



مسیریابی حرکت روبات‌های ماشین‌واره با روش پیشروی سریع

داود جنت و الپس مسیحی*

چکیده:

کلمات کلیدی

مسیریابی روبات عبارتست از یافتن یک توالی حرکت برای روباتی که در محیطی با اشیای ثابت یا متحرک از یک نقطه آغازین شروع شده و بدون برخورد با موانع به یک نقطه پایانی ختم می‌گردد. چنین مسئله‌ای در مسیریابی خودروهای هوشمند و AGV ها نیز مطرح می‌شود. این مبحث از منظر محدودیت‌های سینماتیکی به دو دسته هولونومیک و غیرهولونومیک تقسیم می‌شود. در ناحیه عاری از مانع، در حالت اول روبات آزادانه می‌تواند از هر نقطه به نقطه دیگر حرکت کند اما در حالت دوم به دلیل وجود محدودیت‌های سینماتیکی دامنه حرکتی روبات محدودتر می‌شود، به عنوان مثال یک ماشین نمی‌تواند به پهلو حرکت کند. در این مقاله به وسیله روش پیشروی سریع - که یک روش حل عددی برای معادله دیفرانسیل جزئی غیر خطی Eikonal است - بدون در نظر گرفتن محدودیت غیرهولونومیک روبات ماشین‌واره، مسیری ایمن و هموار ایجاد می‌شود و سپس با اصلاح مسیر به دست آمده توسط مفهوم جدیدی به نام مانع مجازی و تطبیق آن با محدودیت غیرهولونومیک روبات ماشین‌واره، مسیری جهت حرکت آن برنامه ریزی می‌شود. روش ارائه شده، کوتاه‌ترین مسیر ممکن را پیدا کرده و در عین حال دقیق، سریع و مستقل از شکل موانع است. نتایج مقایسات این روش با روش جستجوی گراف برتری آن را از نظر طول مسیر و زمان حل نشان داده است.

روبات متحرک،
روبات ماشین‌واره،
مسیریابی روبات،
پیشروی سریع، مانع مجازی

۱. مقدمه

هر چند الگوریتم‌های برنامه‌ریزی حرکت از نظر جزئیات تفاوت‌های زیادی با هم دارند اما در بیشتر آن‌ها یک چارچوب کلی وجود دارد. در گام اول روبات را که ممکن است دارای یک شکل پیچیده هندسی باشد توسط روش مینکوفسکی به یک نقطه تبدیل می‌کنند و فضای موجود نیز به یک فضا به نام فضای پیکربندی تبدیل می‌شود.

مسیریابی روبات عبارت است از یافتن یک توالی حرکت برای روباتی که در محیطی با اشیای ثابت یا متحرک از یک نقطه آغازین شروع شده و بدون برخورد با موانع به یک نقطه پایانی ختم می‌گردد. در مسائل برنامه‌ریزی حرکت توصیف کاملی از هندسه روبات، محیط و موانع ارائه می‌شود، و هدف پیدا کردن مسیری است که روبات بتواند بدون برخورد با موانع موجود در محیط مسیری را از نقطه ابتدا به نقطه هدف طی کند [۱].

این امر باعث می‌شود تا مسئله اصلی تبدیل به یک مسئله مسیریابی برای یک نقطه تبدیل شود [۲]. پیکربندی^۲ یک روبات (یا جسم) مجموعه‌ای از پارامترها است که به صورت یکتا موقعیت هر نقطه بر روی روبات (یا جسم) را تعیین می‌کند. این موضوع برای اولین بار در برنامه‌ریزی حرکت در [۲] استفاده شد. به عنوان مثال، فضای پیکربندی یک روبات بازویی مجموعه‌ای است شامل زوایای مفاصل آن $(\theta_1, \theta_2, \dots)$.

تاریخ وصول: ۸۹/۶/۹

تاریخ تصویب: ۹۰/۴/۷

داود جنت، دانش‌آموخته کارشناسی ارشد، بخش مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس، jannat.davood@gmail.com

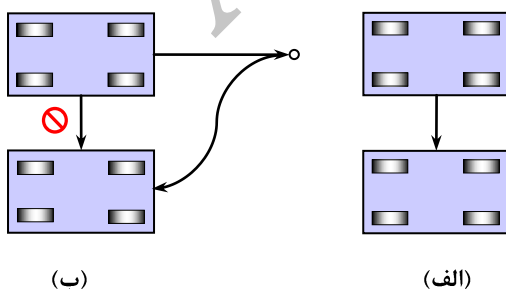
*نویسنده مسئول مقاله: دکتر الپس مسیحی، استادیار، بخش مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس، masehian@modares.ac.ir

² Configuration space

می‌توان نشان داد. جهت A را نیز می‌توان با بیان دو معادله مستقل از این سه زاویه به دست آورد. هر چند محدودیت‌های هولونومیک باعث می‌شوند که ابعاد فضای پیکربندی کاهش پیدا کند اما در ماهیت مساله برنامه‌ریزی حرکت تاثیری ندارند [۳].

در مسیریابی با محدودیت غیرهولونومیک روبات نمی‌تواند بعضی از نقاط فضای پیکربندی را از نقطه‌ای خاص به دست آورد. بنابراین نمی‌توان مانند حالت هولونومیک یک الگوریتم جامع ارائه کرد، اما می‌توان با استفاده از الگوریتم‌های ارائه شده برای مسیریابی هولونومیک، یک مسیر ممکن برای روبات پیدا کرده و با توجه به محدودیت‌های غیرهولونومیک روبات، در مسیر به دست آمده تغییراتی ایجاد کرد تا مسیری ممکن برای روبات پیدا شود.

روبات A با محدودیت غیرهولونومیک را وقتی که در حال حرکت است در نظر بگیرید. هر نقطه از فضای پیکربندی آن مانند q یک معادله مشتقی از زمان است. یک رابطه غیر قابل انتگرال‌گیری بین نقطه پیکربندی روبات و سرعت آن به صورت $G(q, \dot{q}) = 0$ برقرار خواهد بود. G یک تابع همگن است که یک محدودیت غیرهولونومیک نامیده می‌شود. یک محدودیت غیرهولونومیک باعث می‌شود که سرعت‌های قابل دسترس در هر پیکربندی q با m بعد به یک بردار $(m-1)$ بعدی کاهش پیدا کند. اگر k معادله غیرهولونومیک برای یک روبات وجود داشته باشد فضای سرعت قابل دسترس برای روبات $(m-k)$ بعدی خواهد بود. شکل ۱ نشان دهنده تفاوت بین روبات هولونومیک و غیرهولونومیک است. در شکل ۱(الف) روبات فاقد محدودیت هولونومیک است و به راحتی به وضعیت کناری حرکت می‌کند، اما در شکل ۱(ب) روبات دارای محدودیت غیرهولونومیک بوده و نمی‌تواند به پهلو حرکت کند و برای رفتن به وضعیت کناری باید از طریق یک نقطه واسط مسیر نشان داده شده را طی کند (همانند پارک دوپل اتوموبیل).



شکل ۱. مقایسه روبات هولونومیک با غیرهولونومیک

با این که سیستم حرکتی بسیاری از روبات‌ها همانند وسایل غیرهولونومیک است اما بسیاری از مقالات موجود راه حل برای

مسائل برنامه‌ریزی حرکت از نظر محدودیت‌های سینماتیکی به دو دسته هولونومیک^۱ و غیرهولونومیک^۲ دسته‌بندی می‌شوند. در حالت هولونومیک روبات هیچ گونه محدودیت حرکتی ندارد و آزادانه در محیط عاری از مانع می‌تواند حرکت کند، اما وجود محدودیت‌های غیرهولونومیک باعث می‌شود تا روبات نتواند آزادانه در محیط عاری از مانع حرکت کند.

