

مسیریابی روبات‌های ماشین واره یدک‌کش با روش پیشروی سریع (FMM)

داود جنت^۱ و الپس مسیحی^{۲*}

اطلاعات مقاله	چکیده
<p>واژگان کلیدی: مسیریابی روبات، روبات یدک‌کش، روش پیشروی سریع، مانع مجازی.</p>	<p>چکیده: روبات یدک‌کش روبات ماشین‌واره‌ای است که یک یا چند یدک فاقد نیروی محرکه را به دنبال خود می‌کشد. اضافه شدن هر یدک یک محدودیت سینماتیکی غیرهولونومیک به مساله مسیریابی اضافه می‌کند که باعث پیچیده‌تر شدن مساله می‌شود. با به‌کاربردن مفهوم اندازه معادل (ES)، مسئله برنامه‌ریزی حرکت یک روبات یدک‌کش تبدیل به مسئله برنامه‌ریزی حرکت روبات ماشین‌واره می‌شود. مقدار پارامتر ES باتوجه به تعداد یدک‌ها، ابعاد آن‌ها و همچنین نحوه اتصال و فاصله اتصال تعیین می‌شود. در این مقاله به وسیله روش پیشروی سریع - که یک روش عددی برای حل معادله دیفرانسیل جزئی غیرخطی Eikonal است - و با استفاده از مفهوم مانع مجازی یک الگوریتم مسیریاب برای روبات ماشین‌واره ارائه شده است که با استفاده از ES می‌توان آن را برای روبات‌های یدک‌کش تعمیم داد. الگوریتم ارائه شده سریع، دقیق، مستقل از شکل موانع و آسان در پیاده‌سازی است. علاوه بر آن روش مذکور با دو روش جستجوی شبکه‌ای و RRT غیرهولونومیک مقایسه شده و برتری آن از نظر سرعت حل نشان داده شده است.</p>

۱- مقدمه

بر روی روبات (یا جسم) را به‌صورت یکتا تعیین می‌کند. در فضای پیکربندی روبات تبدیل به یک نقطه می‌شود و مسیر بین دو پیکربندی متفاوت برای یک روبات نقطه‌ای تعیین می‌شود [۲].

مسائل برنامه‌ریزی حرکت از نظر محدودیت‌های سینماتیکی به دو دسته هولونومیک و غیرهولونومیک دسته‌بندی می‌شوند. در حالت هولونومیک روبات هیچ‌گونه 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 و لذا مجبور به حرکت در امتداد مسیری می‌شود که محدودیت‌های مذکور را ارضا نماید. در نتیجه، یافتن یک مسیر عاری از تصادم با موانع

مسیریابی روبات یکی از مباحث مهم و به روز حوزه روباتیک است که هدف آن یافتن یک مسیر ممکن و حتی الامکان بهینه در محیطی با اشیای ثابت یا متحرک است که روبات بتواند با پیمودن آن از وضعیت شروع خود به یک وضعیت مطلوب و نهایی بدون برخورد با موانع منتقل شود [۱]. در مسائل برنامه‌ریزی حرکت توصیف کاملی از هندسه روبات، محیط و موانع ارائه می‌شود، و در بیشتر الگوریتم‌های مسیریابی، از مفهوم فضای پیکربندی جهت مسیریابی استفاده می‌شود. پیکربندی یک روبات (یا جسم) مجموعه‌ای از پارامترها است که موقعیت هر نقطه

* پست الکترونیک نویسنده مسئول: masehian@modares.ac.ir

۱. کارشناس ارشد مهندسی صنایع

۲. استادیار، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس

در زمینه ربات‌های ماشین‌واره تاکنون تحقیقات متنوعی انجام شده است، درحالی‌که کارهای زیادی برای مسیریابی ربات‌های یدک‌کش در ادبیات موجود نیست، زیرا مسیریابی ربات یدک‌کش بسیار پیچیده‌تر از مسیریابی یک ربات ماشین‌واره است.

اولین مدل ربات ماشین‌واره که در آن تنها حرکت رو به جلو وجود دارد توسط دوبینز در سال ۱۹۵۷ ارائه شد [۶]، که خانواده‌ای از منحنی‌ها با کمترین طول ممکن را جهت مسیریابی تعریف کرد. با الهام از این مدل، مدل ربات ماشین‌واره R&S ارائه شد که علاوه بر حرکت رو به جلو، حرکت رو به عقب نیز داشت [۷]. الگوریتم‌های ارائه شده در مراجع [۸] و [۹] قابلیت مسیریابی برای هر دو نوع ربات‌های ماشین‌واره و یدک‌کش را دارا هستند.

در مرجع [۱۰] الگوریتمی ارائه شده است که می‌توان آن را برای تمامی ربات‌های غیرهولونومیک به‌کار برد. این الگوریتم در سه مرحله اجرا می‌شود: ۱- پیدا کردن یک مسیر ممکن برای ربات در فضای پیکربندی مربوط به آن، ۲- تقسیم متوالی مسیر به دو مسیر و پیدا کردن مسیری ممکن از مسیرهای تقسیم شده به‌نحوی که محدودیت‌های غیرهولونومیک را ارضا کند، ۳- بهینه کردن مسیر به‌دست آمده جهت کوتاه‌تر شدن طول آن.

در مرجع [۱۱] با الهام از روش نقشه‌مسیر احتمالی (PRM) برای ربات ماشین‌واره یک الگوریتم مسیریاب ارائه شده است، و در مرجع [۱۲] روش ارائه شده براساس PRM در محیط دو بعدی و بدون در نظر گرفتن محدودیت غیرهولونومیک ربات ماشین‌واره، یک مسیر کنترلی ایجاد کرده و سپس مسیر کنترلی به‌دست آمده را با توجه به پیکربندی ربات اصلاح می‌کند.

در این مقاله روش جدیدی برای برنامه‌ریزی مسیر ربات‌های یدک‌کش ارائه شده است: ابتدا به‌وسیله روش پیشروی سریع - که یک روش عددی برای حل معادله دیفرانسیل جزئی غیرخطی Eikonal است - یک مسیر برای ربات ماشین‌واره محاسبه می‌شود و سپس با استفاده از مفهوم اندازه معادل، برای ربات‌های یدک‌کش

برای ربات هولونومیک به مراتب ساده‌تر از مسیریابی برای ربات غیرهولونومیک است.

جهت مقایسه بهتر محدودیت‌های هولونومیک و غیرهولونومیک می‌توان به صندلی چرخدار و ویلچر اشاره کرد، که صندلی چرخدار (با چرخ‌های گوی و کاسه‌ای) به‌راحتی از یک موقعیت و جهت به هر موقعیت و جهت دیگری حرکت می‌کند، در حالی که یک ویلچر نمی‌تواند به پهلو حرکت کند (شکل ۱).

مثال دیگری از محدودیت غیرهولونومیک اتومبیل است که نمی‌تواند به پهلو حرکت کند. بنابراین ربات‌های ماشین‌واره و یدک‌کش نیز متعلق به دسته ربات‌های غیرهولونومیک هستند و قادر نیستند در هر جهتی حرکت کنند.



(ب)

(الف)

شکل ۱- (الف) نشان دهنده محدودیت هولونومیک است و صندلی قادر است به هر طرف حرکت کند. (ب) نشان دهنده محدودیت غیرهولونومیک است که در آن صندلی قادر نیست به طرفین حرکت کند.

علی‌رغم نقش و کاربردهای بسیاری که ربات‌های غیرهولونومیک در عرصه صنعت دارند (به‌طور مثال در زمینه حمل و نقل و جابه‌جایی مواد)، بیشتر تحقیقات در حوزه مسیریابی ربات‌ها برای انواع هولونومیک انجام شده است که مهمترین دلیل آن پیچیدگی مسائل مسیریابی ربات‌های غیرهولونومیک است [۳]. بیشتر روش‌های ارائه شده در برنامه‌ریزی حرکت ربات‌های هولونومیک برگرفته از چند شیوه کلاسیک نقشه‌مسیر (اسکلت‌یابی)، میدان‌های پتانسیل و تجزیه سلولی می‌باشند [۴-۵].

در شکل ۲، l_1 فاصله میان دو اکسل روبات یدک کش، و l_2 و l_3 به ترتیب نشان دهنده فاصله بین مرکز اکسل یدک اول و دوم با مرکز اکسل روبات یدک کش و یدک جلویی آن‌ها هستند. θ_i زاویه محور اصلی هر جزء سیستم با افق بوده و φ نیز زاویه فرمان یدک کش می‌باشد که اندازه آن، کوچک‌تر از $\pi/2$ است و اگر مقداری مخالف صفر داشته باشد روبات یدک کش بر روی یک دایره به شعاع $R = l_1 / \tan(\varphi)$ حرکت می‌کند.

