

برنامه ریزی (Planning) مسیر ربات با استفاده از الگوریتم مبتنی بر

نمونه گیری *ARRT

چکیده

پیشرفت در ساخت تجهیزات دقیق و اطمینان از صحت عملکرد ربات ها، موجب افزایش تمایل انسان به استفاده از سیستم های رباتیکی شده است. هدف اصلی در برنامه ریزی مسیر ، یافتن مسیر حرکتی برای ربات از نقطه شروع تا هدف، بدون برخورد با موانع و با کمترین در سال است. هزینه های اخیر، الگوریتم های مبتنی بر نمونه گیری مورد توجه بسیاری از پژوهشگران قرار گرفته است. در این گزارش الگوریتم مبتنی بر نمونه گیری جدیدی برای یافتن مسیر ربات ارائه شده است. الگوریتم جدید با استفاده از سه استراتژی ، به متفاوت نمونه برداری از محیط می پردازد. به روز شدن میزان سوگیری نسبت به هدف و سوگیری نسبت به جستجوی محیط های کشف نشده در هر مرحله، موجب عملکرد مناسب این روش در محیط های مختلف شده است. آزمایشات مختلف نشان می دهند که الگوریتم *ARRT، در مقایسه با روش های RRT و *RRT و RRT* توسعه یافته، در یافتن مسیر کوتاه تر در محیط، موفق تر عمل میکند.

1- مقدمه

امروزه با ساخت ربات های مختلف، مسئله برنامه ریزی مسیر ربات، بسیار مورد توجه قرار گرفته است. به طور خلاصه، مسئله برنامه ریزی مسیر ، یافتن مسیری است که ربات را از نقطه شروع به نقطه هدف هدایت می کند. در بسیاری از کاربردها برای اینکه ربات بتواند وظیفه خود را بدرستی انجام دهد، نیاز به حرکت در محیط دارد. الزام حرکت در محیط موجب پرسیدن این سوال می شود که، ربات برای انجام وظیفه خود چه مسیری را طی کند که علاوه بر ایمن بودن آن، با کمترین هزینه به هدف خود برسد. با توجه به محدودیت انرژی، تسريع در انجام کارها و عدم برخورد ربات با موانع موجود در محیط، اهمیت این سوال بهتر درک می شود. پاسخگویی به این پرسش، که پرسش اساسی در زمینه تحقیقاتی مهمی تحت عنوان برنامه ریزی مسیر است، از دیدگاه های

مختلف صورت گرفته است. رویکرد های حل مسئله برنامه ریزی مسیر را میتوان به چند دسته ۱ کلی تقسیم کرد: ۱) بلادرنگ اگر ربات از وقایع محیطی، مکان و حرکت موانع، از قبل اطلاع نداشته باشد، آنگاه به یافتن مسیری ایمن به سوی هدف مبادرت کند، در این حالت برنامه ریزی بلادرنگ نامیده می شود. ۲) کامل و کامل احتمالاتی: ارائه یک مسیر، در صورت وجود، کامل بودن الگوریتم نامیده می شود. کامل بودن احتمالاتی نیز به این معنی است که اگر الگوریتم تا زمان طولانی اجرا شود، در صورت وجود یک مسیر، آن را خواهد یافت. از جمله روش های مطرح برای حل مسئله برنامه ریزی مسیر، الگوریتم های مبتنی بر نمونه گیری هستند. در این روش ها با نمونه برداری از فضای پیکربندی، به ساخت درختی از نقطه شروع، مبادرت می شود. باش رسیدن یکی از شاخه های درخت به نقطه هدف الگوریتم خاتمه می یابد. این روش در ابتدا توسط Lavalle در سال ۱۹۹۱ تحت عنوان درخت جستجوی سریع تصادفی (RRT) مطرح گردید. در این گزارش روش نمونه گیری جدیدی برای هدایت درخت جستجوی تصادفی به سوی هدف ارائه می شود. این روش با استفاده از سه استراتژی و با تعریف نرخ نمونه برداری و به روز شدن آن در هر مرحله، سعی در ایجاد تعادل بین میزان جستجوی فضای اطراف هدف و جستجوی فضاهای کشف نشده برقرار کند. با این شیوه نمونه برداری از سوگیری ثابت نسبت به هدف و جستجوی کاملاً تصادفی فضاهای کشف نشده جلوگیری به عمل می آید. سوگیری نسبت به هدف به این معناست که درخت تمایل به حرکت به سوی هدف با نمونه برداری از نواحی اطراف هدف دارد. سوگیری نسبت به جستجوی محیطه ای کشف نشده هم بدین معنا است که درخت تمایل به رشد در تمامی محیط دارد. در این گزارش پس از بررسی الگوریتم های برنامه ریزی مسیر در بخش ۲، در بخش ۳ به ارائه روش جدید مبتنی بر نمونه گیری پرداخته می شود. انجام شبیه سازی و بررسی عملکرد روش ارائه شده در این گزارش در مقایسه با دیگر روش های مطرح در بخش ۴ صورت می پذیرد. سرانجام در بخش ۵، نتیجه گیری و ارائه راه حل ها برای کارهای آتی، توضیح داده میشوند