یکی از متداول‌ترین نوع روبات که کاربرد فراوانی در عرصه‌های مختلف دارد، روبات ماشین‌واره^۳ است که در دسته روبات‌های غیرهولونومیک قرار می‌گیرد.

اکثر کارهای انجام شده در زمینه برنامه‌ریزی حرکت مربوط به روبات‌های هولونومیک است، در حالی که برنامه‌ریزی حرکت روبات‌های ماشین‌واره به دلیل وجود محدودیت‌های غیرهولونومیک پیچیده‌تر از حالت هولونومیک است و در این دو دهه اخیر بیشتر مورد توجه محققین قرار گرفته است. بیشتر الگوریتم‌های ارائه شده جهت مسیریابی روبات‌های غیرهولونومیک از الگوریتم‌های هولونومیک ملهم شده‌اند.

در ادامه این مقاله محدودیت‌های هولونومیک و غیرهولونومیک تشریح شده و مدل روبات ماشین‌واره و تحقیقات انجام شده در زمینه مسیریابی این روبات‌ها ارائه می‌شود. در بخش ۲ روش پیشروی سریع^۴ و الگوریتم آن معرفی می‌شود. در بخش ۳ روشی جدید برای مسیریابی روبات ماشین‌واره بر اساس پیشروی سریع ارائه شده و نتایج آن با شبیه‌سازی نشان داده می‌شود. در بخش ۴ مقایسه روش جدید با یک روش موجود انجام گرفته و در بخش ۵ نیز نتیجه‌گیری انجام شده است.

۱-۱. محدودیت‌های هولونومیک و غیرهولونومیک

یک محدودیت هولونومیک رابطه تعادلی بین پارامترهای فضای پیکربندی q به صورت $F(q) = 0$ است. این روابط در قالب معادلاتی می‌توانند فضای پیکربندی را بیان کرده، بعضی از پارامترها را حذف کنند و ابعاد فضای پیکربندی را کاهش دهند. هر یک از روابط به دست آمده بین پارامترها یک بعد از فضای پیکربندی را کاهش می‌دهد. یک مجموعه k -تایی از محدودیت‌های مستقل هولونومیک ابعاد فضای پیکربندی را k تا کاهش می‌دهد [۳].

مفاصل چرخشی و کشویی بازوهای مکانیکی از معمول‌ترین محدودیت‌های هولونومیک هستند. به عنوان مثال جسم سه بعدی A آزادانه می‌تواند در فضا حرکت کند، اما اگر حرکت آن را به چرخش حول یک محور محدود کنیم آنگاه جهت A را با سه زاویه

¹ Holonomic

² Nonholonomic

³ Car-like robot

⁴ Fast Marching Method (FMM)

مقدار سرعت در جهت محورهای x و y به ترتیب $\dot{x} = v \cos(\theta)$ و $\dot{y} = v \sin(\theta)$ است. با تقسیم این دو مقدار خواهیم داشت:

$$\frac{\dot{x}}{\dot{y}} = \frac{\cos(\theta)}{\sin(\theta)} \Rightarrow y \cos(\theta) - x \sin(\theta) = 0 \quad (2)$$

معادله (۲) معادله غیرهولونومیک روبات ماشین‌واره است. مقدار θ (زاویه لحظه ای) برابر است با:

$$\dot{\theta} = \frac{v}{L} \tan(\phi) \quad (3)$$

مسیر ممکن برای روبات ماشین‌واره باید مسیری هموار باشد، بدین معنا که میزان انحنای مورد نیاز جهت حرکت روبات ارضا شود. یک مسیر ممکن می‌تواند مسیری متشکل از خط مستقیم و کمان‌های دایره‌ای شکل باشد [۴]. شعاع کمان‌های دایره‌ای باید بزرگتر از شعاع مینیمم باشد که مقدار آن $\rho_{\min} = L/\tan(\phi_{\max})$ است. اولین مدل روبات ماشین‌واره که در آن تنها حرکت رو به جلو وجود دارد توسط Dubins در سال ۱۹۵۷ ارائه شد [۷]. که خانواده ای از منحنی‌ها با کمترین طول ممکن را جهت مسیریابی تعریف کرد. با الهام از این مدل، مدل روبات Reeds and Shepp (R&S) ارائه شد که علاوه بر حرکت رو به جلو، حرکت رو به عقب نیز داشت [۸]. در [۹] مجموعه نقاط قابل دسترسی روبات ماشین‌واره به دست آمده است.

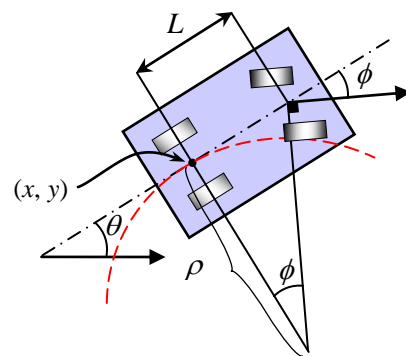
در [۱۰] ثابت شده است که مساله پیدا کردن کوتاه‌ترین مسیر با محدودیت انحنای در محیط دو بعدی از مسائل NP-Hard است و هیچ الگوریتم دقیقی برای این دسته از مسائل وجود ندارد. در نتیجه محققان زیادی جهت حل این مسئله به روش‌های ابتکاری روی آورده‌اند.

در [۱۱] الگوریتمی برای روبات ماشین‌واره بر اساس گراف دیدنگار^۵ ارائه شده است. در این الگوریتم ابتدا در فضای پیکربندی دو بعدی یک مسیر بهینه برای روبات نقطه‌ای شکل پیدا می‌شود که کوتاه‌ترین مسیر ممکن برای روبات نقطه‌ای است. سپس مسیر به دست آمده به صورت محلی با محدودیت‌های غیرهولونومیک برآورد می‌شود: در صورتی که مسیر نتواند محدودیت‌های غیرهولونومیک را ارضا کند، حذف می‌شود و دومین مسیر کوتاه برای روبات نقطه‌ای برآورد می‌شود. مسیر به دست آمده برای روبات ماشین‌واره مسیر نزدیک بهینه است. در این الگوریتم موانع باید چند ضلعی باشند. پیچیدگی زمان حل این الگوریتم $O(nk+m \log n)$ که k

مسیریابی روبات‌های هولونومیک ارائه کرده‌اند [۴]. اکثر روش‌های ارائه شده در برنامه‌ریزی حرکت هولونومیک ملهم از چند شیوه کلاسیک نقشه مسیری^۱ (اسکلت‌یابی)، میدان‌های پتانسیل^۲ و تجزیه سلولی^۳ می‌باشند [۵، ۶].

۲-۱. روبات ماشین‌واره

روبات ماشین‌واره یک مثال ملموس و متداول برای محدودیت غیرهولونومیک است. روبات ماشین‌واره حرکت رو به جلو، رو به عقب و گردشی می‌تواند داشته باشد اما حرکت به راست یا چپ برای این نوع روبات غیر ممکن است پس روبات نمی‌تواند در محیط آزادانه حرکت کند و هر موقعیت و جهتی را به راحتی به دست آورد. شکل ۲ مدل روبات ماشین‌واره را نشان می‌دهد. فضای پیکربندی این روبات $R^2 \times S^1$ است که با سه پارامتر (x, y, θ) نشان داده می‌شود که (x, y) نشان دهنده مختصات نقطه‌ی مرکز اکسل (محور) عقب روبات ماشین‌واره است و θ مقدار زاویه بین محور مثبت x و جهت سرعت روبات است. ϕ نشان دهنده زاویه فرمان می‌باشد که در شکل ۲ مقدار آن منفی است و در تمامی حالات اندازه آن از $\frac{\pi}{2}$ کوچکتر است ($|\phi_{\max}| < \frac{\pi}{2}$). وقتی $|\phi| > 0$ باشد روبات بر روی یک مسیر دایره‌ای شکل حرکت می‌کند که مرکز آن نقطه تقاطع اکسل عقب و جلوی روبات است و اندازه شعاع آن ρ است. L و v به ترتیب نشان دهنده فاصله بین اکسل عقب و جلوی روبات و مقدار سرعت روبات هستند.



