

جابه‌جایی اجسام توسط یک سیستم چندرباتی توزیع‌شده با استفاده از یک معماری ترکیبی

طاہر حکمت‌فر و الیپس مسیحی

بین‌ربات‌ها است. در بحث همکاری، عامل‌ها سعی می‌کنند تا با ترکیب مناسب عملیات خود به عنوان یک گروه واحد، یک وظیفه خاص را انجام دهند. ارتباط بین عامل‌ها نیز به عنوان مکانیزمی ساده و کارا به هماهنگی رفتار آنها کمک می‌کند تا بتوانند به عنوان سیستمی هماهنگ و منسجم وظیفه محول را به طور همکارانه انجام دهند [۲].

در بیشتر تحقیقات انجام‌شده در این حوزه محیط کار دوعدی فرض شده و از رانش برای جابه‌جایی اجسام استفاده شده است [۳] و [۴] اما چنین محدودیت‌هایی مانع استفاده از نتایج تحقیقات در محیط واقعی می‌شود. از طرفی محیط کاری واقعی عموماً حاوی موانع متعددی است که در برخی تحقیقات مانند [۵] و [۶] از وجود این موانع نیز صرف نظر شده است. معماری مورد استفاده در سیستم به کار رفته در بیشتر تحقیقات متمرکز است. چنین سیستم‌هایی علی‌رغم فراهم کردن مزایایی مانند سادگی و تولید راه حل بهینه، مشکلاتی در رابطه با مقیاس‌پذیری و تحمل‌پذیری خطای پایین، توان پردازشی و پهنای باند محدود دارند. در سیستم‌های غیر متمرکز با عامل‌های کاملاً خودمختار نیز با وجود افزایش کارایی و مقیاس‌پذیری در حل مسأله، به دلیل عدم دسترسی به اطلاعات سراسری خطر رخداد بن‌بست و عدم وصول جواب بهینه وجود دارد.

در تحقیق پیش رو سعی شده است با استفاده از یک سیستم چندرباتی همگن متشکل از ربات‌های KUKA youBot یک جسم جابه‌جاشونده سنگین در محیط سه‌بعدی حاوی موانع ایستا جابه‌جا شود. رویکرد پیشنهادی برای حل این مسأله، یک رویکرد ترکیبی است که هدف آن بهره‌گیری از مزایای معماری متمرکز و توزیع‌شده به صورت هم‌زمان است. با این کار ربات‌ها می‌توانند با هزینه پردازشی کمتر و بدون گیرکردن در وضعیت بن‌بست به یک جواب نسبتاً بهینه برسند. در بخش دوم مقاله، مسأله جابه‌جایی اجسام توسط سیستم چندرباتی در محیط حاوی موانع تعریف شده و فرضیات مربوط به مسأله بیان می‌شود. در این فصل به کارگیری از سیستم‌های چندرباتی و روش‌های مورد استفاده در جابه‌جایی اجسام در تحقیقات مختلف مورد بررسی قرار می‌گیرد. در بخش سوم معماری توسعه داده شده برای حل مسأله به همراه اجزای آن تشریح می‌شود. در بخش چهارم نحوه شبیه‌سازی مسأله ذکر شده و ربات و شبیه‌ساز به کار رفته معرفی می‌گردد. همچنین نتایج آزمایشات انجام‌شده ارائه و با روش قدرتمند RRT مقایسه می‌گردد. بخش پنجم نیز شامل نتیجه‌گیری مقاله و کارهای آتی می‌باشد.

۲- جابه‌جایی اجسام

۲-۱- تعریف جابه‌جایی اجسام

جابه‌جایی اجسام یکی از مسایلی است که استفاده از سیستم‌های چندرباتی در آن می‌تواند مفید باشد. استفاده از ربات‌های متحرک قابلیت‌های بیشتری برای حل مسأله فراهم می‌کند اما اضافه‌شدن فاکتورهایی مانند حرکت، سینماتیک و کنترل به مسأله باعث پیچیدگی

چکیده: مقاله پیش رو بر روی حل همکارانه مسأله جابه‌جایی اجسام توسط سیستم‌های چندرباتی توزیع‌شده تمرکز دارد. دو چالش مهم برنامه‌ریزی مسیر و همکاری ربات‌ها باعث دشوارشدن این مسأله شده است. در این مسأله ربات‌ها باید با خودداری از موانع و با بهره‌گیری از یک مکانیزم هماهنگی و همکاری مناسب، جسم را از مسیر عاری از تصادم به نقطه هدف برسانند. رویکرد ارائه‌شده در این مقاله متشکل از یک ساختار دولایه است که از مزایای هر دو معماری متمرکز و غیر متمرکز بهره می‌برد. لایه سراسری با آگاهی کامل از اجزای محیط امکان رسیدن به جواب بهینه را با استفاده از الگوریتم جدید ORT فراهم می‌کند. لایه محلی نیز با انجام محلی برخی از پردازش‌ها باعث کاهش آسیب‌پذیری، بار پردازشی سیستم مرکزی و هزینه کلی سیستم می‌شود. هماهنگی مورد نیاز بین ربات‌ها در این لایه به کمک ارتباط رادیویی برقرار می‌شود و برای برنامه‌ریزی مسیر حرکت محلی ربات‌ها از ترکیب الگوریتم‌های میدان پتانسیل و Tangent Bug استفاده شده است. رویکرد پیشنهادی بر روی یک سیستم چندرباتی متشکل از ربات‌های KUKA youBot و با استفاده از شبیه‌ساز Webots پیاده‌سازی شده است. به منظور بررسی کارایی رویکرد پیشنهادی، نتایج حاصل از آزمایشات مختلف با الگوریتم ORT و همچنین الگوریتم RRT به دست آمده و مقایسه گردید. این نتایج نشانگر کارایی مناسب رویکرد پیشنهادی است.

کلیدواژه: برنامه‌ریزی مسیر، جابه‌جایی همکارانه اجسام، سیستم‌های چندرباتی، هماهنگی، همکاری.

۱- مقدمه

یکی از وظایف سیستم‌های رباتی در محیط‌های صنعتی امروز مانند انبارها، کارخانه‌ها و اسکله‌ها، جابه‌جایی اجسام است. مسأله جابه‌جایی اجسام سنگین توسط این سیستم‌ها یکی از مسایل پرطرفدار و در عین حال پیچیده در حوزه رباتیک است که خود زیرمسایل متعددی مانند برنامه‌ریزی حرکت و انتقال پایدار جسم را در برمی‌گیرد [۱]. هدف برنامه‌ریزی حرکت ایجاد مسیری بدون تصادم برای حرکت ربات یا جسم است. انتقال پایدار بر روی تعیین روش انتقال، مطابق با شرایط جسم و مسیر تمرکز دارد. اگرچه استفاده از سیستم‌های چندرباتی در مسأله جابه‌جایی اجسام مزایای زیادی به دنبال دارد، با این حال بهره‌گیری از این سیستم‌ها چالش‌هایی مانند هماهنگی، همکاری و برقراری ارتباط را به مسأله جابه‌جایی اجسام اضافه می‌کند. برای کسب اطمینان از این که ربات‌ها تصمیمات مغایر با هم اتخاذ نکنند، نیاز به یک مکانیزم هماهنگی

این مقاله در تاریخ ۲۹ تیر ماه ۱۳۹۳ دریافت و در تاریخ ۱۹ اردیبهشت ماه ۱۳۹۴ بازنگری شد.

طاہر حکمت‌فر، دانشکده فنی مهندسی، دانشگاه تربیت مدرس، تهران، (email: taheer.hekmatfar@modares.ac.ir).

الیپس مسیحی، دانشکده فنی مهندسی، دانشگاه تربیت مدرس، تهران، (email: masehian@modares.ac.ir).

فضای پیکربندی آزاد مسأله C_{free} نیز به صورت ترکیبی از پیکربندی‌های مجاز و آزاد برای هر ربات r_i ($r_i \cap O = \emptyset$) تعریف می‌شود

$$C_{free} = C \setminus C_{obs} \quad (۶)$$

در نهایت راه حل مسأله جابه‌جایی اجسام به صورت رشته‌ای از پیکربندی‌های $\langle Q_1, Q_2, \dots, Q_N \rangle \in C_{free}$ تعریف می‌گردد به نحوی که حالت مسأله را از $q_{init} \in Q_1$ به $q_{goal} \in Q_N$ تغییر دهد.

در این مقاله فرض شده است که جسم جابه‌جاشونده مکعب‌شکل بوده و به قدری سنگین است که یک ربات به تنهایی توانایی حمل آن را ندارد. در انتقال جسم هیچ محدودیت زمانی برای ربات‌ها تعیین نشده و ربات‌ها میزان زمان مورد نیاز برای انجام وظایف را در اختیار دارند. ربات‌های مورد استفاده به حسگرهای خاصی مجهز هستند که امکان دریافت اطلاعات مورد نیاز آنها را از محیط فراهم می‌کند. ربات‌ها از موقعیت فعلی خود مطلع هستند و می‌توانند با یکدیگر و سیستم مرکزی ارتباط برقرار کرده و اطلاعات جدیدی دریافت کنند. موانع موجود در محیط دارای اشکال مشخصی هستند اما هیچ محدودیتی از نظر اندازه و تعداد ندارند. این موانع ایستا بوده و با برخورد ربات‌ها یا اجسام جابه‌جاشونده با آنها می‌توانند تکان بخورند. سیستم مرکزی که ناظر نیز نامیده می‌شود، اطلاعات کاملی نسبت به محیط دارد و می‌تواند بخشی از اطلاعات مورد نیاز ربات‌ها را در اختیار آنها قرار دهد. واحد مرکزی از مبدأ و مقصد اجسام جابه‌جاشونده و همچنین موقعیت موانع و تک‌تک ربات‌ها در هر لحظه مطلع است. در سیستم مفروض، محدودیت‌های دینامیکی مانند سرعت و شتاب در نظر گرفته نشده و در واقع سیستم مورد استفاده شبه‌ایستا^۱ است.

۲-۳ سیستم‌های چندرباتی در مسأله جابه‌جایی اجسام

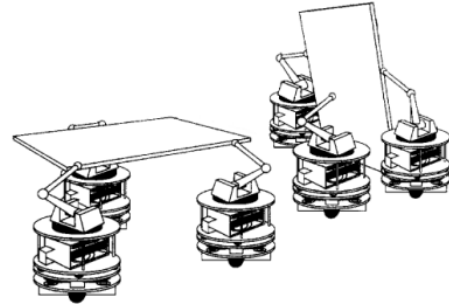
یک سیستم چندرباتی یک سیستم چندعاملی است که از مجموعه‌ای از عامل‌ها که هر کدام یک ربات است تشکیل شده است. سیستم‌های چندرباتی برای انجام کارهایی که به تنهایی توسط یک ربات قابل انجام نیست، بالاخص در محیط‌هایی با خصوصیات عدم قطعیت، اطلاعات ناقص، کنترل توزیع‌شده و محاسبات غیر هم‌زمان به کار می‌روند و با ایجاد افزونگی، افزایش پوشش و ظرفیت و انعطاف‌پذیری و در حالت کلی، با همکاری در انجام وظایف باعث می‌شوند تا کارها به صورت سریع‌تر، ارزان‌تر و با قابلیت اطمینان بیشتری انجام شوند [۱۱] و [۱۲].

به طور کلی در استفاده از سیستم‌های چندعاملی دو رویکرد کلی پیشنهاد شده است: (۱) رویکرد متمرکز و (۲) رویکرد غیر متمرکز.

در معماری متمرکز هر ربات مجموعه‌ای از محرک‌ها، عملگرها و برخی حلقه‌های بازخورد محلی است و از یک کنترل واحد برای هدایت و کنترل ربات‌ها استفاده می‌شود. لازمه این معماری ارتباط دوطرفه بین واحد مرکزی و ربات‌ها است تا ربات‌ها اطلاعات جمع‌آوری شده توسط حسگرها از محیط را برای واحد مرکزی ارسال کنند و این واحد با انجام پردازش‌های لازم، فرمان‌های مربوط به ربات‌ها را برای آنها تعریف و ارسال کند [۱۳]. مزیت اصلی این رویکرد سادگی پیاده‌سازی و امکان انجام عملیات بهینه به دلیل آگاهی واحد مرکزی از همه اطلاعات است [۱۴] اما مقیاس‌پذیری و تحمل‌پذیری پایین خطا را می‌توان جزو مشکلات آن دانست. به دلیل محدودبودن توان پردازشی سیستم مرکزی کارایی کلی سیستم در این رویکرد با افزایش تعداد ربات‌ها کاهش می‌یابد [۱۱] و [۱۳] و در مقابل، رویکرد غیر متمرکز فاقد کنترل‌کننده مرکزی بوده و



(الف)



(ب)

شکل ۱: دو نمونه از مسأله جابه‌جایی اجسام، (الف) جابه‌جایی یک جسم مکعب‌شکل [۹] و (ب) جابه‌جایی صفحات چوبی [۱۰].

حل آن می‌شود [۷] و [۸]. در شکل ۱ دو نمونه از این مسأله نمایش داده شده است. در ادامه این بخش با این مسأله و تحقیقات پیشین انجام‌شده بر روی آن بیشتر آشنا می‌شویم.