روبات یدک کش که یک روبات ماشین‌واره است دارای فضای پیکربندی $R^2 \times S^1$ است که با سه پارامتر (x, y, θ) نشان داده می‌شود، که در آن (x, y) نشان دهنده مختصات نقطه مرکز اکسل عقب روبات ماشین‌واره، و θ مقدار زاویه بین محور مثبت x و خط واصل بین مرکز دو اکسل روبات است. هر یدک دارای فضای پیکربندی S^1 است که نشان دهنده زاویه بین محور اصلی هر یدک (که عمود بر اکسل آن است) با جهت مثبت محور x است. با اضافه شدن هر یدک، یک بعد به ابعاد فضای پیکربندی و یک محدودیت غیرهولونومیک دیگر اضافه می‌شود که این امر باعث پیچیده‌تر شدن مدل می‌شود. در واقع، بُعد فضای پیکربندی یک روبات یدک کش با k یدک $k+3$ می‌شود. معادلات حرکت روبات یدک کش که به صورت استاندارد به یدک‌هایش متصل است عبارتند از [۱۳]:

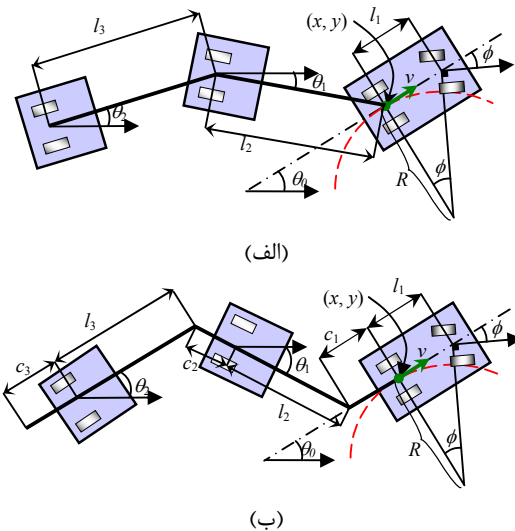
$$\begin{aligned} \dot{x} &= v \cdot \cos \theta_0 \\ \dot{y} &= v \cdot \sin \theta_0 \\ \dot{\theta}_0 &= \frac{v}{l_1} \tan \varphi \\ \dot{\theta}_1 &= \frac{v}{l_2} \sin(\theta_1 - \theta_0) \\ &\vdots \\ \dot{\theta}_i &= \frac{v}{l_{i+1}} \left(\prod_{j=1}^{i-1} \cos(\theta_{j-1} - \theta_j) \right) \sin(\theta_{i-1} - \theta_i) \end{aligned} \quad (1)$$

از اولین الگوریتم‌های مسیریابی روبات یدک کش در محیطی با وجود مانع، روش جستجوی شبکه‌ای است. در این مدل برای روبات یدک کش شش حرکت در نظر گرفته شده است، که عبارتند از چهار حرکت پیچیدن به چپ و

تعمیم داده می‌شود. در بخش ۲ ساختار و روابط حاکم بر روبات یدک کش معرفی می‌شود، در بخش ۳ روش پیشروی سریع (Fast Marching Method: FMM) بیان می‌شود و در بخش ۴ رویکرد مسیریابی جدید بر پایه روش پیشروی سریع ارائه می‌شود. بخش‌های ۵، ۶ و ۷ نیز به ترتیب شبیه‌سازی و تحلیل حساسیت، مقایسه و نتیجه‌گیری را ارائه می‌کنند.

۲- روبات یدک کش

روبات یدک کش روبات ماشین‌واره‌ای است که یک یا چند یدک فاقد نیروی محرکه را به دنبال خود می‌کشد (شکل ۲).



شکل ۲- (الف) روبات یدک کش با اتصال استاندارد؛ (ب) روبات یدک کش با اتصال غیر استاندارد.

روبات‌های یدک کش از نظر نحوه اتصال یدک‌ها به دو دسته استاندارد و غیراستاندارد دسته‌بندی می‌شوند: در حالت استاندارد یدک به وسیله یک میله رابط به مرکز اکسل (محور) عقب یدک جلویی و یا یدک کش وصل می‌شود (شکل ۲-الف)، اما در حالت غیراستاندارد یدک به وسیله یک میله رابط به میله امتداد داده شده از مرکز اکسل یدک جلویی و یا روبات یدک کش وصل می‌شود (شکل ۲-ب).

۳- روش پیشروی سریع (FMM)

فرض کنیم در پارکینگی بزرگ و عاری از مانع قرار داریم که سرعت حرکت در آن یک واحد سرعت است و قصد داریم از نقطه شروع A به نقطه B محل خودروی خود حرکت کنیم (شکل ۳-الف). کوتاه‌ترین فاصله زمانی جهت پیمودن مسیر، پیمایش خط راستی است که نقطه A را به نقطه B وصل می‌کند. راه دیگر یافتن این مسیر آن است که از نقطه A با سرعت ثابت یک واحد بر ثانیه دوایر زمانی رسیدن به نقاط دیگر پارکینگ را گسترش دهیم تا به نقطه B برسیم. هر دایره مکان هندسی نقطای است که از نقطه A دارای فاصله زمانی برابر هستند. با رسیدن به نقطه B می‌توان با حرکت عمود بر دوایر به نقطه A رسید که نتیجه همان خط مستقیم است.

حال فرض کنیم که نصف پارکینگ پوشیده از برف است به طوری که سرعت حرکت در آن برابر $0/5$ واحد است، و قصد داریم از نقطه A که در قسمت برفی قرار دارد با صرف کمترین زمان ممکن به نقطه B که در قسمت خشک قرار دارد حرکت کنیم. با گسترش دوایر زمانی از نقطه A که مکان هندسی نقطای از پارکینگ است که در زمان t بعد از حرکت از نقطه A به آن می‌رسیم، شکل ۳-ب حاصل می‌شود که با حرکت عمود بر دوایر (یعنی در جهت بیشترین کاهش گرادیان) از نقطه B تا رسیدن به نقطه A کوتاه‌ترین مسیر زمانی بین دو نقطه A و B به دست خواهد آمد.

حال فرض کنیم که در پارکینگ اتومبیلی مابین دو نقطه A و B قرار دارد که به عنوان یک مانع تلقی شده و سرعت عبور از آن صفر واحد است. با گسترش دوایر زمانی از نقطه A نتیجه شکل ۳-ج خواهد شد [۱۹].

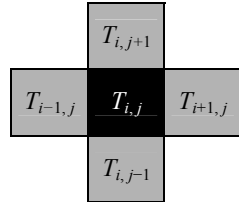
باتوجه به مطالب گفته شده می‌توان مسئله پیدا کردن کوتاه‌ترین مسیر را به مسئله مسیریابی روبات‌تعمیم داد، بدین صورت که برای حرکت در فضای عاری از مانع سرعت یک واحد را در نظر گرفت و برای عبور از موانع سرعت صفر واحد در نظر گرفت، و همان‌طور که در شکل

راست با دو سرعت مثبت و منفی و دو حرکت مستقیم با دو سرعت مثبت و منفی. در این روش فضای پیکربندی سلول‌بندی (شبکه بندی) شده و از پیکربندی شروع با گسترش شش منحنی سلول‌های پیکربندی تا رسیدن به یک همسایگی از پیکربندی هدف مورد پیمایش قرار می‌گیرند [۸].

در [۱۴] جهت ساده‌تر شدن مسئله، با در نظر گرفتن مفروضاتی برای پارامترهای مدل روبات یدک‌کش، یک راه حل دقیق جهت مسیریابی بین دو پیکربندی در محیط عاری از موانع ارائه شده است که می‌توان از آن به عنوان یک مسیریاب محلی برای مسیریابی در محیط‌های با مانع استفاده کرد. در [۱۵] براساس الگوریتمی در چند سطح ارائه شده است که محدودیت‌های غیرهولونومیک به صورت گام به گام اضافه می‌شوند. براساس الگوریتم RRT نیز یک الگوریتم غیرهولونومیک ارائه شده است که در آن یک نقطه به تصادف از فضای پیکربندی انتخاب می‌شود، سپس پیکربندی انتخابی جهت ارضای محدودیت‌های غیرهولونومیک روبات یدک‌کش تغییر داده می‌شود [۹].

ارائه مفهوم اندازه معادل (ES) برای روبات یدک‌کش توسط یوان و همکاران تا حد زیادی باعث آسان‌تر شدن حل مسائل مسیریابی روبات‌های یدک‌کش شده است. ES به تعداد یدک‌ها و اندازه روبات یدک‌کش و هر یک از یدک‌ها بستگی دارد که برای راحتی محاسبه آن اندازه تمامی آن‌ها یکسان در نظر گرفته می‌شود. در صورتی که موانع به اندازه نصف ES بزرگ شوند مساله مسیریابی روبات یدک‌کش به مساله مسیریابی روبات ماشین‌واره تبدیل می‌شود [۱۶]. با استفاده از مفهوم ES و الگوریتم ژنتیک و الهام از الگوریتم PRM دو الگوریتم مسیریابی برای روبات یدک‌کش ارائه شده است [۱۷ و ۱۸].

روش پیشروی سریع با سلول‌بندی محیط در رزولوشن (تفکیک‌پذیری) دلخواه شروع می‌شود. هدف الگوریتم، محاسبه فاصله زمانی هر سلول از سلول شروع می‌باشد. نمایش زمان عبور از سلول (i, j) با متغیر $T_{i,j}$ ، سلول‌های همسایه آن مطابق شکل ۴ نشان داده می‌شوند [۲۰].



شکل ۴- همسایه‌های سلول $T_{i,j}$.