شکل ۲. روبات ماشین‌واره

با توجه به شکل ۲ و نمادهای معرفی شده داریم:

$$\rho = \frac{L}{\tan(\phi)} \Rightarrow \tan(\phi) = \frac{L}{\rho} \quad (1)$$

⁴ Reachable set

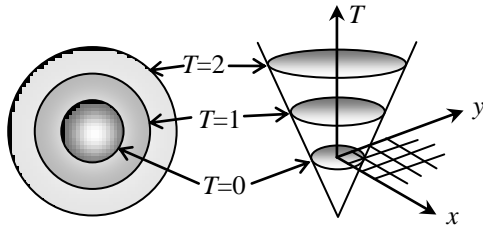
⁵ Visibility graph

¹ Roadmap

² Potential fields

³ Cell decomposition

که در آن Γ_0 نشان دهنده حالت اولیه منحنی در زمان صفر است. معادله (۵) در حالتی که سرعت مثبت است معادله Eikonal نامیده می‌شود. شکل ۳ نشان دهنده حرکت یک دایره با سرعت واحد $F = 1$ عمود بر منحنی آن است.



شکل ۳. حرکت یک منحنی دایره‌ای شکل در جهت عمود بر منحنی با سرعت واحد

روش پیشروی سریع (FMM) یک روش عددی جهت حل معادله دیفرانسیل جزئی غیر خطی Eikonal (معادله ۵) است که توسط Sethian ارائه شده است. پیچیدگی زمان حل این الگوریتم $O(n \log n)$ است که n تعداد سلول‌های تقسیم بندی شده محیط است. روش پیشروی سریع به صورت زیر می‌باشد:

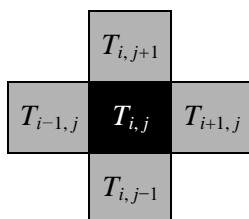
$$\max(D_{ij}^{-x}T, 0)^2 + \min(D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, 0)^2 + \min(D_{ij}^{+y}T, 0)^2 = \frac{1}{F_{ij}^2} \quad (6)$$

که در آن

$$D_{ij}^{-x} = \frac{T_{i,j} - T_{i-1,j}}{\Delta x}, \quad D_{ij}^{+x} = \frac{T_{i+1,j} - T_{i,j}}{\Delta x} \quad (7)$$

$$D_{ij}^{-y} = \frac{T_{i,j} - T_{i,j-1}}{\Delta y}, \quad D_{ij}^{+y} = \frac{T_{i,j+1} - T_{i,j}}{\Delta y}$$

در معادلات (۶) و (۷) مقدار $T_{i,j}$ مقدار زمان عبور از سلول (i, j) است که مجهول است و باید مقداری را بگیرد تا در معادله (۶) صدق کند. Δx و Δy فواصل سلول بندی محیط هستند. شکل ۴ نشان دهنده همسایه‌های سلول $T_{i,j}$ جهت حل معادله (۶) است [۱۹].



شکل ۴. همسایه‌های سلول $T_{i,j}$

تعداد موانع و n تعداد رئوس موانع است. در [۱۲] نیز بر اساس گراف دیدنگار یک الگوریتم مسیریاب برای روبات ماشین‌واره ارائه شده است. مسیر به دست آمده ترکیبی از خطوط مستقیم، کمان‌های دایره‌ای و کمان‌های clothoid است [۱۳، ۱۴]. در [۱۵] الگوریتمی ارائه شده است که می‌توان آن را برای تمامی روبات‌های غیرهولونومیک به کار برد.

این الگوریتم در سه مرحله اجرا می‌شود: ۱- پیدا کردن یک مسیر ممکن برای روبات در فضای پیکربندی مربوط به آن، ۲- تقسیم متوالی مسیر به دو مسیر و پیدا کردن مسیری ممکن از مسیرهای تقسیم شده به نحوی که محدودیت‌های غیرهولونومیک را ارضا کند، ۳- بهینه کردن مسیر به دست آمده جهت کوتاه‌تر شدن طول آن. در [۴] الگوریتم ارائه شده برای روبات ماشین‌واره محیط را به سلول‌های محدب تقسیم‌بندی کرده و آن را به صورت یک گراف ارتباطی بیان می‌کند. با یکی از روش‌های جستجو، سلول‌هایی که نقطه ابتدا و انتها را به هم وصل می‌کند تعیین می‌شوند و به وسیله روش ارائه شده در [۱۶] یک مسیر ممکن در هر سلول و نقطه اتصال دو سلول پیدا می‌شود.

در [۱۷] با الهام از روش PRM^۱ برای روبات ماشین‌واره یک الگوریتم مسیریاب ارائه شده است، و در [۱۸] روش ارائه شده بر اساس PRM در محیط دو بعدی و بدون در نظر گرفتن محدودیت غیرهولونومیک روبات ماشین‌واره، یک مسیر کنترلی ایجاد کرده و سپس مسیر کنترلی به دست آمده را با توجه پیکربندی روبات اصلاح می‌کند.

۲. روش پیشروی سریع

منحنی بسته Γ را تصور کنید که تمامی نقاط آن در جهت عمود بر منحنی و با سرعت مثبت F در حال حرکت هستند. هدف یافتن موقعیت مکانی و شکل منحنی در زمان‌های مختلف بعد از آغاز حرکت است. برای این منظور می‌توان زمان $T(x, y)$ زمانی که منحنی Γ از نقطه $P(x, y)$ می‌گذرد را محاسبه کرد. در حالت یک بعدی داریم:

$$F = \frac{dx}{dT} \Rightarrow F \frac{dT}{dx} = 1 \quad (4)$$

که F سرعت حرکت و عمود بر منحنی است. x و T به ترتیب بیانگر فاصله و زمان طی شده از نقطه شروع هستند. در ابعاد بالاتر معادله حرکت به صورت زیر بیان می‌شود:

$$|\nabla T| \cdot F = 1; \quad T(\Gamma_0) = 0 \quad (5)$$

¹ Probabilistic Roadmap Method (PRM)

۳. روش جدید مسیریابی روبات ماشین‌واره

در این بخش به معرفی روش جدیدی برای برنامه‌ریزی حرکت روبات ماشین‌واره با استفاده از روش پیشروی سریع می‌پردازیم. لازم به ذکر است که روش پیشروی سریع برای اولین بار است که برای حل مسائل با محدودیت غیرهولونومیک مورد استفاده قرار می‌گیرد.

در این الگوریتم ابتدا مسیری پیوسته و هموار توسط FMM ایجاد می‌شود و سپس مسیر به دست آمده با محدودیت غیرهولونومیک روبات تطبیق داده می‌شود.

مهمترین مزیت FMM نسبت به دیگر الگوریتم‌های ارائه شده برای روبات‌های هولونومیک این است که FMM می‌تواند مسیری هموار ایجاد کند [۲۵]. در [۲۶] ثابت شده است که انحناى مسیر به دست آمده از روش FMM با مقدار $1/F$ رابطه دارد و مقدار آن از معادله (۸) تبعیت می‌کند.

$$R \geq \frac{\inf \{1/F\}}{\sup \{\|\nabla(1/F)\|\}} \quad (8)$$

برای افزایش شعاع R می‌توان صورت کسر معادله (۸) را افزایش داد و یا اینکه مخرج کسر را کاهش داد. به وسیله فیلتر میانگین^۵ می‌توان مخرج کسر را کاهش داد و میزان شعاع حداقل را افزایش داد [۲۷]. در شکل ۵ نحوه عملکرد فیلتر میانگین و تغییرات ایجاد شده در انحناى مسیر با اعمال فیلتر میانگین نشان داده شده است. به وسیله FMM می‌توان یک مسیر هموار با انحناى کافی به وجود آورد و مشکل انحناى مسیر را حل کرد. همان‌طور که در شکل‌های ۵(ب) و ۵(ج) نشان داده شده است مسیر به وجود آمده یک مسیر کاملاً هموار با انحناى کافی برای روبات ماشین‌واره است و روبات ماشین‌واره در امتداد آن می‌تواند حرکت کند، اما مشکلی که در این دو شکل وجود دارد حرکات رو به پهلوئی است که روبات هولونومیک در مسیر دارد و روبات ماشین‌واره قادر به انجام آن نیست.