۲-۲ تعریف مسأله و مفروضات

در این بخش مسأله جابه‌جایی اجسام توسط سیستم چندرباتی همگن تعریف می‌شود. در این مسأله که جهان W به صورت سه‌بعدی ($W = \mathbb{R}^3$) تعریف شده است، n ربات بازویی متحرک یکسان $R = \{r_i; i = 1, \dots, n\}$ قصد دارند جسم جابه‌جاشونده M را در حضور مجموعه $O = \{o_j; j = 1, \dots, l\}$ از موانع ایستا از موقعیت q_{init} به موقعیت q_{goal} منتقل کنند. موقعیت جسم جابه‌جاشونده و هر یک از موانع در مسأله از (۱) و (۲) قابل وصول است

$$M = \{(x, y, z, \theta) \mid (x, y, z) \in R^3 \text{ is occupied by } M \text{ with an angle of } \theta \text{ at time } t\} \quad (۱)$$

$$o_j = \{(x, y, z, \theta) \mid (x, y, z) \in R^3 \text{ is occupied by } o_j \text{ with an angle of } \theta\}. \quad (۲)$$

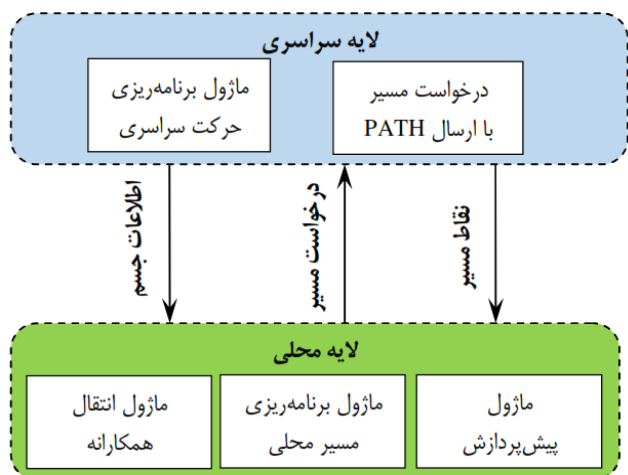
اگر ربات r_i مجهز به یک بازو با درجه آزادی D_f باشد، فضای پیکربندی این ربات (C_r) ، $D_f + ۴$ بعدی خواهد بود. بنابراین فضای پیکربندی مرکب مسأله (C) به صورت (۳) مشخص شده و هر پیکربندی در این فضا بر اساس (۴) تعریف می‌شود که در آن (x, y, z) نشانگر موقعیت مرکز ثقل ربات و θ_p نشانگر زاویه پایه ربات در سیستم مختصاتی است. هر کدام از مقادیر $(\theta_1, \theta_2, \dots, \theta_{D_f})$ زاویه مفصل مورد نظر را مشخص می‌کنند

$$C = C_{r_1} \times C_{r_2} \times \dots \times C_{r_n} \quad (۳)$$

$$Q_{r_i}(t) = (x, y, z, \theta_p, \theta_1, \theta_2, \dots, \theta_{D_f}). \quad (۴)$$

ناحیه پیکربندی موانع C_{obs} ، زیرمجموعه‌ای از فضای پیکربندی است که ربات‌های قرارگرفته در آن با حداقل یک مانع در تصادم هستند

$$C_{obs} = \{q \in C \mid R \cap O \neq \emptyset\}. \quad (۵)$$



شکل ۳: نمایش شکل کلی معماری پیشنهادی.

جابه‌جایی بررسی مفهوم بستار^۸ ضروری است. بستار نحوه محدود کردن جسم توسط ربات‌ها در زمان انتقال آن را تعریف می‌کند. در تشکیل بستار توزیع نیروی وزن متناسب، رهانشدن جسم و اصطکاک بین جسم و ربات باید در نظر گرفته شود [۱۵].

یکی از متداول‌ترین و البته پیچیده‌ترین روش‌های انتقال جسم گرفتن است. در این روش که در ربات‌های بازویی یا چندانگشتی بیشتر کاربرد دارد، بستار نیرو^۹ یا بستار شکل^{۱۰} ایجادشده در تمامی لحظات انتقال جسم باید در حالت تعادل باشد [۴]. تحقیقاتی مانند [۲۳] تا [۲۶] را می‌توان به عنوان نمونه‌هایی برای استفاده از این رویکرد در جابه‌جایی اجسام نام برد.

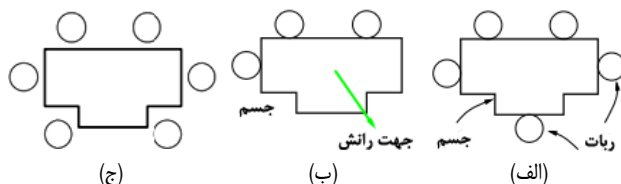
در روش رانش ربات‌ها با هل دادن جسم، اقدام به انتقال آن می‌کنند که این رویکرد هیچ تضمینی درباره بستار نیرو یا بستار شکل ارائه نمی‌دهد. رانش جعبه یکی از ساده‌ترین نمونه‌هایی است که در آن از روش رانش برای جابه‌جایی جسم استفاده می‌شود [۴]. بیشتر تحقیقات انجام‌شده در حوزه جابه‌جایی اجسام از این رویکرد استفاده کرده‌اند که برخی از این تحقیقات عبارتند از [۱]، [۳]، [۲۷] و [۲۸].

در روش محاصره، ربات‌ها با ایجاد بستار جسم و به کمک شناور بودن جسم در داخل بستار یا با کشیده شدن جسم، آن را به مقصد منتقل می‌کنند. در بستار جسم، جسم به نحوی محدود می‌شود که فقط دامنه کوچکی برای جابه‌جایی داشته باشد و نتواند آزادانه از موقعیت فعلی به موقعیت دیگری خارج از محدوده حرکت کند [۴]. در پژوهش‌های [۴]، [۱۷] و [۲۹] این رویکرد اعمال شده است.

۳- معماری مسأله

۳-۱ توسعه یک معماری جهت حل مسأله

همان‌طور که در تعریف مسأله اشاره شد در این تحقیق قصد داریم تا یک معماری توزیع‌شده برای جابه‌جایی اجسام سنگین توسط یک سیستم چندرباتی همگن متشکل از ربات‌های KUKA ارائه دهیم. حالت کلی معماری پیشنهادی که در شکل ۳ نشان داده شده است، از دو لایه سراسری (یا متمرکز) و محلی (یا توزیع‌شده) تشکیل شده که لایه سراسری توسط واحد مرکزی یا ناظر و لایه محلی توسط تک‌تک ربات‌ها



شکل ۲: روش‌های جابه‌جایی اجسام توسط سیستم‌های چندرباتی، (الف) روش گرفتن، (ب) روش رانش و (ج) روش محاصره [۴].

کنترل وظایف بر عهده خود ربات‌ها است و تحمل‌پذیری بالای خطا، قابلیت اطمینان زیاد و توزیع بار پردازشی را می‌توان به عنوان ویژگی‌های آن نام برد. به دلیل افزایش تحمل‌پذیری خطا در زمان افزایش تعداد ربات‌ها، این رویکرد نسبت به معماری متمرکز برای همکاری ربات‌ها مناسب‌تر است [۱۳] و [۱۵].

رویکرد غیر متمرکز در حالت کلی شامل ۳ مدل رهبر- پیرو^۱، تقسیم وظیفه^۲ و کاملاً توزیع‌شده^۳ است [۱۴]. در مدل اول یکی از اعضای گروه به عنوان رهبر تعیین شده و بقیه اعضای گروه وظیفه تبعیت از وی را دارند [۷]. مزیت این مدل سادگی آن در ایجاد هماهنگی و همکاری بین ربات‌ها است اما این روش نسبت به از کار افتادن ربات رهبر آسیب‌پذیر و در آن از حداکثر توانایی ربات‌های پیرو استفاده نمی‌شود [۱۴]. در تحقیقات مختلفی مانند [۳] و [۱۶] تا [۱۸] در مدل تقسیم وظیفه، وظایف به چند دسته مجزا تقسیم شده و هر دسته به یک گروه سپرده می‌شود. این کار باعث سادگی حل مسأله می‌شود [۱۴]. روش محدود کردن و حرکت دادن^۴ که در [۷] ارائه شده، یکی از روش‌هایی است که از این مدل بهره می‌گیرد و در آن هر کدام از وظایف کنترل مکان جسم و کنترل زاویه جسم به یک گروه مجزا از ربات‌ها سپرده می‌شود.

در مقابل، رویکرد کاملاً توزیع‌شده فاقد مشکلات معمول رویکردهای متمرکز و رهبر- پیرو یعنی آسیب‌پذیری و هزینه بالا بوده و مقیاس‌پذیری و تحمل خطای بالا و مبادله کم اطلاعات را می‌توان جزو نقاط قوت آن دانست [۹] و [۱۱]. در یک سیستم کاملاً توزیع‌شده کارها به صورت غیر صریح و خودمختارانه بین همه ربات‌ها پخش شده و هر ربات به طور مستقل در مورد وظیفه خود تصمیم‌گیری می‌کند. با وجود تمام مزایای مطرح‌شده، به دلیل محدود بودن اطلاعات ربات‌ها به اطلاعات دریافتی از حسگرهایشان، ممکن است ربات‌ها در این مدل نتوانند به طور بهینه فعالیت کرده و حتی در وضعیت بن‌بست گیر کنند [۱۹] و [۲۰]. این رویکرد نیز در تحقیقاتی مانند [۱۹] و [۲۰] مورد استفاده قرار گرفته است.

۲-۴ روش‌های جابه‌جایی اجسام

روش‌های جابه‌جایی اجسام به کمک سیستم‌های چندرباتی در حالت کلی به سه دسته گرفتن^۵، رانش^۶ و محاصره^۷ تقسیم می‌شود که در شکل ۲ نمایش داده شده‌اند [۴]. البته روش‌هایی مانند پرتاب کردن جسم [۲۱] یا لرزاندن سطح به منظور جابه‌جا شدن جسم روی آن [۲۲] نیز مطرح هستند اما این روش‌ها چندان کاربرد ندارند. در همه روش‌های

1. Leader-Follower
2. Task Division
3. Pure Distributed
4. Constrain and Move
5. Grasping
6. Pushing
7. Caging

8. Closure
9. Force Closure
10. Form Closure

۳-۲ لایه سراسری

اگرچه در جابه‌جایی اجسام با یک سیستم کاملاً توزیع‌شده، ربات‌ها وظیفه دارند خود مسیر و نحوه حرکت جسم را تعیین کنند، عدم بهینگی عملکرد سیستم‌های محلی به دلیل نقصان اطلاعات می‌تواند منجر به گرفتارشدن در بیشینه‌های محلی شود. لذا در پژوهش پیش رو به منظور وصول بهینه جواب ممکن و جلوگیری از وقوع بن‌بست، بخشی از وظایف توسط سیستم مرکزی انجام می‌شود. در روش پیشنهادی برنامه‌ریزی حرکت ربات‌ها در دو سطح انجام می‌شود. در سطح اول که توسط سیستم مرکزی یا ناظر اجرا می‌شود، در زمان نیاز ربات‌ها به یک مسیر سراسری برای انتقال جسم، یک مسیر نسبتاً بهینه و عاری از تصادم وسط الگوریتم ORT^1 ایجاد می‌شود. شبه‌کد مربوط به لایه سراسری در شکل ۶ نمایش نمایش داده شده است.

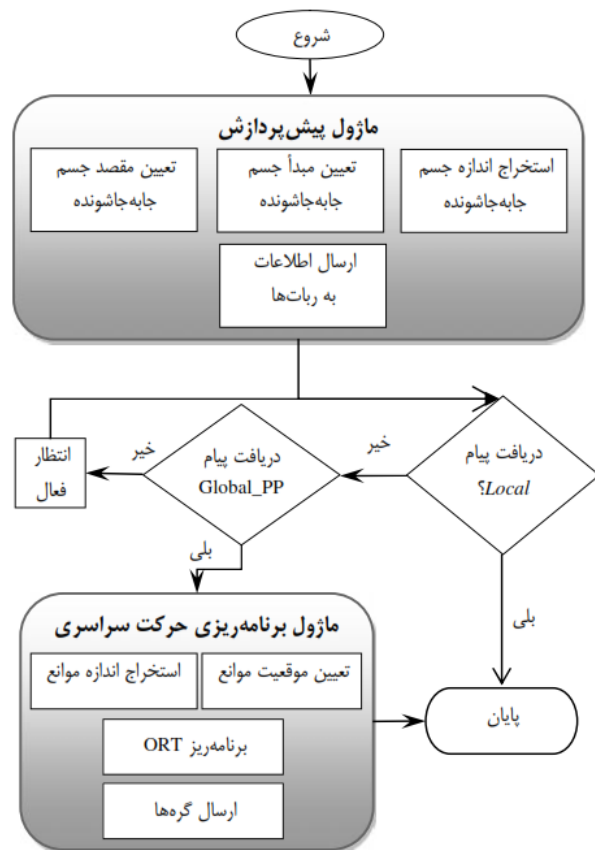
در ماژول پیش‌پردازش با توجه به آگاهی ناظر نسبت به همه اجزای محیط، پیکربندی اولیه جسم شامل موقعیت، جهت و اندازه جسم توسط ناظر استخراج شده و برای ربات ارسال می‌شود. پس از ارسال اطلاعات به ربات‌ها ناظر در حالت انتظار قرار می‌گیرد و این انتظار تا زمانی که ربات‌ها نیاز یا عدم نیاز خود به مسیر سراسری را به ناظر اعلام کنند ادامه دارد.

دریافت پیامی با عنوان $Local_PP$ نشانگر عدم نیاز ربات‌ها به مسیر سراسری است. با این حال اگر ناظر پیامی به صورت $Global_PP$ دریافت کند، وارد فاز برنامه‌ریزی سراسری شده و پس از استخراج موقعیت و اندازه موانع اقدام به ایجاد یک مسیر عاری از تصادم به کمک الگوریتم ORT برای انتقال جسم می‌کند. این مسیر به صورت رشته‌ای از نقاط منفصل که با فاصله‌ای به اندازه طول گام حرکتی ربات هستند، توسط سیستم رادیویی به ربات‌ها اطلاع داده می‌شود.

همان‌طور که ذکر شد در مواقعی که فاصله مبدأ جسم از مقصد از یک حد آستانه بیشتر باشد یا موانعی مسیر مستقیم بین مبدأ و مقصد جسم را مسدود کند، ربات‌ها نیازمند یک مسیر سراسری خواهند بود تا از طریق یک مسیر نسبتاً بهینه جسم را منتقل کرده و در وضعیت بن‌بست گیر نکنند. در تحقیق پیش رو برای تولید این مسیر از الگوریتم ORT که در [۳۰] معرفی شده، استفاده می‌شود. البته به منظور استفاده از این الگوریتم، تغییراتی در آن اعمال شده تا الگوریتم متناسب با مسأله مطرح‌شده در تحقیق باشد.

ORT (شکل ۷) یک الگوریتم مبتنی بر نمونه‌برداری است که با هدف ارائه مسیر نزدیک به بهینه برای مسایل مختلف برنامه‌ریزی حرکت شامل سیستم‌های تک‌رباتی و چندرباتی در بین موانع متحرک و ثابت و با رویکرد متمرکز معرفی شده است. مسیر تولیدشده توسط این الگوریتم متشکل از تعدادی نقطه است که با فاصله‌ای برابر با طول گام حرکتی ربات ایجاد شده و در واقع جهت حرکت کلی جسم را تعیین می‌کنند. ممکن است مسیر مذکور بهینه نباشد زیرا الگوریتم ORT مبتنی بر نمونه‌برداری بوده و کیفیت جواب به نمونه‌های انتخابی بستگی دارد.

