

طراحی مسیر روی خط برای ربات خودمختار زیرسطحی در یک بستر تقریباً ناشناخته به روش درخت جستجو سریع تصادفی محلی

احسان طاهری^۱، محمدحسین فردوسی^۲، محمد دانش^۳

ferdowsi@mut.ac.ir

- ۱- دانشجوی دکتری برق کنترل، دانشکده مهندسی برق و الکترونیک، دانشگاه صنعتی مالک اشتر
۲- دانشیار دانشکده مهندسی برق و الکترونیک، دانشگاه صنعتی مالک اشتر
۳- دانشیار دانشکده مهندسی مکانیک، دانشگاه صنعتی اصفهان

چکیده

تقریباً ناشناخته بودن محیط بیکران دریاها و نبود نقشه دقیق قبلی از فضای کاری یکی از چالش‌های فعلی ربات‌های خودمختار زیرسطحی در انجام ماموریت‌های محوله می‌باشد. هدف این پژوهش، طراحی مسیر بصورت روی خط برای یک ربات خودمختار زیرسطحی از موقعیت و سرعت ابتدایی به موقعیت و سرعت هدف در یک فضای کاری تقریباً ناشناخته با حضور موانع از پیش نامعلوم می‌باشد. برای این منظور، الگوریتم طراحی مسیر درخت جستجو سریع تصادفی محلی برای ربات معرفی می‌گردد. الگوریتم مذکور شامل سه جزء به هم پیوسته: ماژول طراحی مسیر، ماژول طراحی مسیر محلی بلادرنگ و ماژول تشخیص مانع می‌باشد. هر یک از گره‌های تولیدی توسط ماژول طراحی مسیر و شاخه‌های مربوطه از منظر قیدهای سینودینامیکی ربات توسط کنترل‌کننده سطح پایین و مدل دینامیکی غیرخطی بررسی می‌شوند. چنانچه گره و شاخه بطور همزمان قیدهای مذکور را در بازه زمانی مدنظر برآورده نمایند، به همراه سیگنال‌های کنترلی توسط الگوریتم طراحی مسیر برای طراحی مسیر نهایی ذخیره می‌گردند. چنانچه ماژول تشخیص مانع ربات، مانعی از پیش نامعلوم را بر روی مسیر طراحی شده تشخیص دهد، ماژول طراحی مسیر محلی بلادرنگ به منظور اجتناب از برخورد با آن، درخت جستجو سریع تصادفی محلی را از موقعیت و وضعیت فعلی ربات به نزدیک‌ترین گره درخت طراحی شده توسعه می‌دهد. الگوریتم پیشنهادی به منظور ارزیابی بروی یک کامپیوتر صنعتی تک بورد پیاده‌سازی شده و بصورت زمان حقیقی در محیط xPC-Target متلب مورد آزمون پردازنده در حلقه قرار گرفت. نتایج آزمون نشان می‌دهد که ربات به کمک الگوریتم طراحی مسیر پیشنهادی نه تنها قادر به اجتناب از برخورد با موانع از پیش نامعلوم می‌باشد، بلکه به دلیل ماهیت تصادفی آن، از سرعت مطلوبی در طراحی مسیر در محیط‌های متشکل از موانع بسیار، بصورت روی خط برخوردار است.

واژگان کلیدی: ربات خودمختار زیرسطحی، طراحی مسیر، بستر تقریباً ناشناخته، درخت جستجو سریع تصادفی محلی، آزمون پردازنده در حلقه.

