

طراحی مسیر درخت جستجوی سریع تصادفی برای ربات عمود پرواز بدون سرنشین در بستر تست پردازشگر در حلقه

احسان طاهری*

۱- دکتری مهندسی برق کنترل، مجتمع دانشگاهی مکانیک، دانشگاه صنعتی مالک اشتر، شاهین شهر، ایران
(دریافت مقاله: ۱۴۰۰/۰۶/۱۵ تاریخ پذیرش: ۱۴۰۰/۰۷/۳۰)

چکیده

طراحی مسیر یکی از گلوگاه‌های باز تحقیقاتی جهت ایجاد خودمختاری در ربات‌های بدون سرنشین می‌باشد. در فضاهای شهری و محیط‌های بسته با افزایش تعداد موانع و قیدهای پیرامونی، پیچیدگی محاسباتی مورد نیاز برای الگوریتم‌های طراحی مسیر کامل با مرتبه $O(n^2)$ افزایش می‌یابد. هدف این مقاله ارائه یک الگوریتم طراحی مسیر مبتنی بر نمونه برداری تصادفی برای یک ربات عمود پرواز بدون سرنشین و تست زمان واقعی آن در محیط xPC-Target می‌باشد. برای این منظور، روش درخت جستجو سریع تصادفی پیشنهاد می‌گردد. الگوریتم پیشنهادی کامل احتمالاتی بوده و همچنین امکان اعمال قیدهای غیرهولونومیک در گره‌های درخت جستجوی مسیر را دارا می‌باشد. این امر سبب می‌گردد که تنها مسیر با حجم محاسبات محدود طراحی گردد، بلکه امکان اعمال قیدهای دیفرانسیلی حرکتی و در نهایت طراحی یک مسیر عملیاتی با قابلیت رهگیری فراهم گردد. جهت اعتبار سنجی و اعتبار بخشی به الگوریتم طراحی مسیر پیشنهادی پیش از انجام تست‌های پر ریسک و پر هزینه میدانی، بستر تست پردازنده در حلقه در محیط نرم افزار xPC-Target متلب در نظر گرفته شده است. پردازنده هدف مد نظر در این تست، تکبرد صنعتی مدل SBC84710 شرکت Axiomtek می‌باشد. در این تست دو سناریو آزمون با پیچیدگی متفاوت تدوین شده است. در سناریو اول یک محیط متشکل از موانع با تعداد زیاد و در سناریو دوم چند گذرگاه باریک در فضای کاری در نظر گرفته شده است. نتایج نشان می‌دهند که الگوریتم درخت جستجو سریع تصادفی پیشنهادی جهت طراحی مسیر برای ربات عمود پرواز مد نظر با بهره‌گیری از ماهیت تصادفی خود قادر به طراحی مسیر سریع برای ربات می‌باشد. همچنین به دلیل درگیر نمودن مدل دینامیکی ربات در فرایند تولید گره‌های تصادفی و شاخه‌های درخت جستجو، در مسیر طراحی شده هر دو نوع قید سینماتیکی و دینامیکی لحاظ شده‌اند و لذا منجر به طراحی یک مسیر عملیاتی با قابلیت رهگیری بالا توسط ربات شده است.

واژه‌های کلیدی: طراحی مسیر، الگوریتم درخت جستجوی سریع تصادفی، سینودینامیک، ربات عمود پرواز، پردازشگر درحلقه، کامپیوتر صنعتی تکبرد.

Designing the rapidly exploring random tree path planning algorithm for the vertical take-off and landing vehicle on the processor-in-the-loop test platform

Ehsan Taheri*

Abstract

Path design is one of the open research bottlenecks to develop autonomy in unmanned robots. In urban spaces and closed environments, with increasing the number of obstacles and constraints, the computational complexity required for the completed path planning algorithms increases with order $O(n^2)$. The main purpose of this manuscript is to present a path planning algorithm based on the random sampling method for the vertical take-off and landing vehicle (VTLV) and its real-time test in the xPC-Target toolboxes. For this purpose, the rapidly exploring random tree (RRT) algorithm is presented. The proposed algorithm is completely probabilistic and also the non-holonomic constraints of the VTLV are integrated into the vertices of the search tree. This not only allows the path to be planned with a limited computational complexity but also allows the differential motion constraints are integrated into the planned path. In order to validate and evaluate the proposed path planning algorithm before performing risky and costly field tests, the processor-in-the-loop (PIL) test, in MATLAB xPC-Target software environment is considered. The target processor in the PIL test is the SBC84710 which is produced by Axiomtek. In the PIL test, two test scenarios with different complexity have been developed. In the first scenario, a configuration space consisting of a large number of obstacles and in the second test scenario, several narrow passages are considered. The results show that the proposed rapidly exploring random tree algorithm is able to plan a primary path for the robot by using its random nature in a rapid manner. Also, due to the involvement of the robot dynamic model in the process of generating random vertices and search tree edges, both types of kinematic and dynamic (kinodynamic) constraints have been considered in the planned path and therefore lead to the design of a practical path with high intercept ability through the robot.

Key words: Path Planning, Rapidly Exploring Random Tree (RRT), Kinodynamic, Vertical Take-Off and Landing Vehicle (VTLV), Processor-in-the-Loop (PIL), Single Board Computer (SBC).

مقدمه

مطابق با آخرین نقشه راه‌های منتشر شده در حوزه ربات‌های پرنده بدون سرنشین، ربات‌های عمود پرواز با بهره‌گیری از ویژگی‌های منحصر به فردی که به یومن پیشرفتهای روزافزون صورت پذیرفته در عرصه تکنولوژیهای جدید، بلاخص در فیلد تجهیزات نرم افزاری و سخت افزاری منجر به خودمختاری، نقش انکارناپذیری در پیاده‌سازی اهداف و ماموریت‌های مطرح شده در این حوزه را برعهده خواهند داشت [۱]. جهت ایجاد خودمختاری در اینگونه ربات‌های بدون سرنشین نیاز به همکاری حداقل چهار ماژول: طراحی مسیر، کنترل، هدایت، و ناوبری می‌باشد [۲]. طراحی مسیر در [۳] به مفهوم تبدیل نیازها و خواسته‌های سطح بالای کاربر که از آن تحت قالب ماموریت یاد می‌شود به نیازها و خواسته‌های سطح پایین که از آن تحت قالب جابجایی و حرکت یاد می‌گردد، تعریف می‌گردد. الگوریتم‌های طراحی مسیر از منظرهای مختلفی قابل طبقه‌بندی می‌باشند. به عنوان مثال در [۴، ۵]، الگوریتم‌های طراحی مسیر از منظر نحوه عملکرد به سه دسته: سراسری (مبتنی بر اطلاعات حاصل از نقشه)، محلی (مبتنی بر اطلاعات حاصل از سنسور) و ترکیبی (مبتنی بر اطلاعات حاصل از سنسور و نقشه) تقسیم‌بندی شده است. در [۶]، الگوریتم‌های طراحی مسیر از منظر گذر زمان به دو دسته کلاسیک و مدرن طبقه‌بندی شده است. گروهی از روش‌های طراحی مسیر مدرن، مبتنی بر نمونه‌برداری افزایشی می‌باشند، که در این مقاله بیشتر به آنها خواهیم پرداخت. الگوریتم‌های طراحی مسیر مدرن مبتنی بر نمونه‌برداری افزایشی به دو دسته اصلی گراف محور مانند: نقشه راه احتمالاتی، و درخت محور مانند: درخت جستجوی سریع تصادفی، درخت متراکم جستجوی سریع و درخت توسعه یافته در فضا تقسیم می‌گردند. الگوریتم نقشه راه احتمالاتی توسط کواراکی ارائه گردید [۷]. این روش متشکل از دوفاز بوده و برای مسائل چندپرس و جو متناسب است. فاز آموزش یا پیش‌پردازش بصورت غیربهنگام و فاز جستجو بصورت بهنگام انجام می‌شود. الگوریتم درخت جستجوی سریع تصادفی توسط لاول در سال ۱۹۹۸ ارائه گردید [۸]. این روش متشکل از یک فاز بوده و بصورت روی-خط پیاده‌سازی می‌شود. این روش مناسب برای مسائل تک پرس و جو است و بر خلاف نقشه راه احتمالاتی که گراف محور است، بر پایه رشد درخت در فضای پیکربندی بنا نهاده می‌شود. الگوریتم‌های طراحی مسیر از منظرهای مختلفی: الگوریتم‌های طراحی مسیر بهینه [۹، ۱۰]، الگوریتم‌های طراحی مسیر