یکی از نقاط ضعف روش‌های مبتنی بر نمونه‌برداری تصادفی مانند PRM یا RRT کیفیت پایین جواب آنها از نظر ناهمواری مسیر است. اگرچه می‌توان به کمک هموارسازی مسیر بر این مشکل فایز آمد، اما در سیستم‌های چندرباتی و در محیط‌های پیچیده هموارسازی مسیر دشوار بوده و نیاز به پردازش زیادی دارد. این در حالی است که الگوریتم ORT تلاش می‌کند تا یک درخت با شاخه‌های صاف ایجاد کرده و مسیر نهایی

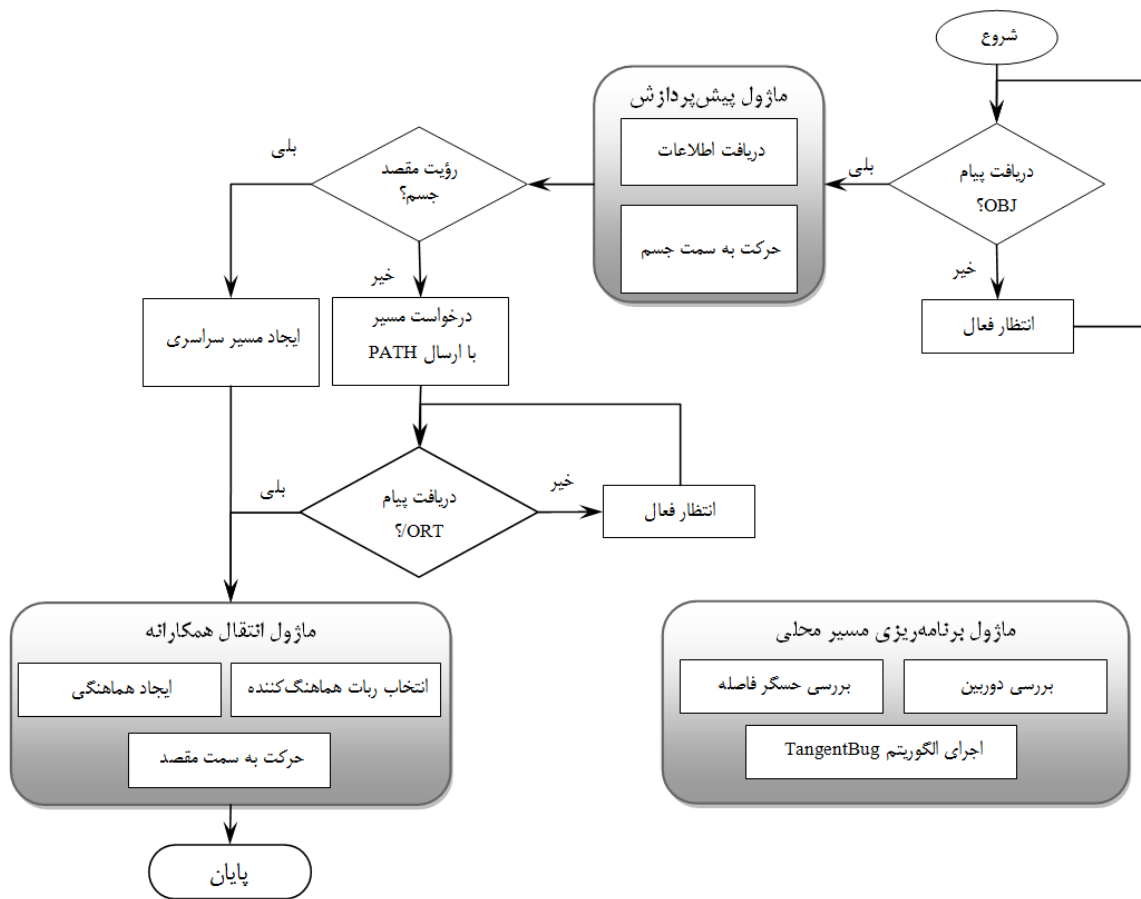


شکل ۴: بخش‌های مختلف لایه سراسری.

اجرا می‌شود. دلایل مختلفی برای استفاده از چنین ساختاری وجود دارد. سیستم مرکزی به ربات‌ها کمک می‌کند تا به یک جواب بهینه دست یابند. این در حالی است که سیستم محلی مورد استفاده در ربات‌ها به دلیل فقدان اطلاعات لازم نمی‌تواند به تنهایی به جواب مناسبی دست یابد و چه بسا در حالت بن‌بست گیر کند. این ساختار با تقسیم پردازش‌های مورد نیاز بین سیستم مرکزی و ربات‌ها باعث کاهش بار پردازشی سیستم مرکزی شده و نیاز سیستم به یک واحد مرکزی قدرتمند و همچنین بهره‌مندی از پهنای باند بالا را کاهش می‌دهد. همچنین با افزایش تحمل‌پذیری خطا در سیستم، در صورت بروز مشکل در واحد مرکزی و عدم دریافت مسیر سراسری از ناظر، ربات‌ها می‌توانند با هماهنگی ایجادشده به صورت محلی، جسم را به نقطه هدف منتقل کنند.

کاملاً مشخص است که لایه سراسری و لایه محلی به طور مجزا و در سیستم‌های متفاوت اجرا می‌شوند و ارتباط آنها از طریق ارسال پیام‌های متفاوت برقرار می‌شود. لایه سراسری وظیفه دارد اطلاعات جسم شامل موقعیت و زاویه فعلی، موقعیت و زاویه هدف و همچنین اندازه جسم را در اختیار لایه محلی قرار دهد. ربات‌ها بر اساس وضعیت مسأله ممکن است نیاز به یک مسیر سراسری داشته باشند که لایه محلی این مسأله را از طریق ارسال یک درخواست به لایه سراسری اعلام می‌کند. آخرین نوع پیغام که ممکن است بین دو لایه رد و بدل شود نیز مربوط به اعلام نقاط مسیر حرکت جسم و همچنین نقاط مناسب برای ربات‌ها در این مسیر است.

بر اساس توضیحات ارائه‌شده درباره مسأله می‌توان فهمید که عملیات انجام‌شده در هر لایه با یکدیگر متفاوت بوده و از طریق چندین ماژول انجام می‌گیرد که این ماژول‌ها در شکل‌های ۴ و ۵ به طور دقیق مشخص شده‌اند. هر لایه در ادامه به طور کامل تشریح خواهد شد.



شکل ۵: بخش‌های مختلف لایه محلی.

در این روش از سه استراتژی نمونه‌برداری برای توسعه درخت جستجو استفاده شده که عبارتند از:

- ۱: حرکت مستقیم به سمت هدف
 - ۲: استراتژی انتخاب نقطه تصادفی از ناحیه اثر هدف^۱ (TIR)
 - ۳: استراتژی نمونه‌گیری یکنواخت از فضای آزاد هر لایه (C_{free})
- دو استراتژی اول باعث گسترش درخت جستجو به سمت هدف و استراتژی سوم منجر به توسعه درخت در فضاهای آزاد جدید می‌شود. زمانی که هدف از گره فعلی q_i قابل رؤیت باشد (یعنی یک خط مستقیم بین نقاط q_i و q_{goal} وجود داشته باشد که با هیچ مانعی برخورد نداشته باشد) از استراتژی اول استفاده می‌شود. در این استراتژی نقطه تولیدشده به نقطه هدف به طور مستقیم متصل می‌شود و نقاط جدید بر روی خط واصل q_{new} و q_{goal} و با فاصله γ گام ربات ایجاد و به درخت جستجو اضافه می‌گردد. بررسی رؤیت‌پذیری یک نقطه توسط تابع $is_visible$ و تولید گره در مسیر مستقیم نیز توسط تابع $direct$ انجام می‌شود.

استراتژی دوم که توسط تابع $exploit$ اجرا می‌گردد، به منظور محدودکردن درخت جستجو و حرکت به سمت هدف ارائه‌شده است. در این استراتژی از محدوده اطراف هدف که ناحیه اثر یا TIR نامیده می‌شود، نمونه‌برداری شده که شعاع نمونه‌برداری به طور تصادفی و بین 1 تا مقدار t_{LB} تعیین می‌شود. این ناحیه مجموعه‌ای از نواحی دیسک‌شکل حول نقطه هدف است. شعاع هر ناحیه به صورت ضربی از یک مقدار از پیش تعریف شده γ که گام حرکتی نام دارد، تعریف می‌شود. نمونه‌برداری

Algorithm 1: Global_Planner ($Final_cfg$)

```

Input:
Obj: The moveable objects
Final_cfg: The final configuration of moveable object
Obs: A set of static obstacles
Robots: The set of mobile manipulators
Output:
Global_Path: The global path to transport the object
1 //Preprocessing
2 Init_cfg = EXTRACT_CFG(Obj)
3 SEND_TO_ROBOTS(Init_cfg, Final_cfg)
4 Do
5 Message ← RECIEVER()
6 Until (Message = "Global_PP") OR (Message = "Local_PP")
7 //Path planning
8 If (Message = "Global_PP")
9 For i = 1 to N //N = Obstacle number
10 obs_cfg[i] = EXTRACT_CFG(Obs[i])
11 End
12 Global_Path = ORT_PLANER
    (init_cfg, obs_cfg, Final_cfg)
13 SEND_TO_ROBOTS("G_PP")
14 SEND_TO_ROBOTS(Global_Path)
15 End
16 TERMINATE()
    
```

شکل ۶: الگوریتم ۱.

به کمک جستجو بر روی این درخت تولید شود. بنابراین مسیر نهایی ایجادشده توسط الگوریتم ORT نیاز به هموارسازی نداشته و خود یک مسیر هموار می‌باشد.

این الگوریتم بر اساس انتخاب تصادفی نقاط عمل می‌کند و وجه تمایز آن نسبت به روش‌های دیگر در نحوه نمونه‌برداری از محیط، اتصال نقاط جدید به درخت جستجو و تشکیل این درخت است.

1. Target Influence Region

Algorithm 3: Local_Planner()

```

Input:
  Init_Cfg: The initial configuration of box
  Final_cfg: The final configuration of box
1 Do
2   Message ← RECIEVER()
3   Until (Message = "Object_CFG")
4   Init_cfg ← RECIEVER() //Preprocessing
5   Final_cfg ← RECIEVER()
6   GO_TO(Init_cfg) //Local Path Planning
7   Target_Visibility = CHECK_TARGET(Final_cfg)
8   If (Target_Visibility = False)
9     SEND_TO_SUPERVISOR("Global_PP")
10  Do
11    message ← RECIEVER()
12    Until (message = "G_PP")
13    Global_Path ← RECIEVER()
14  Else
15    SEND_TO_SUPERVISOR("Local_PP")
16    Global_Path = GENERATE_PATH(Init_cfg, Final_cfg)
17  End
18 TRANSPORT (Global_Path) //Cooperative transportation
    
```

شکل ۸: الگوریتم ۳.

مبدأ و هدف جسم را دریافت کرده و بر اساس موقعیت اولیه جسم شروع به حرکت به طرف آن می‌کنند. حرکت ربات به صورت محلی و در مازول برنامه‌ریزی حرکت محلی در تابع GO_TO و با استفاده از ترکیب الگوریتم‌های Tangent Bug و میدان پتانسیل و اطلاعات حاصل از حسگرهای دوربین و فاصله‌سنج تعبیه‌شده در ربات‌ها انجام می‌گیرد.

پس از رسیدن ربات‌ها به مجاورت جسم جابه‌جاشونده، هر ربات وضعیت مقصد جسم را رصد می‌کند و در صورتی که مقصد جسم فاصله‌ای بیشتر از یک حد آستانه داشته باشد یا یکی از ربات‌ها مانعی در جهت حرکت جسم از مبدأ به مقصد رؤیت کند، ربات‌ها نیاز خود به مسیر سراسری را توسط ارسال پیغام Global_PP به ناظر اعلام خواهند کرد. در حالت اول این درخواست توسط ربات با کوچک‌ترین شماره و در حالت دوم این درخواست توسط رباتی که مانع را رؤیت کرده است انجام می‌گیرد. در این مرحله همه ربات‌ها در حالت انتظار قرار می‌گیرند و این انتظار تا زمان دریافت پیام حاوی پرچم ORT ادامه می‌یابد. پیام‌های بعد از این پرچم حاوی نقاط مسیر حرکت جسم از مبدأ به مقصد و همچنین وضعیت آزادبودن اطراف جسم برای گرفتن از آن در هر کدام از این نقاط است.

با دریافت پیام مذکور، جهت حرکت ربات‌ها مشخص می‌شود. در صورتی که ربات‌ها نیازی به مسیر سراسری تعریف‌شده توسط ناظر نداشته باشند (مقصد نزدیک باشد و هیچ مانعی در مسیر وجود نداشته باشد)، ربات‌ها پس از ارسال پیغام Local_PP به ناظر، فاصله مبدأ و مقصد جسم و همچنین اختلاف زاویه این دو وضعیت را محاسبه کرده و با تقسیم این مقادیر به مقدار از پیش تعریف شده‌ای که طول گام ربات‌ها را نشان می‌دهد، تعداد گام‌های لازم برای حرکت جسم به سمت هدف و همچنین نقاط قرارگیری جسم در هر گام را مشخص می‌کنند. در واقع در این حالت ربات‌ها خود مسیر سراسری جسم را ایجاد می‌کنند.

در این مرحله مازول مسیریابی محلی در کنترل‌کننده ربات‌ها به کار افتاده و ربات‌ها با تعقیب نقاط مسیر تعیین‌شده که مانند اهداف میانی در حرکت ربات‌ها عمل می‌کنند، جسم را تا زمانی که به نقطه مقصد برسد منتقل می‌کنند. در هر گام، ربات‌ها از برقراری ارتباط برای ایجاد هماهنگی در حرکات خود استفاده می‌کنند. در این فاز نیز ربات‌ها برای حرکت خود از ترکیب الگوریتم‌های میدان پتانسیل و Tangent Bug استفاده می‌کنند. در زمان حمل جسم و در صورت روبه‌رو شدن هر کدام از ربات‌ها با موانع، ربات مذکور زاویه حرکت خود را به بقیه ربات‌ها اعلام

Algorithm 2: ORT_main (q_{init}, q_{goal})

```

Input:
  qinit: Initial configuration, qgoal: goal configuration
  {V,E}: Vertices and edges of search tree
Output:
  P: A Sequence of configurations defining a collision-free-path from
  qinit to qgoal
    
```

```

1 V ← {qinit}; E ← ∅
2 While (True)
3   If tlb < qnew.t
4     tlb ← tlb + 1
5   End If
6   (qnew, qnear, success) ← exploit(tL,B)
7   If (!success) then
8     (qnew, qnear) ← explore()
9   End If
10  qnew.t ← tnear + 1
11  qroot ← root (qnew, qnear)
12  {V,E} ← connect (qnew, qroot)
13  If is_visible (qnew, qgoal) then
14    direct (qnew, qgoal)
15    break; //End algorithm
16  End If
17 End While
18 P ← backward search in {V,E}.
19 return P
    
```

شکل ۷: الگوریتم ۲.