جهت حل این مشکل از ابتکار جدیدی به نام *مانع مجازی بهره جسته ایم*. مانع مجازی مانعی است که خودمان در اطراف روبات قرار می‌دهیم تا محدودیت غیرهولونومیک روبات را لحاظ نماییم. مانع مجازی در روبات Dubins به شکل U بوده و از حرکت روبات به پهلو و رو به عقب در نقاط شروع و انتهای مسیر جلوگیری می‌کند، و در روبات R&S به شکل دو مستطیل موازی بوده و فقط مانع حرکت روبات به پهلو در دو نقطه آغاز و انتهای مسیر می‌شود.

در جدول ۱ الگوریتم پیشروی سریع در فضای دو بعدی نشان داده شده است [۲۰]. الگوریتم FMM کوتاه‌ترین مسیر ممکن را پیدا کرده و الگوریتمی مستقل از شکل مانع است [۲۱]. بر خلاف الگوریتم A^* که مسیری پیوسته اما غیر هموار متشکل از خطوط تولید می‌کند، FMM می‌تواند مسیری با انحناى^۱ مناسب ایجاد کند [۲۲]. در [۲۳] به وسیله نمودار وُرونوی^۲ و FMM یک الگوریتم مسیریابی ارائه شده که روبات نقطه‌ای با یک فاصله اطمینان نسبت به موانع حرکت می‌کند. در [۲۴] بر اساس تبدیل لگاریتمی فاصله در FMM الگوریتمی ارائه شده که مسیری هموار^۳ ایجاد می‌کند و همانند نمودار ورونوی مسیر ایجاد شده دارای یک فاصله اطمینان از موانع است. در [۲۵] یک الگوریتم بهنگام^۴ بر اساس FMM ارائه شده است که موانع در آن چند ضلعی است.

جدول ۱. الگوریتم پیشروی سریع در فضای دو بعدی

گام صفر:

سه مجموعه برای سلول‌ها تعریف می‌شوند:

Alive: مجموعه‌ای از سلول‌ها که مقدار T آن‌ها محاسبه شده و قطعی و تغییر ناپذیر است.

Trial: مجموعه‌ای از سلول‌ها که یک مقدار تقریبی از T برای آنها محاسبه شده و در مراحل بعدی محاسبات مقدار قطعی T آن محاسبه می‌شود و به مجموعه *Alive* اضافه می‌شود.

Far: مجموعه‌ای از سلول‌ها که هیچ مقداری برای آن‌ها محاسبه نشده است.

گام اول:

- تمامی سلول‌های با مقدار $T = \infty$ به مجموعه *Far* تعلق می‌گیرند.
- سلول‌های شروع با مقدار $T = 0$ از مجموعه *Far* حذف شده و به مجموعه *Alive* اضافه می‌شود.
- همسایه‌های سلول شروع به مجموعه *Trial* اضافه می‌شود و مقدار T آن‌ها از معادله (۶) محاسبه می‌شود که البته مقدار تمامی آن‌ها برابر با $1/F_{ij}$ است.

گام دوم:

تکرار قدم‌های زیر:

- نقطه p که دارای مقدار مینیمم در مجموعه *Trial* است، از مجموعه *Trial* حذف می‌شود و به مجموعه *Alive* اضافه می‌شود.
- مقدار T نقاط همسایگی نقطه p به روز می‌شود. نقاط همسایگی نقطه p یکی از سه حالت زیر را دارند:
 ۱. متعلق به مجموعه *Alive* است که در این حالت مقدار T آن قبلاً محاسبه شده و قطعی شده است و مقدار آن دیگر محاسبه نمی‌شود.
 ۲. متعلق به مجموعه *Trial* است که در این حالت مقدار T آن از معادله (۶) دوباره محاسبه می‌شود و مقدار جدید را می‌گیرد.
 ۳. متعلق به مجموعه *Far* است که در این صورت مقدار T آن از معادله (۶) محاسبه می‌شود و از مجموعه *Far* حذف می‌شود و به مجموعه *Trial* اضافه می‌شود.

¹ Curvature

² Voronoi diagram

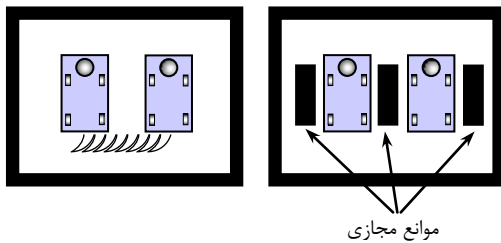
³ Smooth

⁴ Online

⁵ Average Filter

قدم های الگوریتم ارائه شده بدین شرح است:

ابتدا به وسیله FMM و موانع مجازی یک مسیر هموار برای روبات هولونومیک ایجاد می‌شود. سپس مسیر به دست آمده برای روبات هولونومیک به وسیله منحنی اسپلاین^۱ برازش می‌شود تا یک مسیر ایمن برای روبات غیرهولونومیک ماشین‌واره ایجاد شود. لازم به ذکر است که چنانچه با وجود موانع مجازی مسیری پیدا نشود، نمی‌توان نتیجه گرفت که هیچ مسیری موجود نیست زیرا روبات R&S می‌تواند با حرکات رفت و برگشتی به نقاط پیکربندی کناری برود (شکل ۶). در مورد روبات Dubins این مساله صادق نیست اما اگر نقطه آغاز و انتها بسیار نزدیک به یکدیگر باشند باعث می‌شود که موانع مجازی ایجاد شده در این دو نقطه با یکدیگر تداخل داشته باشند و مسیری پیدا نشود.

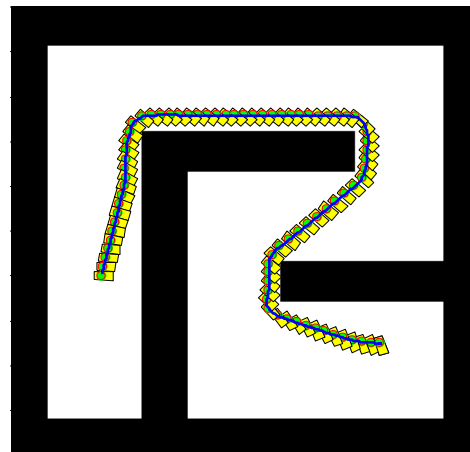


(ب)

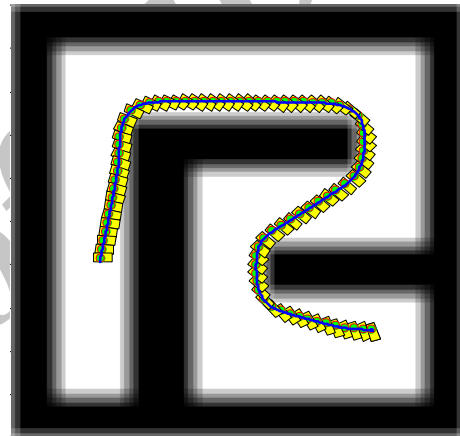
(الف)

