اولاً، نحوه تغییر طول پایه‌ها که متغیری پیوسته است و ثانیاً ترتیب بازو بسته شدن قفل لینک‌ها که متغیری گسسته می‌باشد. با این توضیح، الگوریتم پرواز پرندگان پیوسته برای بهینه‌سازی عامل اول و الگوریتم پرواز پرندگان گسسته برای بهینه‌سازی عامل دوم بکار گرفته شده است.

در این قسمت برای روشن شدن موضوع، از یک مثال استفاده می‌کنیم. اگر نقطه شروع حرکت، A، در وضعیت کاملاً عمودی بازو باشد و مختصات نقطه مقصد، B، نیز داده شده باشد. در این صورت طول پایه‌های هر لینک مقدار معینی تغییر کرده است. حال اگر این تغییر طول‌ها را با ترتیب‌های متفاوتی اعمال کنیم، مسیر حرکت مجری نهایی کاملاً متفاوت خواهد بود. شکل‌های ۴ و ۵ به وضوح تاثیر ترتیب سوئیچینگ قفل‌ها را بر روی مسیر حرکت، نشان می‌دهند. با مشاهده این شکل‌ها مشخص می‌شود که ترتیب بازو بسته کردن قفل مفاصل، تغییر زیادی در شکل مسیر حرکت بازو ایجاد می‌نماید.

در این مساله، برای محدود نمودن فضای پاسخ، فرض می‌شود که بخواهیم، قفل هر لینک فقط یک بار باز و سپس بسته شود پس برای یک بازوی ده درجه آزادی، تنها ده سوئیچ در نظر گرفته می‌شود. در ادامه این بخش دو استراتژی برای طراحی مسیر این بازو در نظر گرفته شده که به تفصیل راجع به آن بحث خواهد شد.



شکل ۶ شماتیک بازوی ۱۰ درجه آزادی با ترتیب سوئیچینگ

$$S=[۶, ۱, ۷, ۴, ۹, ۵, ۸, ۳, ۱۰, ۲]$$

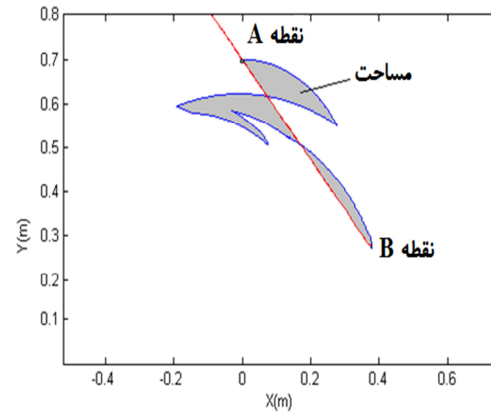
با تمرکز بر روی این استراتژی در بخش محاسبه سینماتیک معکوس، می‌توان ذرات را به صورت $X_i = [l_1^i, l_2^i, l_3^i, \dots]$ تعریف کرد که در آن l_1^i طول پایه لینک i ام است. در روند بهینه‌سازی، هر جا طول پایه ای از قید رابطه (۱۲) تخلف کند، سرعت آن ذره صفر قرار داده شده و همچنین مقدار ذره در تکرار پیشین را جایگزین مقدار فعلی می‌کنیم. در حل سینماتیک معکوس دو تابع هزینه به صورت زیر، در کنار هم بهینه می‌شوند.

$$Fun1 = \sqrt{(X_B - X_{final})^2 + (Y_B - Y_{final})^2} \quad (17)$$

$$Fun2 = \sqrt{\sum_{i=1}^n (B_l l_1^i - A_l l_1^i)^2} \quad (18)$$

که در آن (X_B, Y_B) موقعیت نقطه B و (X_{final}, Y_{final}) موقعیت نهایی مجری نهایی می‌باشد و l_1^i طول پایه لینک i ام می‌باشد. در رابطه (۱۷) فاصله بین موقعیت مطلوب و موقعیت واقعی مجری نهایی محاسبه می‌شود. در حالی که در رابطه (۱۸) مجموع اندازه تغییر طول‌های تمامی لینک‌های بازو مد نظر قرار می‌گیرد. علت اضافه کردن $Fun2$ در مسئله بهینه‌سازی آن است که اگر به تنهایی و فقط $Fun1$ را در نظر بگیریم، پاسخ‌های بسیار زیادی برای سینماتیک معکوس به دست می‌آید. تابع $Fun2$ تعداد زیادی از جواب‌ها ناخواسته را حذف کرده و شکلی از بازو را ارائه می‌دهد که بدون پیچ و تاب باشد. در صورتی که پاسخ دارای پیچ و تاب باشد، در بخش دوم که ترتیب سوئیچینگ را به دست می‌آید، هرگز نمی‌توان ترتیبی را پیدا کرد که بازو مسیر مستقیم پیماید. در نتیجه می‌توان گفت که وارد کردن $Fun2$ باعث یکنواخت کردن حرکت بازو و بهبود نتایج خواهد شد.