از این نواحی اثربخشی بیشتری نسبت به نمونه‌برداری از کل فضای آزاد دارد و باعث می‌شود درخت جستجو فقط در جهت هدف گسترش پیدا کند و از تولید گره‌های زاید جلوگیری شود.

استراتژی سوم برای جستجو در فضاهای جدید طراحی شده است. این استراتژی که توسط تابع explore و در شرایطی که نقطه هدف از آخرین گره جدید تولیدشده قابل رؤیت نباشد و انتخاب نقطه تصادفی جدید از نواحی انتخابی f بار با شکست روبه‌رو شود، به منظور رشد درخت جستجو به سمت جدید اجرا می‌شود. در این استراتژی نمونه تصادفی q_{rand} از C_{free} به صورت تصادفی یکنواخت انتخاب می‌شود. استفاده از توزیع یکنواخت باعث می‌شود که تمام نقاط محیط احتمال یکسانی برای انتخاب داشته باشند و الگوریتم در بهینه محلی گرفتار نشود.

علاوه بر تفاوت الگوریتم ORT در نحوه تولید گره‌های جدید، نحوه اتصال گره‌ها و تشکیل درخت جستجو نیز در این الگوریتم متفاوت است. در این الگوریتم، گره تصادفی تولیدی q_{rand} لزوماً به درخت جستجو اضافه نمی‌شود، بلکه ابتدا نزدیک‌ترین گره از درخت جستجو به q_{rand} یعنی q_{near} انتخاب می‌شود و گره جدید q_{new} بر روی خط واصل q_{near} و q_{rand} و به فاصله طول گام ربات γ از q_{near} ایجاد می‌شود. در زمان اتصال گره جدید q_{new} به درخت جستجو به جای اتصال این گره به q_{near} ، الگوریتم درخت جستجو را از گره اول پیمایش کرده و در صورت وجود یک خط راست بین گره q_{new} و هر کدام از گره‌های درخت جستجو، q_{new} را به گره مذکور متصل کرده و آن گره را q_{root} می‌نامد و گره‌های جدید را بر روی خط واصل q_{new} و q_{root} ایجاد کرده و به درخت جستجو اضافه می‌کند. گره ریشه یک گره جدید توسط تابع root مشخص می‌شود و عمل اتصال به درخت نیز توسط تابع connect انجام می‌گیرد.

۳-۳ لایه محلی

این لایه در تک‌تک ربات‌ها به صورت مجزا اجرا می‌شود. الگوریتم ۳ (شکل ۸) نشان‌دهنده فرمان‌های مربوط به این لایه است. در مازول پردازش اولیه، کنترل‌کننده ربات‌ها اطلاعات ارسالی از ناظر درباره موقعیت

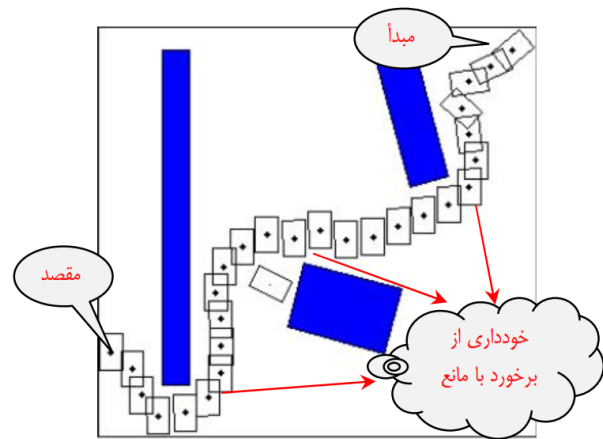
فاصله تعبیه‌شده بر روی ربات‌ها استفاده می‌کند. این الگوریتم برای ربات‌هایی به صورت نقطه‌ای کامل است و در صورت وجود مسیر قابل قبول، حتماً آن را شناسایی خواهد کرد. میزان بهینگی مسیر پیدا شده در این الگوریتم به محدوده حسگرهای فاصله آن و تعداد حسگرهای تعبیه‌شده در ربات بستگی دارد که با افزایش این محدوده الگوریتم جواب‌های بهینه‌ای خواهد یافت. عملکرد الگوریتم Tangent Bug بر اساس شناسایی نقاط منفصل^۱ است. این نقاط محدوده ابتدایی و انتهایی یک مانع را که در شعاع حسگرهای ربات قرار دارند شناسایی می‌کند [۳۱]. در زمان حرکت به سمت نقطه هدف نهایی، در صورت شناسایی موانع در مسیر مربوط، الگوریتم با تولید نقاط هدف میانی باعث تغییر مسیر حرکت ربات و عبور از موانع می‌شود.

در سیستم توسعه داده شده به منظور کاهش بار پردازشی در زمان شناسایی موانع در اطراف ربات، به جای پردازش اطلاعات همه دوربین‌ها و حسگرهای فاصله ربات، فقط دوربین و حسگرهای فاصله‌ای که در راستای مسیر حرکت ربات به سمت مقصد آن هستند در هر لحظه انتخاب و اطلاعات آنها پردازش می‌شود. این کار با ایده‌گرفتن از روش بینایی^۲ فعال [۳۲] انجام شده است.

الگوریتم Tangent Bug یک الگوریتم کامل است اما این الگوریتم مشکلاتی در زمان استفاده در ربات‌های واقعی که اشکالی غیر از دایره دارند به همراه دارد. در بیشتر تحقیقات انجام‌شده به کمک الگوریتم Tangent Bug، ربات به صورت یک نقطه یا یک دایره تصور شده است. این در حالی است که ربات مورد استفاده در این تحقیق دارای سطح مقطعی به شکل مستطیل است. از طرفی با توجه به محدود بودن تعداد حسگرهای فاصله بر روی ربات، حسگرها نمی‌توانند تمامی اطراف ربات را پوشش دهند و نقطه هدف میانی ایجادشده توسط Tangent Bug در زمان مواجه شدن با موانع بر اساس اطلاعات فقط تعداد محدودی از حسگرها تعیین می‌شود و از این رو همچنان امکان برخورد ربات با موانع وجود دارد.

با توجه به موارد مذکور، در این تحقیق از یک میدان پتانسیل محلی نیز برای تصحیح حرکات ربات و اطمینان حاصل کردن از عدم وقوع تصادم استفاده شده است. در روش میدان پتانسیل^۳ به کمک یک تابع پتانسیل ربات به سمت نقطه هدف هدایت می‌شود. ماهیت این رویکرد به این نحو است که نقاط موجود در مسیر جواب به سمت نقطه هدف میانی تعریف‌شده توسط الگوریتم Tangent Bug حالت جاذبه داشته و موانع موجود در مسیر حالت دافعه دارند. این نیروها باعث ایجاد یک شبکه نیرو می‌شوند که بر روی ربات اثر می‌گذارد و با تولید یک مسیر عاری از تصادم، ربات را به سمت هدف پیش می‌برد.

این رویکرد بسیار ساده بوده و به راحتی می‌تواند خود را با مسایل مختلف تطبیق دهد. اما مشکل اصلی این رویکرد گیرکردن ربات در نقاط کمینه محلی است. همچنین تعریف توابع پتانسیل برای محاسبه نیروی جاذبه و دافعه دقیق دشوار می‌باشد [۳۳]. وجود مشکل نقاط کمینه محلی باعث شد که در این تحقیق از میدان پتانسیل به تنهایی استفاده نگردد. در واقع الگوریتم Tangent Bug مانع گیرکردن ربات در نقاط بهینه محلی شده و الگوریتم میدان پتانسیل نیز خروجی‌های الگوریتم Tangent Bug را به نحوی تصحیح می‌کند که از برخورد اجسام با موانع جلوگیری شود.



شکل ۹: عبور ربات از مسیر عاری از تصادم.

می‌کند تا با تغییر هدف میانی از برخورد ربات‌ها با موانع جلوگیری شود و همچنین هماهنگی حرکت همه ربات‌ها حفظ شود. حرکت به سمت هدف تا زمان رسیدن به نقطه هدف جسم ادامه می‌یابد و پس از رسیدن به نقطه مذکور، ربات‌ها با هماهنگی ایجادشده توسط ربات رهبر، جسم را بر روی زمین قرار داده و به کار خود خاتمه می‌دهند. انتقال همکارانه جسم نیز در تابع TRANSPORT انجام می‌گیرد.

۳-۳-۱ برنامه‌ریزی حرکت محلی

مسیر ایجادشده توسط سیستم مرکزی برای انتقال جسم به صورت رشته‌ای از نقاط است که امکان گرفتن از جسم در آنها برای ربات‌ها فراهم شده باشد. در واقع به منظور کاهش پیچیدگی در ساخت مسیر، کل مسیر مورد نیاز برای حرکت ربات‌ها به طور دقیق بررسی نمی‌شود و فقط امکان گرفتن از جسم در نقاط ایجادشده بررسی می‌گردد. بنابراین امکان رخ دادن تصادم در زمان حرکت ربات‌ها برای انتقال جسم مابین این نقاط وجود دارد و ربات‌ها نیازمند یک برنامه‌ریز مسیر محلی برای جلوگیری از رخ دادن این تصادم هستند. این برنامه‌ریز نه تنها در زمان انتقال جسم به ربات‌ها کمک می‌کند، بلکه در زمان حرکت یک ربات به تنهایی (حرکت به سمت جسم) نیز مانع برخورد ربات با موانع، جسم جابه‌جاشونده و ربات‌های دیگر موجود در محیط می‌شود.

در این تحقیق به منظور طراحی برنامه‌ریز حرکت محلی مورد نظر از ترکیب الگوریتم‌های میدان پتانسیل و Tangent Bug استفاده شده است. در این حالت نقطه هدف به الگوریتم برنامه‌ریز حرکت محلی داده می‌شود و این الگوریتم در صورت شناسایی مانع در مسیر با تعریف اهداف میانی، باعث حرکت ربات‌ها از یک مسیر عاری از تصادم به نقطه هدف اصلی می‌شود. در زمان رؤیت موانع، هدف میانی ابتدا توسط الگوریتم Tangent Bug تعریف می‌شود، سپس در صورت نیاز به تغییر، توسط الگوریتم میدان پتانسیل تغییر داده می‌شود تا ربات نه تنها در حالت بن‌بست گیر نکند، بلکه با هیچ مانعی نیز برخورد نداشته باشد.

شکل ۹ نمونه‌ای از نحوه عملکرد برنامه‌ریز محلی طراحی‌شده را نشان می‌دهد که در آن ربات شماره ۱ که با علامت ♦ مشخص شده است، می‌بایست خود را از مبدأ مشخص‌شده به مقصد معین برساند. در این شکل مستطیل‌های توپر نشانگر موانع استاتیکی و مستطیل‌های با علامت‌های ♦ و نقطه به ترتیب نشانگر ربات‌های شماره ۱ و ۲ هستند. همان‌طور که مشخص است ربات ۱ در حین حرکت با رسیدن به موانع ایستا و ربات شماره ۲ بدون برخورد به این اجسام از کنار آنها عبور می‌کند. الگوریتم Tangent Bug به عنوان یک الگوریتم ساده و کارا برای برنامه‌ریزی مسیر محلی مطرح است که از اطلاعات حاصل از حسگرهای

1. Discontinuity
2. Active Vision
3. Potential Field

می‌شود. نخ مذکور به طور مداوم این لیست را چک می‌کند و هر زمان که یک عضو جدید توسط فرایند اصلی کنترل‌کننده ربات به لیست اضافه شود، ربات را در جهت اعلام‌شده به حرکت در می‌آورد.

۳-۳-۲ ارتباط و هماهنگی محلی

یکی از ساده‌ترین روش‌های ایجاد هماهنگی در یک سیستم چندعاملی استفاده از ارتباط بین عامل‌ها است. در تحقیق پیش رو نیز از ارتباط بین ربات‌ها برای ایجاد هماهنگی بین آنها استفاده شده است. ارتباط بین ربات‌ها از طریق ارسال پیام‌های رادیویی بین آنها صورت می‌گیرد. هر پیام به صورت رشته‌ای از کاراکترها ایجاد و ارسال می‌شود. عبارت زیر ساختار پیام‌های مورد استفاده را نشان می‌دهد

$$"Reciever|Emitter|Option1|Option2|..." \quad (11)$$

با توجه به این که هر ربات و همین طور سیستم مرکزی دارای یک شماره منحصر به فرد هستند، با قراردادن این شماره‌ها در بخش فرستنده یا گیرنده پیام، مبدأ و مقصد آن را مشخص می‌نماییم. اولین کاراکتر در هر پیام نشانگر گیرنده و کاراکتر دوم نیز نمایشگر فرستنده آن است. بقیه کاراکترها اختیاری بوده و مشخص‌کننده محتوای پیام هستند. در زمان ارسال پیام‌های همه‌پخشی از کاراکتر صفر در بخش گیرنده پیام استفاده می‌شود. همه ربات‌ها قادر به دریافت همه پیام‌ها هستند اما ربات‌ها در صورت یکسان بودن اولین کاراکتر پیام با شماره خود یا برابر بودن این مقدار با صفر پیام مربوط را پردازش خواهند کرد.

ادامه پیام، بسته به محتوای آن می‌تواند شامل چندین بخش دیگر نیز باشد که محتوای اصلی پیام را تشکیل می‌دهند. برای نمونه پیامی که از ناظر به همه ربات‌ها به صورت همه‌پخشی ارسال می‌شود می‌تواند به صورت "۰۶۴|-۱٫۲|-۰٫۱۴|۰٫۳|۱۰۰۰" باشد. این پیام به همه ربات‌ها چهار مقدار (۰٫۳، ۰٫۱۴، -۱٫۲، -۰٫۶۴) را اعلام می‌کند. این که این چهار مقدار نمایشگر چیست، بستگی به پرچم ارسالی قبل از این مقادیر دارد.