حال برای محاسبه مقدار $T_{i,j}$ هر سلول، از رابطه زیر استفاده می‌شود:

$$\max(D_{ij}^{-x}T, 0)^2 + \min(D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, 0)^2 + \min(D_{ij}^{+y}T, 0)^2 = \frac{1}{F_{ij}^2} \quad (۴)$$

که در آن F_{ij} سرعت حرکت در سلول (i, j) است و

$$D_{ij}^{-x}T = \frac{T_{i,j} - T_{i-1,j}}{\Delta x}, \quad D_{ij}^{+x}T = \frac{T_{i+1,j} - T_{i,j}}{\Delta x} \quad (۵)$$

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

در معادله (۵)، Δx و Δy فواصل سلول بندی محیط در جهت افقی و عمودی بوده و مقدار $T_{i,j}$ مجهول است و باید مقداری را بگیرد تا در معادله (۴) صدق کند. گام‌های الگوریتم پیشروی سریع در فضای دو بعدی به صورت زیر انجام می‌شود، که در آن برای سهولت، متغیر $T_{i,j}$ با T نشان داده شده است [۲۱].

گام ۱:

سه مجموعه برای سلول‌ها تعریف می‌شوند: *Alive*: مجموعه‌ای از سلول‌ها که مقدار T آن‌ها محاسبه شده و قطعی و تغییرناپذیر 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.

۳- نشان داده شد کوتاه‌ترین مسیر بین دو نقطه شروع و پایان بدون برخورد با موانع به دست خواهد آمد.

باتوجه به معادله سرعت اجسام $T = x/F$ که در آن T ، x و F به ترتیب نماد زمان، مسافت و سرعت هستند، زمان طی شده بین دو نقطه رابطه مستقیم با مسافت بین دو نقطه و رابطه عکس با سرعت حرکت دارد، که می‌توان به صورت زیر بیان کرد:

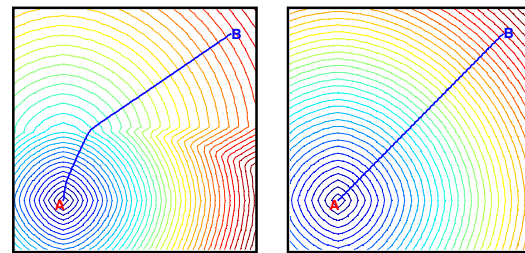
$$F = \frac{dx}{dT} \Rightarrow F \frac{dT}{dx} = 1 \quad (۲)$$

رابطه ۲ در ابعاد بالاتر به صورت زیر بیان می‌شود:

$$|\nabla T| \cdot F = 1; \quad T(S) = 0 \quad (۳)$$

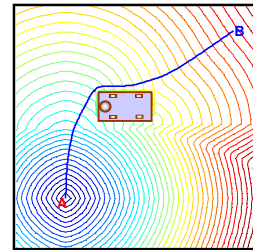
که در آن $T(S)$ بیانگر زمان نقطه شروع است. رابطه (۳) وقتی که مقدار F مثبت باشد به معادله دیفرانسیل جزئی غیرخطی Eikonal معروف است که دارای جواب صریح نیست [۲۰].

روش پیشروی سریع (FMM) یک روش عددی جهت حل معادله دیفرانسیل جزئی غیرخطی Eikonal (رابطه (۳)) است که توسط جیمز ستیان از دانشگاه برکلی کالیفرنیا ارائه شده است. پیچیدگی زمان حل این الگوریتم $O(n \log n)$ است که n تعداد سلول‌های تقسیم‌بندی شده محیط است.



(ب)

(الف)



(ج)

شکل ۳- (الف) پارکینگ عاری از مانع؛ (ب) نیمی از پارکینگ پوشیده از برف است؛ (ج) علاوه بر برف یک اتومبیل نیز در پارکینگ وجود دارد.

روبات نقطه‌ای با یک فاصله اطمینان نسبت به موانع حرکت می‌کند.

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

گام ۲:

تمامی سلول‌های با مقدار $T = \infty$ به مجموعه *Far* تعلق می‌گیرند.

سلول 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 (۴) محاسبه می‌شود که البته مقدار تمامی آن‌ها برابر با $\frac{1}{F_{ij}}$ است.

گام ۳:

تکرار قدم‌های زیر:

نقطه p که دارای مقدار حداقل در مجموعه *Trial* است، از مجموعه *Trial* حذف می‌شود و به مجموعه *Alive* اضافه می‌شود.

مقدار T سلول‌های 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 (۴) دوباره محاسبه می‌شود و مقدار جدید را می‌گیرد.

۳. متعلق به مجموعه *Far* است که در این صورت مقدار T آن از معادله (۴) محاسبه می‌شود و از مجموعه *Far* حذف می‌شود و به مجموعه *Trial* اضافه می‌شود.

الگوریتم FMM کوتاه‌ترین مسیر ممکن را پیدا کرده و الگوریتمی مستقل از شکل مانع است [۲۲]. بر خلاف الگوریتم A^* که مسیری پیوسته اما غیرهموار متشکل از خطوط تولید می‌کند، FMM می‌تواند مسیری با انحنای^۱ مناسب ایجاد کند [۲۳]. در [۲۴] به‌وسیله نمودار ورونوی^۲ و FMM یک الگوریتم مسیریابی ارائه شده که

۴- روش جدید مسیریابی روبات یدک‌کش

در بخش ۲ روبات یدک‌کش معرفی شد و مفهوم اندازه معادل *ES* نیز بیان شد. با بزرگ‌تر کردن ابعاد موانع به مقدار نصف اندازه معادل ($ES/2$) روبات یدک‌کش به یک روبات نقطه‌ای با محدودیت غیرهولونومیک روبات ماشین‌واره تبدیل می‌شود. بنابراین با پیدا کردن یک مسیر ممکن برای روبات ماشین‌واره، می‌توان به‌وسیله پارامتر *ES* آن را برای روبات یدک‌کش تعمیم داد.

باتوجه به مطالب گفته شده در مورد پارامتر *ES* در [۱۶] و مدل ارائه شده از روبات یدک‌کش با اتصالات استاندارد در بخش ۲، مقدار *ES* به‌صورت زیر محاسبه می‌شود:

$$ES = \sqrt{\left(\frac{l}{\tan \varphi_{\max}}\right)^2 + N \cdot l^2} - \frac{l}{\tan \varphi_{\max}} + \frac{D}{2} \quad (۶)$$

که در آن φ_{\max} مقدار حداکثر زاویه فرمان روبات یدک‌کش، N تعداد یدک‌های متصل به یدک‌کش، D پهنای روبات و یدک‌ها، و l نیز فاصله بین دو اکسل عقب و جلوی روبات یدک‌کش یا طول محور رابط یدک‌ها و روبات به یکدیگر است (رجوع شود به شکل ۲-الف).

با بزرگ‌کردن موانع به اندازه *ES* مسئله مسیریابی روبات یدک‌کش تبدیل به مسئله مسیریابی یک روبات ماشین‌واره می‌شود که شعاع حداقل حرکت آن $\sqrt{\left(\frac{l}{\tan \varphi_{\max}}\right)^2 + N \cdot l^2}$ خواهد بود، که l طول اتصال بین یدک‌ها و نیز روبات با اولین یدک است.

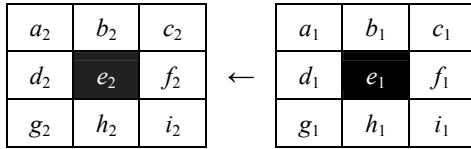
۴-۱- ایجاد یک مسیر هموار

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 در روش FMM، انحنای مسیر (که عکس

¹ Curvature
² Voronoi diagram

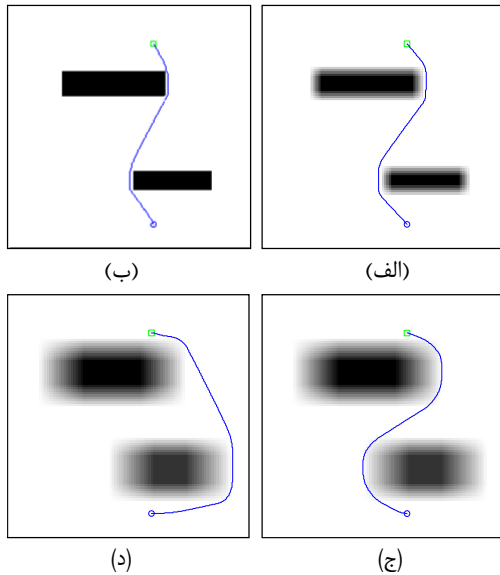
در این رابطه، $g(s, t)$ مقدار $1/F$ برای هر یک از سلول‌های داخل پنجره S_{xy} بوده و $f(x, y)$ مقدار $1/F$ جدید برای سلول (x, y) است [۲۸]. در شکل ۵ یک فیلتر میانگین 3×3 نشان داده شده است که در آن مقدار جدید $1/F$ برای سلول مرکزی پنجره S_{xy} برابر است با:

$$e_2 = \frac{a_1 + b_1 + c_1 + d_1 + e_1 + f_1 + g_1 + h_1 + i_1}{9}$$



شکل ۵- نحوه اعمال فیلتر میانگین.

در شکل ۶ تأثیر تغییر اندازه فیلتر میانگین و افزایش مقدار $1/F$ بر انحنای مسیر ایجاد شده نشان داده شده است.