با توجه به توضیحات بالا مشاهده شد که در حل سینماتیک معکوس با یک مسئله بهینه‌سازی چند هدفه روبرو هستیم. دو روش متفاوت برای حل این مسئله وجود دارد. یکی ترکیب



شکل ۶ مساحت بین مسیر مطلوب و یک مسیر دلخواه

در این بخش می‌توان ذرات را به شکل $S_i = [s_1, s_2, \dots, s_n]$ معرفی کرد. به عنوان مثال اگر ذره ای را به صورت زیر در نظر بگیریم،

$$S = [10, 5, 8, 3, 9, 2, 7, 6, 1, 4]$$

در ابتدا تمامی لینک‌ها بجز لینک شماره ۱۰ قفل هستند و با کشیدن کابل‌ها تغییر طول مشخصی به پایه‌های آن لینک داده می‌شود. سپس بار دیگر همه لینک‌ها قفل شده و لینک ۵ باز می‌شود و به همین ترتیب تا لینک ۴ ادامه پیدا می‌کند. در این جا ذکر دو نکته ضروری است. اول اینکه تعداد سوئیچ‌ها برابر تعداد لینک‌ها یعنی ۱۰ فرض شده است در حالی که افزایش تعداد سوئیچ‌ها و یا اضافه کردن سوئیچ‌های تکراری ممکن باعث بهتر شدن جواب مسئله شود. این فرض تنها برای ساده کردن بهینه‌سازی انجام شده است و بهینه‌سازی بدون این قیود، در دست بررسی می‌باشد، که در کارهای آتی ارائه خواهد شد. علاوه بر این باید توجه داشت که جدا کردن بخش گسسته و پیوسته مسئله ممکن است رسیدن به پاسخ بهینه مطلق را دشوار سازد اما می‌توان انتظار رسیدن به یک پاسخ بهینه محلی را داشت. هدف اصلی از بیان این، زمینه سازی برای ارائه روشی کاملتر تحت عنوان استراتژی دوم است. به علاوه با مقایسه نتایج دو استراتژی، کارایی روش پیشنهادی روشن تر می‌شود.

توابع هزینه و دیگری استفاده از الگوریتم پرواز پرندگان چند هدفه در ادامه این بخش نحوه بکارگیری این روش‌ها مختصراً توضیح داده خواهد شد.

۴-۱-۱- ترکیب وزنی توابع هزینه

در این روش تابع هزینه جدید، که ترکیب خطی از دو تابع هدف مورد نظر می‌باشد را به صورت زیر ایجاد می‌کنیم:

$$f = w_1 \times Fun1 + w_2 \times Fun2 \quad (19)$$

ضرائب w_1 و w_2 با سعی و خطا به ترتیب برابر ۰.۲۵ و ۰.۷۵ قرار داده شدند. این مقادیر منجر به حداقل شدن هر دو تابع هزینه $Fun1$ و $Fun2$ خواهد شد و خواسته پژوهش را برآورده می‌کند. برای تعیین ضرایب به صورت سعی و خطا، ابتدا مقداری برابر به ضرایب w_1 و w_2 داده شده و سپس ضریب w_1 کاهش و ضریب w_2 افزایش داده شده است و در هر بار وضعیت بازو و فاصله مجری نهایی از نقطه هدف را بررسی شده است. این کار تا جایی ادامه یافته که فاصله مجری نهایی از نقطه هدف شروع به افزایش نماید. در این وضعیت، ضرایب بهینه استخراج شده‌اند. لازم به ذکر است که دو تابع مورد نظر در کل فضای کاری بازو تقریباً از یک مرتبه هستند و چنانچه اختلافی در مرتبه آنها وجود داشته، با تنظیم ضرایب وزنی جبران شده‌اند. به همین علت است که با وجود اهمیت بیشتر تابع $Fun1$ ضریب مربوط به آن کمتر از ضریب تابع $Fun2$ می‌باشد.

۴-۱-۲- روش پرواز پرندگان چند هدفه^۱

الگوریتم پرواز پرندگان چند هدفه یا برداری صورتی از الگوریتم بهینه‌سازی پرواز پرندگان است که برای حل مسائل بهینه‌سازی چند هدفه بکار می‌رود. اگر مسئله‌ای در نظر گرفته شود که دو تابع هدف در آن وجود داشته باشد، دو گروه از ذرات ایجاد کرده و هر کدام را برای بهینه‌سازی یکی از این توابع مامور می‌شوند.

فرمولاسیون کلی پرواز پرندگان برداری با M گروه ذرات، هر کدام شامل N ذره، به صورت زیر است [۱۹]:

$${}^{[j]}X_i(t+1) = {}^{[j]}X_i(t) + {}^{[j]}V_i(t+1) \quad (20)$$

$${}^{[j]}V_i(t+1) = {}^{[j]}V_i(t) + c_1 r_1 \{ {}^{[j]}P_i - {}^{[j]}X_i(t) \} + c_2 r_2 \{ {}^{[k]}P_{gb} - {}^{[j]}X_i(t) \} \quad (21)$$

که در آن ${}^{[j]}X_i(t)$ ، ${}^{[j]}V_i(t)$ ، ${}^{[k]}P_{gb}$ ، ${}^{[j]}P_i$ به ترتیب موقعیت فعلی ذره i ام گروه j ام، سرعت ذره i ام، بهترین موقعیت تجربه شده تاکنون توسط ذره i ام و بهترین موقعیت یافت شده در گروه k ام در تکرار t ام می‌باشد. در اینجا نیز مشابه الگوریتم پرواز پرندگان استاندارد، c_p ضریب یادگیری ادراکی، c_g ضریب یادگیری اجتماعی و w ضریب اینرسی می‌باشد. r_1 و r_2 اعداد تصادفی بین صفر و یک هستند. در الگوریتم پرواز پرندگان برداری رفتار هر گروه از ذرات تحت تاثیر بهترین ذره گروه k ام می‌باشد. پارامتر k را می‌توان به روش‌های متفاوتی تعیین کرد که هر کدام نشان دهنده سبکی از انتقال اطلاعات بین گروه‌ها می‌باشد.