کامل [۱۱]، الگوریتم طراحی مسیر تصادفی [۷]، الگوریتم طراحی مسیر سینودینامیکی [۱۲]، طراحی مسیر برای گروهی از ربات‌ها [۱۳]، طراحی مسیر با در نظر گرفتن پیچیدگی زمانی و محاسباتی [۱۴]، الگوریتم طراحی مسیر هیوریستیک [۱۵]، طراحی مسیر متغیر با زمان [۱۶]، طراحی مسیر با در نظر گرفتن عدم قطعیت‌های فضای پیکربندی [۱۷]، طراحی مسیر محلی [۱۸] (مبتنی بر اطلاعات سنسور)، و ... طبقه‌بندی می‌گردند. در این خصوص در عمده تحقیقات گذشته همچون روش درخت جستجوی سریع تصادفی بهینه چند بعدی [۱۹، ۲۰]، درخت جستجو سریع تصادفی زیربهینه [۲۱، ۲۲]، درخت جستجو سریع تصادفی حریم [۲۳-۲۵] و ... دینامیک ربات جهت مسئله طراحی مسیر نادیده گرفته شده و مسئله بصورت هولونومیک تعریف می‌گردد. این امر سبب شده علی‌رغم حل مسئله طراحی مسیر، مسیر طراحی شده برای ربات بدلیل عدم اعمال قیدهای حرکتی در رشد شاخه‌های درخت جستجو قابل رهگیری نباشد. در [۲۶] قیدهای دیفرانسیلی ربات در طراحی مسیر لحاظ شده منتها از تابع متریک اقلیدسی بمنظور سنجش فاصله گره‌های درخت استفاده شده که خود باعث به تله افتادن درخت جستجو در مینیم‌های محل و یا در موانع به فرم مقعر شده است. طراحی مسیر برای ربات‌های خودمختار بدون سرنشین بدلیل وجود همزمان قیدهای سینماتیکی، قیدهای دینامیکی، قیدهای محیطی و قیدهای مد نظر کاربر امری چالش برانگیز می‌باشد. از جمله قیدهای دینامیکی مطرح در این حوزه می‌توان به: درجات آزادی بالا نسبت به تعداد عملگر در اختیار، دامنه محدود گشتاورها و نیروهای تولیدی، دامنه محدود شتابها و سرعت‌های خطی و زاویه‌ای، کولپینگ بین درجات آزادی، مدل دینامیکی غیرخطی و پارامترهای متغیر با زمان، و از جمله قیدهای سینماتیکی و محیطی می‌توان به: فضای پیکربندی از پیش ناشناخته، موانع متغیر با زمان، گذرگاه‌های باریک و فضاهای تو در تو، اغتشاشات محیطی و نویزهای اندازه‌گیری، و از جمله قیدهای مد نظر کاربر می‌توان به: نقاط ممنوعه حرکتی اشاره داشت. هدف این مقاله کاهش پیچیدگی محاسباتی مسئله طراحی مسیر در حضور موانع محیطی و قیدهای حرکتی در ربات عمود پرواز بدون سرنشین به کمک الگوریتم درخت جستجوی سریع تصادفی و اعتبار سنجی و اعتبار بخشی بدان به کمک تست پردازشگر درحلقه می‌باشد. برای این منظور الگوریتم درخت جستجو سریع تصادفی مبتنی بر نمونه برداری تصادفی افزایشی پیشنهاد می‌گردد. موضوع مذکور

اولیه تحقیق پیشرو برای اولین بار در نوزدهمین کنفرانس بین المللی انجمن هوافضای ایران به چاپ رسیده است. نظر به فراهم شدن امکان چاپ مقالات منتخب از سوی نشریه علمی پژوهشی مهندسی هوانوردی، نسبت به ارتقاء سطح کیفی مقاله مد نظر اقدام لازم صورت پذیرفت و نتایج برای مجله مد نظر ارسال گردید.

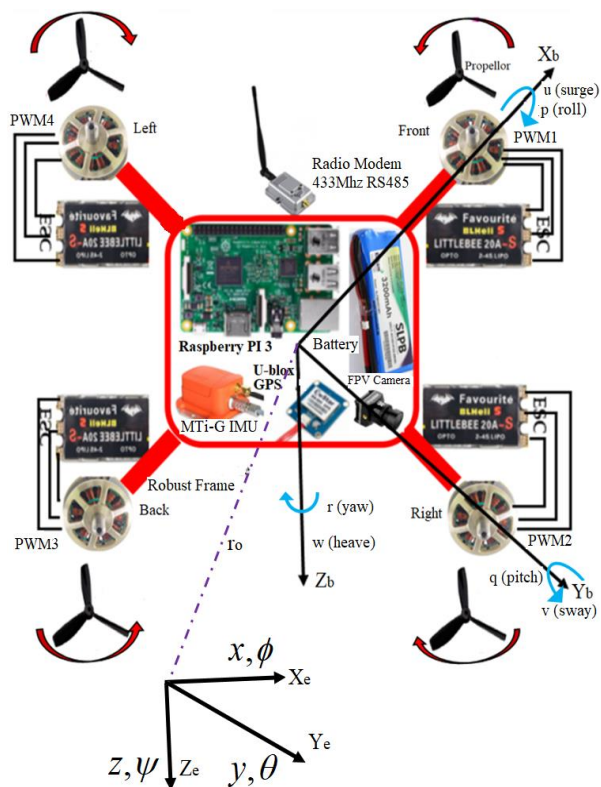
۲- معادلات حاکم بر حرکت ربات عمود پرواز

روش اوایلر-لاگرانژ یا روش اوایلر-نیوتن راهکارهای متداولی برای بیان معادلات حاکم بر حرکت ربات عمود پرواز مد نظر می‌باشند [۲۷-۳۰]. پیش از پرداختن به بیان معادلات حاکم و تحلیل آنها، دستگاه‌های مختصات بدنی و اینرسی مورد نیاز معرفی می‌گردند. پیکربندی و دستگاه‌های مختصات مورد استفاده در تحلیل پایداری حرکتی حلقه باز ربات عمود پرواز مد نظر در شکل ۱ آورده شده‌اند. برای توصیف حرکت انتقالی و دورانی ربات مد نظر در فضای سه بعدی نیاز به شش درجه آزادی می‌باشد. موقعیت و وضعیت ربات ($\eta \in R^6$) در دستگاه مختصات اینرسی:

$$\eta = [\eta_1, \eta_2] = (x, y, z, \phi) \quad (1)$$

و سرعت‌های خطی و دورانی ربات ($v \in R^6$) توسط بردار ذیل توصیف می‌گردند:

$$v = [v_1, v_2] = (u, v, w, p, q, r) \in R^6 \quad (2)$$



شکل ۱- پیکربندی ربات عمود پرواز و دستگاه‌های مختصات بدنی و اینرسی مورد استفاده

هنگامی چالش برانگیزتر می‌گردد که درجه پیچیدگی محاسباتی و زمانی الگوریتم‌های طراحی مسیر کامل با افزایش تعداد موانع از مرتبه $O(n^2)$ افزایش یابد. این امر توانایی پیاده سازی اینگونه الگوریتم‌ها، بصورت روی خط را محدود می‌نماید. همچنین نکته چالشی دیگر در هدف پیش‌رو امکان اعمال همزمان قیدهای سینماتیکی و قیدهای دینامیکی در فرایند طراحی مسیر می‌باشد. این امر سبب طراحی مسیر با قابلیت رهگیری توسط ربات می‌گردد. در نهایت الگوریتم پیشنهادی پیش از پیاده سازی بر روی ربات و انجام تست‌های پریسک و پری هزینه میدانی، بصورت تست روی خط پردازشگر در حلقه مورد ارزیابی قرار می‌گیرد. بعبارت دیگر انگیزه اصلی این تحقیق، پاسخگویی به چالش‌های پیشروی ذکر شده:

۱- کاهش پیچیدگی محاسباتی و زمانی الگوریتم طراحی مسیر و در نهایت فراهم کردن امکان پیاده سازی روی خط در محیط عملیاتی با تراکم بالایی از موانع خاص.