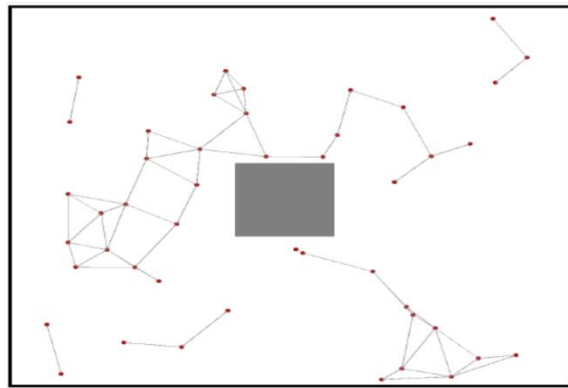
2- مروری بر پژوهش های گذشته

تحقیقات انجام شده در زمینه برنامه ریزی مسیر، که حاصل تلاش پژوهشگران بسیاری با دیدگاه های متفاوت است، به دسته های مختلفی تقسیم می شود ۱) روش حشره: در این روش ربات از موقعیت هدف آگاه بوده و در هر لحظه بنا بر درک موانع موجود در محیط قادر به تصمیم گیری است. اساس این الگوریتم حرکت مستقیم به سوی هدف و دور زدن موانع است. توسعه های مختلفی از این الگوریتم، نظیر الگوریتم های حشره ۱، حشره ۲، حشره مماسی و حشره برداری ارائه شده است. ۲) میدان پتانسیل: در این روش با تعریف یک میدان پتانسیل مجازی برای محیط، که دارای مینیمم در نقطه هدف است، سعی در هدایت ربات به سوی هدف می شود. بنابر موقعیت موانع و ربات، ممکن است در نقاطی به غیر از هدف نیز میدان پتانسیل صفر گردد. این موضوع تحت عنوان مینیمم محلی میدان پتانسیل مطرح شده و از ضعف های اساسی این روش است. در رویکرد جدید، روش

میدان پتانسیل مبتنی بر منطق فازی ارائه شده است. (۳) روش های تکاملی: پژوهشگران هوش مصنوعی نیز رباتها را عامل هوشمند در نظر گرفته و سعی در حل این مسئله با استفاده از روش های پردازش تکاملی نظیر GA داشته‌اند. نویسنده در با استفاده از روش های پردازش تکاملی، مسئله برنامه ریزی مسیر را مورد بررسی قرار داده است. (۴) نقشه راه دقیق: روش مطرح دیگر، روش نقشه راه است. در این روش، ابتدا سعی می شود که در نقشه محیط شاهراهایی رسم شوند، سپس با اتصال نقاط شروع و پایان به این شاهراه، مسیری امن برای ربات بدست آورد. در روش نقشه راه دقیق، شاهراهها با استفاده از دو روش نمودار ورونوی و گراف دیدنگار ساخته میشوند. مسیر بدست آمده از روش گراف دیدنگار، کوتاهترین مسیر است. اما به دلیل نزدیکی بیش از حد مسیر به مرز موانع امنیت بسیار پایینی داشته و در محیط هایی با ابعاد بالا دارای پیچیدگی محاسباتی بسیاری است. استفاده از نمودار ورونوی، روش دیگری برای ساخت شاهراه باشد. نمودار ورونوی بر اساس فاصله بین دو مانع ساخته می شود. با توجه به تعریف این نمودار، مسیر حاصل تا حد ممکن از مرز موانع دور است، لذا این مسیر بهینه نیست. (۵) روش های جستجوی تصادفی: یکی از مهمترین روش های مطرح در برنامه‌ریزی مسیر، استفاده از جستجوی تصادفی است. مهم ترین جستجوی تصادفی که در سالهای اخیر توجه بسیاری از پژوهشگران را به خود معطوف کرده است، روش نقشه‌راه احتمالاتی (PRM) است. این روش در سال ۱۹۹۶ توسط kavradi و همکارانش ارائه شده است. در سال ۱۹۹۱، Lavalley، با معرفی درخت جستجوی تصادفی RRT رویکرد جدیدی در استفاده از جستجوی تصادفی ارائه کرد. روش RRT با استفاده از ساخت درخت با شاخه‌های تصادفی در محیط، سعی در حل مسئله برنامه ریزی مسیر دارد. الگوریتم های ارائه شده توسط این افراد به الگوریتم های مبتنی بر نمونه گیری شهرت دارند. پاسخگویی در زمان کمتر، عدم نیاز به تعریف دقیق فضا و قابلیت بکارگیری برای انواع مختلف رباتها، موجب شده است که این الگوریتم ها از جمله محبوبترین روش های مطرح در برنامه‌ریزی مسیر باشند.