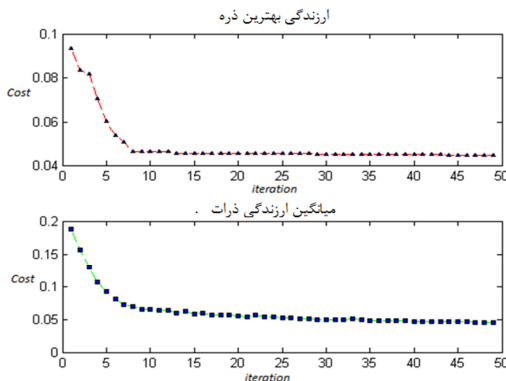
در این مسئله روش الگوریتم پرواز پرندگان برداری برای بهینه‌سازی طول پایه‌های هر لینک بکار می‌رود به طور که برای هر تابع هزینه، یک گروه ذرات تعریف می‌شود. گروه اول با گشتن در فضای جستجو بدنبال حداقل‌سازی $Fun1$ و گروه دوم در تلاش برای یافتن حداقل تابع $Fun2$ می‌باشد. در هر تکرار، موقعیت ذرات یک گروه تحت تاثیر بهترین ذره گروه خودشان و بهترین ذره گروه دیگر بروز می‌شود. در نهایت بهترین ذره گروه اول به عنوان پاسخ بهینه انتخاب می‌شود زیرا اهمیت تابع $Fun1$ بیشتر است.

۴-۲- استراتژی دوم

در استراتژی دوم، بخش پیوسته و گسسته مسئله ترکیب شده و روش بهینه‌سازی پرواز پرندگان هیبریدی معرفی خواهد شد. عبارت دیگر در این روش دو تابع هزینه به‌طور همزمان بهینه می‌شوند تا بازو از یک نقطه از پیش تعیین شده شروع به حرکت کرده و بر روی خط مستقیم به سمت نقطه مقصد برود. در این روش، ذرات به صورت زیر تعریف می‌شوند:

1. VEPSO

که مقدار سرعت ماکزیمم، بیشتر از این مقدار انتخاب شود، موقعیت ذرات بطور ناگهانی تغییر می‌کند و الگوریتم همگرا نمی‌شود. b_1 و b_2 در معادلات (۱۳) و (۱۴) برابر ۲ و مقدار پارامتر w به صورت خطی و کاهنده در نظر گرفته شده است. کاهش خطی مقدار w منجر به جستجوی وسیع در فضای جواب، در تکرارهای آغازین شده و کم کم دایره جستجو محدودتر شده تا مقدار مینیمم مطلق یافت شود. بدین ترتیب از افتادن در دام مینیمم‌های محلی جلوگیری به عمل خواهد آمد. مقادیر حداقل و حداکثر w به ترتیب برابر $w_{\min}=0.4$ و $w_{\max}=0.95$ انتخاب شده‌اند. شکل ۷ ارزیابی بهترین ذره و میانگین ارزیابی ذرات با در نظر گرفتن تابع هزینه ترکیب وزنی نشان می‌دهد.



شکل ۷ ارزیابی بهترین ذره و میانگین ارزیابی ذرات با در نظر گرفتن ترکیب وزنی توابع هزینه با معادله $cost = 0.25 \times Fun1 + 0.75 \times Fun2$

در پرواز پرندگان چند هدفه، دو گروه ذرات، هر کدام شامل ۴۰ ذره انتخاب شده و با توجه به پیچیدگی مسئله حداکثر تعداد تکرارها برابر ۱۵۰ قرار داده شد. گروه اول ذرات بدنبال حداقل سازی تابع $Fun1$ و گروه دوم بدنبال مقدار مینیمم $Fun2$ می‌گردند. در جدول ۱ نتایج بهینه‌سازی چند هدفه توسط پرواز پرندگان برداری و تابع هزینه ترکیب وزنی برای سه نقطه هدف مختلف نشان داده شده است. لازم به ذکر است که موقعیت آغازین بازو در حالتی است که بازو به صورت کاملاً عمودی، یعنی در موقعیت $A(0,0.7)$ قرار دارد.

$$X_i = [s_1, s_2, s_3, \dots, s_n, l_1^1, l_1^2, l_1^3, \dots, l_1^n]$$

در هر ذره $2n$ عضو وجود دارد، در n عضو اول، اعداد صحیح (گسسته) که نشان دهنده ترتیب سوئیچینگ‌ها می‌باشد، قرار گرفته است و در n عضو دوم، طول پایه‌ها قرار گرفته‌اند. در روش پرواز پرندگان هیبریدی معرفی شده در این مقاله، ارزیابی هر ذره از جمع وزنی دو تابع هدف تعیین می‌شود. اولاً، اختلاف بین موقعیت مجری نهایی و موقعیت مطلوب، مطابق رابطه (۱۷) و ثانیاً، مساحت خطا مطابق رابطه (۱۶). توجه شود که رابطه (۱۸) در اینجا لازم نمی‌باشد زیرا به طور همزمان بدنبال مقدار تغییر طول پایه و ترتیب سوئیچینگ می‌گردد که بازو بر روی خط راست حرکت کند. بعبارت دیگر رابطه (۱۸) زمانی لازم است که دو بخش مسئله بهینه‌سازی از هم جدا باشند. تابع هدف مسئله از این قرار است:

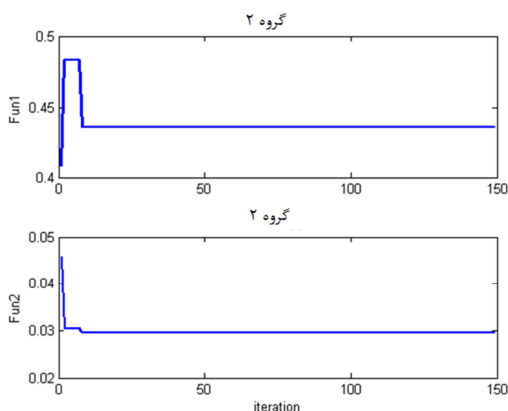
$$cost = w_c \times Fun1 + w_a \times Area \text{ cost} \quad (22)$$

ضرائب وزنی w_c و w_a به صورت سعی و خطا و به ترتیب برابر ۰.۸۵ و ۰.۱۵ می‌باشند. برای تنظیم ضرایب، فاصله مجری نهایی از نقطه هدف را در محدوده معین نگه داشته و با افزایش ضریب مساحت، نتایج را بهبود می‌دهیم. برای بروز رسانی ذرات در هر تکرار، از معادله (۱۵) برای بخش گسسته ذرات و از معادله (۱۳) و (۱۴) برای بخش پیوسته ذرات استفاده شده است. نتایج این استراتژی در بخش پنجم آورده شده است. مقایسه بین نتایج این استراتژی حل با استراتژی اول نشان از توانایی روش پیشنهاد شده دارد.