شکل ۶- (الف) مسیر ایجاد شده توسط FMM با شعاع حداقل $1/2947$ واحد؛ (ب) اعمال فیلتر میانگین 5×5 ، مقدار شعاع حداقل $3/3180$ واحد؛ (ج) اعمال فیلتر میانگین 19×19 ، مقدار شعاع حداقل $4/8165$ واحد؛ (د) اعمال فیلتر میانگین 19×19 و افزایش مقدار $1/F$ به اندازه ۳ واحد، مقدار شعاع حداقل $7/8546$ واحد.

شعاع پیمایش می‌باشد) با مقدار $1/F$ (که عکس سرعت بوده و تابعی از زمان است) رابطه داشته و مقدار آن از رابطه (۷) تبعیت می‌کند:

$$R \geq \frac{\inf \{1/F\}}{\sup \{\|\nabla(1/F)\|\}} \quad (7)$$

می‌دانیم که در محیط بدون مانع، کوتاه‌ترین مسیر بین دو نقطه یک خط راست است که شعاع آن بینهایت بوده و به عبارتی انحنای صفر دارد. رابطه (۷) ارتباطی بین تابع هزینه زمانی (یعنی $1/F$) و انحنای مسیر به دست آمده ایجاد می‌کند، و از این جهت مفید است که می‌توان محاسبه کرد آیا مسیر به دست آمده توسط روبات ماشین‌واره قابل پیمایش است یا خیر. در واقع، کافیست که حداقل شعاع به دست آمده از مسیر بهینه تولید شده R_{lim} را با حداقل شعاع چرخش روبات^۱ R_{rob} مقایسه کرد: اگر $R_{rob} \leq R_{lim}$ باشد آنگاه مسیر تولید شده یک مسیر بهینه شدنی برای روبات است. در غیر این صورت امکان پیمایش مسیر توسط روبات وجود ندارد. برای اجتناب از این امر بایستی شعاع حداقل R را افزایش داد [۲۷]. برای افزایش R می‌توان صورت کسر رابطه ۷ را به وسیله افزایش مقدار $1/F$ افزایش داد و یا این که مخرج کسر را به وسیله اعمال فیلتر میانگین^۲ کاهش داد.

فیلترهای میانگین از معمول‌ترین ابزارهای هموار و ملایم^۳ کردن تصاویر دیجیتال و از بین بردن نویزهای تصویر محسوب می‌شوند. با اعمال فیلتر میانگین مقدار $1/F$ در هر سلول برابر میانگین سلول‌های همسایه آن سلول است. فرض کنید سلول به مختصات (x, y) در مرکز پنجره S_{xy} به طول m و عرض n سلول قرار دارد. با اعمال فیلتر میانگین، مقدار $1/F$ در سلول (x, y) از رابطه (۸) به دست می‌آید:

$$f(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)}{m \cdot n} \quad (8)$$

¹ Steering radius

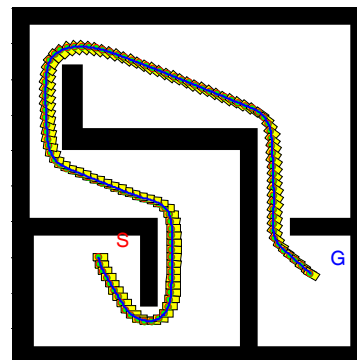
² Average filter

³ Image smoothing

۴-۲- اعمال محدودیت غیرهولونومیک

همان‌طور که اشاره شد می‌توان با استفاده از روش پیشروی سریع یک مسیر با انحنا کافی ایجاد کرد به نحوی که یک روبات ماشین‌واره بتواند مماس بر آن حرکت کند. در واقع تضمین می‌شود که انحنا هیچ نقطه‌ای از مسیر بزرگ‌تر از انحنا قابل پیمایش توسط روبات نیست. تنها مشکل باقیمانده، حرکات رو به پهلوئی است که روبات در دو پیکربندی شروع و پایان خواهد داشت.

همان‌طور که در شکل ۷ نشان داده شده است مسیر ایجاد شده برای روبات یک مسیر هموار با انحنا کافی است اما حرکت رو به پهلو در نقطه شروع مسیر S مانع از آن می‌شود که مسیر به‌دست آمده برای روبات ماشین‌واره شدنی و قابل پیمایش باشد.

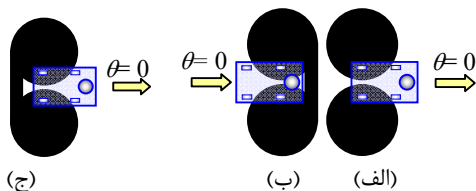


شکل ۷- مسیریابی روبات هولونومیک.

به‌منظور جلوگیری از حرکات به سمت طرفین روبات در دو نقطه شروع و پایان، از مفهوم جدیدی که آن را *موانع مجازی* نامگذاری کرده‌ایم استفاده می‌شود. برای روبات‌های ماشین‌واره R&S که قادر به حرکت به جلو و عقب می‌باشند، موانع مجازی به شکل دو دایره در طرفین نقاط شروع و پایان روبات قرار می‌گیرند تا از حرکات جانبی آن جلوگیری کنند. در عین حال، شعاع این دوایر به اندازه حداقل شعاع قابل پیمایش روبات در نظر گرفته می‌شود تا در صورت نیاز روبات بتواند از همان ابتدا چرخشی مماس بر آن دوایر انجام دهد (شکل ۸-الف).

موانع مجازی برای روبات ماشین‌واره Dubins که تنها مجاز به حرکت رو به جلو می‌باشد به شکل دو دایره می‌باشند که از پشت به همدیگر متصل شده‌اند و بدین ترتیب به‌طور خودکار مانع حرکت روبات به عقب خواهند شد. شکل ۸-ب و ج دو مانع مجازی برای نقاط شروع و پایان روبات Dubins را نشان می‌دهند. جزئیات بیشتر در مورد موانع مجازی در مرجع [۲۹] آورده شده است.

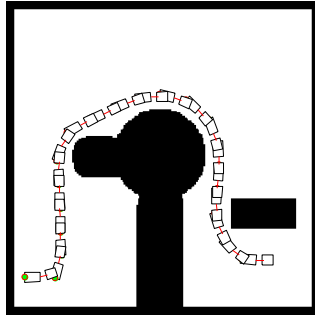
در الگوریتم مانع مجازی، ابتدا به‌وسیله FMM و موانع مجازی یک مسیر هموار برای روبات هولونومیک ایجاد می‌شود، و سپس مسیر به‌دست آمده برای روبات هولونومیک به‌وسیله منحنی اسپلاین برازش می‌شود تا یک مسیر ایمن برای روبات غیرهولونومیک ماشین‌واره ایجاد شود. در شکل ۹-الف اضافه شدن موانع مجازی در دو نقطه شروع و پایان باعث تولید یک مسیر شدنی برای روبات ماشین‌واره Dubins می‌شود.



شکل ۸- (الف) مانع مجازی روبات R&S؛ (ب) مانع مجازی روبات Dubins در نقطه پایان حرکت؛ (ج) مانع مجازی روبات Dubins در نقطه شروع حرکت.

۵- شبیه‌سازی و تحلیل حساسیت

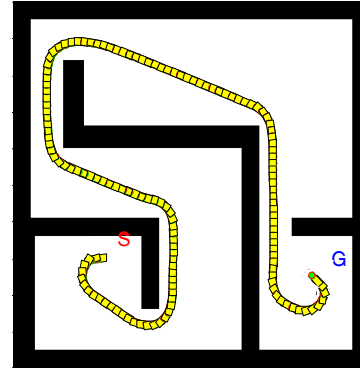
نتایج شبیه‌سازی روش ارائه شده برای روبات ماشین‌واره (بدون یک‌کش) در شکل ۹ نشان داده شده است. در شکل ۱۰ نیز مراحل شبیه‌سازی الگوریتم مسیریابی برای یک روبات یک‌کش (با یک یک‌کش) نشان داده شده است.



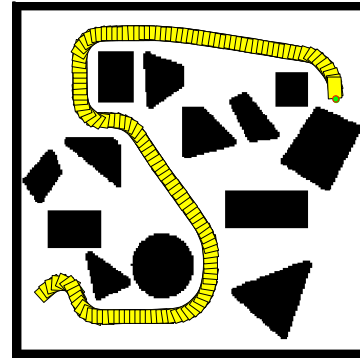
(ج)

شکل ۱۰- (الف) مسیریابی در حالت هولونومیک و بدون حضور موانع مجازی؛ (ب) حرکت روبات یدک‌کش در حضور موانع مجازی؛ (ج) شبیه‌سازی روبات یدک‌کش.

همان‌طور که از رابطه ۶ پیداست اندازه معادل ES ارتباط مستقیمی با جذر تعداد یدک‌ها (N) دارد. به‌منظور ارزیابی تغییرات تعداد یدک‌ها بر اندازه معادل و مسیر به‌دست آمده برای روبات یدک‌کش، تحلیل حساسیتی برای پارامتر N انجام شد بدین ترتیب که یک مسئله واحد برای تعداد ۱ تا ۱۰ یدک حل شد (گرچه در عمل تعداد یدک‌ها بسیار کمتر می‌باشد). شکل ۱۱ پیکربندی اولیه و نهایی این مسئله را نشان می‌دهد.