2-1- نقشه راه احتمالاتی (PRM)

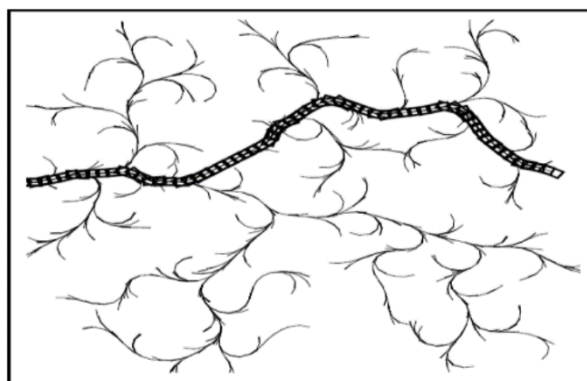
این الگوریتم در دو فاز به حل مسئله برنامه ریزی مسیر می‌پردازد. در فاز اول که فاز یادگیری خوانده می شود، با استفاده از نمونه برداری با توزیع یکنواخت، نقاطی از فضای پیکربندی آزاد انتخاب می شود. اتصال بین نمونه ها موجب ایجاد یک شاهراه در فضا می شود. پس از ساخت نقشه راه، در فاز دوم که فاز سوال می شود، سعی می شود که پیکربندی اولیه و هدف نیز به نقشه‌راه متصل گردند. با این اتصالات نقشه‌راه کامل میگردد. در سالهای اخیر توسعه های فراوانی برای بهبود این روش ارائه شده است. نمونه‌ای از مراحل ساخت نقشه-راه با استفاده از روش PRM در شکل (1) نمایش داده شده است.



شکل 1- ساخت نقشه راه به روش PRM در فاز یادگیری.

2-2- درخت جستجوی تصادفی (RRT)

این الگوریتم نیز با استفاده از جستجوی تصادفی فضای پیکربندی به حل مسئله برنامه ریزی مسیری می‌پردازد. در این روش، ریشه درخت در پیکربندی اولیه قرار دارد. ابتدا یک نقطه به صورت تصادفی با توزیع یکنواخت انتخاب می‌شود. اگر نقطه انتخاب شده به فضای موانع تعلق داشته باشد، نقطه دیگری انتخاب می‌شود. با انتخاب نقطه از فضای آزاد، در راستای خط واصل بین این نقطه و ریشه درخت، نقطه جدیدی تحت عنوان x_{new} که به اندازه گام ربات از ریشه فاصله دارد، انتخاب می‌شود. اگر بتوان x_{new} را با پاره خطی که تماماً به فضای آزاد تعلق دارد، به ریشه وصل کرد، x_{new} به عنوان گره جدید به درخت اضافه می‌شود. نمایی از درخت توسعه یافته با این روش در شکل (2) نمایش داده شده است.



شکل 2- درخت جستجوی تصادفی RRT.

2-3- روش RRT^*

روش RRT^* با روندی مشابه روش RRT از محیط نمونه برداری می کند. اما تفاوت آن با روش RRT در شیوه توسعه درخت جستجوی تصادفی است. پس از نمونه برداری کاملاً تصادفی از محیط، در صورت معتبر بودن نقطه جدید، نقطه به درخت اضافه می شود. حال با در نظر گرفتن دیسکی به مرکز گره جدید، گره‌هایی از درخت که هزینه آنها بیشتر از هزینه اتصال آنها به گره جدید است، مسیرشان تصحیح می شود. مسیر بدست آمده با اجرای این روش، دارای مسافت کوتاهتری نسبت به سایر روشها است. توسعه‌ای که به روش RRT^* داده شده است، روش RRT^* توسعه‌یافته است. در این روش، در هر تکرار الگوریتم، یک متغیر تصادفی تولید شده و بنا بر مقدار آن تصمیم گرفته می شود که مستقیم به سوی هدف حرکت کرده، و یا به صورت تصادفی به جستجوی فضا پرداخته شود. ضعف اساسی این روش این است که سوگیری نسبت به هدف به صورت تصادفی و بدون توجه به نوع محیط صورت می‌گیرد.