۵- مقایسه نتایج و بحث

۵-۱- استراتژی اول

برای اجرای الگوریتم پرواز پرندگان پیوسته، گروهی شامل ۵۰ ذره در نظر گرفته شده است. حداکثر تعداد تکرارها برابر ۵۰ و حداکثر سرعت ذرات نیز برابر ۰.۰۱ انتخاب شده است تا حداکثر تغییر موقعیت یک ذره از ۰.۰۱ بیشتر نشود. در صورتی



شکل ۹ بهترین مقدار $Fun1$ و $Fun2$ در گروه دوم از پرواز پرنندگان برداری

جدول ۲ نتایج بهینه‌سازی با پرواز پرنندگان گسسته را بعد از حل سینماتیک معکوس نشان می‌دهد. برای بکارگیری درست روش پرواز پرنندگان، باید ضرائب w ، $C1$ و $C2$ در معادله (۱۵) به دقت تعیین شوند. واضح است که تغییرات اندک در مقدار یکی از این پارامترها، منجر به کاهش و یا افزایش سرعت و کیفیت روند بهینه‌سازی خواهد شد. به‌عنوان مثال کاهش پارامترهای C باعث جستجوی جامع در تمام فضای جواب شده و افزایش آنها جستجوی محلی را در پی خواهد داشت.

جدول ۲ نتایج پرواز پرنندگان گسسته در مقابل نتایج الگوریتم ژنتیک گسسته

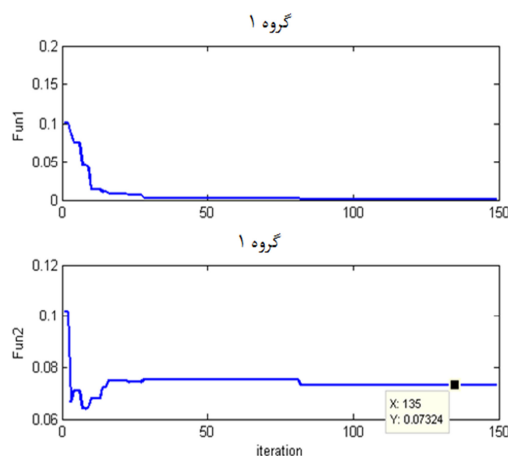
| نقطه هدف (X,Y) | پرواز پرنندگان گسسته | | الگوریتم ژنتیک گسسته |
|----------------|----------------------|----------------------|----------------------|
| | Area cost | Time of optimization | Area cost |
| (۰/۱, ۰/۲) | ۰/۰۱۴۷ | ۳۸/۷۳۶۶۲ | ۰/۰۱۶۸ |
| (۰/۳, ۰/۱) | ۰/۲۲۱۹ | ۲۴/۹۳۰۲۵۴ | ۰/۲۲۲۲ |
| (۰/۵, ۰/۲) | ۰/۰۹۸۷ | ۲۳/۹۵۶۲ | ۰/۰۹۸۴ |

در این پژوهش، برای فرار از دام مینیمم‌های محلی یک تابع خطی افزایشنده برای C و یک تابع خطی کاهنده برای پارامتر w در نظر گرفته شده است. فرم کلی این توابع به‌صورت زیر است:

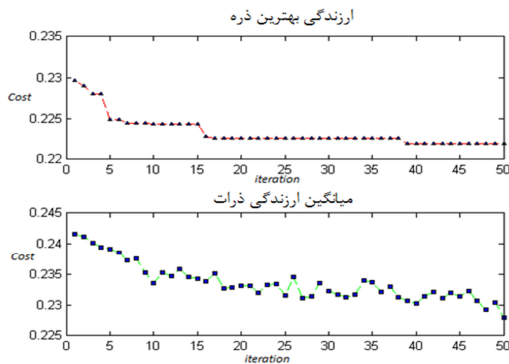
جدول ۱ بهینه‌سازی چند هدفه توسط پرواز پرنندگان برداری و ترکیب وزنی توابع هزینه

| نقطه هدف (X,Y) | پرواز پرنندگان پیوسته | | | | | |
|----------------|------------------------|--------|----------------------|-----------------------|---------|--------|
| | ترکیب وزنی توابع هزینه | | | پرواز پرنندگان برداری | | |
| | Fun1 | Fun2 | Time of optimization | Fun1 | Fun2 | time |
| (۰/۱, ۰/۲) | ۱/۰۴E-۰۵ | ۰/۰۷۸۲ | ۱/۴۰۸۹ | ۰/۰۰۳ | ۰/۰۹۴۸ | ۵/۰۹۴۲ |
| (۰/۳, ۰/۱) | ۲/۸۹E-۰۴ | ۰/۰۵۹۲ | ۱/۱۹۰۹ | ۷/۲E-۰۴ | ۰/۰۷۳۲ | ۳/۹۲۱۷ |
| (۰/۵, ۰/۲) | ۴/۷۶E-۰۵ | ۰/۰۳۷۶ | ۱/۲۱۶۱ | ۰/۰۰۱۱ | ۰/۰۵۹۱۴ | ۳/۸۸۳۹ |