قبل از ارسال پیام اصلی، یک رشته کاراکتری که نقش پرچم را دارد در بخش *Option* پیام درج و ارسال می‌شود. این پیام مشخص می‌کند که پیام‌های بعدی حاوی چه نوع مفاهیمی هستند. همچنین پس از آخرین پیام ارسالی، یک پرچم دیگر برای گیرنده ارسال می‌شود تا گیرنده از اتمام پیام‌ها مطمئن شود. در این پیام‌ها نیز به کمک دو کاراکتر ابتدایی، فرستنده و گیرنده پیام مشخص می‌شود. دو پرچم *OBJ ... OBJ* و *ORT ... ORT* به ترتیب برای اعلام موقعیت و زاویه جسم در مبدأ و مقصد و نیز گره‌های تولیدشده در الگوریتم *ORT* در نظر گرفته شده‌اند.

ربات‌ها با دریافت مشخصات و موقعیت جسم به طرف آن حرکت می‌کنند و پس از رسیدن به موقعیت مذکور، به منظور ایجاد هماهنگی محلی هر یک به اندازه *ID* ثانیه منتظر مانده و سپس شماره خود را در پیام درج کرده و به صورت همه‌پخشی ارسال می‌کنند. با این کار همه ربات‌ها از حضور یکدیگر مطلع می‌شوند. در این مرحله ربات با کوچک‌ترین شماره به عنوان ربات هماهنگ‌کننده انتخاب می‌شود و با توجه به این که همه ربات‌ها از شماره ربات‌های عضو گروه مذکور مطلع هستند، همگی به صورت خودکار ربات هماهنگ‌کننده را خواهند شناخت. با توجه به پویا بودن روش انتخاب ربات هماهنگ‌کننده، هیچ مشکلی در زمان خرابی یا تغییر ربات‌ها رخ نخواهد داد.

در این معماری، بلندکردن، انتقال جسم و بر روی زمین گذاشتن آن همگی توسط فرمان‌های ربات هماهنگ‌کننده انجام می‌شود. این فرمان‌ها به صورت مقادیر از پیش تعیین شده‌ای در بخش *Option* پیام‌ها درج و ارسال می‌شود. این پیام‌ها در جدول ۱ قابل مشاهده است.

در این الگوریتم ابتدا نقطه اعلام‌شده از طرف الگوریتم *Tangent Bug* به عنوان هدف انتخاب می‌شود و مقدار جاذبه و دافعه وارد بر ربات بر اساس این نقطه محاسبه می‌گردد و در صورت نیاز تغییراتی بر روی این نقطه اعمال می‌شود.

بردارهای جاذبه و دافعه مورد استفاده در این تحقیق به صورت (۷) و (۸) تعریف می‌شود

$$Attr_Vector = Q_{dest} \begin{vmatrix} x \\ y \end{vmatrix} - Q_{robot} \begin{vmatrix} x \\ y \end{vmatrix} \quad (7)$$

$$Rep_Vector = Distance(Q_{robot} \begin{vmatrix} x \\ y \end{vmatrix} - CollisionP \begin{vmatrix} x \\ y \end{vmatrix}) \quad (8)$$

بردار جاذبه به صورت تفاضل نقطه هدف و موقعیت فعلی ربات محاسبه می‌شود. در (۸)، *CollisionP* یک نقطه بر روی مانع است که کمترین فاصله را با ربات داشته و توسط یکی از حسگرهای فاصله تعبیه‌شده بر روی ربات مشخص شده است. این نقطه محتمل‌ترین نقطه برای وقوع تصادم مابین ربات و مانع است. در محاسبه بردار دافعه از ضریب *Distance_Value* استفاده می‌شود. این ضریب که بر اساس تفاوت میزان فاصله محاسبه‌شده از مانع نسبت به حدود آستانه *range1* و *range2* توسط (۹) محاسبه می‌شود، تأثیر مانع را در مقدار پتانسیل محاسبه‌شده تعیین می‌کند

$$distance = \begin{cases} \alpha / (range1 - Min_Sen_Value) & \text{for } range1 > Min_Sen_Value \\ \beta / (range2 - Min_Sen_Value) & \text{for } range2 > Min_Sen_Value \end{cases} \quad (9)$$

بردار جابه‌جایی نهایی نیز بر اساس بردارهای دافعه و جاذبه به صورت (۱۰) محاسبه می‌شود. ضریب *W* اعمال‌شده باعث می‌شود تأثیر نقطه میانی تعیین‌شده توسط *Tangent Bug* در مقدار پتانسیل افزایش یابد

$$Dest_Vector = W(Attr_Vector \begin{vmatrix} x \\ y \end{vmatrix}) + Rep_Vector \begin{vmatrix} x \\ y \end{vmatrix} \quad (10)$$

در الگوریتم *Tangent Bug* فقط حسگرهایی که در جهت حرکت ربات قرار دارند بر روی جهت حرکت آن ربات اثر دارند اما با استفاده از تابع میدان پتانسیل هر حسگری که نزدیک بودن بیش از حد به یک مانع را اعلام کند، در جهت حرکت ربات تأثیرگذار خواهد بود. حدود آستانه تعریف‌شده به صورت تجربی و بر اساس آزمایشات مختلف تعیین شده‌اند. اهداف و نقاط میانی تولیدشده توسط برنامه‌ریز محلی در اختیار یک نخ مجزا به نام *Move_Thread* قرار می‌گیرد. این نخ با دریافت نقاط میانی ربات را به سمت آن حرکت می‌دهد. هدف از استفاده از یک نخ مجزا برای اعمال حرکات بر روی ربات‌ها به وجود آوردن امکان تغییر مسیر در زمان حرکت ربات‌ها به صورت لحظه به لحظه است. اگر کل حرکت ربات توسط کنترل‌کننده اصلی ربات انجام شود، به دلیل خطی بودن برنامه، کنترل‌کننده نمی‌تواند مقصد حرکت را قبل از رسیدن به مقصد قبل تغییر دهد. این نخ در زمانی که ربات‌ها با یکدیگر اقدام به جابه‌جایی جسم می‌کنند و یک ربات به دلیل نزدیک شدن به مانع، جهت حرکت جدید خود را به بقیه ربات‌ها اعلام می‌کند، کاملاً مفید است. این نخ در ابتدای شروع کار سیستم شروع به کار کرده و در حالت انتظار برای دریافت یک مقصد از طرف کنترل‌کننده اصلی ربات می‌ماند. مبدأ و مقصد ربات در یک لیست مشترک مابین فرایند اصلی کنترل‌کننده ربات و این نخ درج



شکل ۱۰: ربات بازویی متحرک KUKA youBot مورد استفاده در تحقیق.

ربات‌هایی است که از قبل در شبیه‌ساز Webots پیاده‌سازی شده است. ربات KUKA youBot که در شکل ۱۰ نمایش آمده است، یک ربات بازویی کوچک با ۶ درجه آزادی است که به منظور فراهم کردن امکان جابه‌جایی به یک پایه متحرک هولونومیک که مجهز به ۴ چرخ سوئدی یا مکانیوم^۱ است، تجهیز شده است. در این تحقیق حسگرهای مختلفی به این ربات اضافه شده که امکان دریافت اطلاعات از محیط را فراهم می‌کند. اجسام جابه‌جاشونده و موانع موجود در محیط توسط ۸ دوربین رنگی تعبیه‌شده بر روی ربات و همچنین ۴۸ حسگر فاصله تشخیص داده می‌شود. موانع، اجسام جابه‌جاشونده و ربات‌ها با رنگ‌های خاص توسط سیستم بینایی از یکدیگر قابل تمایز هستند. ربات‌ها می‌توانند از سیستم موقعیت‌یاب جهانی و قطب‌نما برای تعیین موقعیت و جهت حرکت خود در هر لحظه استفاده نمایند. قابلیت گرفتن توسط اتصال‌گرهایی که در انتهای بازوی ربات تعبیه شده‌اند فراهم شده است. با استفاده از فرستنده و گیرنده رادیویی نیز ربات‌ها می‌توانند با یکدیگر یا با سیستم مرکزی ارتباط برقرار کرده و اقدام به ارسال و دریافت پیام نمایند.

محیط شبیه‌سازی طراحی‌شده در شبیه‌ساز Webots یک محیط بسته با ابعاد 10×10 مترمربع فرض شده که توسط دیوارهای دور آن از فضای بیرونی متمایز می‌شود. بنابراین ربات‌ها در زمان انجام وظیفه خود محدود به حرکت در فضای مشخصی هستند و حق ندارند جسم را از محدوده خارج کنند. برخی از مفروضاتی که در انجام شبیه‌سازی لحاظ شده‌اند عبارتند از حداکثر سرعت حرکت پایه ربات و بازوی ربات که به ترتیب ۵ متر در ثانیه و 0.5 متر در ثانیه است. میزان خطای فاصله قابل قبول برای رسیدن به موقعیت هدف در زمان حرکت ربات 0.02 متر و میزان خطای دوران قابل قبول برای رسیدن به وضعیت هدف در زمان حرکت ربات نیز ۳ درجه فرض شده است. هیچ خطایی در ارتباط رادیویی وجود ندارد. جسم انتخاب‌شده برای جابه‌جایی یک مکعب مستطیل با ابعاد یک جسم مکعب‌شکل با $0.4 \times 0.18 \times 0.06$ است که با رنگ سبز از بقیه اشیای موجود در محیط قابل تمایز است.

سناریوهای طراحی‌شده در دو دسته طبقه‌بندی شده‌اند. در نمونه‌های گروه اول هیچ مانعی در محیط وجود ندارد. در حالی که محیط مربوط به نمونه‌های گروه دوم دارای تعداد دلخواهی مانع با اندازه دلخواه و در موقعیت‌های دلخواه هستند. در طراحی سناریوها سعی شده تا با طراحی موانعی متفاوت با یکدیگر و طرح محیط‌های متفاوت و قراردادن ربات‌ها، موانع و جسم جابه‌جاشونده در موقعیت‌های مختلف، جنبه‌های مختلف

جدول ۱: پیام‌های مورد استفاده برای هماهنگی محلی ربات‌ها.

Option	محتوا
LPP (Local Path Planning)	اعلام ساخت مسیر سراسری به صورت محلی
Y (Yes)	موافقت با پیغام قبلی هماهنگ‌کننده
GPP (Global Path Planning)	درخواست مسیر سراسری از ناظر
Arrive (A)	تأیید رسیدن به نقطه هدف میانی
RT (Ready to Take)	آمادگی برای برداشتن جسم
S (Stop)	توقف حرکت
T (Take)	دستور برداشتن جسم
P (Put)	گذاشتن جسم بر روی زمین

در زمان بلندکردن جسم، ربات هماهنگ‌کننده پیغام RT را با هدف کسب اطلاع از این که ربات‌ها در مکان‌های مشخص‌شده قرار داشته و از جسم گرفته و آماده بلندکردن جسم هستند، ارسال می‌کند و منتظر جواب ربات‌های مذکور می‌ماند. بقیه ربات‌ها در صورت آمادگی برای برداشتن جسم، پیغام Y را برای ربات هماهنگ‌کننده ارسال می‌کنند. پس از این که ربات هماهنگ‌کننده از آمادگی همه ربات‌ها برای برداشتن جسم مطمئن شد، پیغام T را به صورت همه‌پخشی ارسال می‌کند. همه ربات‌ها با دریافت این پیغام اقدام به بلندکردن جسم خواهند کرد.

شروع به حرکت ربات‌ها پس از مکث ۳ ثانیه‌ای بعد از دریافت پیام T انجام می‌شود. در زمان حرکت با توجه به حرکت گام به گام ربات‌ها، هر ربات با طی گام مذکور و رسیدن به هدف میانی پیامی با محتوای A ارسال می‌کند. زمانی که ربات هماهنگ‌کننده از وصول این پیام از طرف تک‌تک ربات‌ها مطمئن شد، خود نیز یک پیام حاوی کاراکتر A برای بقیه ارسال می‌کند و اعضای گروه با دریافت این پیام اقدام به حرکت به سمت نقطه هدف میانی بعدی می‌کنند. زمانی که ربات‌ها به نقطه هدف اصلی برسند، با هماهنگی کامل و پس از دریافت پیام P از طرف ربات هماهنگ‌کننده، با یکدیگر اقدام به زمین گذاشتن جسم کرده و به کار خود پایان می‌دهند.

۴- شبیه‌سازی، نتایج و تحلیل

با توجه به محدودیت‌های موجود و عدم دسترسی به ربات‌های مورد نیاز، از یک محیط شبیه‌سازی به نام Webots در انجام این تحقیق بهره گرفته شده است. Webots یک محیط توسعه به منظور مدل‌کردن، برنامه‌ریزی و شبیه‌سازی ربات‌های متحرک است. این برنامه امکان طراحی محیط‌های رباتیک به کمک یک یا چندین ربات یکسان یا غیر یکسان را می‌دهد. تمامی خصوصیات اشیای موجود در محیط مانند شکل، رنگ، اندازه، جرم، جنس و ... قابل تعیین است. شبیه‌ساز امکان اضافه‌کردن حسگرها و محرک‌های جدید به ربات‌ها را فراهم نموده است. کنترل‌کننده ربات می‌تواند توسط کاربر برنامه‌نویسی شده و رفتار ربات در یک محیط فیزیکی واقعی آزمایش شود. در این محیط مسایل فیزیکی مانند نیرو، تکانه و گرانش در نظر گرفته شده‌اند. در انجام تحقیق پیش رو از نسخه ۷.۱.۱ Webots به همراه زبان برنامه‌نویسی جاوا بهره گرفته شده است.

هسته اصلی مسأله جابه‌جایی یک جسم توسط یک سیستم چندرباتی، ربات‌های مورد استفاده در آن است. از این رو انتخاب یک ربات مناسب که قابلیت مورد نیاز برای جابه‌جایی اجسام به صورت تیمی را داشته باشد حایز اهمیت است. در این پژوهش بر اساس نیازمندی‌های مسأله، ربات KUKA youBot برای استفاده در مسأله انتخاب شد. این ربات جزو

رویکرد پیشنهادی با نتایج حاصل از سناریوهای فاقد مانع، مشخص و ساده‌بودن تعیین مسیر در این محیطها است. در این محیطها هدف اصلی نشان‌دادن نحوه هماهنگی ربات‌ها است. از طرفی با توجه به استفاده از ربات‌های بازویی در این تحقیق، مقایسه روش انتقال جسم از طریق گرفتن و بلندکردن توسط این ربات‌ها در مقابل روش‌های دیگر مانند رانش چندان معنی‌دار نیست زیرا به دلیل وضعیت بازوی این ربات‌ها مناسب‌ترین روش انتقال جسم توسط این ربات‌ها، گرفتن است.