3- روش درخت جستجوی سریع تصادفی با زاویه ($ARRT^*$)

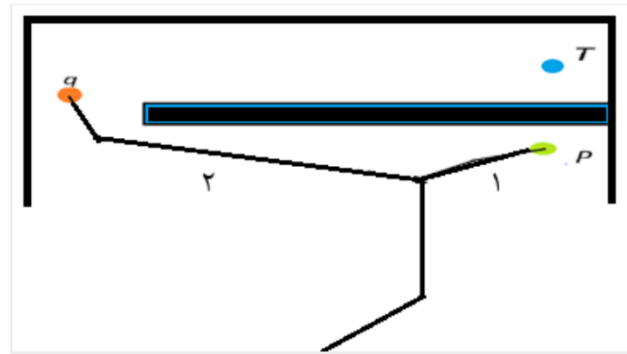
سوگیری نسبت به جستجوی محیطهای ناشناخته موجب رهایی از بهینه محلی می شود، البته مسیر بدست آمده نزدیک به بهینه نیست. از طرفی سوگیری نسبت به هدف نیز میتواند کوتاهی مسیر را در پی داشته باشد. استفاده از این دو سوگیری به صورت مناسب، میتواند مسیر حاصله را به بهینگی نزدیک کند. در این روش از سه استراتژی نمونه برداری به صورت زیر استفاده می شود .

1. اتصال آخرین گره درخت به هدف
2. نمونه برداری از قوس فرضی با شعاع $|ST|$ و زاویه α
3. انتخاب تصادفی از کل فضا

3-1- اتصال آخرین گره درخت به هدف

در روش های ذکر شده، در صورت انتخاب هدف به عنوان نقطه تصادفی، نزدیکترین گره درخت را یافته و سپس سعی می شود آنها را به هم وصل کند. این امر موجب می‌گردد که در وضعیتی نظیر شکل (3)، درخت رشد مناسبی نداشته باشد. عملکرد روش های پیشین بدین صورت است که با انتخاب هدف (T) به عنوان نقطه تصادفی، سعی می شود نزدیکترین گره درخت یعنی P را به T وصل کند، که این کار امکانپذیر نیست. لذا شاخه 2 که دارای شرایط بهتری برای رشد درخت است، توسعه نمی‌یابد. انتخاب آخرین گره درخت برای اتصال

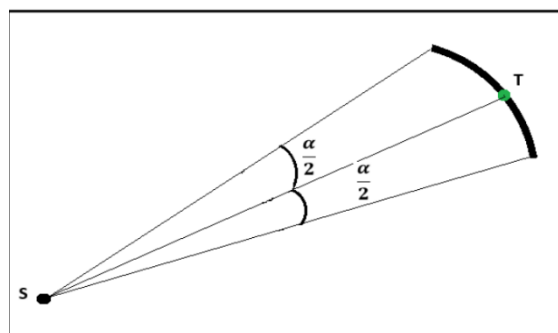
به هدف در صورت امکان، موجب ایجاد مسیری مستقیم به سوی هدف شده و از جستجوی درخت اجتناب می کند. این استراتژی علاوه بر کاهش طول مسیر، موجب کاهش زمان اجرای الگوریتم میگردد. لذا با استفاده از استراتژی اول، با انتخاب نقطه q و اتصال آن به T شاخه 2 به سوی هدف رشد می کند.



شکل 3- تصحیح مسیر با استفاده از استراتژی اول.

3-2- نمونه برداری از قوس فرضی با شعاع $|ST|$ و زاویه α

در این استراتژی یک قوس فرضی به مرکز نقطه شروع و شعاعی به طول فاصله نقطه شروع S تا نقطه هدف T ($|ST|$) و زاویه α مانند شکل 4 در نظر گرفته شده و نقاط تصادفی از روی این قوس انتخاب می شود. سپس به اندازه گام ربات به سوی این نقطه تصادفی حرکت می شود. اگر نقطه جدید به فضای آزاد تعلق داشته باشد، به درخت اضافه می شود. مقدار زاویه α در شروع الگوریتم یک مقدار پیش فرض مانند $\pi/4$ است، ولی در طول اجرای الگوریتم این مقدار بنا بر نرخ موفقیت میتواند، بزرگتر و یا کوچکتر شود.



شکل 4- تعیین کمان به مرکز S و شعاع $|ST|$ با زاویه α .