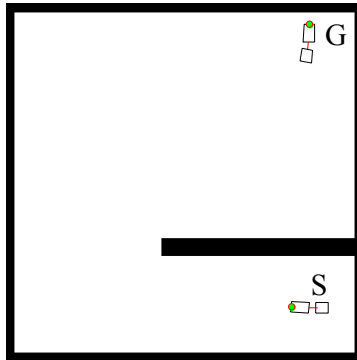


(الف)

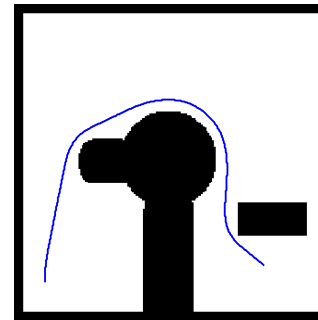


(ب)

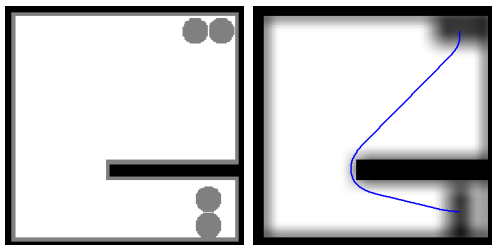
شکل ۹- (الف) اصلاح مسیر به‌دست آمده در شکل (۷) با اعمال موانع مجازی و ارضای محدودیت غیرهولونومیک روبات ماشین‌واره Dubins؛ (ب) مثالی دیگر از مسیریابی روبات ماشین‌واره.



شکل ۱۱- مسئله مسیریابی بین دو پیکربندی $S(x_1, y_1, \pi)$ و $G(x_2, y_2, \pi/2)$.

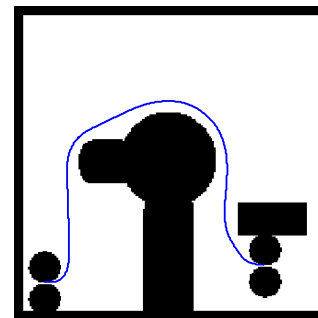


(الف)

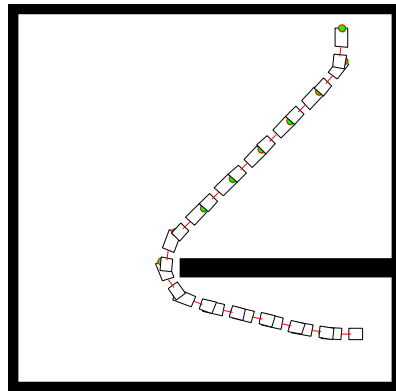


(ب)

(الف)



(ب)

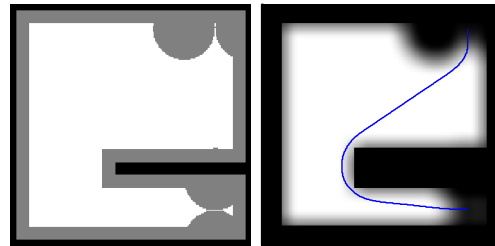


(ب)

شکل ۱۳- (الف) مسیر به‌دست آمده از یک تا ده یدک؛ (ب) شبیه‌سازی حرکت روبات یک یدک بر روی مسیر.

پس از حل این مسئله برای تعداد حالات از یک یدک تا ده یدک، مسیرهای متفاوتی با انحنای متفاوت ایجاد شده که مجموعه آن‌ها در شکل ۱۳-الف نشان داده شده‌اند. در این شکل، مسیرهای حالات از یک یدک تا ده یدک به ترتیب از راست به چپ نمایش داده شده‌اند که در آن می‌توان افزایش شعاع حداقل مسیر را به تناسب افزایش تعداد یدک‌ها مشاهده کرد. شکل ۱۳-ب نیز شبیه‌سازی حرکت روبات یدک‌کش با یک یدک را نمایش می‌دهد.

برای کلیه حالات ۱ تا ۱۰ یدکی، مقادیر اندازه معادل، شعاع حداقل روبات یدک‌کش و طول مسیر به‌دست آمده در روش پیشروی سریع مطابق جدول ۱ حاصل شد. در شکل ۱۱ نمودار روند صعودی اندازه معادل روبات و شعاع حداقل حرکت روبات نشان داده شده است، که طبق رابطه (۶) با جذر تعداد یدک‌ها متناسب است.

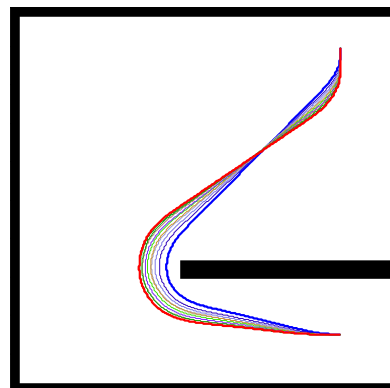


(د)

(ج)

شکل ۱۲- حل مسئله شکل ۱۱: (الف) فضای کاری (سیاه) و فضای پیکربندی (خاکستری) روبات یدک‌کش با یک یدک؛ (ب) اعمال فیلتر میانگین و مسیریابی؛ (ج) فضای کاری (سیاه) و فضای پیکربندی (خاکستری) روبات یدک‌کش با ده یدک؛ (د) اعمال فیلتر میانگین و مسیریابی.

شکل‌های ۱۲-الف و ۱۲-ب حل این مسئله را برای یک یدک به تصویر می‌کشند، که در شکل ۱۲-الف فضای پیکربندی (خاکستری) با بزرگ‌تر شدن موانع به اندازه $ES/2$ و نیز افزوده شدن موانع مجازی (دوایر) ایجاد شده، و در شکل ۱۲-ب پس از اعمال فیلتر میانگین بر روی فضای پیکربندی و اجرای الگوریتم، مسیر نهایی پیدا شده است. شکل‌های ۱۲-ج و ۱۲-د همین مراحل را برای حالت ده یدکی نشان می‌دهند.



(الف)

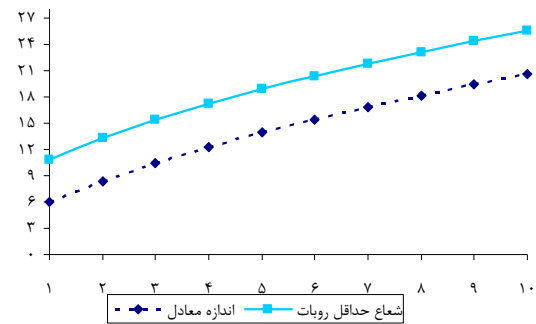
جدول ۱- نتایج به‌دست آمده از حل الگوریتم مسیریابی روبات یدک‌کش با تعداد یدک‌های متفاوت.

پیشروی سریع		اندازه معادل	تعداد یدک
طول مسیر	شعاع حداقل		
۲۴۴,۱۰۵	۱۰,۸۸۹	۵,۹۲۹	۱
۲۵۱,۸۶۶	۱۳,۳۳۶	۸,۳۸۶	۲
۲۵۶,۴۹۷	۱۵,۴۰۰	۱۰,۴۵۰	۳
۲۶۰,۸۱۴	۱۷,۲۱۷	۱۲,۲۶۷	۴

۲۶۱,۵۸۹	۱۸,۸۶۱	۱۳,۹۱۱	۵
۲۶۶,۰۲۶	۲۰,۳۷۲	۱۵,۴۲۲	۶
۲۶۹,۱۸۱	۲۱,۷۷۸	۱۶,۸۲۸	۷
۲۷۳,۰۵۴	۲۳,۱۰۰	۱۸,۱۵۰	۸
۲۷۴,۱۳۲	۲۴,۳۴۹	۱۹,۳۹۹	۹
۲۷۷,۱۵۴	۲۵,۵۳۸	۲۰,۵۸۸	۱۰

مطابق جدول ۱، با اضافه شدن هر یک طول مسیر نیز افزایش می‌یابد که دلیل آن افزایش ابعاد موانع در ازای بزرگ‌تر شدن اندازه معادل ربات یدک‌کش و نیز بزرگ‌تر شدن ابعاد موانع مجازی به موجب افزایش شعاع انحنای مسیر است. در شکل ۱۳-الف، ده مسیر ممکن نشان داده شده است و در آن مسیرهای با یک یدک و ۱۰ یدک ضخیم‌تر نشان داده شده‌اند.

مطابق جدول ۱، با اضافه شدن هر یک طول مسیر نیز افزایش می‌یابد که دلیل آن افزایش ابعاد موانع در ازای بزرگ‌تر شدن اندازه معادل ربات یدک‌کش و نیز بزرگ‌تر شدن ابعاد موانع مجازی به موجب افزایش شعاع انحنای مسیر است. در شکل ۱۳-الف، ده مسیر ممکن نشان داده شده است و در آن مسیرهای با یک یدک و ۱۰ یدک ضخیم‌تر نشان داده شده‌اند.



شکل ۱۴- روند تغییرات ES و شعاع حداقل ربات یدک‌کش.

۶- مقایسه با روش جستجوی شبکه‌ای

جهت ارزیابی عملکرد الگوریتم پیشروی سریع برای ربات‌های یدک‌کش، مقایسه‌ای با روش‌های جستجوی شبکه‌ای [۸] و RRT غیرهولونومیک [۳۰] انجام گرفته است.

۶-۱- مقایسه با الگوریتم جستجوی شبکه‌ای^۱

الگوریتم جستجوی شبکه‌ای در سال ۱۹۹۳ توسعه داده شده و جزء اولین و شناخته شده‌ترین الگوریتم‌های مسیریابی ربات‌های ماشین‌واره و یدک‌کش در حضور موانع می‌باشد.