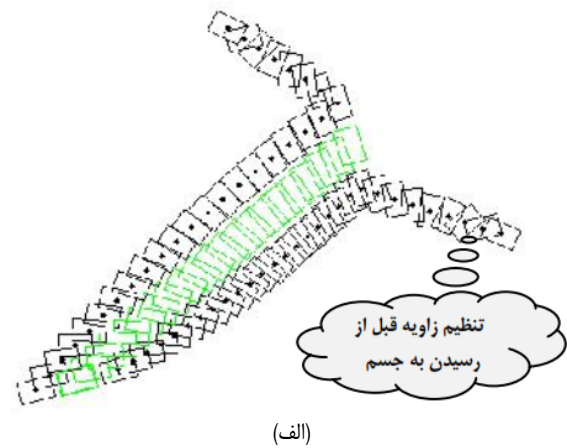
نتایج حرکت ربات‌ها و انتقال جسم با استفاده از هر دو روش در جدولی در ادامه مقاله مشخص شده است. تفاوت دو مسیر طی‌شده توسط ربات‌ها در این دو روش، نشانگر میزان بهینگی الگوریتم پیشنهادی است. همچنین میزان خطای دورانی و فاصله‌ای ربات‌ها در قراردادن جسم در موقعیت هدف مشخص‌شده نیز مبنای دیگری برای سنجش کارایی رویکرد پیاده‌سازی شده است. در همه مسایل نمایش داده شده، مستطیلی که هیچ علامتی در داخل آن وجود ندارد، جسم جابه‌جاشونده و مستطیل با علامت‌هایی در مرکز، ربات‌ها هستند. در زمان استفاده از الگوریتم‌های RRT و ORT در لایه سراسری، فاصله هر نقطه تولیدشده با نقاط مجاور یا طول گام حرکتی ۲۰ سانتی‌متر در نظر گرفته شده است.

در شکل ۱۱ سه نمونه از مسایلی که در آن ربات‌ها جسم جابه‌جاشونده را در غیاب موانع به نقطه هدف منتقل می‌کنند مشاهده می‌شود. در نمونه‌های الف و ب از یک سیستم دورباتی استفاده شده است، حال آن که سیستم مورد استفاده در مسأله سوم یک سیستم سه‌رباتی است.

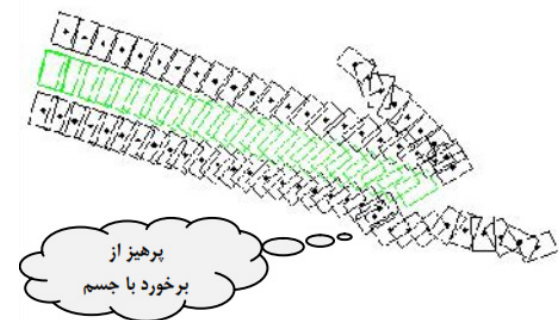
همان‌طور که در شکل ۱۱ مشخص شده است، ربات‌ها قبل از رسیدن به مجاورت جسم جابه‌جاشونده، زاویه خود با جسم را تنظیم می‌کنند و این تنظیم زاویه در حین حرکت انجام می‌شود. از طرفی با توجه به این که ربات‌ها باید در مجاورت وجه خاصی از جسم قرار بگیرند تا جسم را بردارند، در زمان نزدیک‌شدن ربات‌ها به جسم در صورتی که در سمت آن وجه قرار نداشته باشند و نیاز به دورزدن جسم داشته باشند، جسم جابه‌جاشونده را یک مانع فرض کرده و بدون برخورد با آن، آن را دور می‌زنند تا در نهایت به وجه مورد نظر خود برسند. در شکل ۱۲ مراحل حل مسأله نمایش داده شده در شکل ۱۱-ج در محیط شبیه‌ساز Webots ترسیم شده است. عملکرد هماهنگ ربات‌ها در لحظه برداشتن و زمین گذاشتن جسم در این شکل کاملاً مشهود است. اطلاعات مربوط به حمل جسم در سه مسأله نمایش داده شده در شکل ۱۱ در جدول ۲ آمده است. با توجه به مقادیر درج‌شده در جدول ۲، مشخص است که ربات‌ها پس از انتقال جسم آن را با کمی خطا در موقعیت هدف قرار داده‌اند. البته این خطا ناچیز بوده و قابل چشم‌پوشی است.

در بخش دوم سناریوها، تعدادی مانع ایستا به صورت دلخواه در محیط قرار داده شده و ربات‌ها می‌بایست بدون برخورد با این موانع خود را به مجاورت جسم جابه‌جاشونده برسانند و در ادامه به صورت هماهنگ جسم را از یک مسیر عاری از تصادم به نقطه هدف برسانند. در شکل‌های ۱۳ تا ۱۶، ۴ مسأله مختلف نمایش داده شده است. شکل‌هایی که با نمایه ۱ مشخص شده‌اند، نشانگر مسیر نسبتاً بهینه‌ای هستند که توسط الگوریتم ORT ایجاد و توسط ربات‌ها و جسم طی شده است. شکل‌های با نمایه ۲ نیز نمایشگر مسیر طی‌شده توسط جسم و ربات‌ها در زمان استفاده از الگوریتم RRT است.

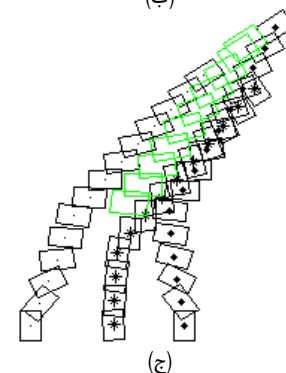
همان‌طور که از شکل‌های نمایش داده شده برای هر مسأله مشخص است، در هر ۴ مسأله به کمک هر دو الگوریتم RRT و ORT ربات‌ها توانسته‌اند با طی یک مسیر عاری از تصادم جسم را به نقطه هدف منتقل کنند. در زمان انتقال جسم در صورتی که یکی از ربات‌ها بیش از حد به



(الف)



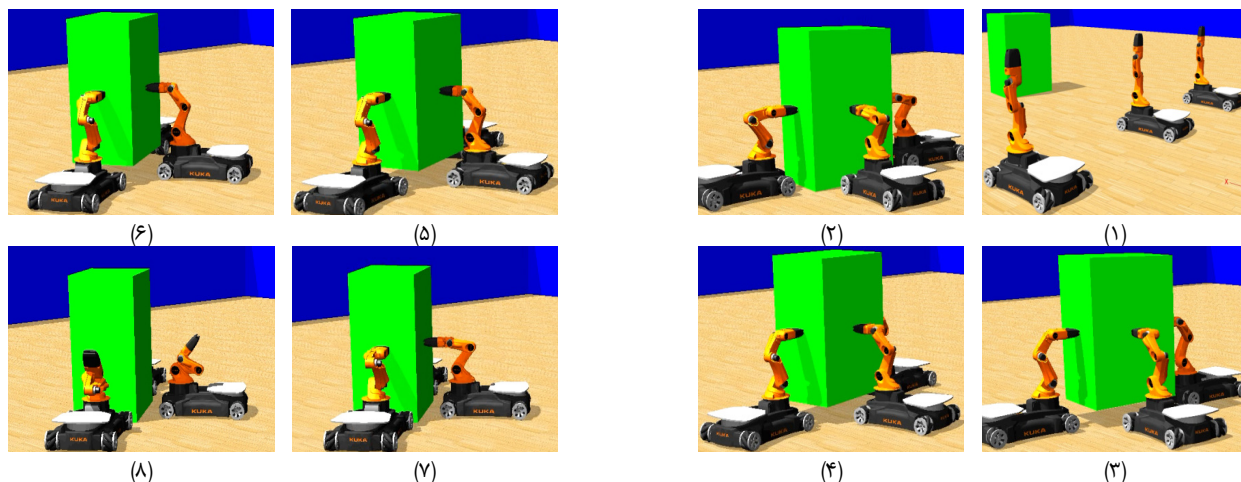
(ب)



(ج)

شکل ۱۱: جابه‌جایی اجسام در غیاب موانع، (الف) نمونه شماره ۱، (ب) نمونه شماره ۲ و (ج) نمونه شماره ۳.

معماری ارائه‌شده به طور کامل بررسی شود. با توجه به این که هدف محقق ارائه یک روش کامل برای انجام کار است و بهینگی جواب در درجه دوم اهمیت قرار دارد، مبنای ارزیابی کارایی معماری و الگوریتم توسعه داده شده انجام درست کار است. با این حال برای پی‌بردن به برتری روش پیشنهادی مبتنی بر الگوریتم ORT، نسبت به روش‌های دیگری که از الگوریتم‌های جستجوی محیط استفاده می‌کنند و ارزیابی میزان بهینگی مسیر طی‌شده توسط ربات‌ها برای انتقال جسم مورد نظر، سناریوهای طراحی‌شده برای محیط حاوی مانع، یک بار به کمک الگوریتم RRT نیز اجرا شده و نتیجه آنها ثبت گردیده است. الگوریتم RRT یکی از الگوریتم‌های شناخته‌شده و پرکاربرد برای تعیین مسیر است. این الگوریتم جزو الگوریتم‌های مبتنی بر نمونه‌برداری محسوب می‌شود که با تولید نقاط تصادفی در محیط، یک درخت از مبدأ تعریف‌شده به نقطه هدف تعیین‌شده ایجاد می‌کند. اثبات شده که این الگوریتم کامل است. با توجه به این که قابلیت اصلی الگوریتم ORT بی‌نیازی از هموارسازی مسیر نهایی است، در زمان استفاده از الگوریتم RRT نیز از هیچ روش هموارسازی مسیر استفاده نشده است. علت عدم سنجش



شکل ۱۲: مراحل مربوط به جابه‌جایی جسم در شکل ۸-ب.

جدول ۲: اطلاعات مربوط به انتقال جسم در مسایل نمایش داده شده در شکل ۸

نمونه	موقعیت اولیه ربات ۱	موقعیت اولیه ربات ۲	موقعیت اولیه ربات ۳	موقعیت اولیه جعبه (m)	زاویه اولیه جعبه (Deg)	موقعیت هدف (m)	زاویه هدف (Deg)	خطا در هدف (m)	انحراف زاویه (Deg)	زمان حل (s)
الف	(۹،۴،۵)	(۴،۵،۸)	-	(۶،۵،۶)	۱۰،۵	(۲،۲)	۱۹،۵	۰،۱۰۱۸	۲،۶	۱۷۶
ب	(۶،۵)	(۹،۲)	-	(۷،۳)	۳۰	(۶،۲)	۷،۵	۰،۱۰۱۲	۲،۵	۱۴۹
ج	(۰،۵،۱)	(۱،۸،۱)	(۱،۵،۳)	(۲،۳)	۸۰	(۶،۴)	۱۲،۵	۰،۱۰۱۰	۲،۱	۱۲۶

جدول ۳: اطلاعات مربوط به انتقال جسم در مسایل نمایش داده شده در شکل‌های ۱۰ تا ۱۳.

نمونه	موقعیت اولیه ربات ۱	موقعیت اولیه ربات ۲	موقعیت اولیه جعبه (m)	زاویه اولیه جعبه (Deg)	موقعیت هدف (m)	زاویه هدف (Deg)	خطا در هدف (m)	انحراف زاویه (Deg)
۴	(۶،۸،۵)	(۳،۵،۹)	(۵،۸)	۰	(۵،۲)	۹۰	۰،۱۰۱۶	۲،۳
۵	(۷،۰،۵)	(۸،۱،۵)	(۱،۵،۴)	۰	(۳،۸)	۰	۰،۱۰۱۸	۲،۸
۶	(۲،۵،۵)	(۶،۵،۹)	(۳،۷،۷)	۰	(۹،۲۵،۲)	۹۰	۰،۱۰۱۷	۲،۶
۷	(۳،۵،۲)	(۰،۵،۱،۵)	(۱،۵،۳)	۲۷۰	(۹،۲،۵)	۰	۰،۱۰۱۲	۲،۹

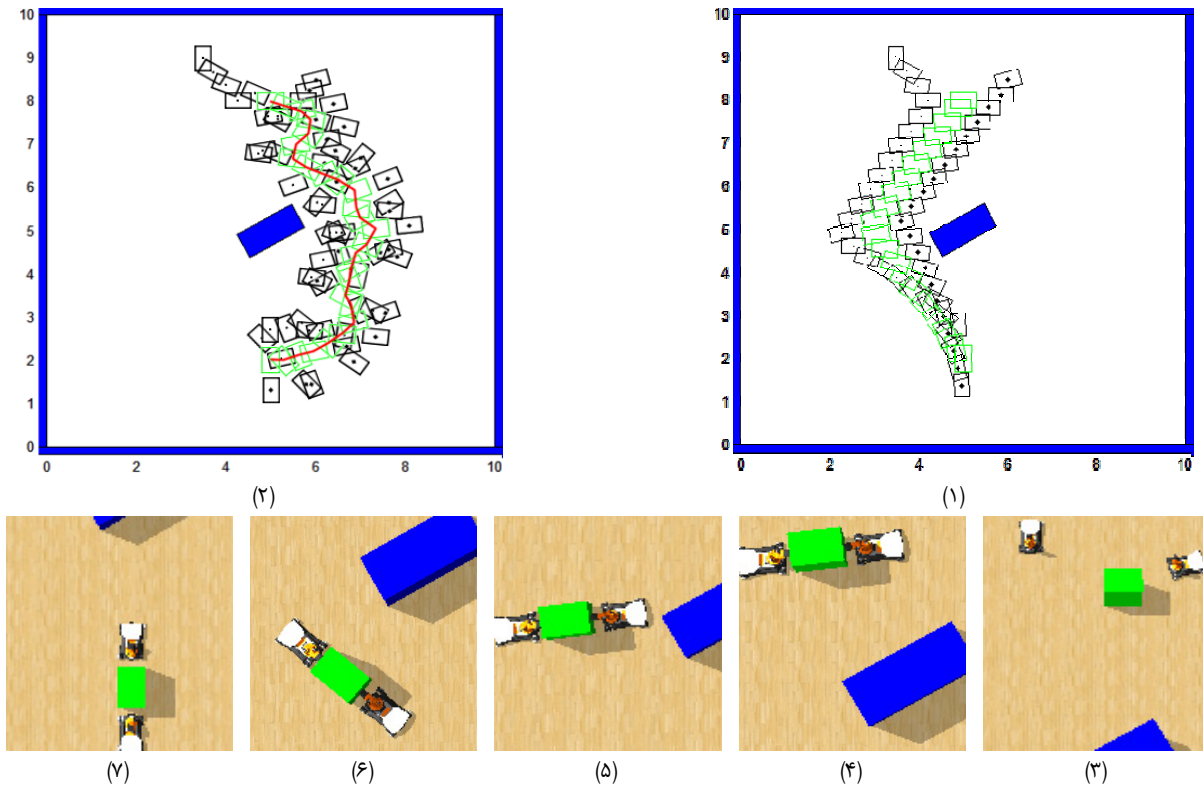
جسم و موانع می‌شود.