شکل ۶. حذف برخی از جواب‌های موجود به خاطر وجود موانع مجازی

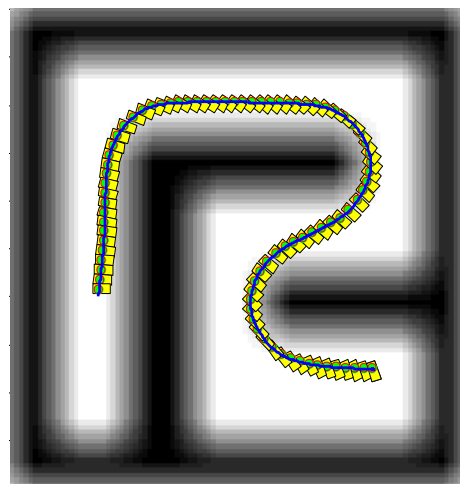
در شبیه‌سازی‌های ارائه شده تعداد سلول‌ها $(100 \times 100 \times 36)$ است که x و y هر دو به 100 سلول و θ به 36 سلول تقسیم شده‌اند. در شکل ۷ یک روبات ماشین‌واره Dubins با حداقل شعاع چرخشی ۱ شبیه‌سازی شده است. در شکل ۷(الف) مسیر روبات هولونومیک بدون موانع مجازی نشان داده شده است. در شکل ۷(ب) یک مسیر برای روبات هولونومیک با اضافه کردن موانع مجازی ایجاد شده است، که در شکل ۷(ج) با منحنی اسپلاین برازش شده و یک مسیر ایمن برای روبات Dubins به دست آمده است. برای مسیریابی روبات R&S می‌توان دو حالت زیر را در نظر گرفت: حالت اول برای مسیری که جهت حرکت روبات در نقطه آغاز و پایان مشابه باشد. شکل ۸ نشان دهنده ایجاد مسیر برای حالت اول است. در این حالت مسیر به دست آمده برای روبات هولونومیک به وسیله منحنی اسپلاین برازش می‌شود و جهت مناسب حرکت روبات ماشین‌واره تعیین می‌شود. به عبارت دیگر، روبات نقطه آغاز مسیر را با حرکت رو به جلو ترک کند و با حرکت رو به جلو نیز وارد نقطه پایان شود، و یا بالعکس (با



(الف)



(ب)

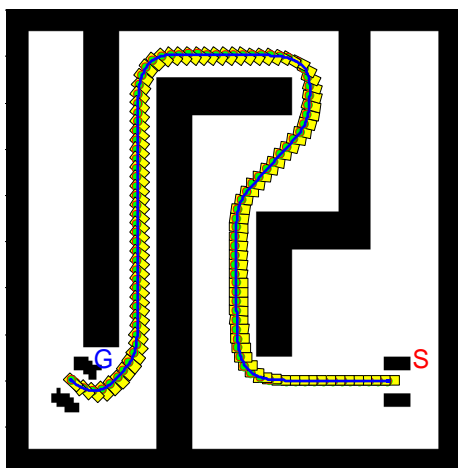


(ج)

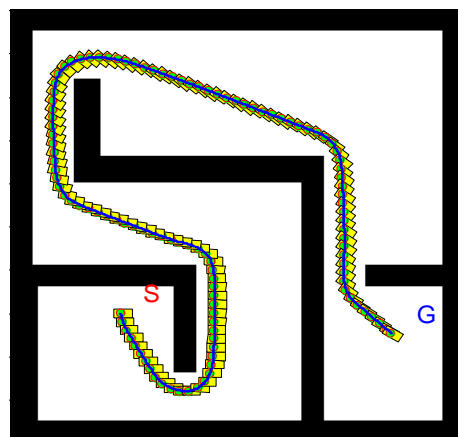
شکل ۵. (الف) مسیر بدون استفاده فیلتر میانگین که شعاع حداقل 0.7722 است. (ب) افزایش انحنا با فیلتر میانگین 5×5 که مقدار حداقل شعاع 4.1687 است. (ج) افزایش بیشتر انحنای مسیر با فیلتر میانگین 11×11 که مقدار حداقل شعاع 7.7380 است

^۱ Spline

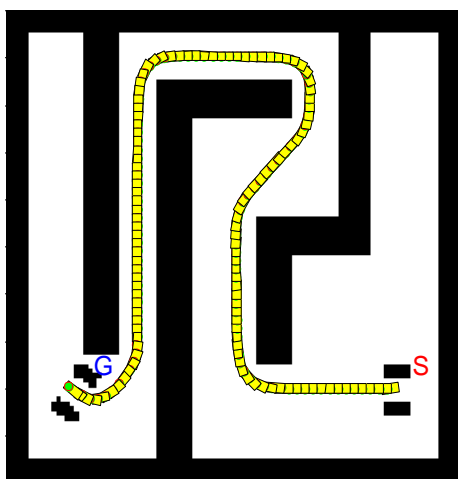
حرکت رو به عقب شروع شود و با حرکت رو به عقب نیز به پایان برسد).



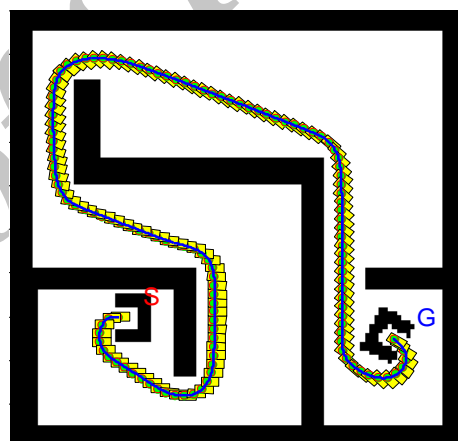
(الف)



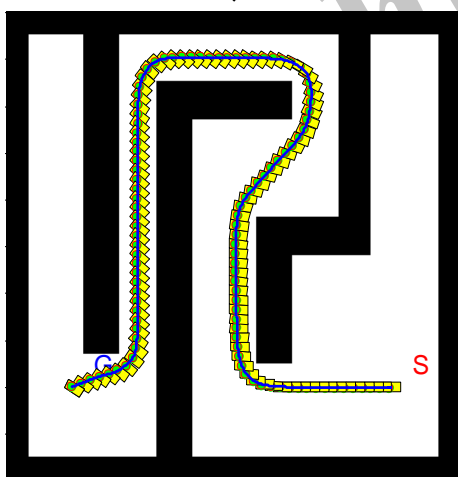
(الف)



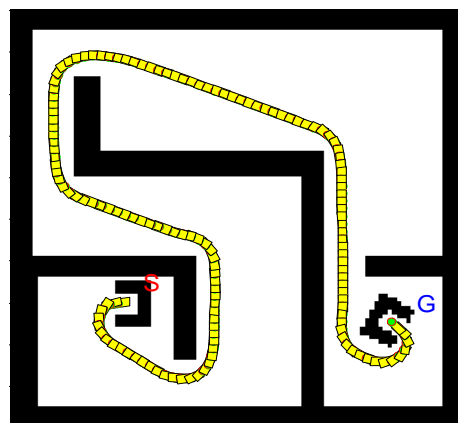
(ب)



(ب)



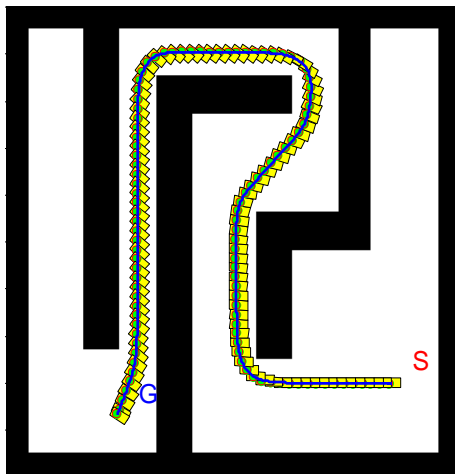
(ج)