۲- گسترش شاخه‌ها و گره‌های درخت جستجوی سریع تصادفی مبتنی بر نمونه برداری افزایشی در محیط عملیاتی جهت طراحی سریع مسیر اولیه.

۳- امکان اعمال همزمان قیدهای دینامیکی ربات در کنار قیدهای سینماتیکی جهت طراحی مسیر با قابلیت رهگیری میدانی.

۴- افزایش سطح اعتبار کدهای الگوریتم طراحی مسیر طراحی شده، به کمک بستر تست پردازشگر در حلقه پیش از انجام تست‌های پریسک و پری هزینه میدانی.

می‌باشد. روند ارائه مطالب در این مقاله بدین صورت خواهد بود، که ابتدا در بخش دوم به بیان معادلات حاکم بر سینماتیک و دینامیک ربات خواهیم پرداخت. معادلات مذکور بر رشد شاخه‌ها و گره‌های درخت جستجو اعمال شده تا مسیر با قابلیت رهگیری طراحی شود. در بخش سوم مسئله طراحی مسیر و سپس الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی پیشنهادی معرفی می‌گردند. در بخش چهارم بستر تست پردازنده در حلقه xPC-Target، مادر برد صنعتی تکبرد SBC84710 ساخت شرکت Axiomtek و سناریوهای تست تعریف شده و اثربخشی الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی پیشنهادی در طرح‌ریزی مسیر حرکت ربات مد نظر مورد نقد و بررسی قرار خواهد گرفت. در بخش پنجم همانند تمامی مقالات به جمع‌بندی و نتیجه‌گیری موضوع خواهیم پرداخت. شایان ذکر است که نسخه

۲-۲- دینامیک حرکت

با توجه به معادلات حرکت نیوتن-اویلر در توصیف نیروها و ممان‌های اعمالی بر ربات و با انجام فرضیات ذیل:

۱- چارچوب ربات مد نظر متقارن و صلب است، ۲- مرکز جرم ربات بر مرکز دستگاه مختصات بدنی منطبق است، ۳- پروانه-های هر یک از چهار روتور صلب می‌باشند، ۴- نیروی تراست و درگ متناسب با مربع سرعت دوران پروانه‌ها می‌باشد، ۵- از دینامیک پروانه‌ها و روتورها صرف نظر می‌گردد. معادلات حرکت بصورت کلی ذیل بیان می‌گردد:

$$\begin{aligned} \text{Newton's law} &\Rightarrow m(v_2 \wedge v_1 + \dot{v}_1) = f_B \\ &= [f_x f_y f_z]^T \in R^3 \\ \text{Euler's law} &\Rightarrow I \cdot \dot{v}_2 + v_2 \wedge (I \cdot v_2) = \tau_B \\ &= [\tau_x \tau_y \tau_z]^T \in R^3 \end{aligned} \quad (۸)$$

و $m=2.1\text{kg}$ که

$$I = \begin{bmatrix} I_{xx}=0.00612\text{kgm}^2 & & \\ & I_{yy}=0.00612\text{kgm}^2 & \\ & & I_{zz}=0.0137\text{kgm}^2 \end{bmatrix}^T$$

باشد. معادله شماره هشت در فرم فضای حالت:

$$\begin{aligned} M_B \dot{v} + C_B(v)v = \Lambda &= [f_B \quad \tau_B]^T \\ M_B &= \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix}, C_B = \begin{bmatrix} 0_{3 \times 3} & -mS(v_1^B) \\ 0_{3 \times 3} & S(Iv_2^B) \end{bmatrix} \end{aligned} \quad (۹)$$

که M_B ماتریس اینرسی ثابت و قطری، C_B ماتریس کوریولیس و $S(\cdot)$ ماتریس نیمه پادمتقارن با تعریف ذیل می‌باشد.

$$S(k) = -S^T(k) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (۱۰)$$

جدول ۱- متغیرها و پارامترهای مورد استفاده در تدوین معادلات

حرکت ربات عمود پرواز	
پارامترها	
موقعیت و وضعیت ربات	$\eta = [\eta_1 \quad \eta_2]$
سرعت خطی و دورانی ربات	$v = [v_1 \quad v_2]$
ماتریس انتقال از دستگاه بدنی به اینرسی	$J: R^6 \rightarrow R^{6 \times 6}$
ماتریس نیروها و ممان‌ها	$\Lambda = [f_B \quad \tau_B]^T$
متغیرها	
$I_{xx} = I_{yy} = 0.00612\text{kgm}^2$	$m = 2.1\text{kg}$
$I_{zz} = 0.0137\text{kgm}^2$	$l = 0.2\text{m}$
$J_r = 1.23 \times 10^{-5}\text{kgm}^2$	$g = 9.81 \text{ms}^{-2}$
$b = 3.21 \times 10^{-5}\text{kgm}^2$	$d = 6.45 \times 10^{-7}\text{kgm}^2$

ربات عمود پرواز مد نظر همانطور که در شکل ۱ نشان داده شده است شامل دو جفت ملخ (۳،۱) و (۴،۲) است که در خلاف جهت یکدیگر می‌چرخند. دوران این روتورها تولید کننده مانورهای مختلف ربات می‌باشند. صرفاً با دوران این روتورها نیروی لیفت تولید می‌گردد، در حالی که برای سایر مانورها نیاز به تغییر سرعت روتورهای مربوط به آن مانور است. گشتاور حاصل از دوران روتورهای ۲ و ۴ در خلاف جهت هم و حول محور X است. لذا با تغییر سرعت این روتورها می‌توان زاویه غلتش را تغییر داد. بدلیل اینکه نیروی لیفت حاصل از روتورهای ۱ و ۳ هم‌جهت می‌باشند، گشتاور آنها حول محور Y در خلاف جهت هم دیگر است و با تغییر سرعت روتورهای مذکور می‌توان زاویه چرخش را تغییر داد.

۲-۱- سینماتیک حرکت

بطور کلی بمنظور بیان معادلات حاکم بر سینماتیک حرکت ربات با در نظر گرفتن ویژگی‌های هندسی حرکت ماتریس انتقال $J: R^6 \rightarrow R^{6 \times 6}$ بصورت ذیل تعریف می‌گردد:

$$\dot{\eta} = J(\eta)v \rightarrow \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (۳)$$

که ارتباط بین موقعیت و سرعت خطی ربات:

$$J_1(\eta_2) = C_z(\psi), C_y(\theta), C_x(\phi) \in SO(3) \quad (۴)$$

همچنین ماتریس انتقال بین وضعیت و سرعت دورانی ربات بصورت ذیل بیان می‌گردد:

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (۵)$$

با بیان تفصیلی رابطه شماره ۴ بصورت ذیل:

$$J_1(\eta_2) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} = \quad (۶)$$

$$\begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

و با توجه به معادلات ۳ و ۵-۶ معادلات حاکم بر سینماتیک حرکت ربات بصورت ذیل خواهد شد.