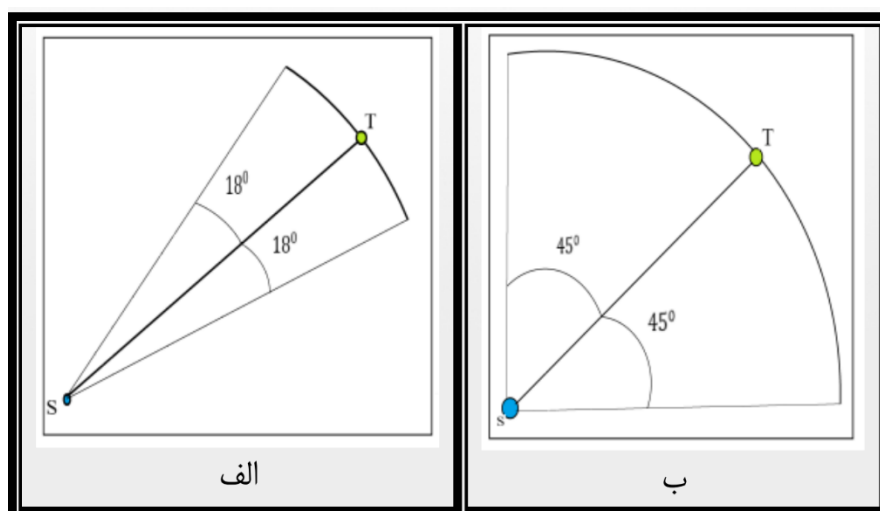
نرخ موفقیت بنا بر تعداد نمونه برداری موفق از قوس تعیین می شود. Na تعداد نمونه برداری مجاز با استفاده از این استراتژی است که در ابتدا تعیین می شود (مثلا $Na = 5$)، Ns تعداد نمونه های انتخاب شده تا رسیدن به یک نقطه تصادفی جدید است که با استفاده از این نقطه تصادفی بتوان گره معتبری برای اتصال به درخت ایجاد کرد. α زاویه قوس است که مقدار اولیه آن به صورت دستی و مقادیر به روز شده آن در طول الگوریتم، بر حسب نرخ موفقیت محاسبه می شود. مقدار نرخ موفقیت η به صورت زیر تعریف می شود:

$$\eta = 1 - Ns / Na$$

آشکار است که اگر بتوان آخرین گره را به هدف وصل کرد نرخ موفقیت برابر یک است و اگر تعداد نمونه برداری با این استراتژی برابر Na باشد ($Na = Ns$) نرخ موفقیت برابر صفر خواهد بود. حال با استفاده از تعریف نرخ موفقیت در گامهای دوم به بعد مقدار α را برای مرحله بعد با فرمول زیر به روز می شود.

$$\alpha = \pi * (1 - \eta)$$

این تغییر مقدار α متناسب با نرخ موفقیت، تغییر مقدار قوس فرضی را در پی دارد که موجب کاهش سوگیری نسبت به هدف و افزایش سوگیری نسبت به جستجوی محیط های ناشناخته می گردد. دقت شود که مقدار η در هر مرحله به بازه $[0,1]$ تعلق دارد. برای درک بهتر این استراتژی به مثال زیر توجه کنید. مثال: فرض کنید مقدار اولیه α 36 درجه و مقدار N برابر 10 باشد. اگر بعد از نمونه برداری با این استراتژی، مقدار Ns برابر 5 شود. یعنی بعد از 4 مرتبه نمونه برداری ناموفق از قوس، در مرتبه پنجم نمونه معتبری برای اتصال به درخت انتخاب گردید. آنگاه مقدار η با توجه به فرمول (1) برابر $\eta = (1 - 5/10) = 0.5$ است. بنابراین مقدار زاویه قوس برای مرحله بعد برابر $\alpha = 180 * (1 - 0.5) = 90$ است. (شکل 5)



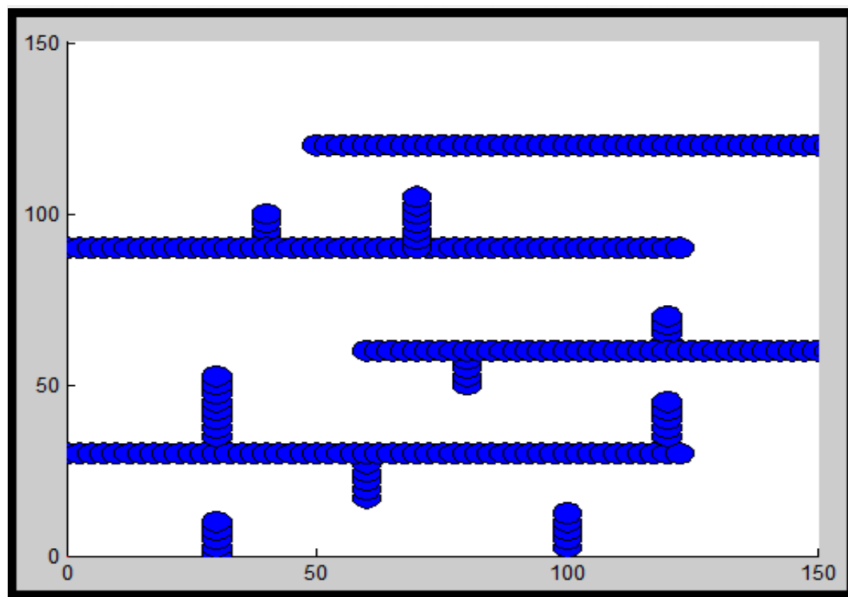
شکل 5- (الف) زاویه اولیه قوس، (ب) تصحیح زاویه قوس پس از نمونه برداری