در این روش شش حرکت برای ربات ماشین‌واره در نظر گرفته می‌شود: دو حرکت برای گردش به چپ با شعاع

حداقل ربات، دو حرکت برای گردش به راست با شعاع حداقل ربات، و دو حرکت مستقیم. حرکات گردشی و مستقیم ربات می‌توانند در دو جهت رو به جلو و یا رو به عقب باشند و به ترتیب با $L \pm$ و $R \pm$ و $S \pm$ نشان داده می‌شوند که علامات مثبت و منفی به ترتیب بیان‌گر حرکت به سمت جلو و عقب ربات ماشین‌واره است.

در این الگوریتم، یک درخت T از نقاط قابل دسترس از نقطه جاری به تدریج ساخته می‌شود به طوری که در هر تکرار، شش کمان جدید که متناسب با شش حرکت مذکور هستند به درخت اضافه می‌شوند (البته نقاطی که در داخل موانع می‌افتند حذف می‌شوند).
گره‌هایی از درخت که هنوز به آن‌ها رجوع نشده است (یعنی گسترش^۲ نیافته‌اند) در لیستی به نام OPEN ذخیره شده و براساس یک تابع هزینه به صورت صعودی مرتب می‌شوند. الگوریتم از نقطه ابتدایی q_{start} شروع کرده و آن را به عنوان ریشه^۳ درخت در نظر می‌گیرد و لیست OPEN را براساس نقاط قابل دسترس از آن تعریف می‌کند. سپس نقطه‌ای از لیست OPEN که کمترین هزینه را داراست انتخاب و گسترش داده می‌شود و بعد از لیست حذف می‌شود. نقاط جدیدی که از گسترش این نقطه به دست می‌آیند به درخت T و به لیست OPEN اضافه می‌شوند.

علاوه بر ساخت درخت، محیط بایستی به شکل منظمی سلول‌بندی شود، که اندازه هر سلول توسط کاربر تعیین می‌شود. نقش این سلول‌بندی آن است که هنگام گسترش درخت، اگر یک نقطه جدید داخل سلولی بیفتد که قبلاً توسط یک گره اشغال شده است، آن نقطه جدید به درخت و لیست OPEN اضافه نشده و در نتیجه در هر سلول حداکثر

² Expand

³ Root

¹ Grid search

$$Cost = \sum_m a \cdot m + \sum_k b \cdot k + \sum_r c \cdot r \quad (9)$$

مسئله نشان‌داده شده در شکل ۱۱ با الگوریتم جستجوی شبکه‌ای نیز حل شده و مقادیر طول مسیر و میزان خطای آن (یعنی فاصله نزدیک‌ترین گره تا پیکربندی هدف) در جدول ۲ آورده شده‌اند.

با مقایسه طول مسیرهای به‌دست آمده از دو الگوریتم پیشروی سریع و جستجوی شبکه‌ای با یکدیگر در شکل ۱۵، مشاهده می‌شود که در تمامی حالات طول مسیر به‌دست آمده از الگوریتم پیشروی سریع کوتاه‌تر از الگوریتم جستجوی شبکه‌ای است. علاوه بر آن، الگوریتم پیشروی سریع قادر است به‌طور دقیق و با جهت‌گیری خواسته شده به پیکربندی هدف دست پیدا کند درحالی‌که الگوریتم جستجوی شبکه‌ای تنها یک همسایگی از پیکربندی هدف (و نه خود هدف) را پیدا می‌کند و هیچ کنترلی بر جهت‌گیری نهایی روبات ندارد (یعنی روبات با جهت خواسته شده به پیکربندی هدف نمی‌رسد). شکل ۱۶-الف و ۱۶-ب حل مسئله شکل ۱۱ با یک و ده یدک را با روش جستجوی شبکه‌ای نشان می‌دهد.

یک گره درخت قرار خواهد داشت. در واقع، از گسترش نقاطی که در همسایگی نقطه‌ای هستند که قبلاً زیر شاخه‌های آن گسترش پیدا کرده است جلوگیری می‌شود. اندازه سلول‌های شبکه باید به حدی کوچک انتخاب شود که فضای پیکربندی با دقت مناسبی جستجو شود و اندازه همسایگی پیکربندی هدف نیز باید به حدی بزرگ انتخاب شود که حداقل یک گره از درخت بتواند در آن واقع شود.

جستجو و گسترش درخت تا وقتی ادامه می‌یابد که یکی از دو شرط توقف اتفاق بیفتد: (۱) لیست OPEN خالی شود و یا تعداد گره‌های درخت T از مقدار تعیین شده توسط کاربر بیشتر شود؛ یا (۲) یک نقطه در همسایگی نقطه هدف پیدا شود که اندازه این همسایگی توسط کاربر تعریف می‌شود.

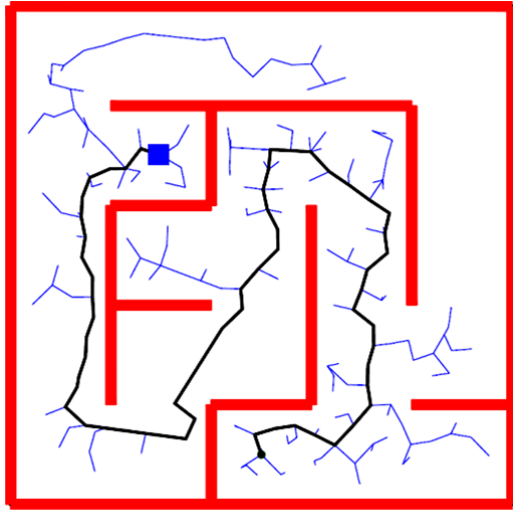
تابع هزینه الگوریتم جستجوی شبکه‌ای در رابطه (۹) نشان داده شده است، که تابعی از تعداد حرکات روبات (m)، تعداد تغییرات جهت فرمان (k) و تعداد تغییرات جهت حرکت (r) (یعنی برعکس شدن علامت حرکت) است، و a ، b و c به ترتیب بیان‌گر هزینه یک حرکت، هزینه تغییر جهت فرمان، و هزینه یک تغییر علامت حرکت هستند:

جدول ۲- مقایسه الگوریتم‌های مسیریابی پیشروی سریع و جستجوی شبکه‌ای برای حل مسئله شکل ۱۱.

جستجوی شبکه‌ای		پیشروی سریع		اندازه معادل	تعداد یدک
خطا	طول مسیر	طول مسیر	شعاع حداقل		
۲,۳۹۳	۲۸۲,۲۳۴	۲۴۴,۱۰۵	۱۰,۸۸۹	۵,۹۳۹	۱
۲,۴۳۲	۲۹۳,۲۹۱	۲۵۱,۸۶۶	۱۳,۳۳۶	۸,۳۸۶	۲
۲,۷۴۸	۲۹۰,۲۸۳	۲۵۶,۴۹۷	۱۵,۴۰۰	۱۰,۴۵۰	۳
۱,۲۹۵	۲۹۷,۵۰۰	۲۶۰,۸۱۴	۱۷,۲۱۷	۱۲,۲۶۷	۴
۱,۶۳۶	۳۲۵,۸۹۵	۲۶۱,۵۸۹	۱۸,۸۶۱	۱۳,۹۱۱	۵
۲,۱۳۲	۳۶۸,۰۰۸	۲۶۶,۰۲۶	۲۰,۳۷۲	۱۵,۴۲۲	۶
۳,۰۷۷	۲۹۰,۷۸۶	۲۶۹,۱۸۱	۲۱,۷۷۸	۱۶,۸۲۸	۷
۳,۸۱۵	۳۰۸,۴۲۵	۲۷۳,۰۵۴	۲۳,۱۰۰	۱۸,۱۵۰	۸
۴,۶۶۲	۳۰۵,۹۸۵	۲۷۴,۱۳۲	۲۴,۳۴۹	۱۹,۳۹۹	۹
۱,۰۶۸	۳۲۰,۹۲۰	۲۷۷,۱۵۴	۲۵,۵۳۸	۲۰,۵۸۸	۱۰

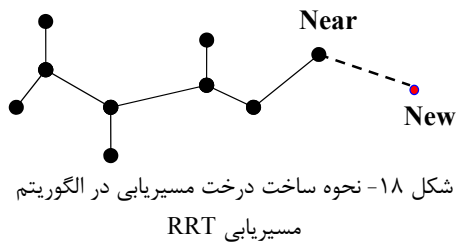
۲-۶- مقایسه با الگوریتم RRT

الگوریتم درخت سریع تصادفی^۱ یکی از الگوریتم‌های نمونه‌گیری تصادفی است که از یک پیکربندی منحصر به فرد شروع می‌شود و با گسترش یک درخت کل فضای پیکربندی را مورد پوشش قرار می‌دهد (شکل ۱۷) [۳۰].