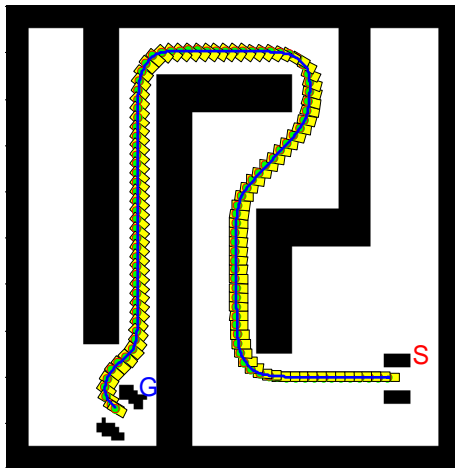
(ج)

شکل ۸. (الف) مسیریابی روبات هولونومیک بدون حضور موانع مجازی ($R_{min} = ۴,۱۸۳۳$) در ۰,۸۳۸ ثانیه. (ب) مسیریابی روبات هولونومیک با موانع مجازی ($R_{min} = ۴,۰۲۳۷$) در ۰,۸۵۵ ثانیه. (ج) مسیریابی روبات R&S ($R_{min} = ۴,۰۲۳۷$) در ۰,۰۲۴ ثانیه

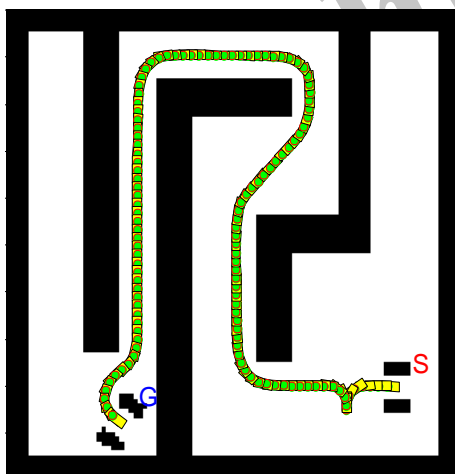
شکل ۷. (الف) مسیریابی روبات هولونومیک بدون موانع مجازی ($R_{min} = ۳,۴۲۷۵$) در ۱,۰۱۴ ثانیه. (ب) مسیریابی روبات هولونومیک با موانع مجازی ($R_{min} = ۲,۱۸۹۳$) در ۱,۴۶۵ ثانیه. (ج) مسیریابی روبات ماشین‌واره Dubins ($R_{min} = ۲,۱۸۹۳$) در ۰,۰۱۸ ثانیه



(الف)



(ب)

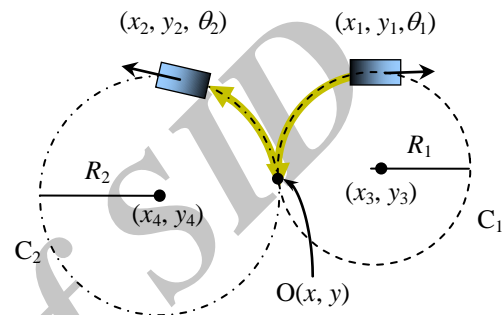


(ج)

شکل ۱۰. (الف) مسیریابی روبات هولونومیک بدون حضور موانع مجازی $R_{min} = 411803$ با زمان حل ۰/۸۴۲ ثانیه. (ب) مسیریابی روبات هولونومیک با موانع مجازی $(R_{min} = 410128)$ با زمان حل ۰/۸۷۳ ثانیه. (ج) مسیریابی روبات R&S با ایجاد تغییر در جهت آن (در قسمت اول مسیر $R_{min} = 218764$ و در قسمت دوم $R_{min} = 410128$ است) با زمان حل ۰/۰۲۵ ثانیه

حالت دوم برای مسیریست که جهت حرکت ابتدا و انتهای آن متفاوت است؛ یعنی روبات با حرکت رو به جلو نقطه شروع را ترک می‌کند و با حرکت رو به عقب وارد نقطه پایان می‌شود و یا بالعکس.

در این حالت روبات باید در محلی مناسب تغییر جهت داده و جهت خود را متناسب با جهت حرکت کند. برای به دست آوردن مسیر مناسب به منظور تغییر جهت می‌توان از رویکرد نشان داده شده در شکل ۹ استفاده کرد.

شکل ۹. تغییر جهت روبات از زاویه θ_1 به زاویه θ_2

در شکل ۹، (x_1, y_1, θ_1) پیکربندی اولیه روبات است. معادله دایره C_1 را با شعاع R_1 که بزرگتر و یا مساوی با شعاع حداقل است مماس بر پیکربندی اولیه روبات به دست آورده و سپس دایره C_2 را به نحوی که هم بر دایره C_1 و هم بر پیکربندی (x_2, y_2, θ_2) مماس باشد به دست می‌آوریم. در این شکل روبات با حرکت رو به عقب تا نقطه $O(x, y)$ و سپس حرکت رو به جلو می‌تواند به پیکربندی نهایی برسد. در نهایت برای مسیریابی کلی، دو کمان دایره ای به دست آمده را با منحنی اسپلاین به دو قسمت ابتدایی و انتهایی مسیر وصل کرده و مسیر نهایی به دست می‌آید. شکل ۱۰ نشان دهنده نتایج شبیه سازی حالت دوم است.

۴. مقایسه الگوریتم موانع مجازی با الگوریتم

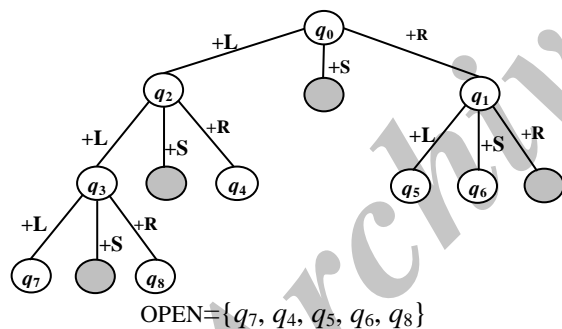
جستجوی شبکه‌ای

Barraquand و Latombe با استفاده از منحنی‌های Reeds-Shepp یک مدل ساده جهت برنامه‌ریزی حرکت روبات‌های ماشین‌واره در یک محیط دو بعدی محدود شده گسترش داده‌اند [۶ و ۲۸]. در این مدل برای روبات ماشین‌واره شش حرکت در نظر گرفته شده است. دو حرکت برای گردش به چپ $(\pm L)$ ، دو حرکت برای گردش به راست $(\pm R)$ ، و دو حرکت مستقیم $(\pm S)$ ، که هر کدام از این سه حرکت می‌توانند در دو جهت رو به جلو و یا رو به عقب باشند. علامات مثبت و منفی نشان دهنده جهت سرعت روبات ماشین‌واره است.

است بدین معنی که اگر اندازه‌ی سلول‌ها به اندازه کافی کوچک باشد و مسیری وجود داشته باشد الگوریتم آن را پیدا خواهد کرد. جهت اعمال روش جستجوی شبکه‌ای برای روبات Dubins تنها باید از سه حرکت +L، +R، و +S استفاده کرد. در شکل ۱۲ یک مساله توسط الگوریتم جستجوی شبکه‌ای (GS) با دو درجه تفکیک پذیری^۳ متفاوت حل شده که در هر دو حالت الگوریتم به یک همسایگی از پیکربندی هدف دست پیدا کرده است.

جهت مقایسه دو الگوریتم جستجوی شبکه‌ای (GS) با الگوریتم مانع مجازی (VO) یک مساله مشابه در سلول‌بندی‌های مختلف توسط دو الگوریتم حل شده است که نتایج آن در جدول ۳ آورده شده است. در تمامی ۹ مساله همسایگی نقطه هدف برای الگوریتم GS ۱ واحد در نظر گرفته شده است.