3-3- نمونه برداری از انتخاب تصادفی از کل فضا

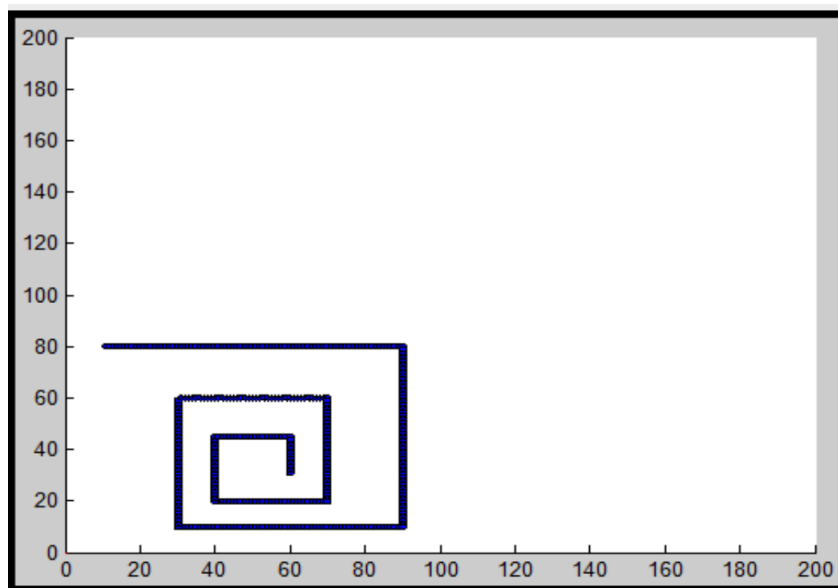
اگر در استراتژی قبل $Na = Ns$ شود و نمونه معتبری از فضا انتخاب نشود، آنگاه از این استراتژی استفاده می شود. در این استراتژی نمونه برداری مانند روش RRT، به صورت کاملاً یکنواخت از محیط انجام میگیرد. اکنون پس از تعیین یک نقطه معتبر از فضا برای اتصال آن به درخت، از روش ارائه شده در الگوریتم RRT^* برای انتخاب گره والد استفاده می شود. در این روش پس از تعیین مقدار شعاع دیسک بر اساس روش RRT^* بنابر گام ربات و پارامترهای فضای پیکربندی، یک دایره فرضی حول نقطه جدید در نظر گرفته می شود. حال نقطه جدید به گره های از درخت وصل می شود که دارای کمترین هزینه باشد.