$$\begin{aligned} \dot{x} &= w[s\phi s\psi + c\phi c\psi s\theta] - v[c\phi s\psi - c\psi s\phi s\theta] + u[c\psi c\theta] \\ \dot{y} &= v[c\phi c\psi + s\phi s\psi s\theta] - w[c\psi s\phi - c\phi s\psi s\theta] + u[c\theta s\psi] \\ \dot{z} &= w[c\phi c\theta] - u[s\theta] + v[c\theta s\phi] \\ \dot{\phi} &= p + r[c\phi t\theta] + q[s\phi t\theta] \\ \dot{\theta} &= q[c\phi] - r[s\phi] \\ \dot{\psi} &= r \frac{c\phi}{c\theta} + q \frac{s\phi}{c\theta} \end{aligned} \quad (۷)$$

و سیگنال‌های کنترلی $U = [U_1, U_2, U_3, U_4]^T \in R^4$ اعمالی به ربات:

$$\begin{aligned} U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= lb(-\Omega_2^2 + \Omega_3^2) \\ U_3 &= lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 &= d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{aligned} \quad (19)$$

همچنین معادلات حرکت در دستگاه مختصات هیبرید (معادلات حرکت خطی در دستگاه اینرسی و معادلات حرکت دورانی در دستگاه بدنی) بصورت ذیل خواهد بود:

$$\begin{aligned} \ddot{x} &= (s\psi s\phi + c\psi s\theta c\phi) \frac{U_1}{m} \\ \ddot{y} &= (-c\psi s\phi + s\psi s\theta c\phi) \frac{U_1}{m} \\ \ddot{z} &= -g + (c\theta c\phi) \frac{U_1}{m} \\ \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} qp + \frac{U_4}{I_{zz}} \end{aligned} \quad (20)$$

معادلات حرکت شماره ۲۰ در صورت کوچک بودن حرکت-های دورانی قابل ساده سازی می‌باشد. عبارت دیگر حاصلضرب سرعت‌های زاویه‌ای قابل نظر می‌باشند.

۳- الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی

درخت جستجوی سریع تصادفی در زمره الگوریتم‌های ابتکاری است، که بدلیل سرعت، سادگی و حجم محاسبات محدود، مناسب برای اجرای بلادرنگ در فضاهایی با ابعاد بالا و ربات‌های با درجات آزادی بالا است [۱۲، ۳۱، ۳۲]. در این روش ابتدا یک مسیر اولیه بسرعت طراحی می‌شود، سپس در طول مسیر از درخت‌های جستجوی سریع تصادفی محلی برای عبور از موانع از قبل ناشناخته و همچنین اعمال قیدهای بهینه‌سازی استفاده می‌شود. ساختار اصلی الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی از پنج تابع اصلی ذیل تشکیل شده است: ۱- تابع نمونه‌برداری تصادفی، ۲- تابع هدایت شاخه درخت، ۳- تابع انتخاب گره والد یا نزدیک‌ترین همسایه، ۴- تابع بررسی عدم برخورد با مانع و ۵- تابع اندازه‌گیری تشکیل می‌شود. با توجه به پنج تابع معرفی شده، توصیف مفهومی نحوه عملکرد الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی در شکل ۲ و الگوریتم ۱ آورده شده است.

با توجه به مقادیر ارائه شده در جدول ۱ ماتریس‌های اینرسی و کوریولیس بصورت ذیل خواهد بود.

$$M_B = \begin{bmatrix} 2.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00612 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00612 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0137 \end{bmatrix} \quad (11)$$

$$C_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 2.1w & -2.1v \\ 0 & 0 & 0 & -2.1w & 0 & 2.1u \\ 0 & 0 & 0 & 2.1v & -2.1u & 0 \\ 0 & 0 & 0 & 0 & 0.0137r & -0.00612q \\ 0 & 0 & 0 & -0.0137r & 0 & 0.00612p \\ 0 & 0 & 0 & 0.00612q & -0.00612p & 0 \end{bmatrix} \quad (12)$$

و بردار نیروها و ممان‌های $\Lambda = [f_B \ \tau_B]^T$ رابطه ۹ بصورت:

$$\begin{aligned} \Lambda &= G_B(\eta) + O_B(v)\Omega + E_B\Omega^2 \\ \Omega^{Overall} &= -\Omega_1^{front} + \Omega_2^{right} - \Omega_3^{back} + \Omega_4^{left} \\ \Omega &= [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T \end{aligned} \quad (13)$$

گسترش یافته‌اند، که $\Omega (rad\ s^{-1})$ سرعت چرخش پروانه‌ها و G_B نیروی ناشی از جاذبه بصورت:

$$G_B(\eta) = [mgs\theta, -mgc\theta s\phi, -mgc\theta s\phi, 0, 0, 0]^T \quad (14)$$

و O_B گشتاور ژيروسکپی بوده و بصورت ذیل بیان می‌گردد:

$$O_B(v) = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

$J_{TP} [NmS^2] \rightarrow$ total inertia of rotors

E_B ماتریس نیرو و ممان، $l(m)$ فاصله مرکز جرم از مرکز روتورها، $d(kg\ m^2)$ ضریب درگ و $b(kg\ m^2)$ ضریب تراست می‌باشد.

$$E_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \quad (16)$$

با توجه به رابطه ۹ و ۱۳ خواهیم داشت:

$$\dot{v} = M_B^{-1}(-C_B(v)v + G_B(\xi) + O_B(v)\Omega + E_B\Omega^2) \quad (17)$$

معادلات حرکت در دستگاه بدنی:

$$\begin{aligned} \dot{u} &= (vr - wq) + gs\theta \\ \dot{v} &= (wp - ur) - gc\theta s\phi \\ \dot{w} &= (uq - vp) - gc\theta s\phi + \frac{U_1}{m} \\ \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \end{aligned} \quad (18)$$

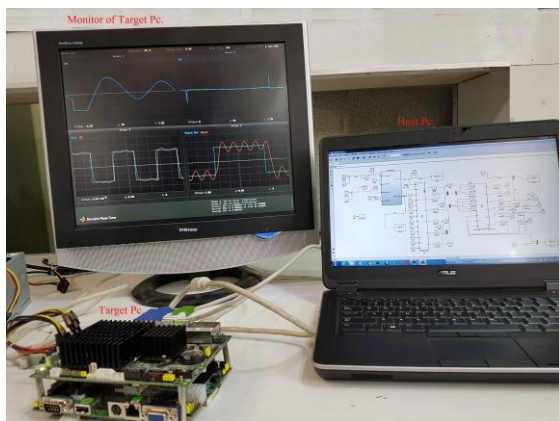
به کمک تکنیک نمونه برداری تدریجی افزایشی بستر تست پردازنده در حلقه معرفی می‌گردد. در بستر تست پردازنده در حلقه، از پردازنده هدف SBC84710 که در ربات واقعی نیز استفاده می‌گردد، استفاده شده است. لذا امکان تست زمان واقعی الگوریتم طراحی مسیر طراحی شده فراهم می‌گردد. در بستر تست مذکور از دو سناریوی تست با پیچیدگی متفاوت استفاده شده است.

۴-۱-بستر تست xPC-Target

xPC-Target یک محیط شبیه‌سازی مبتنی بر معماری x86 است، که امکان تست مدل‌های Stateflow و Simulink بصورت زمان واقعی و بر روی سخت افزار واقعی مورد استفاده در ربات را فراهم می‌سازد. به عبارت دیگر xPC-Target یک هسته زمان واقعی است که بر روی کامپیوتر هدف اجرا می‌شود و منتظر می‌ماند که یک فرمان بصورت ارتباط سریال یا توسط ارتباط شبکه از کامپیوتر میزبان دریافت گردد. برای فراهم کردن بستر تست مذکور نیاز به یک پردازنده میزبان، یک پردازنده هدف، بلوک‌های I/O، کامپایلر Visual C/C++ و نرم‌افزار متلب می‌باشد.