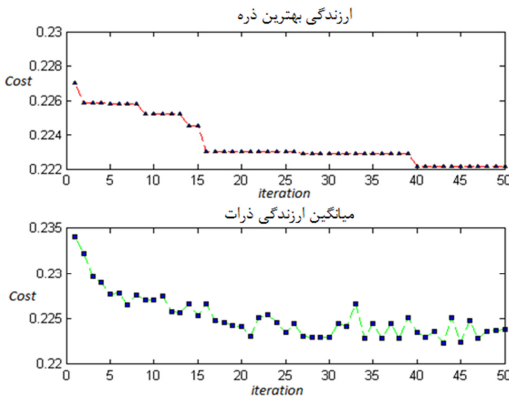
اگرچه الگوریتم پرواز پرنندگان برداری بمراتب پیچیده‌تر از روش ترکیب وزنی می‌باشد، اما نتایج آن بخوبی روش ترکیب وزنی نیست. علاوه بر زمان رسیدن به پاسخ آن نیز، بیشتر و تنظیم ضرائب بکار رفته در معادلات (۲۰) و (۲۱) بسیار دشوارتر می‌باشد. شکل‌های ۸ و ۹ به ترتیب نشان دهنده بهترین مقدار $Fun1$ و $Fun2$ در گروه اول و دوم می‌باشد. در شکل ۸ می‌توان مشاهده کرد که مقدار بطور یکنواخت کاهش یافته است زیرا گروه اول مسئول یافتن مقدار حداقل $Fun1$ می‌باشد و مقدار $Fun2$ در همین گروه، در ۱۰ تکرار اول کاهش یافته و سپس افزایش پیدا کرده است. این افزایش بدان سبب است که این دو تابع برضد هم عمل می‌کنند و افزایش یکی منجر به کاهش دیگری خواهد شد و چون گروه اول مسئول کاهش مقدار $Fun1$ است، ناچار به افزایش $Fun2$ شده است. بطور خلاصه، الگوریتم بگونه‌ای عمل می‌کند که مقدار تابع $Fun1$ را حداقل کند در حالی که مقدار تابع $Fun2$ بیش از حد افزایش پیدا نکند.



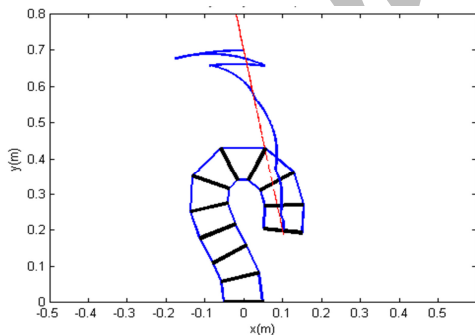
شکل ۸ بهترین مقدار $Fun1$ و $Fun2$ در گروه اول از پرواز پرنندگان برداری



شکل ۱۰ ارزندگی بهترین ذره و میانگین ارزندگی ذرات در بهینه‌سازی با پرواز پرندگان گسسته



شکل ۱۱ ارزندگی بهترین ذره و میانگین ارزندگی ذرات در بهینه‌سازی با الگوریتم ژنتیک گسسته

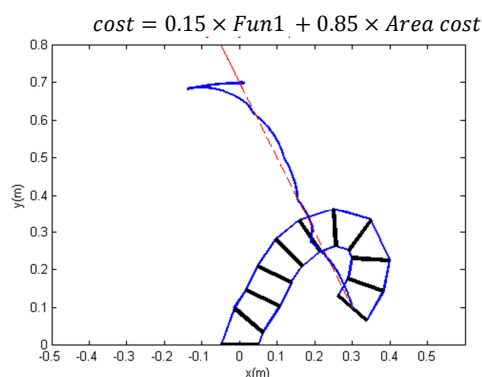


شکل ۱۲ شماتیک بازوی ۱۰ درجه آزادی و مسیر حرکت بهینه شده توسط پرواز پرندگان گسسته برای نقطه هدف اول

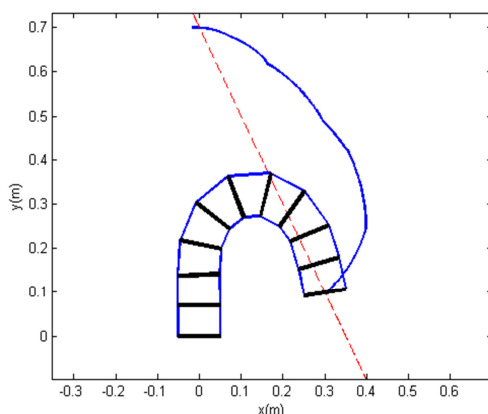
$$c_t = c_{min} + \frac{c_{max} - c_{min}}{total\ iteration} \times t \quad (23)$$