4- آزمایشات

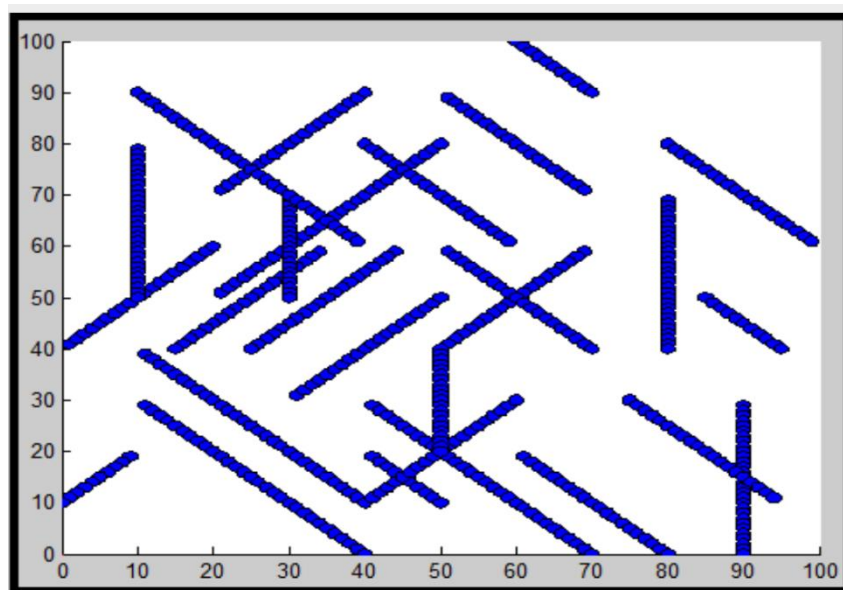
برای بررسی عملکرد روش $ARRT^*$ ارائه شده در این گزارش، چهار محیط مختلف در نظر گرفته شده است. روند انجام آزمایشات بدین صورت است که یک محیط ثابت در نظر گرفته می شود و چهار الگوریتم RRT و RRT^* و RRT^* توسعه یافته و $ARRT^*$ در 20 مرتبه متفاوت اجرا شده و نتایج ثبت شده اند. از میان نتایج بدست آمده باید به کوتاه ترین مسیر بدست آمده و میانگین طول مسیرها توجه کرد. دقت شود که در الگوریتم های تصادفی مقدار میانگین مسافت مسیر، که به روند رشد درخت مربوط است از اهمیت ویژه ای برخوردار است. این آزمایشات در محیط چهار مختلف که در شکل های 6 و 7 و 8 و 9 دیده می شود، انجام شده اند. محیط آزمایش اول، نمونه ای از محیط های ساختاریافته است. هدف از انجام این آزمایش بررسی رفتار الگوریتم ها در محیط های منظم است. هدف از انجام آزمایش دوم بررسی عملکرد الگوریتم ها در محیطی شبه هزارتو است. این آزمایش برای نشان دادن تاثیر سوگیری هدف نسبت به هدف در ساخت مسیر، انجام گرفته است. هدف از انجام آزمایش سوم، بررسی عملکرد الگوریتم ها در محیط غیرساختاریافته و پیچیده است. محیط آزمایش چهارم متشکل از موانع چندضلعی مختلف است. از نتایج حاصل اجرای آزمایشات در جداول 1 و 2 نمایش داده شده اند.



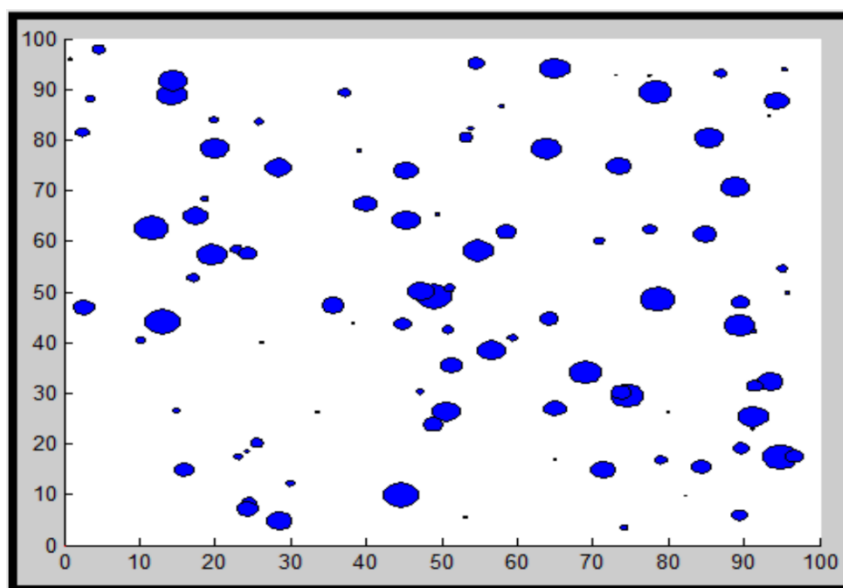
شکل 6- محیط آزمایش اول.



شکل 7- محیط آزمایش دوم.



شکل 8- محیط آزمایش سوم.



شکل 9- محیط آزمایش چهارم.

آزمایش				الگوریتم
اول	دوم	سوم	چهارم	
۶۸۱/۷	۵۹۷/۲	۲۳۸/۲	۸۵۲/۱	RRT
۶۷۷/۷	۵۹۵	۲۲۴/۲	۸۱۰/۷	RRT*
۶۷۶	۵۸۰/۵	۲۱۵/۲	۷۷۱	RRT* توسعه یافته
۶۴۹/۵	۵۳۹/۵	۲۱۴/۲	۶۹۰/۶	ARRT*

جدول 1: کوتاه ترین مسیر بدست آمده از انجام آزمایشات.

آزمایش				الگوریتم
اول	دوم	سوم	چهارم	
۲۱۰	۴۹۵	۶۳۰	۸۰۰	RRT
۲۰۵	۴۹۰	۶۳۰	۷۶۰	RRT*
۲۰۵	۴۷۵	۶۴۰	۷۲۰	RRT* توسعه یافته
۱۹۵	۴۱۵	۶۰۰	۶۱۵	ARRT*

جدول 2: میانگین طول مسیرهای بدست آمده از انجام آزمایشات.