شکل ۳- پردازنده هدف SBC84710 مورد استفاده در تست پردازنده در حلقه در محیط xPC-Target نرم افزار متلب



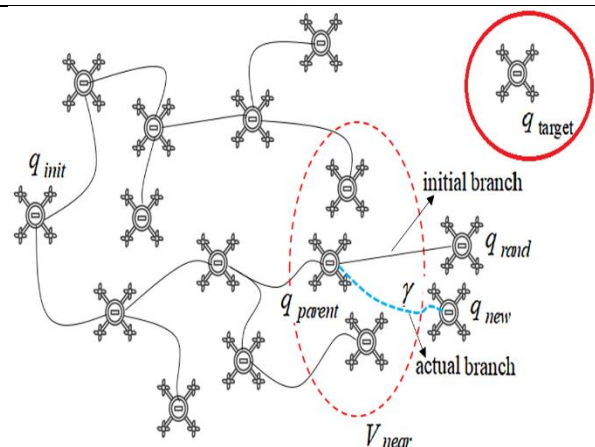
شکل ۴- چیدمانی پردازنده و مانیتور هدف، پردازنده میزبان و ارتباط توسط پورت TCP/IP در تست پردازنده در حلقه

Algorithm 1: The pseudo-code of the standard RRT

```

1:  $(x, x_{free}) \leftarrow$  Configuration/Free space
2:  $V \leftarrow \{q_{init}, q_{goal}\}$ 
3:  $E \leftarrow \{\emptyset\}$ 
4:  $R(V, E) \leftarrow$  Search Tree
5:  $(N, \beta) \leftarrow$  Max. number of iteration and threshold of goal vertex.
6:  $R_c \leftarrow L^2 norm(q_{goal}, q_{new})$ 
7: while  $i < N$  and  $R_c \geq \beta$  do
8:    $q_{rand} \leftarrow$  The random sampling is generated through the function number 1 ( $x_{free}$ ).
9:    $V_{near} \leftarrow$  The best candidate parent vertices are determined through the function number 3.
10:   $q_{parent} \leftarrow$  The parent vertex is determined through the function number 2.
11:   $[q_{new}, \mathfrak{S}] \leftarrow$  The edge and new vertex are generated through the function number 4.
12:  if  $(q_{new}, \mathfrak{S})$  located in  $x_{free}$  then  $\triangleright$  by Fun.#5
13:     $V \leftarrow \{V \cup q_{new}\}$ 
14:     $E \leftarrow \{E \cup \mathfrak{S}\}$ 
15:     $R \leftarrow$  Update(R)
16:  endif
17:   $R_c \leftarrow$  Update ( $R_c$ )
18: endwhile

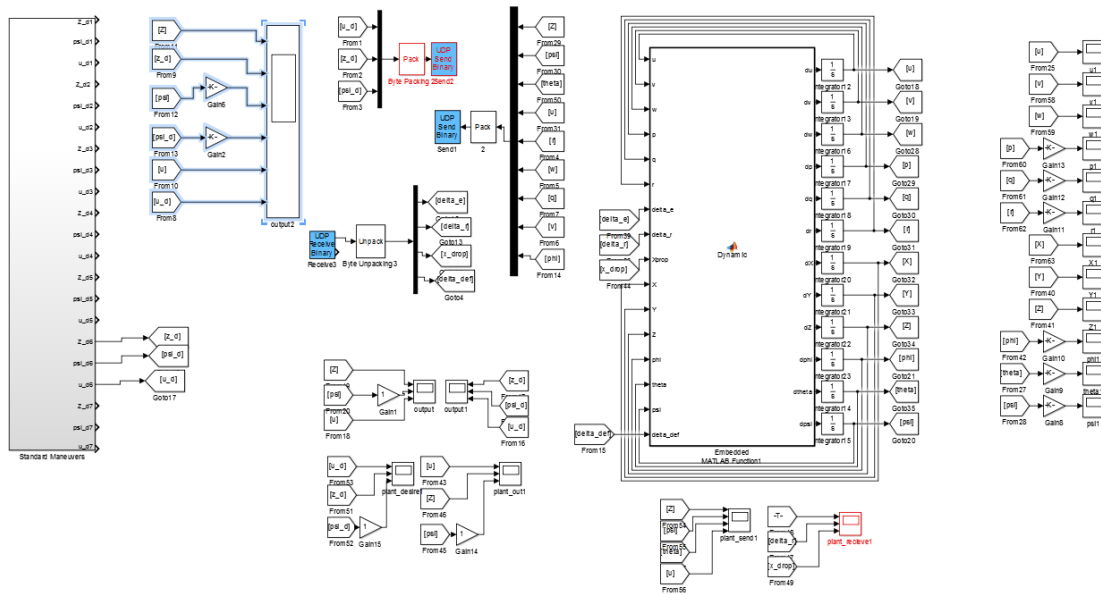
```



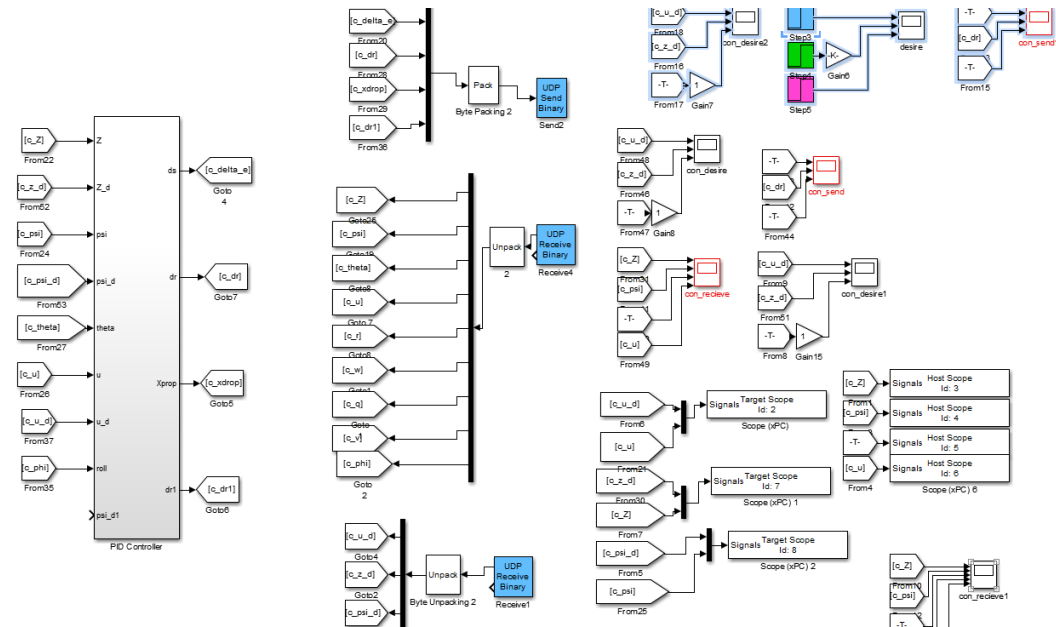
شکل ۲- معرفی مفهومی گره تصادفی، گره جدید، گره والد، مجموعه گره‌های کاندید، یال واقعی و یال اولیه در درخت طراحی مسیر پیشنهادی

۴- نتایج آزمون پردازنده در حلقه

جهت ارزیابی عملکرد الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی پیشنهادی، در اعمال همزمان قیده‌های سینماتیکی و دینامیکی (سینودینامیک) ربات در فرایند طراحی مسیر و همچنین بمنظور بررسی اثربخشی الگوریتم پیشنهادی در کاهش پیچیدگی محاسباتی و زمانی



شکل ۵- پیاده سازی مدل دینامیکی ربات به همراه بلوک‌های ارتباط با شبکه UDP بر روی پردازنده میزبان



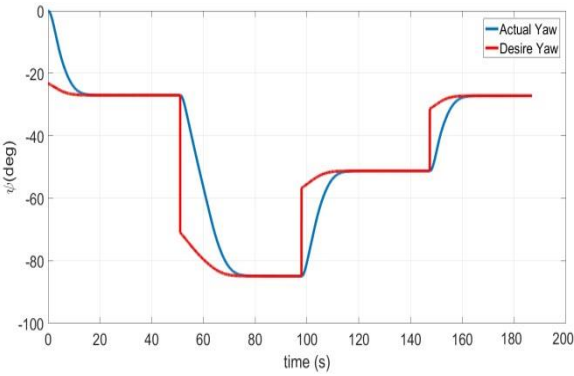
شکل ۶- پیاده سازی واحد طراحی مسیر پیشنهادی همراه با بلوک‌های ارتباط با شبکه UDP بر روی پردازنده هدف

سرعت ۱۰ مگابیت بر ثانیه فراهم است، شکل ۴. مدل سیمولینک الگوریتم طراحی مسیر پیشنهادی بر روی پردازنده هدف و مدل دینامیکی ربات بر روی پردازنده میزبان پیاده سازی می‌گردد، شکل ۵-۶. در ادامه به معرفی بیشتر سناریوهای تست مد نظر و تحلیل نتایج آن خواهیم پرداخت.