$$W = W_{max} - \frac{W_{max} - W_{min}}{total\ iteration} \times t \quad (24)$$

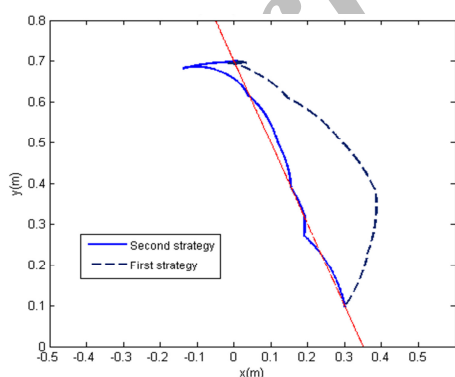
که در آن c_t مقدار ضریب در تکرار t ام است و c_{min} و c_{max} به ترتیب مقادیر آنها در تکرار اول و آخر می‌باشد. بعد از آزمایش و تکرارهای متوالی، مقادیر تجربی این ضرایب برابر $c_{1max} = c_{2max} = 0.9$, $c_{1min} = c_{2min} = 0.2$, $W_{max} = 0.6$ و $W_{min} = 0.9$ تعیین شده و تعداد جمعیت ذرات در پرواز پرندگان گسسته برابر ۴۰ و حداکثر تعداد تکرارها ۵۰ انتخاب شده است. جدول ۲ مقادیر مساحت بین مسیره‌ها برای ۳ نقطه هدف متفاوت نشان می‌دهد. برای نشان دادن توانایی و عملکرد الگوریتم پرواز پرندگان گسسته، نتایج آن با الگوریتم ژنتیک مقایسه شده است. برای داشتن یک مقایسه صحیح جمعیت اولیه الگوریتم ژنتیک برابر ۴۰، ضریب تولید نسل برابر ۰/۰۲ و حداکثر تولید نسل برابر ۵۰ انتخاب شده است. با توجه به نتایج جدول ۲ می‌توان مشاهده کرد که الگوریتم بهینه‌سازی پرواز پرندگان نتایج بهتری را ارائه می‌دهد. شکل‌های ۱۰ و ۱۱ به ترتیب ارزندگی بهترین ذره و میانگین ارزندگی ذرات را در الگوریتم در پرواز پرندگان گسسته و الگوریتم ژنتیک نشان می‌دهد. در شکل ۱۲ مسیر حرکت بازو برای داده‌های ردیف اول جدول ۲ نشان داده شده است. لازم به ذکر است که بازوی کابلی با مفصل قفل شونده، در حالتی که حق تکرار سوئیچینگ روی یک لینک را نداشته باشیم، توانایی حرکت دقیق بر روی مسیر پیش فرض را ندارد. زیرا در هر لحظه تنها یک لینک بازو می‌باشد. با این حال مسیر خط راست برای نشان دادن عملکرد الگوریتم پیشنهادی در نظر گرفته شده و در تحقیقات امکان تکرار سوئیچ روی یک لینک اضافه خواهد شد.



شکل ۱۴ مسیر حرکت بازو در الگوریتم پرواز پرنندگان هیبریدی برای نقطه هدف دوم



شکل ۱۵ مسیر حرکت بازو به دست آمده از استراتژی اول برای نقطه هدف دوم



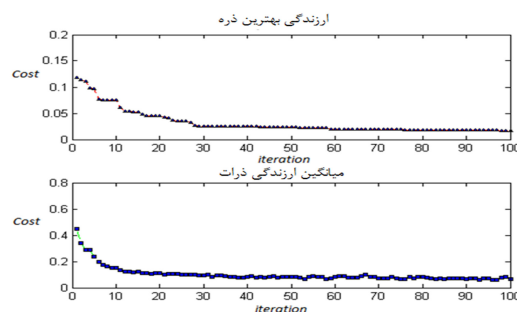
شکل ۱۶ مسیر حرکت بازو در استراتژی اول و استراتژی دوم (الگوریتم پرواز پرنندگان هیبریدی) برای نقطه هدف دوم

۲-۵- استراتژی دوم

برای اجرای الگوریتم پرواز پرنندگان هیبریدی تمامی پارامترها مشابه قبل می‌باشند مگر حداکثر تکرارها که برابر ۱۰۰ و جمعیت گروه ذرات که برابر ۷۰ انتخاب شده‌اند. علت افزایش جمعیت گروه آن است که دو وظیفه متفاوت به طور همزمان در یک گروه ذرات در حال انجام شدن می‌باشد. جدول ۳ حاوی نتایج بهینه‌سازی با این استراتژی برای همان سه نقطه هدف گذشته می‌باشد. با توجه به نتایج می‌توان مشاهده کرد که تابع هزینه مساحت در این روش بسیار کمتر از مقدار آن در روش قبلی می‌باشد. این تفاوت را می‌توان آشکارا در دومین (۰.۳، ۰.۱) و سومین (۰.۵، ۰.۲) نقطه هدف مشاهده کرد و همچنین زمان حل نیز به طور چشمگیری کاهش پیدا کرده است. شکل ۱۳ ارزیابی بهترین ذره و میانگین ارزیابی ذرات را در روش پرواز پرنندگان هیبریدی نشان می‌دهد. در شکل های ۱۴ و ۱۵ مسیر حرکت بازو برای استراتژی اول و دوم مربوط به دومین نقطه هدف نشان داده شده است. در شکل ۱۶ هر دو شکل به صورت یکجا نمایش داده شده‌اند تا مقایسه براحتی انجام شود. نهایتاً در جدول ۴ تمامی نتایج حاصل از بهینه‌سازی مربوط به نقطه هدف دوم (۰.۳، ۰.۱) نشان داده شده است. در این جدول، طول پایه‌ها بر حسب متر می‌باشد. در پایان مسیر حرکت بهینه‌سازی شده برای رسیدن به نقطه هدف سوم نشان داده شده تا عملکرد روش معرفی شده به روشنی نشان داده شود.