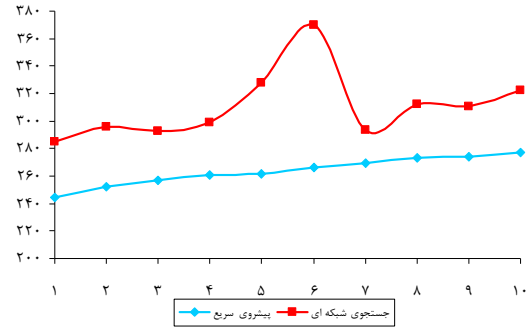


شکل ۱۷- الگوریتم مسیریابی RRT

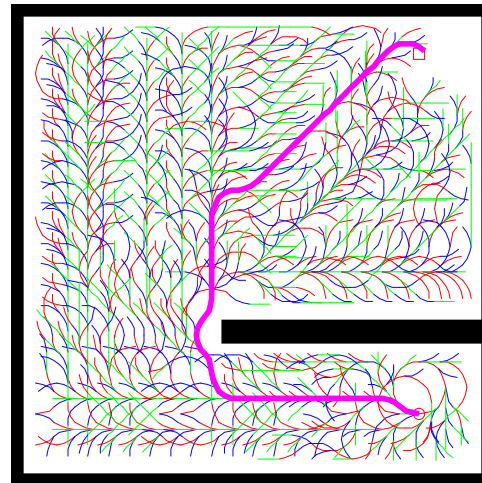
الگوریتم RRT در حالت هولونومیک نقطه شروع به‌عنوان ریشه در نظر گرفته می‌شود و نقاط فضای پیکربندی به‌صورت تصادفی انتخاب می‌شوند و به نزدیکترین گره از درخت گسترش داده شده وصل می‌شوند و این کار تا پوشش کامل فضای پیکربندی انجام می‌شود و در پایان نقطه هدف به درخت وصل شده و با برگشت شاخه به شاخه از پیکربندی هدف به پیکربندی شروع می‌رسد. نحوه انجام الگوریتم در شکل ۲۵ نشان داده شده است.



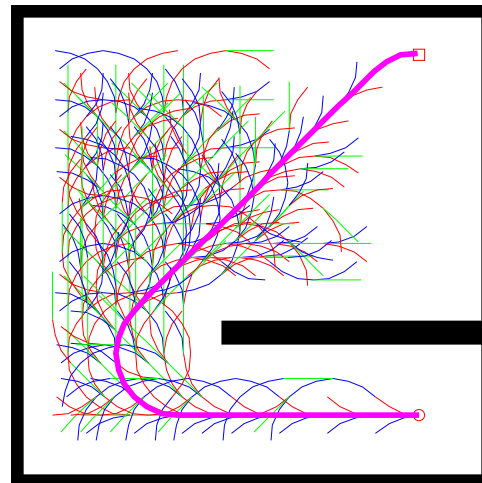
شکل ۱۸- نحوه ساخت درخت مسیریابی در الگوریتم مسیریابی RRT



شکل ۱۵- تغییرات طول مسیرهای به‌دست آمده به ازای تعداد یدک‌های متفاوت.



(الف)



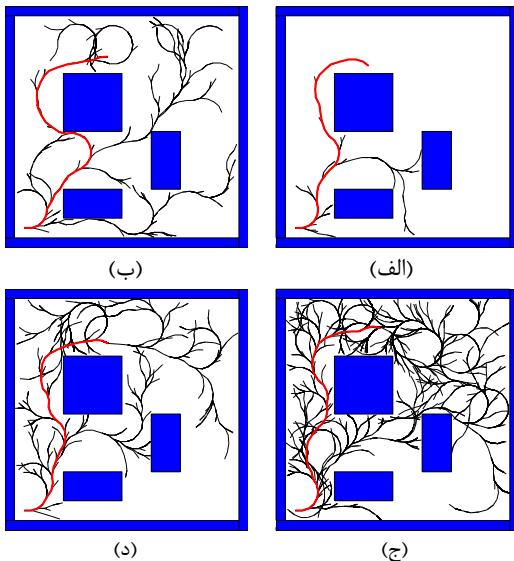
(ب)

شکل ۱۶- (الف) حل مسئله با یک یدک با روش جستجوی شبکه ای با شعاع حداقل ۱۰,۸۸۹ واحد. (ب) حل مسئله با ده یدک با روش جستجوی شبکه ای با شعاع حداقل ۲۵,۵۳۸ واحد.

^۱ Rapidly-exploring Random Tree

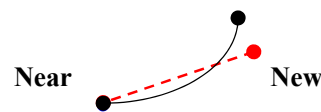
خواهد کرد. در شکل ۲۱ هدف الگوریتم رسیدن به همسایگی یک واحدی پیکربندی هدف است.

مساله‌ای که در شکل‌های ۲۰ و ۲۱ توسط الگوریتم RRT حل شده است توسط الگوریتم VO نیز حل شده که در شکل ۲۲ نشان داده شده است. همان‌طور که در دو شکل ۲۰ و ۲۱ نشان داده شده است الگوریتم RRT به دلیل داشتن ماهیت احتمالی در اجراهای مختلف یک مساله قادر نیست که به جوابی یکسان در زمانی مشخص دست پیدا کند. علاوه بر آن الگوریتم RRT به یک همسایگی از نقطه هدف می‌رشد که هر قدر این همسایگی کوچک‌تر باشد احتمال افزایش زمان محاسبه بیشتر خواهد بود. حداقل زمان جهت محاسبه جواب توسط RRT، $3/716$ ثانیه است و این در حالی است که الگوریتم VO این مساله را در $0/015$ ثانیه با سلول‌بندی 100×100 حل کرده است.



شکل ۲۰- حل یک مساله مسیریابی با الگوریتم RRT با فاصله حداکثر دو واحدی از پیکربندی هدف در فضای پیکربندی دو بعدی؛ الف- زمان حل $3/716$ ثانیه؛ ب- زمان حل $5/403$ ثانیه؛ ج- زمان حل $15/148$ ثانیه؛ د- زمان حل $37/893$ ثانیه.

الگوریتم RRT برای حالت غیرهولونومیک نیز گسترش داده شده است که قادر است به پیکربندی نهایی و یا منطقه‌ای در همسایگی آن برسد. در حالت غیرهولونومیک الگوریتم بعد از انتخاب پیکربندی تصادفی و پیدا کردن نزدیکترین پیکربندی به آن، نزدیکترین نقطه ممکن به پیکربندی جدید را به‌نحوی پیدا می‌کند که مسیر به‌دست آمده بتواند محدودیت غیرهولونومیک روبات را ارضا کند [۳۰]. نحوه اعمال الگوریتم RRT غیرهولونومیک در شکل ۱۹ نشان داده شده است.



شکل ۱۹- نحوه ساخت درخت مسیریابی در الگوریتم مسیریابی RRT در حالت غیرهولونومیک

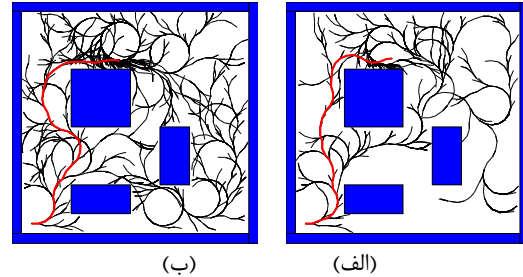
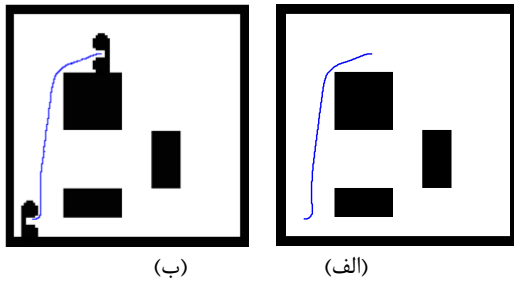
به دلیل تصادفی بودن الگوریتم RRT زمان حل آن نیز کاملاً تصادفی خواهد بود. در شکل ۲۰ یک مساله به‌عنوان نمونه با الگوریتم RRT حل شده که هدف رسیدن به همسایگی نقطه هدف با فاصله حداکثر دو واحد است. میزان فاصله از تابع زیر محاسبه می‌شود.

$$D(q, q_g) = \sqrt{(x - x_g)^2 + (y - y_g)^2 + (\theta - \theta_g)^2} \quad (10)$$

در رابطه ۱۰، $q_g = (x_g, y_g, \theta_g)$ بیان‌گر پیکربندی هدف است و $q = (x, y, \theta)$ نشان‌دهنده پیکربندی جدید درخت RRT است. در صورتی که $D(q, q_g)$ کوچک‌تر از دو باشد الگوریتم متوقف می‌شود و مسیر تعیین می‌شود.

در شکل ۲۰ تمامی ۴ مساله یکسان است اما به دلیل تصادفی بودن الگوریتم RRT زمان حل الگوریتم برای هر ۴ مساله متفاوت است. میزان فاصله تا پیکربندی هدف نیز بر زمان حل تاثیرگذار است. در شکل ۲۰ میزان فاصله تا پیکربندی هدف دو واحد در نظر گرفته شده است در صورتی که میزان فاصله تا پیکربندی هدف به یک واحد کاهش یابد احتمال افزایش زمان الگوریتم افزایش پیدا

توسط الگوریتم RRT حل شده است که نتایج آن در جدول ۳ نشان داده شده است.



شکل ۲۱- حل یک مساله مسیریابی با الگوریتم RRT با فاصله حداکثر فاصله یک واحدی از پیکربندی هدف در فضای پیکربندی دو بعدی؛ الف- زمان حل ۳۲/۳۷۴ ثانیه؛ ب- زمان حل 38.218 ثانیه.