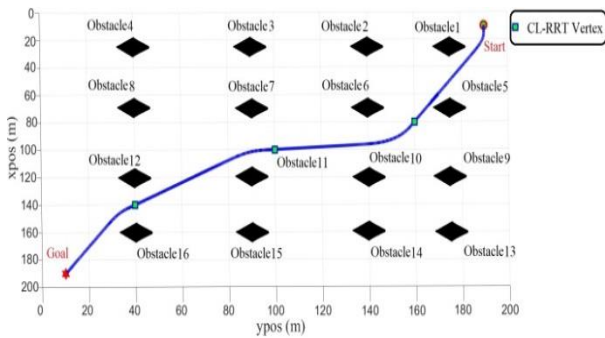
۴-۲- سناریوهای تست

به منظور ارزیابی عملکرد الگوریتم درخت جستجوی سریع تصادفی در طراحی مسیر برای ربات عمود پرواز مد نظر دو سناریوی تست با پیچیدگی متمایز تعریف گردید. سناریوهای مذکور توسط آزمون پردازنده در محیط xPC-

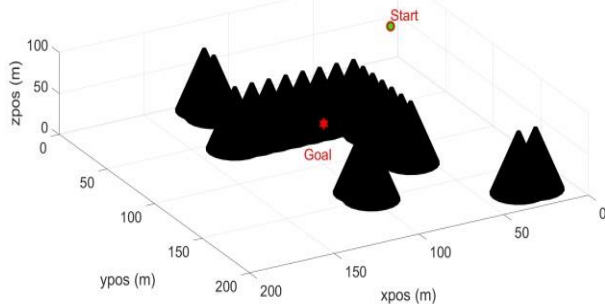
پردازنده میزبان در اینجا یک لپ‌تاپ با ۱۶ گیگابایت RAM و پردازنده Intel Core i7-9750 با ۸ مگابایت حافظه Cache و فرکانس ۴.۵ GHz می‌باشد. پردازنده هدف یک تکبرد صنعتی مدل 84710 ساخت شرکت Axiomtek می‌باشد، شکل ۳. پردازنده صنعتی تکبرد مذکور مجهز به پردازنده VIA V4 Eden/C3/C7 با کارت گرافیکی اینترفیس تصویر-صدا، فرم فکتور ۳.۵ اینچ، ۴ درگاه ارتباط سریالی (3 x RS-232, 1 x RS-232/422/485)، ولتاژ مصرفی ATX/AT و یا ۱۲+ ولتی مستقیم، دو گیگابایت RAM، درگاه‌های miniPCI، Compact Flash، کانکتور باس PC104 و هیستینگ می‌باشد. ارتباط بین پردازنده هدف و میزبان توسط پورت TCP/IP



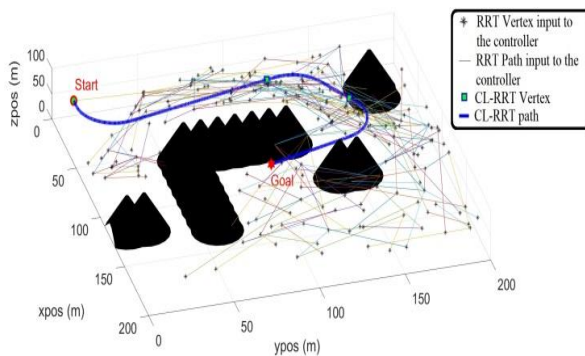
شکل ۹- مسیر مطلوب با خط قرمز رنگ و مسیر واقعی ربات با خط آبی رنگ نمایش داده می‌شود.



شکل ۱۰- نما از بالای فضای پیکربندی تست شماره یک، به همراه مسیر طراحی شده توسط الگوریتم طراحی مسیر پیشنهادی و نقطه مسیره‌ای که با مربعات سبز رنگ نمایش داده شده‌اند.



شکل ۱۱- نمای سه بعدی فضای پیکربندی در سناریو تست شماره دو.



شکل ۱۲- گسترش درخت جستجو سریع تصادفی در فضای پیکربندی تست شماره دو. یال‌های درخت با خطوط نازک، گره‌های درخت با نقطه‌های سیاه رنگ و مسیر طراحی شده با خط آبی ضخیم و موانع با مخروط‌های مشکی رنگ نمایش داده شده است.

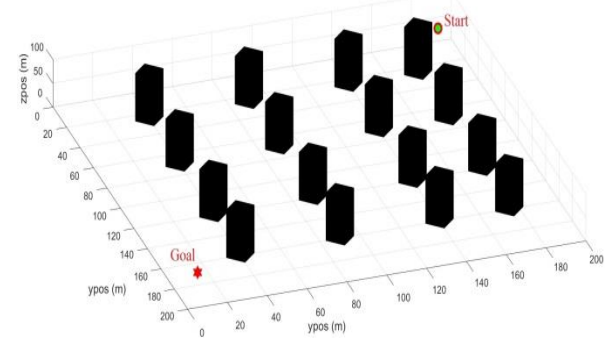
Target نرم افزار متلب و بر روی پردازشگر SBC84710

مورد استفاده در ربات واقعی پیاده سازی می‌گردد. سناریوی شماره یک: در این تست فضای کاری متشکل از تعداد زیادی مانع ثابت است، شکل ۷. در این تست، هریک از موانع به شکل مکعب مستطیل طراحی شده‌اند. وظیفه الگوریتم درخت جستجوی سریع تصادفی پیشنهادی طراحی مسیر سریع برای ربات مد نظر در فضای کاری معرفی شده، با در نظر گرفتن قیدهای سینودینامیکی می‌باشد.

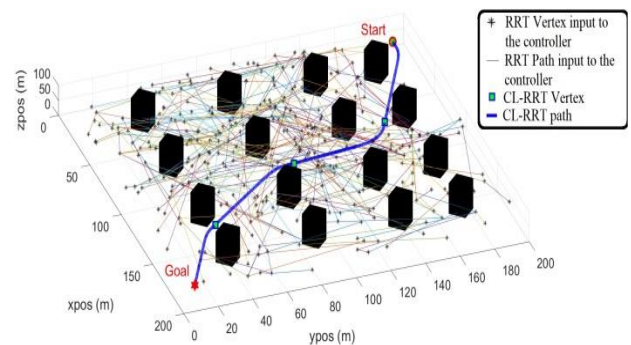
سناریوی شماره دو: در این تست فضای کاری متشکل از تعدادی گذرگاه باریک به همراه تعدادی موانع ثابت می‌باشد، شکل ۱۱. هر یک از موانع در این سناریو بگونه‌ای طراحی شده‌اند که بلعکس سناریو شماره یک، قطر موانع با افزایش طول موانع تغییر می‌نماید و این مسئله طراحی مسیر را با مشکل مواجه می‌سازد.