در جدول ۳ در ستون اول از سمت راست ابعاد سلول‌بندی محیط آورده شده است و ستون دوم زمان ساخت مانع مجازی، ستون سوم زمان محاسبه FMM، ستون چهارم زمان ساخت فضای پیکربندی، و ستون پنجم مجموع زمان کل حل الگوریتم VO را نشان می‌دهند. ستون ششم نیز نشان دهنده زمان حل الگوریتم GS می‌باشد. شکل ۱۳ اختلاف زمان محاسبات را در دو روش نشان می‌دهد.



شکل ۱۱. نمونه ای از درخت T و لیست OPEN

جدول ۳. مقایسه دو الگوریتم VO و GS

ابعاد سلول بندی	زمان الگوریتم VO (ثانیه)				زمان الگوریتم GS (ثانیه)
	زمان کل	ساخت فضای پیکربندی	FMM	ساخت مانع مجازی	
100×100	5.254	0.471	0.11	0.36	0.001
150×150	16.925	1.372	0.44	0.93	0.002
200×200	41.501	3.055	1.29	1.76	0.005
250×250	60.856	6.027	3.06	2.96	0.007
300×300	85.621	10.69	6.27	4.41	0.010
350×350	105.939	17.764	11.51	6.24	0.014
400×400	200.756	27.689	19.45	8.22	0.019
450×450	245.032	42.205	30.84	11.34	0.025
500×500	362.799	61.791	47.63	14.13	0.031

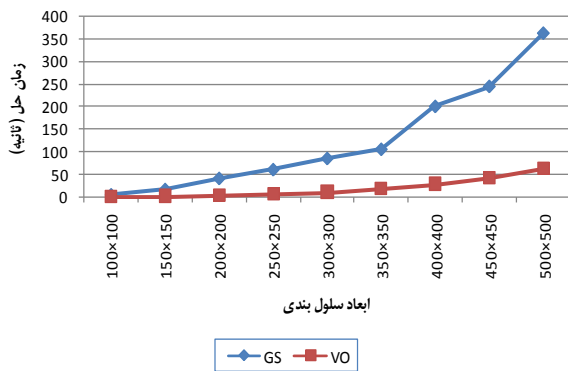
در این الگوریتم یک درخت از نقاط قابل دسترس از نقطه جاری وجود دارد و یک لیست به نام OPEN که در آن نقاطی را که هنوز به آنها رجوع نشده وجود دارد. نقاط موجود در لیست OPEN بر اساس هزینه از کوچک به بزرگ مرتب می‌شوند. الگوریتم از نقطه ابتدایی q_{start} شروع می‌کند و این نقطه را به عنوان ریشه^۱ درخت در نظر می‌گیرد و لیست OPEN را بر اساس نقاط قابل دسترس تعریف می‌کند. این الگوریتم از روش جستجوی best-first search استفاده می‌کند. الگوریتم نقطه‌ای از لیست OPEN را که کمترین هزینه را داراست انتخاب کرده و آن را از لیست OPEN حذف می‌کند و نقاط جدیدی را که از نقطه‌ای که انتخاب کرده و قابل دسترس هستند را به درخت T اضافه می‌کند و بعد از مرتب شدن به لیست OPEN اضافه می‌شوند. این عمل ادامه پیدا می‌کند تا یکی از دو شرط توقف اتفاق بیفتد: (۱) لیست OPEN خالی شود و یا این تعداد گره های درخت T از مقدار تعیین شده توسط کاربر بیشتر شود که از ادامه جستجو منصرف خواهد شد، و یا (۲) الگوریتم یک نقطه در همسایگی نقطه هدف پیدا کند (این همسایگی توسط کاربر تعریف می‌شود). این الگوریتم دقیق نیست و تنها یک مسیر و به همسایگی نقطه هدف پیدا می‌کند.

هزینه‌ی مسیر تابعی از تعداد حرکات روبات، تعداد تغییرات در جهت چرخش و تعداد حرکات برگشتی (حرکت همراه با تغییر جهت) است. برای مثال با در نظر گرفتن ضرایب مثبت a و b و c تابع هزینه برابر خواهد بود با a در تعداد حرکت به اضافه‌ی b در تعداد تغییر چرخش‌ها به اضافه c تا حرکت برگشتی. وقتی که c هر دو صفر باشند الگوریتم کوتاهترین مسیر را پیدا می‌کند و وقتی دو مقدار a و b صفر باشند الگوریتم تعداد حرکات برگشتی را حداقل می‌کند. در شکل ۱۱ نمونه‌ای از درخت T و لیست OPEN که فقط حرکات +L، +R، و +S را در نظر گرفته نمایش داده شده است. الگوریتم مسیریابی در درخت T را که با موانع برخورد دارد را مسدود می‌کند. این الگوریتم همچنین از گسترش نقاطی که در همسایگی نقطه‌ای هستند که قبلاً زیرشاخه‌های آن گسترش پیدا کرده است جلوگیری می‌کند. کاربر باید اندازه همسایگی نقطه هدف و اندازه سلول‌ها جهت شبکه کردن فضای پیکربندی را تعیین کند. اندازه سلول‌های شبکه باید به اندازه کافی کوچک باشد تا فضای پیکربندی با دقت بالایی بررسی شود و نقطه‌ی هدف باید به اندازه کافی بزرگ باشد تا روبات از آن رد نشود. اگر در شکل ۱۱ هزینه هر حرکت ۱ واحد و هزینه تغییر جهت حرکت ۵ واحد در نظر گرفته شود هزینه رسیدن به q_8 از q_0 برابر ۸ واحد است. این الگوریتم کامل از نظر ابعاد سلولی^۲

^۱ Root

^۲ Resolution complete

^۳ Resolution



شکل ۱۳. چگونگی رشد زمان‌های حل دو روش جستجوی گراف و پیشروی سریع با افزایش درجه تفکیک فضای پیکربندی

۵. نتیجه‌گیری

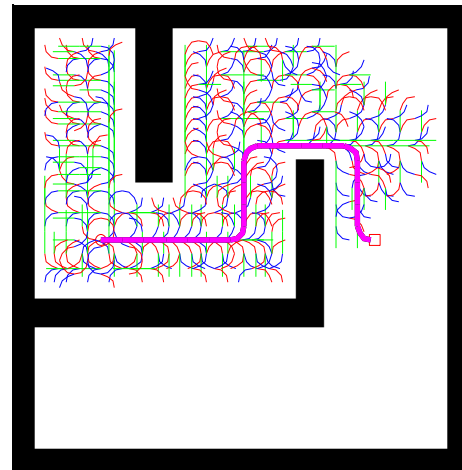
در این مقاله روش جدیدی برای برنامه‌ریزی حرکت روبات ماشین‌واره با استفاده از روش پیشروی سریع (FMM) معرفی شده است. ابتدا به وسیله FMM و رویکرد جدید و مؤثری به نام موانع مجازی (VO) یک مسیر هموار برای روبات بدون لحاظ کردن محدودیت هولونومیک ایجاد می‌شود، و سپس مسیر به دست آمده با در نظر گرفتن حداقل اندازه شعاع چرخش روبات به یک منحنی اسپلاین برازش داده می‌شود تا مسیری ایمن و ممکن برای روبات غیرهولونومیک ماشین‌واره ایجاد شود. روش ارائه شده نسبت به روش‌های موجود سریعتر بوده و دقیق^۱ است، یعنی در صورت وجود جواب حتماً به آن دست پیدا می‌کند. همچنین، پیاده‌سازی روش پیشنهادی جهت مسیریابی روبات ماشین‌واره به سهولت انجام می‌گیرد.

پیدا کردن یک مسیر با انحنای مناسب از مزایای اصلی روش FMM است که باعث بالا رفتن سرعت انجام الگوریتم ارائه شده می‌شود. این الگوریتم مستقل از شکل موانع است که خود یک مزیت نسبت به شماری از الگوریتم‌های دیگر است. از توسعه‌های آتی الگوریتم اعمال مانع مجازی به صورت پویا و هوشمند است به طوری که بسته به وضعیت روبات جهت‌گیری مناسبی پیده کرده و مانع پیدا کردن جواب نشود.