با بررسی نتایج آزمایشات میتوان این طور نتیجه گرفت که الگوریتم * ARRT با تغییر مقدار سوگیری نسبت به هدف و جستجوی فضاهای کشف نشده، میتواند در محیط های مختلف عملکرد قابل قبولی داشته باشد. در هر کدام از سه آزمایش الگوریتم * ARRT مسیری کوتاهتر از سایر الگوریتم ها بدست آورده است. البته شایان توجه است که میانگین طول مسیر بدست آمده در 20 اجرای متفاوت و مستقل الگوریتم ها در سه آزمایش، کمترین میانگین طول مسیر به الگوریتم * ARRT تعلق دارد.

5- نتیجه گیری

در این گزارش الگوریتم مبتنی گیری بر نمونه * ARRT که با توسعه الگوریتم * RRT به یافتن کوتاهترین مسیر میپردازد، ارائه گردید. این روش با استفاده از سه استراتژی مختلف به نمونه برداری از فضای مسئله میپردازد. در استراتژی اول سعی در اتصال آخرین گره درخت به هدف دارد. در استراتژی دوم با تعریف نرخ موفقیت نمونه برداری، تعادل مناسبی میان سوگیری نسبت به هدف و سوگیری نسبت به جستجوی محیطهای کشف نشده، برقرار می کند. در استراتژی سوم، در صورت موفق نبودن دو استراتژی قبل به نمونه برداری کاملاً تصادفی از

محیط مبادرت میوزرد. انجام آزمایشات مختلف نشان میدهند الگوریتم ، $ARRT^*$ در مقایسه با روش های RRT و RRT^* توسعه یافته، در یافتن مسیر کوتاهتر در محیط، موفقتر عمل می کند.

منابع

- [1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun; Principles of robot motion planning: theory, algorithms, and implementation, MIT Press, 2005.
- [2] J. V. D. Berg, M. Lin, D. Manocha, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation", IEEE International Conference on Robotics and Automation (ICRA), pp. 1928-1935, 2008.
- [3] S. Karaman, E. Frazzoli, "Incremental Sampling-based Algorithms for Optimal Motion Planning", Computing Research Repository - CORR , vol. abs/1005.0, 2010.
- [4] S.M., Lavalley, "Rapidly-Exploring Random Tree : A New Tool for Path Planning ", pp. 1-4 ,1998. URL:<http://msl.cs.uiuc.edu/~lavalley/rtrpubs.html>.
- [5] Q. Xu, G. Tang, "Vectorization path planning for autonomous mobile agent in unknown environment", Neural Computing and Applications, Volume 23 , pp. 2129–2135, 2013.
- [6] Y. Koren, J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation", IEEE International Conference on Robotics and Automation (ICRA), pp.1398-1404, 1991.
- [7] M.A. Jaradat, M. Garibeh, E. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field", Soft Computing, Volume 16 , pp. 153–164, 2012.
- [8] R., Kala, "Multi-robot path planning using co-evolutionary genetic programming", Expert Systems with Applications, Volume 39, Issue 3, pp. 3817-3831, 15 February ,2012.
- [9] T. Asano, L. Guibas, J. Hersberger, and H. Imai, "Visibility-polygon search and euclidean shortest paths", in Proc. 26th Annu. Symp. Found. Comput. Sci., pp. 155-164, 1985.
- [10] J. Canny, "A Voronoi method for the piano-movers problem", in Proc. IEEE Int. Conf. Robot. Autom., pp. 530-535, 1985.
- [11] L.E Kavraki, P. Svestka, J-C. Latombe, M.H. Overmars, "Probabilistic roadmaps for path planning in highdimensional configuration spaces", Robotics and Automation, Volume:12 Issue:4, pp. 566-580, 1996.
- [12] M. Elbanhawi, M. Simic, "Sampling-Based Robot Motion planning :A Review", IEEE Access, Volume 2, pp. 56-77, 2014.
- [13] T. Simeon, J-P. Laumond, and C. Nissoux, "Visibility-based probabilistic roadmaps for motion planning", Journal of Advanced Robotics, volume 14, pp. 352-361, 2000.
- [14] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning", International Journal of Robotics Research, vol. 30, iss. 7, pp. 846-894, 2011.
- [15] L. Yang, C. Rongxin, Y. Chenguang, X. Demin, "MultiRobot Path Planning Based on the Developed RRT^* Algorithm", 32nd Chinese control conference, pp. 7049- 7053, 2013