۳-۴- نتایج تست

عملکرد الگوریتم طراحی مسیر پیشنهادی در سناریو تست شماره یک در شکل‌های ۷ الی ۱۰ و در سناریو تست شماره



شکل ۷- نمای سه بعدی فضای پیکربندی در سناریو تست شماره یک.



شکل ۸- گسترش درخت جستجو سریع تصادفی در فضای پیکربندی تست شماره یک. یال‌های درخت با خطوط نازک، گره‌های درخت با نقطه‌های سیاه رنگ و مسیر طراحی شده با خط آبی ضخیم و موانع با مکعب‌های مشکی نمایش داده شده است.

جدول ۲- عملکرد الگوریتم درخت جستجوی سریع تصادفی در هر

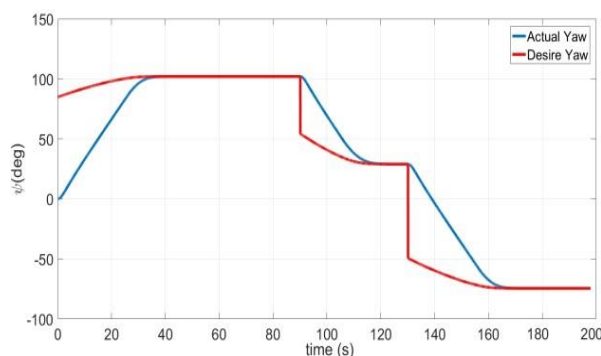
یک از دو سناریوی تست تدوین شده

زمان اجرا	تعداد نقطه مسیر با قید	تعداد گره معتبر درخت	تعداد گره درخت	تعداد کل گره درخت
۱.۰۳	۴	۳۲۷۳	۴۲۳۱	فضای پیکربندی شماره یک
۰.۶	۳	۲۹۵۷	۳۵۸۹	فضای پیکربندی شماره دو

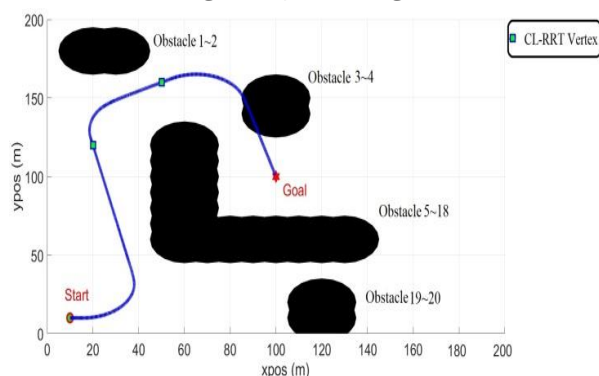
گسترش درخت جستجوی سریع تصادفی پیشنهادی به همراه گره‌ها و یال‌های درخت و مسیر نهایی طراحی شده در سناریوی تست شماره دو آورده شده است. در شکل ۱۲ گره-های درخت جستجو با دایره‌های مشکی توپر، یال‌های درخت جستجو که در فضای پیکربندی گسترش یافته‌اند با خطوط نازک، نقطه مسیره‌های طراحی شده با مربعات سبز رنگ، مسیر نهایی طراحی شده با خط ضخیم آبی رنگ و موانع با مخروط-های سیاه رنگ که تشکیل گذرگاه‌های باریک را داده‌اند نمایش داده شده است. در شکل ۱۳ مسیر مطلوب و مسیر واقعی در رسیدن به هریک از نقطه مسیره‌ها و در نهایت رسیدن به موقعیت مد نظر نشان داده شده است. در شکل ۱۴ نما از بالای فضای پیکربندی به همراه مسیر طراحی شده و نما از بالای فضای پیکربندی در نظر گرفته شده نشان داده شده است. در شکل ۱۴ دیده می‌شود که مسیر طراحی شده در بخشی از مسیر خود $(80 < x < 100, 100 < y < 150)$ با موانع برخورد داشته است. در صورتی که اینطور نیست و این خطای بصری می‌باشد و مسیر با دیواره گذرگاه باریک برخورد نداشته است. به عبارت دیگر به دلیل شکل خاص موانع و کاهش قطر موانع با افزایش ارتفاع آنها، مسیر همان طور که در شکل ۱۲ دیده می‌شود با موانع برخورد ندارد. عملکرد الگوریتم طراحی مسیر پیشنهادی در هریک از دو سناریو تست در جدول ۲ خلاصه می‌گردد. با توجه به اطلاعات ارائه شده در جدول ۲ و نتایج حاصل شده از شکل‌های ۷ الی ۱۴ مشاهده می‌گردد که الگوریتم درخت جستجوی سریع تصادفی در عبور از گذرگاه-های باریک عملکرد بهتری نسبت به فضاهای پیکربندی نامرتب با تعداد موانع زیاد داشته است.

۵- نتیجه‌گیری

در این مقاله الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی به منظور طراحی مسیر قابل اجرا توسط ربات عمود پرواز در یک فضای کاری سه بعدی ارائه گردید. الگوریتم مذکور در کلاس الگوریتم‌های تصادفی مبتنی بر نمونه‌برداری



شکل ۱۳- مسیر مطلوب با خط قرمز رنگ و مسیر واقعی ربات با خط آبی رنگ نمایش داده می‌شود.



شکل ۱۴- نما از بالای فضای پیکربندی تست شماره دو، به همراه مسیر طراحی شده توسط الگوریتم طراحی مسیر پیشنهادی و نقطه مسیره‌های که با مربعات سبز رنگ نمایش داده شده‌اند.

دو در شکل‌های ۱۱ الی ۱۴ نشان داده شده است. در شکل ۷ فضای پیکربندی تست شماره یک نمایش داده شده است. در شکل ۸ گسترش درخت جستجوی سریع تصادفی پیشنهادی به همراه گره‌ها و یال‌های درخت و مسیر نهایی طراحی شده آورده شده است. در شکل ۹ مسیر مطلوب و مسیر رهگیری شده توسط ربات مشخص شده است. در شکل ۹ مشاهده می‌گردد که بین مسیر واقعی و مسیر مطلوب عدم انطباق‌های مشاهده می‌گردد، این امر به دلیل لحاظ شدن قیدهای سینودینامیکی ربات در فرایند طراحی مسیر می‌باشد. به عبارت دیگر در مسیر قرمز رنگ صرفاً قیدهای سینماتیکی اعمال شده و هدف صرفاً عدم برخورد با موانع می‌باشد، در حالی که در مسیر آبی رنگ علاوه بر قیدهای سینماتیکی، قیدهای دینامیکی نیز لحاظ شده و محدودیت حرکتی ربات منجر به عدم انطباق‌ها شده است. نکته دیگری که در شکل ۹ و ۱۰ مشهود است، نقطه مسیره‌های است که الگوریتم جهت رسیدن به موقعیت نهایی از آنها استفاده نموده است. این نقطه مسیره‌ها با مربعات سبز رنگ در شکل ۱۰ نمایش داده شده‌اند. در شکل ۹ مشاهده می‌گردد که مسیر واقعی و مسیر مطلوب ربات از هر نقطه مسیر به نقطه مسیر بعدی عدم انطباق داشته است، که دلیل این امر بیان گردید. در شکل ۱۱

International Journal of Automation and Computing, Vol. 17, No. 3, (2020), 321-352

Ab Wahab, M.N., Nefti-Meziani, S. and Atyabi, A., "A comparative review on mobile robot path planning: Classical or meta-heuristic methods?", *Annual Reviews in Control*, Vol., No (۲۰۲۰).

Yuan, C., Liu, G., Zhang, W. and Pan, X., "An efficient rrt cache method in dynamic environments for path planning", *Robotics and Autonomous Systems*, Vol ۱۳۱, No., (2020), 103595.