جدول ۳ نتایج بهینه‌سازی با الگوریتم پرواز پرنندگان هیبریدی

| نقطه هدف (X,Y) | پرواز پرنندگان هیبریدی | | |
|----------------|------------------------|-----------|----------|
| | Fun1 | Area cost | time |
| (۰/۱، ۰/۲) | ۰/۰۱۳۸ | ۰/۰۱۵۹ | ۲۷/۲۷۹۰۳ |
| (۰/۳، ۰/۱) | ۰/۰۰۳۳ | ۰/۰۱۹۲ | ۳۲/۶۱۱۹ |
| (۰/۵، ۰/۲) | ۰/۰۰۵۷ | ۰/۰۲۰۴ | ۳۲/۹۳۴۷ |



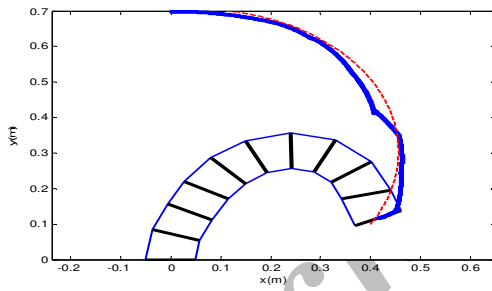
شکل ۱۳ ارزیابی بهترین ذره و میانگین ارزیابی ذرات در الگوریتم پرواز پرنندگان هیبریدی با تابع هزینه

می‌شود، این پارامترها عبارتند از نقطه شروع، نقطه پایانی و شعاع دایره. با مشخص کردن این پارامترها، کلیه نقاط مسیر معین خواهند شد. جدول زیر مشخصات دو مسیر متفاوت را نشان می‌دهد.

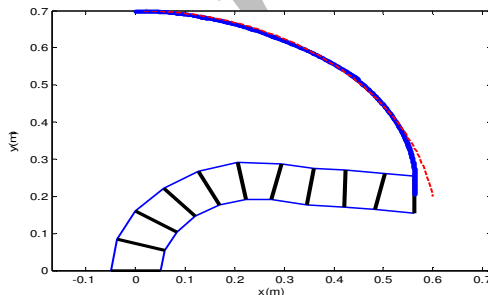
جدول ۵ مشخصات دو منحنی دایروی

| | شعاع دایره | نقطه شروع | نقطه پایانی |
|----------|------------|-----------|-------------|
| مسیر اول | ۰.۴ | (۰.۰.۷) | (۰.۴۰.۱) |
| مسیر دوم | ۰.۶ | (۰.۰.۷) | (۰.۶۰.۲) |

شکل‌های ۱۹ و ۲۰ مسیر حرکت بازو و منحنی مطلوب اول و دوم را نشان می‌دهد. در این شکل‌ها نتایج بهینه‌سازی به روش پرواز پرندگان هیبریدی با ۳۰ ذره بعد از ۵۰ تکرار ارائه شده است. فاصله مجری نهایی از نقطه هدف در انتهای حرکت بازو به ترتیب ۰.۰۲۲ و ۰.۰۳۸ متر است. همانطوری که در این شکل‌ها دیده می‌شود مسیر حرکت بازو عاری از حرکات رفت و برگشتی و زیگزاگی است. بطور کلی می‌توان گفت مسیره‌های دایروی نسبت به مسیره‌های خط راست ساده‌تر می‌باشند. ترتیب سوئیچینگ برای مسیر اول برابر [۹۷۸۱۰۳۱۶۵۲۴] و برای مسیر دوم برابر [۳۷۲۱۰۱۸۴۵۶۹] می‌باشد.



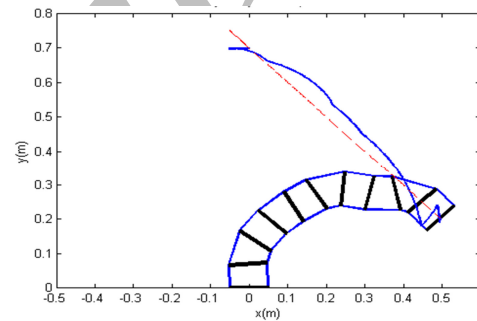
شکل ۱۹ مسیر حرکت بازو بر روی منحنی دایروی اول



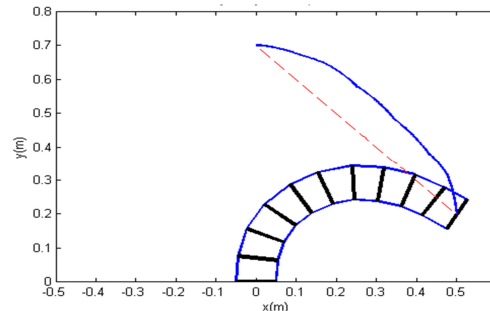
شکل ۲۰ مسیر حرکت بازو بر روی منحنی دایروی دوم

جدول ۴ نتایج بهینه‌سازی با استفاده از روش‌های مختلف برای نقطه هدف دوم

| پرواز پرندگان هیبریدی | الگوریتم ژنتیک گسسته | پرواز پرندگان گسسته | پرواز پرندگان پرنداری | پرواز پرندگان پیوسته |
|-----------------------|----------------------|---------------------|-----------------------|----------------------|
| طول پایه و سوئیچینگ | سوئیچینگ | سوئیچینگ | طول پایه | طول پایه |
| ۱۰ | ۱ | ۲ | ۰/۰۶۱۵ | ۰/۰۶۹۸ |
| ۲ | ۲ | ۸ | ۰/۱۱۴۳ | ۰/۰۶۸۶ |
| ۸ | ۸ | ۱ | ۰/۰۶۳۸ | ۰/۰۸۱ |
| ۹ | ۷ | ۷ | ۰/۰۵۵۹ | ۰/۰۹۲۴ |
| ۷ | ۵ | ۹ | ۰/۱۰۰۲ | ۰/۰۹۷۲ |
| ۶ | ۳ | ۱۰ | ۰/۱۰۱۳ | ۰/۱۰۱۱ |
| ۴ | ۴ | ۵ | ۰/۱۰۸۵ | ۰/۰۸۹۷ |
| ۵ | ۱۰ | ۳ | ۰/۱۰۲۸ | ۰/۰۹۶۸ |
| ۱ | ۹ | ۲ | ۰/۰۷۷۸ | ۰/۰۷۵۱ |
| ۳ | ۶ | ۶ | ۰/۰۵۴۱ | ۰/۰۷۶۳ |