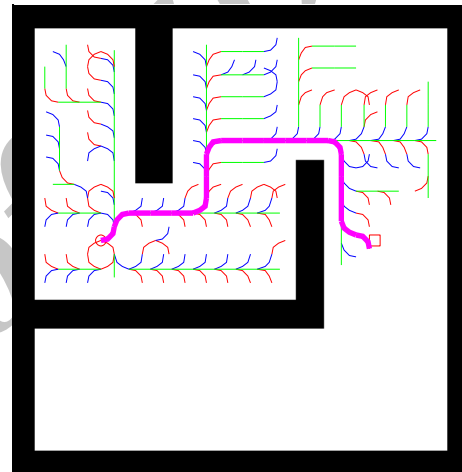
مراجع

- [1] Gonzalez-Banos, H.H., Hsu, D., Latombe, J.C., "Motion Planning Recent Developments", Honda Research Institute USA Inc., USA, 2005.

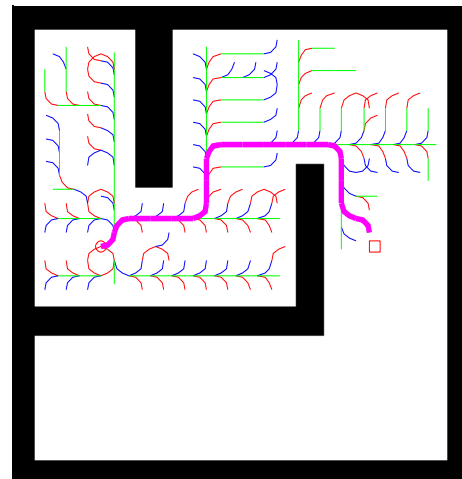
¹ Exact



(الف)



(ب)



(ج)

شکل ۱۲. الگوریتم GS: (الف) سلول‌بندی ۵۰×۵۰ و همسایگی ۱ واحد با زمان حل ۰/۵۱ ثانیه؛ (ب) سلول‌بندی ۲۵×۲۵ و همسایگی ۳ واحد با زمان حل ۰/۲۸ ثانیه؛ (ج) سلول‌بندی ۲۵×۲۵ و همسایگی ۴ واحد با زمان حل ۰/۲۵ ثانیه

- [16] Pin, F.G., Vasseur, H.A., "Autonomous Trajectory Generation for Mobile Robots with Nonholonomic and Steering Angle Constraint", IEEE International Workshop on Intelligent Motion Control, Bogazici University, Istanbul, August 1990.
- [17] Sánchez, A.L., Zapata, R., Arenas, A.B., "Motion Planning for Car-Like Robots using Lazy Probabilistic Roadmap Method", MICAI 2002, pp. 1-10.
- [18] Guang, S., Nancy, A.M., "Randomized Motion Planning for Car-Like Robots with C-PRM", Technical Report TR01-002, Dept. Computer Science, Texas A&M University, March 11, 2001.
- [19] Farag, A.A., Hassouna, M.S., "Theoretical Foundations of Tracking Monotonically Advancing Fronts using Fast Marching Level Set Method", Technical Report Computer Vision and Image Processing Laboratory, ECE Dept., University of Louisville, February 2005.
- [20] Sethian, J.A., "A Fast Marching Level Set Method for Monotonically Advancing Fronts", Proc. National Academy of Science. U.S.A., Vol. 93, No. 4, 1996, pp. 1591-1595.
- [21] Li, Y., *Real-Time Motion Planning of Multiple Agents and Formations in Virtual Environments*, PhD Thesis, Simon Fraser University, Fall 2008.
- [22] Chiang, C.H., Chiang, P.J., Fei, J.C.C., Liu, J.S., "A Comparative Study of Implementing Fast Marching Method and A* Search for Mobile Robot Path Planning in Grid Environment: Effect of Map Resolution", National Science Council under contract NSC 96-2221-E-001-018-MY2, 2007.
- [23] Garrido, S., Moreno, L., Blanco, D., Martin, F., "Exploratory Navigation Based on Voronoi Transform and Fast Marching", 2007.
- [24] Garrido, S., Moreno, L., Blanco, D., Martin, F., "Log of the Inverse of the Distance Transform and Fast Marching Applied to Path Planning", in Proc. IEEE/RJS IROS, 2006, Beijing, China.
- [25] Chiang, C.H., Liu, J.S., "Boundary Following in Unknown Polygonal Environment Based on Fast Marching Method", National Science Council, 2007.
- [26] Cohen, L.D., Kimmel, R., "Global Minimum for Active Contour Models: a Minimal Path Approach", Int. Journal of Computer Vision 24(1), 1997, pp. 57 – 78.
- [27] Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D., "Path Planning for Autonomous Underwater Vehicles", IEEE Trans. Robotics, Vol. 23, No. 2, 2007, pp. 331-341.
- [28] Barraquand, J., Latombe, J.C., "Nonholonomic Multibody Mobile Robots: Controllability and Motion Planning in the Presence of Obstacles", Algorithmica, Vol. 10, 1993, pp. 121-155.
- [2] Lozano-Pérez, T., Wesley, M.A., "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles", 1979.
- [3] Wanseok, Y., "Optimal Approach for Autonomous Parallel Parking of Nonholonomic Car-Like Vehicle", MS thesis, State University of New York, 2006.
- [4] Vasseur, H.A., Pin, F.G., Taylort, J.R., "Navigation of Car-Like Mobile Robot using a Decomposition of the Environment in Convex Cell", in Proc. IEEE ICRA, Sacramento, California, April 1991.
- [5] Latombe, J.C., *Robot Motion Planning*, Kluwer Pub., Boston, MA, 1991.
- [6] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., *Principle of Robot Motion: Theory, Algorithms, and Application*, MIT Press, Cambridge, 2005.
- [7] Dubins, L.E., "On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents", American Journal of Math. Vol. 79, 1957, pp. 497-516.
- [8] Reeds, J.A., Shepp, L.A., "Optimal Paths for a Car that Goes Both Forwards and Backwards," Pacific Journal of Mathematics, Vol. 145, No 2. 1990, pp 367-393.
- [9] Soukres, P., Fourquet, J.Y., Laumond, J.P., "Set of Reachable Positions for a car Philippe", IEEE Transactions on Automatic Control, Vol. 39, No. 8, August 1994.
- [10] Reif, J., Wang, H., "The Complexity of the Two Dimensional Curvature-Constrained Shortest-Path Problem", Workshop on the Algorithmic Foundations of Robotics, Houston, Texas, June 1998.
- [11] Jing, K., Seneviratne, L.D., Earles, S.W., "A Shortest Path Based Path Planning Algorithm for Nonholonomic Mobile Robots", Journal of Intelligent and Robotic Systems, 1999.
- [12] Kito, T., Ota, J., Katsuki, R., Mizuta, T., Arai, T., Ueyama, T., Nishiyama, T., "Smooth Path Planning by using Visibility Graph-Like Method", in Proc. IEEE, Taipei, Taiwan, September 2003.
- [13] Scheuer, A., Fraichard, T., "Continuous-Curvature Path Planning for Multiple Car-Like Vehicles", in Proc. IEEE/RJS IROS, September 1997, Grenoble, France.
- [14] Scheuer, A., Fraichard, T., "Continuous-Curvature Path Planning for Car-Like Vehicles", in Proc. Inc.-Copernicus ERB--IC15-CT96-0702 project, May 25, 1998.
- [15] Laumond, J.P., Jacobs, P.E., Taix, M., Murray, R.M., "A Motion Planner for Nonholonomic Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 10, No. 5, October 1994.