Low, E.S., Ong, P. and Cheah, K.C., "Solving the optimal path planning of a mobile robot using improved q-learning", *Robotics and Autonomous Systems*, Vol. 115, No., (2019), 143-161

Optimal path ", Noreen, I., Khan, A. and Habib, Z planning using rrt* based approaches: A survey and future directions", *Int. J. Adv. Comput. Sci. Appl*, Vol. 7, No. 11, (2016), 97-107

Le, A.V., Arunmozhi, M., Veerajagadheswar, P., Ku, P.-C., Minh, T.H.Q., Sivanantham, V. and Complete path planning for a ", Mohan, R.E tetris-inspired self-reconfigurable robot by the genetic algorithm of the traveling salesman problem", *Electronics*, Vol. 7, No. 12, (2018), 344.

Taheri, E., Ferdowsi, M.H. and Danesh, M., "Closed-loop randomized kinodynamic path planning for an autonomous underwater vehicle", *Applied Ocean Research*, Vol. 83, No., (2019), 48-64

Liu, X.-h., Zhang, D., Zhang, J., Zhang, T. and Zhu, H., "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm", *Cluster Computing*, Vol., No., (2021), 1-15

Chen, J., Du, C., Zhang, Y., Han, P. and Wei, W., "A clustering-based coverage path planning method for autonomous heterogeneous uavs", *IEEE Transactions on Intelligent Transportation Systems*, Vol., No (۲۰۲۱).

Chi, W., Ding, Z., Wang, J., Chen, G. and Sun, L., "A generalized voronoi diagram based efficient heuristic path planning method for rrt in mobile robots", *IEEE Transactions on Industrial Electronics*, Vol., No., (2021).

Fazlollahtabar, H. and Hassanli, S., "Hybrid cost and time path planning for multiple autonomous guided vehicles", *Applied Intelligence*, Vol. 48, No. 2, (2018), 482-498

Cuevas, L., Ramirez, M., Shames, I. and Manzić, risk and uncertainty of C., "Path planning under the environment", in 2021 American Control Conference (ACC), IEEE. Vol., No. Issue, (2021), 4231-4236

Folker, H., Ritchie, M., Charlish, A. and Griffiths, H., "Sensor path planning using reinforcement learning", in Information Fusion, Elsevier. Vol., No. Issue (۲۰۲۰).

افزایشی می‌باشد که از پنج زیرتابع اصلی تشکیل می‌گردد. در طراحی هریک از شاخه‌ها و گره‌های درخت جستجو علاوه بر قیدهای محیطی (سینماتیکی)، قیدهای حرکتی (دینامیکی) نیز لحاظ شده است. این امر سبب می‌گردد مسیر نهایی طراحی شده منطبق بر دینامیک ربات بوده و قابل رهگیری باشد. بمنظور اعتبار سنجی و اعتبار بخشی به کدهای الگوریتم طراحی مسیر پیشنهادی فراتر از سطح شبیه سازی و جهت کاستن از هزینه‌های تست‌های میدانی، بستر تست پردازنده در حلقه پیشنهاد گردید. در این تست از پردازنده واقعی (SBC84710) مورد استفاده در ربات بعنوان پردازنده هدف استفاده گردید. نتایج دو سناریوی تست متفاوت نشان می‌دهد که الگوریتم طراحی مسیر پیشنهادی با بهره‌گیری از ماهیت تصادفی و به‌کمک اعمال همزمان قیدهای سینماتیکی و دینامیکی ربات در فرایند طراحی، قادر به طراحی مسیر سریع برای ربات مد نظر می‌باشد. الگوریتم پیشنهادی نه تنها از حجم محاسبات به دلیل ماهیت تصادفی کاسته است و امکان پیاده سازی روی خط را فراهم آورده است بلکه برخلاف روش‌های متداول که تنها به سینماتیک مسئله توجه می‌نماید، مسیری طراحی می‌کند که قابل رهگیری توسط ربات است.

منابع و مراجع

Cress, J.J., Hutt, M., Sloan, J., Bauer, M., Feller, M. and Goplen, S., "Us geological survey unmanned aircraft systems (uas) roadmap 2014, US Department of the Interior, US Geological Survey, (۲۰۱۵).

Zhao, Y., Zheng, Z. and Liu, Y., "Survey on computational-intelligence-based uav path planning", *Knowledge-Based Systems*, Vol. 158, No., (2018), 54-64

Kavraki, L.E., Svestka, P., Latombe, J.-C. and Overmars, M.H., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996), 566-580

Aggarwal, S. and Kumar, N., "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges", *Computer Communications*, Vol. 149, No (۲۰۲۰), ۲۶۹-۲۷۰

Patle, B., Pandey, A., Parhi, D. and Jagadeesh, A., "A review: On path planning strategies for navigation of mobile robot", *Defence Technology*, Vol. 15, No. 4, (2019), 582-606

Panda, M., Das, B., Subudhi, B. and Pati, B.B., "A comprehensive review of path planning algorithms for autonomous underwater vehicles",

- Kuffner, J.J.a.L., Steven M, "Rrt-connect: An planning", efficient approach to single-query path *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 2, No., (2000), 995--1001
- Taheri, E., Ferdowsi, M.H. and Danesh, M., "Fuzzy greedy rrt path planning algorithm in a complex configuration space", *International Journal of Control, Automation and Systems*, .Vol. 16, No. 6, (2018), 3026-3035
- Tang, Z., Chen, B., Lan, R. and Li, S., "Vector field guided rrt* based on motion primitives for quadrotor kinodynamic planning", *Journal of Intelligent & Robotic Systems*, Vol. 100, No. 3, .9(2020), 1325-133
- Silveira, J., Givigi, S.N., Freire, E.O., Molina, L. and Carvalho, E., "Aggressive motion planning for a quadrotor system with slung load based on rrt", in 2020 IEEE International Systems Conference .(SysCon), IEEE. Vol., No. Issue, 1-7
- P., Li, R., Shi, Y. and He, L., "Indoor ,Zhang navigation for quadrotor using rgb-d camera", in Proceedings of 2018 Chinese Intelligent Systems Conference, Springer. Vol., No. Issue, (2019), 497-506
- Gao, F., Lin, Y. and Shen, S., "Gradient-based trajectory generation for quadrotor online safe flight in complex environments", in 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE. Vol., No. Issue, .(2017), 3681-3688
- Liu, B., Feng, W., Li, T., Hu, C. and Zhang, J., "A variable-step rrt* path planning algorithm for quadrotors in below-canopy", *IEEE Access*, Vol. .8, No., (2020), 62980-62989
- Li, X., Qiu, L., Aziz, S., Pan, J., Yuan, J. and Zhang, B., "Control method of uav based on rrt* environment", in for target tracking in cluttered 2017 7th International Conference on Power Electronics Systems and Applications-Smart Mobility, Power Transfer & Security (PESA), .IEEE. Vol., No. Issue, (2017), 1-4
- Bouزيد, Y., Bestaoui, Y. and Siguerdidjane, H., "Quadrotor-uav optimal coverage path planning in cluttered environment with a limited onboard energy", in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems .(IROS), IEEE. Vol., No. Issue, (2017), 979-984
- Connell, D. and La, H.M., "Dynamic path planning and replanning for mobile robots using rrt", in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE. Vol., No. Issue, (2017), 1429-1434
- Sabatino, F., "Quadrotor control: Modeling, nonlinearcontrol design, and simulation", *KTH, School of Electrical Engineering (EES)*, .(۲۰۱۵), *Automatic Control.*, Vol., No
- Bouabdallah, S., Noth, A. and Siegwart, R., "Pid vs lq control techniques applied to an indoor micro quadrotor", in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). Vol. 3, No. Issue, (2004), 2451-2456 vol.2453
- Bresciani, T., "Modelling, identification and control of a quadrotor helicopter", *MSc Theses*, .(۲۰۰۸), Vol., No. 0280-5316
- F. and Subbarao, K., "Dynamic ,Das, A., Lewis inversion with zero-dynamics stabilisation for quadrotor control", *IET Control Theory & Applications*, Vol. 3, No. 3, (2009), 303-314