شکل ۲۲- حل مساله حل نشان داده شده در شکل های ۲۰ و ۲۱ توسط VO در زمان ۰/۰۱۵ ثانیه؛ الف- مسیر به دست آمده توسط VO در فضای پیکربندی دو بعدی؛ ب- موانع مجازی اضافه شده در فضای پیکربندی دو بعدی جهت حل مساله.

مساله نشان داده شده در دو شکل ۲۰ و ۲۱ با در نظر گرفتن حداکثر فاصله دو واحد از پیکربندی هدف، ده بار

جدول ۳- زمان حل الگوریتم RRT

مسأله	۱	۲	۳	۴	۵
زمان(ثانیه)	۵/۶	۳/۹	۱۴/۰	۴/۱	۳/۷
۶	۷	۸	۹	۱۰	میانگین
۱۱/۹	۵/۴	۳۷/۸	۷/۹	۱۵/۱	۱۱/۰۴

مسیریابی ربات ماشین‌واره می‌شود و سپس با افزودن موانع مجازی به فضای پیکربندی، محدودیت غیرهولونومیک ربات لحاظ شده و در نتیجه با اجرای روش پیشروی سریع، مسیری شدنی و بهینه برای ربات یک‌کش (با تعداد یک‌های دلخواه) پیدا می‌شود. روش مسیریابی ارائه شده در مقایسه با روش‌های دیگر، سریع، دقیق و آسان است.

پیدا کردن یک مسیر با انحنای مناسب از مزایای اصلی روش FMM است و باعث بالا رفتن سرعت انجام الگوریتم ارائه شده می‌شود، چرا که در بسیاری از الگوریتم‌های ارائه شده ابتدا یک مسیر هولونومیک پیدا می‌شود و سپس مسیر با توجه به محدودیت غیرهولونومیک تغییر داده می‌شود اما در الگوریتم مذکور مسیر ایجاد شده دارای انحنای کافی است. علاوه بر آن با استفاده از مفهوم مانع مجازی یک بعد از فضای حل مساله کاهش می‌یابد که باعث افزایش سرعت الگوریتم می‌شود. به عنوان مثال اگر ابعاد مساله در حالت سه بعدی به $۱۰۰ \times ۱۰۰ \times ۳۶$ تقسیم شود

میانگین زمان حل الگوریتم RRT، ۱۱/۰۴ ثانیه است و الگوریتم VO با سلول‌بندی ۱۰۰×۱۰۰ این مساله را در ۰/۰۱۵ ثانیه حل کرده است.

باتوجه به مطالب ارائه شده در اشکال ۲۰ تا ۲۲ و جدول ۳ می‌توان نتیجه گرفت که زمان حل الگوریتم VO بسیار کوتاه‌تر از زمان حل الگوریتم RRT است و جواب به دست آمده از روش VO دقیق‌تر و کوتاه‌تر از جواب الگوریتم RRT است.

۷- نتیجه‌گیری

در این مقاله روش جدیدی برای مسیریابی ربات‌های یک‌کش با استفاده از روش پیشروی سریع (FMM) و مفاهیم اندازه معادل (ES) و موانع مجازی معرفی شده است. ابتدا با افزایش ابعاد موانع به اندازه $ES/2$ ، مسئله برنامه‌ریزی حرکت ربات یک‌کش تبدیل به مسئله

در الگوریتم مذکور تعداد سلول‌های تقسیم‌بندی شده مساله 100×100 است. این الگوریتم مستقل از شکل موانع است که خود مزیتی نسبت به تعدادی از الگوریتم‌های دیگر به شمار می‌آید. از توسعه‌های آتی الگوریتم، طراحی آن برای حالت بهنگام (آنلاین) می‌باشد.

۸- مراجع

- [1] Shuzhi, S.G. (2006). "Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications". CRC Press.
- [2] Lozano-Pérez, T., Wesley, M.A. (1979). "An algorithm for planning collision-free paths among polyhedral obstacles". *Communications of ACM*, Vol. 22, No. 10, pp. 560-570.
- [3] Vasseur, H.A., Pin, F.G., Taylor, J.R. (1991). "Navigation of car-like mobile robot using a decomposition of the environment in convex cell". *Proceedings of the Robotics and Automation IEEE International Conference*, pp. 1496-1502.
- [4] Latombe, J.C. (1991). "Robot Motion Planning", Kluwer Academic Publishers.
- [5] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S. (2005). "Principle of Robot Motion :Theory, Algorithms, and Application". MIT Press, Cambridge.
- [6] Dubins, L.E. (1957). "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, pp. 497-516.
- [7] Reeds, J.A., Shepp, L.A. (1990). "Optimal paths for a car that goes both forwards and backwards". *Pacific Journal of Mathematics*, Vol. 145, No. 2, pp. 367-393.
- [8] Barraquand, J., Latombe, J.C. (1993). "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles". *Algorithmica*, Vol. 10, No. 2, pp. 121-155.
- [9] Kuffner, J.J., LaValle, S.M. (2002). "RRT-connect: An efficient approach to single-query path planning". *Proceedings of the IEEE International Conf. on Robotics and Automation*, Vol. 2, pp. 995-1001.
- [10] Laumond, J.P., Jacobs, P.E., Taix, M., Murray, R.M. (1994). "A motion planner for nonholonomic mobile robots". *Robotics and Automation, IEEE Transactions*, Vol. 10, No. 5, pp. 577-593.
- [11] Sánchez, A.L., Zapata, R., Arenas, A.B. (2002). "Motion planning for car-like robots using lazy probabilistic roadmap method". *Proceedings of the MICAI 2002*, pp. 1-10.
- [12] Guang, S., Nancy, A.M. (2002). "Randomized motion planning for car-like robots with C-PRM". *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Vol. 1, pp. 37-42.
- [13] LaValle, S.M. (2006). "Planning Algorithms". Cambridge University Press.
- [14] Svestka, P., Vleugels, J. (1995). "Exact motion planning for tractor-trailer robots". *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2445-2450.
- [15] Sekhavat, S., Svestka, P., Laumond, J.P., Overmars, M.H. (1998). "Multilevel path planning for non-holonomic robots using semiholonomic subsystems". *The International Journal of Robotics Research*, Vol. 17, No. 8, pp. 840-857.
- [16] Yuan, J., Huang, Y., Sun, F., Kang, Y. (2004). "Computation of equivalent size for tractor-trailer wheeled mobile robot". *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pp. 4788-4792.
- [17] Liu, Z.J., Huang, P., Huang, J.L., Que, J.L. (2004). "Path planning for tractor-trailer mobile robot based on heuristic genetic algorithm". *Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 2, pp. 1119-1124.
- [18] Sun, F., Huang, Y., Yuan, J., Kang, Y., Ma, F. (2007). "A Compound PRM Method for Path Planning of the Tractor-Trailer Mobile Robot". *Proceedings of the IEEE International Conference on Automation and Logistics*, pp. 1880-1885.

- [19] <http://math.berkeley.edu/~sethian/2006/Applications/Robotics/robotics.html>.
- [20] Farag, A.A., Hassouna, M.S. (2005). "Theoretical foundations of tracking monotonically advancing fronts using fast marching level set method". Technical Report, Computer Vision and Image Processing Lab., ECE Dept., University of Louisville.
- [21] Sethian, J.A., (1996). "A fast marching level set method for monotonically advancing fronts". Proceedings of the National Academy of Science, USA, Vol. 93, No. 4, pp. 1591–1595.
- [22] Li, Y. (2008). "Real-time motion planning of multiple agents and formations in virtual environments". PhD Thesis, Simon Fraser University.
- [23] Chiang, C.H., Chiang, P.J., Fei, J.C.C., Liu, J.S. (2007). "A comparative study of implementing fast marching method and A* search for mobile robot path planning in grid environment: effect of map resolution". Proceedings of the ARSO 2007, Advanced Robotics and Its Social Impacts 2007, IEEE Workshop, pp.1-6.
- [24] Garrido, S., Moreno, L., Blanco, D., Martin, F. (2007). "Exploratory navigation based on Voronoi transform and fast marching". Proceedings of the WISP 2007, Intelligent Signal Processing 2007, IEEE International Symposium, pp.1-6.
- [25] Chiang, C.H., Liu, J.S. (2008). "Boundary following in unknown polygonal environment based on fast marching method". Proceedings of the ARSO 2008, Advanced Robotics and Its Social Impacts 2008, IEEE Workshop, pp.1-6.
- [26] Cohen, L.D., Kimmel, R. (1997). "Global minimum for active contour models: a minimal path approach". Int. Journal of Computer Vision, Vol. 24, No. 1, pp. 57–78.
- [27] Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D. (2007). "Path planning for autonomous underwater vehicles". IEEE Transaction on Robotics, Vol. 23, No. 2, pp. 331–341.
- [28] Gonzalez, R.C., Woods, R.E. (1993). "Digital Image Processing", Addison-Wesley, ISBN 0-201-60078-1, pp. 231–235.
- [۲۹] جنت، د. (۱۳۸۹). "برنامه‌ریزی حرکت بهنگام روبات‌های ماشین‌واره منفرد و چندتایی با استفاده از روش پیشروی سریع (FMM)". پایان نامه کارشناسی ارشد، دانشگاه تربیت مدرس.
- [30] Cheng P., Shen Z., LaValle S.M. (2001). "RRT-based trajectory design for autonomous automobiles and spacecraft". Archives of Control Sciences, Vol. 11, No. 3-4, pp. 167-194.