جدول ۳ شامل اطلاعات مربوط به مسایل نمایش داده شده در شکل‌های ۱۳ تا ۱۶ است. طبق جدول ۳ نیز می‌توان به میزان خطا در انتقال جسم جابه‌جاشونده در هر مسأله پی برد. خطای رخ داده با توجه به خطاهای موجود در حرکات فیزیکی چرخ‌ها و بازوهای ربات‌ها، بسیار ناچیز بوده و قابل اغماض است. با توجه به جدول ۳ می‌توان ادعا نمود که ربات‌ها در روش پیشنهادی به صورت قابل قبولی توانسته‌اند جسم را منتقل کنند.

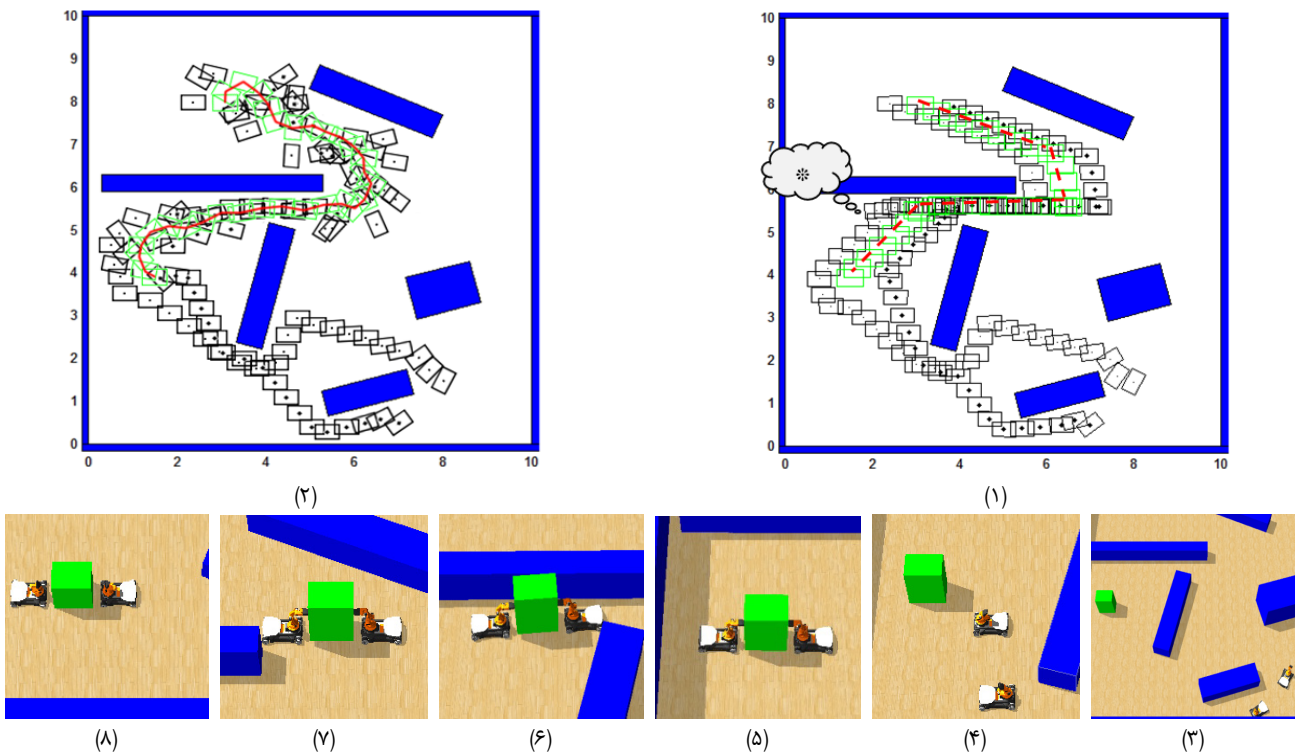
در نهایت در جدول ۴ یک مقایسه بین دو مسیر طی‌شده توسط الگوریتم ORT و الگوریتم RRT ارائه شده است. اولین نکته‌ای که باید به آن توجه کرد این است که با توجه به انتخاب تصادفی نقاط در هر دو الگوریتم ORT و RRT، در هر بار اجرا نتایج متفاوتی به دست می‌آید که این تفاوت در خود مسیر، طول آن و زمان حل مسأله است. بنابراین مسیر انتقال جسم از مبدأ به مقصد یکتا نیست.

نتایج جدول ۴ نشانگر تفاوت در طول دو مسیر مشخص شده است. در حالت کلی طول مسیر طی‌شده توسط الگوریتم ORT کوتاه‌تر است و این مسأله به خاطر انتخاب آگاهانه نقاط و همچنین هموارسازی موجود در ذات این الگوریتم است. این در حالی است که با دقت در راه حل تولیدشده توسط الگوریتم RRT در هر چهار مسأله متوجه می‌شویم که ربات‌ها و جسم در این الگوریتم حرکات اضافی و دوران‌های زاید زیادی دارند. این مسأله به دلیل عدم استفاده از روش‌های هموارسازی در کنار این الگوریتم

یک مانع نزدیک شود، مسیر خود را تصحیح کرده و این مسأله را به ربات‌های دیگر اطلاع می‌دهد. برای نمونه در شکل ۱۴، تغییر مسیر توسط یکی از ربات‌ها و تأثیر آن بر روی حرکت ربات دیگر در موقعیتی که با علامت * نشان داده شده است، کاملاً مشخص است. در این وضعیت رباتی که با نقطه مرکزی بزرگ‌تر مشخص شده است، در زمان حرکت به مانعی که با عدد ۳ مشخص شده، بیش از حد نزدیک گردیده است. بنابراین ربات برای جلوگیری از برخورد با مانع، به سمت چپ حرکت کرده و این مسأله را به ربات دیگر نیز اطلاع داده است که باعث شده تا مسیر انتقال جسم تغییر کند. نکته قابل توجه انتخاب مسیرهای مناسب در زمان حل مسأله است. در شکل ۱۴ ربات‌ها می‌توانند جسم را از مسیر عبوری از پایین مانع ۳ منتقل کنند. اما این مسیر یک مسیر غیر بهینه و نامناسب است که الگوریتم ORT از انتخاب آن خودداری کرده است. مشابه این وضعیت در مسأله نمایش داده شده در شکل ۱۵ نیز قابل رؤیت است. در این مسأله نیز ربات‌ها می‌توانند جسم را از طریق مسیر غیر بهینه عبوری از سمت چپ ربات مانع ۲ عبور دهند اما این مسیر به دلیل غیر بهینگی انتخاب نمی‌شود. نکته دیگر قابل ذکر در انتقال جسم با استفاده از الگوریتم ORT، دوران جسم توسط ربات‌ها در زمان وجود فضای کافی برای دوران است. در صورتی که مسیر نیازمند دوران باشد یا وضعیت جسم در نقطه مبدأ و مقصد از نظر دورانی متفاوت باشد، ربات‌ها در زمان وجود فضای کافی و در طول مسیر و نه به صورت یک‌باره، جسم را دوران می‌دهند. این مسأله نیز مانع وقوع تصادم بین



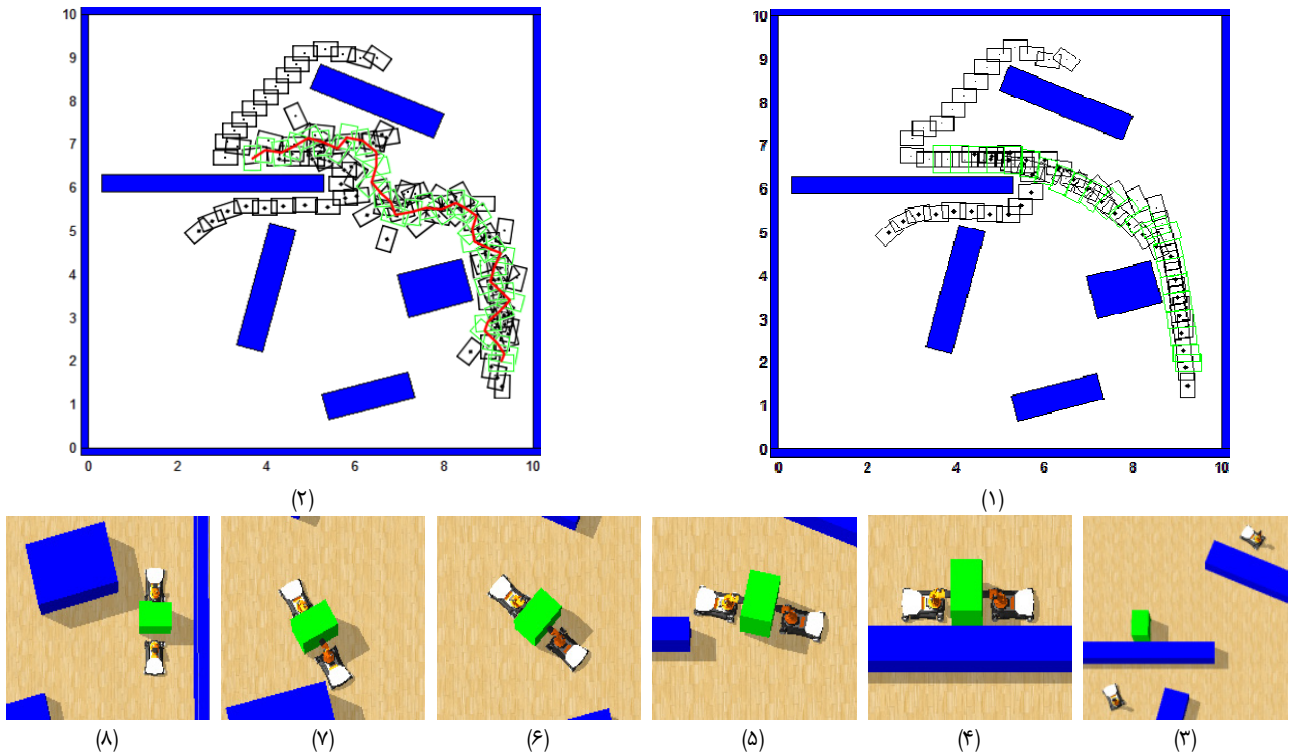
شکل ۱۳: نمونه شماره ۴، (۱) مسیر در الگوریتم ORT، (۲) مسیر در الگوریتم RRT و (۳) تا (۷) نمایشی از حل مسأله در محیط Webots.



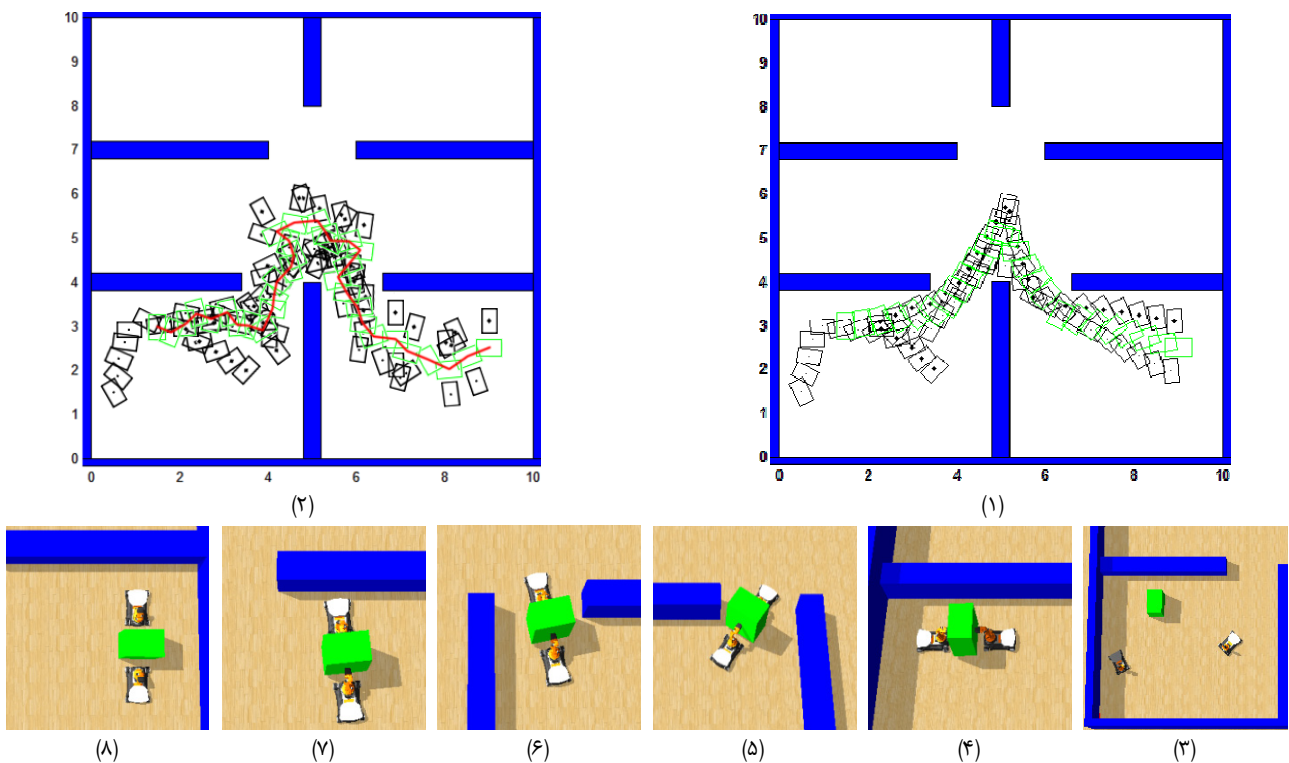
شکل ۱۴: نمونه شماره ۵، (۱) مسیر در الگوریتم ORT، (۲) مسیر در الگوریتم RRT و (۳) تا (۸) نمایشی از حل مسأله در محیط Webots.

جدول ۴: مقایسه مسیر ایجادشده بدون در نظر گرفتن ربات‌ها و مسیر طی شده در مسایل مربوط به شکل‌های ۱۰ تا ۱۳.