شکل ۱۷ مسیر حرکت بازو در الگوریتم پرواز پرندگان هیبریدی برای نقطه هدف سوم



شکل ۱۸ مسیر حرکت بازو به‌دست آمده از استراتژی اول برای نقطه هدف سوم

همانطوری که اشاره شد مسیر خط راست، مسیری نسبتاً پیچیده برای این بازو محسوب می‌شود به همین دلیل در این بخش قطاعی از دایره به‌عنوان مسیر مطلوب انتخاب شده است تا توانایی الگوریتم پیشنهادی مشخص‌تر شود. مسیر مورد نظر با چند پارامتر مشخص

۶- نتیجه گیری

در این مقاله، الگوریتم بهینه‌سازی پرواز پرندگان جهت طراحی مسیر یک بازوی افزونه بکار گرفته شده است. بازوی مارگون مورد مطالعه صفحه‌ای و دارای سیستم قفل بوده و تاکنون تحقیقی در مورد طراحی مسیر این نوع از بازوها صورت نگرفته است. دو استراتژی مختلف برای طراحی مسیر بررسی شده است. در استراتژی اول، حل سینماتیک معکوس از بهینه‌سازی ترتیب بازو و بسته شدن قفلها جدا شده تا فرامین گسسته و پیوسته مسئله بطور مجزا بهینه‌سازی شوند. اما در استراتژی دوم، با معرفی روش جدید پرواز پرندگان هیبریدی، مقدار تغییر طول پایه‌ها و ترتیب سوئیچینگ‌ها بطور همزمان بگونه‌ای پیدا شدند که بازو بر روی مسیر مستقیم حرکت کند. همچنین نتایج حاصل از بهینه‌سازی با الگوریتم پرواز پرندگان گسسته با نتایج بهینه‌سازی توسط الگوریتم ژنتیک مقایسه شده است.

۷- مراجع

- workspace," Journal of Robotic Systems, vol. 16, 1999, pp. 339-351.
- [8] Sciavicco L. and Siciliano B., "A solution algorithm to the inverse kinematic problem for redundant manipulators," Robotics and Automation, IEEE Journal of, vol. 4, 2002, pp. 403-410.
- [9] Zacharia P. T. and Aspragathos N. A., "Optimal robot task scheduling based on genetic algorithms," Robotics and Computer-Integrated Manufacturing, vol. 21, 2005, pp. 67-79.
- [10] Xidias E. K., Zacharia P. T., et al., "Time-optimal task scheduling for articulated manipulators in environments cluttered with obstacles," Robotica, vol. 28, 2010, pp. 427-440.
- [11] da Graça Marcos M., Tenreiro Machado J. A., et al., "Trajectory planning of redundant manipulators using genetic algorithms," Communications in Nonlinear Science and Numerical Simulation, vol. 14, 2009, pp. 2858-2869.
- [12] Khoukhi A., Baron L., et al., "Multi-Objective Trajectory Planning for Redundant Manipulators," 2007, pp. 11-13.
- [13] Kim J. H., Yang J., et al., "A novel formulation for determining joint constraint loads during optimal dynamic motion of redundant manipulators in DH representation," Multibody System Dynamics, vol. 19, 2008, pp. 427-451.
- [14] Khoukhi A., Baron L., et al., "Hybrid neuro-fuzzy multi-objective trajectory planning of redundant manipulators," Control and Intelligent Systems, vol. 37, 2009, pp. 87-96.
- [15] هنرور م., "طراحی، شبیه‌سازی و کنترل یک بازوی مارگون کابلی با مفاصل قفل شونده." پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شریف، ۱۳۸۸.
- [16] Kennedy J. and Eberhart R., "Particle swarm optimization," in Proc. IEEE International Conf. on Neural Networks, 1995, pp. 1942-1948 vol. 4.
- [17] Shi X., Liang Y., et al., "Particle swarm optimization-based algorithms for TSP and generalized TSP," Information Processing Letters, vol. 103, 2007, pp. 169-176.
- [18] Khanesar M. A., Teshnehlab M., et al., "A novel binary particle swarm optimization," in Mediterranean Conference on Control & Automation, Athens 2007, pp. 1-6.
- [19] Babazadeh D., Boroushaki M., et al., "Optimization of fuel core loading pattern design in a VVER nuclear power reactors using Particle Swarm Optimization (PSO)," Annals of Nuclear Energy, vol. 36, 2009, pp. 923-930.
- [1] Choset H. and Henning W., "A follow the leader approach to serpentine robot motion planning," ASCE Journal of Aerospace Engineering, vol. 12, 1999,
- [2] Zhang Y., Wang J., et al., "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," Neural Networks, IEEE Transactions on, vol. 14, 2003, pp. 658-667.
- [3] Conkur E. S., "Path following algorithm for highly redundant manipulators," Robotics and Autonomous Systems, vol. 45, 2003, pp. 1-22.
- [4] Ata A. and Myo T., "Optimal point-to-point trajectory tracking of redundant manipulators using generalized pattern search," Computing Research Repository, vol. 60, 2006,
- [5] Jia Q. X., Chu M., et al., "Research on the optimal algorithm for trajectory planning of a 9-DOF hyper-redundant robot," Journal of Beijing University of Posts and Telecommunications, vol. 31, 2008, pp. 20-25.
- [6] Cai B. and Zhang Y., "Bi-criteria optimal control of redundant robot manipulators using LVI-based primal-dual neural network," Optimal Control Applications and Methods, vol. 31, 2010, pp. 213-229.
- [7] Liang T. C. and Liu J. S., "Improved trajectory planner for redundant manipulators in constrained