نمونه	طول مسیر با الگوریتم ORT	طول مسیر با الگوریتم RRT	تعداد گره‌های ایجادشده با الگوریتم ORT	تعداد گره‌های ایجادشده با الگوریتم RRT	زمان حل با الگوریتم ORT	زمان حل با الگوریتم RRT
۴	۷٫۷۵ متر	۸٫۹۵ متر	۱۶۵۷	۸۹۷	۱۷۳ ثانیه	۱۶۷ ثانیه
۵	۱۰٫۹۵ متر	۱۱٫۸۰ متر	۳۳۹۴	۲۰۵۴	۲۹۴ ثانیه	۲۷۱ ثانیه
۶	۹٫۲۰ متر	۱۰٫۴۵ متر	۲۹۴۴	۱۵۱۲	۳۱۰ ثانیه	۲۸۴ ثانیه
۷	۹٫۷۵ متر	۱۰٫۵۰ متر	۲۸۵۸	۱۶۴۵	۲۷۷ ثانیه	۲۵۲ ثانیه



شکل ۱۵: نمونه شماره ۶، مسیر در الگوریتم ORT، مسیر در الگوریتم RRT و (۳) تا (۸) نمایشی از حل مسأله در محیط Webots.



شکل ۱۶: نمونه شماره ۷، مسیر در الگوریتم ORT، مسیر در الگوریتم RRT و (۳) تا (۸) نمایشی از حل مسأله در محیط Webots.

الگوریتم بسیار بیشتر از گره‌های تولیدشده در الگوریتم RRT باشد. همچنین فرایند انتخاب و اتصال آگاهانه در الگوریتم ORT، باعث شده تا زمان اجرای این الگوریتم نیز بسیار بیشتر از الگوریتم RRT باشد. البته باید توجه کرد که در جدول ۴ منظور از زمان حل، کل زمان لازم برای ایجاد مسیر و انتقال جسم توسط ربات‌ها است. با توجه به این که طول مسیر حرکت در الگوریتم RRT بیشتر است، ربات‌ها به زمان بیشتری برای انتقال جسم نیاز دارند و این مسأله باعث شده تا تفاوت زمانی حل کامل مسأله در دو روش مذکور کمتر شود.

است که در نتیجه باعث شده طول مسیر ایجادشده توسط این الگوریتم نیز بیشتر از مسیر ایجادشده توسط الگوریتم ORT باشد. به طور مشخص یافتن مسیر کوتاه‌تر در الگوریتم ORT هزینه‌های بیشتری نسبت به الگوریتم RRT در پی دارد. این هزینه‌ها در تعداد گره‌های ایجادشده و زمان اجرای الگوریتم مشخص است. در خط ۱۴ الگوریتم ۲، سعی می‌کند تا گره جدید را به یکی از گره‌های تولیدشده تاکنون از ریشه تا والد فعلی گره متصل کرده و گره‌های جدیدی بر روی مسیر اتصال تعریف نماید. این کار باعث می‌شود که مجموع گره‌های تولیدشده در این

۵- تحقیقات آتی و جمع‌بندی

در این مقاله یک رویکرد ترکیبی برای حل مسأله جابه‌جایی اجسام توسط سیستم‌های چندرباتی همگن ارائه شد. رویکرد مذکور به دلیل ساختار دولایه از مزایای هر دو معماری متمرکز و غیر متمرکز بهره می‌گیرد. لایه سراسری وظیفه در اختیار گذاشتن اطلاعات مورد نیاز ربات‌ها را داراست و همچنین این لایه در مواقع لزوم می‌تواند یک مسیر سراسری برای انتقال جسم تعریف و به ربات‌ها اطلاع دهد. برای تعریف مسیر سراسری از الگوریتم ORT که یک الگوریتم مبتنی بر نمونه‌برداری می‌باشد استفاده شده است. لایه محلی که در تک‌تک ربات‌ها پیاده‌سازی شده است، وظیفه کنترل حرکات ربات‌ها را بر عهده دارد. این لایه به کمک ترکیب الگوریتم‌های Tangent Bug و میدان پتانسیل یک مسیر عاری از تصادم برای حرکت هر ربات تولید می‌کند.

رویکرد پیشنهادی در این مقاله به کمک زبان برنامه‌نویسی جاوا و در شبیه‌ساز Webots بر روی ربات‌های KUKA youBot پیاده‌سازی و آزمایش شده است. برای آزمایش رویکرد مذکور سه مسأله جابه‌جایی جسم در غیاب مانع و چهار مسأله جابه‌جایی جسم در حضور موانع طراحی شده است. نتایج حاصل از استفاده از رویکرد پیشنهادی و الگوریتم ORT با نتیجه حاصل از الگوریتم RRT مقایسه شده که طبق نتایج حاصل مشخص است که در همه مسایل رویکرد پیشنهادی به درستی مسأله را حل کرده است. مسیر ایجادشده توسط رویکرد پیشنهادی یک مسیر نسبتاً بهینه است که در زمان استفاده از الگوریتم ORT نسبت به الگوریتم RRT کوتاه‌تر است. البته باید توجه نمود که به دلیل به دست آوردن مسیر مناسب‌تر، رویکرد پیشنهادی در زمان استفاده از الگوریتم ORT نیاز به زمان بیشتری برای حل مسأله دارد.

با توجه به این که روش انتقال جسم در این تحقیق به گرفتن برداشتن محدود شده و این روش در برخی مسایل قابل اعمال نیست، در ادامه این تحقیق می‌توان از ربات‌های دیگر استفاده نمود و روش‌های دیگر انتقال جسم را بررسی کرد. از طرفی با توجه به این که روش برقراری ارتباط در رویکرد پیشنهادی ارتباط صریح است، ممکن است در برخی محیط‌های پرخطا نتوان از چنین ارتباطی استفاده کرد. به همین دلیل توسعه یک روش ارتباط غیر صریح می‌تواند مفید باشد. در مسأله این تحقیق جسم جابه‌جاشونده نیز به اجسام مکعب محدود شده است. این محدودیت را نیز می‌توان در تحقیقات آتی حذف کرده و از رویکرد پیشنهادی برای انتقال اجسام با شکل‌های دلخواه استفاده نمود.

مراجع

- [7] M. Nili Ahmadabadi and E. Nakano, "A 'constrain and move' approach to distributed object manipulation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 157-172, Apr. 2001.
- [8] M. Eslamy and S. A. A. Moosavian, "Dynamics and cooperative object manipulation control of suspended mobile manipulators," *J. of Intelligent & Robotic Systems*, vol. 60, no. 2, pp. 181-199, Nov. 2010.
- [9] A. Zaerpoora, M. N. Ahmadabadi, M. R. Barunia, and Z. D. Wang, "Distributed object transportation on a desired path based on constrain and move strategy," *Robotics and Autonomous Systems*, vol. 50, no. 2-3, pp. 115-128, Feb. 2005.
- [10] O. Khatib, "Mobile manipulation: the robotic assistant," *Robotics and Autonomous Systems*, vol. 26, no. 2-3, pp. 175-183, Feb. 1999.
- [11] I. Mas and C. Kitts, "Object manipulation using cooperative mobile multi-robot systems," in *Proc. of the World Congress on Engineering and Computer Science*, vol. 1, pp. 324-329, 2012.
- [12] E. Yang and D. Gu, *Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey*, Department of Computer Science, Univeristy of Essex, Tech. Report, 2004.
- [13] Z. Wang, E. Nakano, and T. Matsukawa, "Cooperating multiple behavior-based robots for object manipulation," in *Proc. IEEE/RSJ/Int. Conf. on Distributed Autonomous Robotic Systems*, vol. 3, pp. 371-382, Munich, Germany, 12-16 Sep. 1994.
- [۱۴] ع. ا. امینی، یادگیری همکاری در سیستم‌های جابه‌جایی اجسام، پایان‌نامه دوره کارشناسی ارشد مهندسی رباتیک- دانشکده فنی و مهندسی، دانشگاه تهران، ۱۳۸۵.
- [۱۵] آ. زائرپور، طراحی و تحلیل یک پروتکل هماهنگی در یک تیم گسترده رباتیک جهت جابه‌جایی اجسام در یک صفحه دوبعدی، پایان‌نامه دوره کارشناسی ارشد مهندسی رباتیک- دانشکده فنی و مهندسی، دانشگاه تهران، ۱۳۸۰.
- [16] K. Kosuge and T. Oosumi, "Decentralized control of multiple robots handling an object," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 318-323, Osaka, Japan, 4-8 Nov. 1996.
- [17] J. Spletzer, et al., "Cooperative localization and control for multi-robot manipulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 631-636, 29 Oct.-3 Nov. 2001.
- [18] Z. Wang, Y. Takano, Y. Hirata, and K. Kosuge, "Decentralized cooperative object transportation by multiple mobile robots with a pushing leader," in *Distributed Autonomous Robotic Systems 6*, Ed: Springer, pp. 453-462, 2007.
- [19] R. Gross and M. Dorigo, "Towards group transport by swarms of robots," *Int. J. of Bio-Inspired Computation*, vol. 1, no. 1-2, pp. 1-13, Jan. 2009.
- [20] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 85-101, Jan. 2000.
- [21] H. Frank, N. Wellerdick-Wojtasik, B. Hagebeucker, G. Novak, and S. Mahlknecht, "Throwing objects- a bio-inspired approach for the transportation of parts," in *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, pp. 91-96, 17-20 Dec. 2006.
- [22] K. F. Bohringer, V. Bhatt, and K. Y. Goldberg, "Sensorless manipulation using transverse vibrations of a plate," in *Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya*, vol. 2, pp. 1989-1996, 21-27 May 1995.
- [23] S. Cheon, K. Ryu, and Y. Oh, "Object manipulation using robot arm-hand system," in *Proc. 10th Int. Conf. on Ubiquitous Robots and Ambient Intelligence, URAI'13*, pp. 163-166, 30 Oct.-2 Nov. 2013.
- [24] P. Dzitac and A. M. Mazid, "Factors that influence reliable object manipulation," in *Proc. 8th IEEE Conf. on, Industrial Electronics and Applications, ICIEA'13*, pp. 1468-1473, 19-21 Jun. 2013.
- [25] K. Harada, M. Kaneko, and T. Tsuji, "Rolling-based manipulation for multiple objects," in *Proc. IEEE Int. Conf. on Robotics and Automation, ICRA'00*, vol. 4, pp. 457-468, San Francisco, CA, USA, 24-28 Apr. 2000.
- [26] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International J. of Robotics Research*, vol. 27, pp. 157-173, 2008.
- [27] M. J. Mataric, M. Nilsson, and K. Simsarin, "Cooperative multi-robot box-pushing," in *Proc. IEEE/RSJ Int. Conf. on, Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots'*, vol. ???, pp. 556-561, 5-9 Aug. 1995.
- [28] Y. Wang and C. W. de Silva, "Multi-robot box-pushing: single-agent q-learning vs. team q-learning," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 3694-3699, Beijing, China, 9-15 Oct. 2006.
- [29] Z. Wang, Y. Hirata, and K. Kosuge, "An algorithm for testing object caging condition by multiple mobile robots," in *Proc. IEEE Int.*
- [1] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *Robotics and Automation, IEEE Trans. on*, vol. 19, no. 2, pp. 223-237, Apr. 2003.
- [2] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, the MIT Press, 1999.
- [3] M. Nemrava and P. Cermak, "Solving the box-pushing problem by master-slave robots," *J. of Automation, Mobile Robotics & Intelligent Systems*, vol. 2, no. 3, pp. 32-37, 2008.
- [4] Z. Wang and V. Kumar, "Object closure and manipulation by multiple cooperating mobile robots," in *Proc. 6th Int. Symp. on Distributed Autonomous Robotic Systems*, vol. 1, pp. 394-399, 2002.
- [5] A. Khozaee, A. H. Aminaiee, and A. Ghaffari, "A swarm robotic approach to distributed object pushing using fuzzy controllers," in *Proc. IEEE Int. Conf. on Robotics and Biomimetics, ROBIO'09*, pp. 1117-1122, Bangkok, Thailand, 2009.
- [6] A. C. Satici and M. W. Spong, "Nonholonomic cooperative manipulation of polygonal objects in the plane," in *Proc. IEEE 51st Annual Conf. on Decision and Control, CDC'12*, pp. 2439-2446, 10-13 Dec. 2012.

الیبس مسیحی تحصیلات خود در مقاطع کارشناسی و کارشناسی ارشد را در رشته مهندسی صنایع به ترتیب در سال‌های ۱۳۷۳ و ۱۳۷۶ در دانشگاه علم و صنعت ایران و دکتری مهندسی صنایع را در سال ۱۳۸۳ در دانشگاه تربیت مدرس به اتمام رساند و هم‌اکنون عضو هیأت علمی دانشکده مهندسی صنایع و سیستم‌ها در دانشگاه تربیت مدرس می‌باشد. راهنمایی حدود ۴۰ دانشجوی کارشناسی‌ارشد و دکتری و نگارش دو کتاب و بیش از ۱۰۰ مقاله در کتاب‌ها، مجلات، و کنفرانس‌های بین‌المللی حاصل فعالیت‌های پژوهشی وی می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: رباتیک و برنامه‌ریزی حرکت، اتوماسیون و ساخت و تولید پیشرفته، هوش مصنوعی، و بهینه‌سازی ترکیبیاتی و فراابتکاری.

Conf. on Intelligent Robots and Systems, pp. 3022-3027, 2-6 Aug. 2005.

[۳۰] ی. کاتبی، برنامه‌ریزی حرکت و تخصیص کار در سیستم‌های چندرباتی، رساله دوره دکتری مهندسی صنایع- صنایع، دانشگاه تربیت مدرس، ۱۳۹۲.

[31] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, the MIT Press, 2005.

[32] V. J. Lumelsky, *Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World*, John Wiley & Sons, 2005.

[33] J. B. Mbede, X. Huang, and M. Wang, "Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators," *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 61-72, Jul. 2000.

ظاهر حکمت‌فر در سال ۱۳۸۹ مدرک کارشناسی مهندسی فناوری اطلاعات خود را از دانشگاه تبریز و در سال ۱۳۹۳ مدرک کارشناسی ارشد مهندسی فناوری اطلاعات (گرایش سیستم‌های تکنولوژی اطلاعات) خود را از دانشگاه تربیت مدرس تهران دریافت نمود. در حال حاضر وی تحصیلات دوره دکتری تخصصی خود را در رشته مهندسی فناوری اطلاعات (گرایش مدیریت سیستم‌های اطلاعاتی) در دانشگاه تهران پیگیری می‌نماید. زمینه‌های تحقیقاتی مورد علاقه نامبرده شامل موضوعاتی مانند الگوریتم‌های هیوریستیک، مسیریابی حرکت، برنامه‌ریزی حرکت، داده‌کاوی و سیستم‌های توصیه‌گر می‌باشد.