تاریخ دریافت مقاله : ۹۷/۱۱/۲۴
تاریخ پذیرش مقاله : ۹۸/۰۴/۰۶

۱- مقدمه

در دو دهه اخیر ربات‌های خودمختار زیرسطحی بعنوان عامل کاهش‌دهنده ریسک و چند برابر کننده توان در حوزه‌های مختلف نظامی، تجاری و تحقیقاتی در انجام طیف گسترده‌ای از ماموریت‌ها از جمله: نظارت-تجسس-جمع آوری اطلاعات^۱، مقابله با مین^۲، حمل بار^۳، بازرسی-شناسایی^۴، مقابله با زیردریایی^۵ و ... مورد توجه طیف فزاینده‌ای از تحقیقات قرار گرفته‌اند. این امر در کشور عزیز ما با داشتن بیش از 2500 کیلومتر مرز آبی و وجود حجم بالایی از ذخایر هیدروکربوری در مناطق فراساحل از اهمیت خاصی برخوردار می‌باشد [۱]. اکثر این گونه از ربات‌ها از شناور مادر به آب‌اندازی و آب‌گیری می‌شوند و به منظور اخذ تصمیم نیازمند حمایت اپراتور انسانی می‌باشند. به عبارت دیگر، بزرگترین چالش فعلی پیش روی این حوزه، مقوله سطح تحقق فناوری^۶ خودمختاری در این گونه ربات‌ها است. به منظور افزایش سطح خودمختاری در این ربات‌ها جهت دستیابی به ماندگاری بیشتر و همچنین کاهش هزینه‌های مالی نظارت انسانی، ایجاد قابلیت طراحی مسیر در آنها اجتناب‌ناپذیر می‌باشد. ایجاد قابلیت طراحی مسیر در این گونه از ربات‌ها به دلایل مختلف همچون قیدهای سینماتیکی: فضاهای کاری متغیر با زمان، موانع محیطی متعدد از پیش معلوم و نامعلوم، فضای پیکربندی چند بعدی و همچنین قیدهای دینامیکی: دامنه محدود نیروها، گشتاورها، شتاب‌ها، سرعت‌ها، درجات آزادی بالا، تعداد محدود عملگرها، اغتشاشات خارجی، دینامیک غیرخطی، اثرات سطح و جریانات زیرسطحی در آب‌های کم عمق، عدم قطعیت‌های ساختاری و غیرساختاری، ضرایب هیدرودینامیکی متغیر با زمان و ... امری چالش برانگیزی می‌باشد.

در این خصوص، بمنظور طراحی مسیر برای ربات‌های خودمختار زیرسطحی در بسیاری از تحقیقات صورت پذیرفته شده، تنها قیدهای سینماتیکی با فرض موانع از پیش معلوم در فرآیند طراحی مسیر لحاظ می‌شود.

الگوریتم درخت جستجوی سریع تصادفی حریص فازی^۷ در [۲] معرفی شده است. در الگوریتم مذکور تابع اندازه‌گیری اقلیدسی متداول با یک تابع متاهایوریستیکی جایگزین می‌گردد. بدین ترتیب درخت جستجو علاوه بر سنجش کمیت گرہ‌ها و شاخه‌های والد درخت، به سنجش کیفیت آنها نیز می‌پردازد. توانایی عبور درخت از گذرگاه‌های باریک و فضاهای ماز شکل با حجم محاسبات محدود بصورت روی خط بارزترین ویژگی مثبت و در مقابل عدم لحاظ نمودن قیدهای سینودینامیکی ربات در فرآیند رشد درخت از جمله مشکلات روش پیشنهادی است. سکوه‌های شارژ متحرک در [۳] جهت افزایش مداومت کاری گروهی از ربات‌های بدون سرنشین زیرسطحی خودمختار پیشنهاد شده است. به منظور طراحی مسیر برای این گروه، فضای پیکربندی شبکه بندی شده و از الگوی فروشنده دوره گرد با در نظر گرفتن جریانات سطحی و زیرسطحی استفاده شده است. طراحی مسیر برای یک ربات بدون سرنشین زیرسطحی خودمختار در یک فضای کاری سه بعدی متغیر با زمان در حضور موانع ثابت و متحرک به کمک الگوریتم تکاملی تفاضلی در [۴] ارائه شده است. در این روش ابتدا مسیر به کمک یک جمعیت اولیه طراحی شده و سپس به کمک تابع هزینه‌ای که زمان و مسافت طی شده را ملاک قرار می‌دهد شروع به بهینه سازی مسیر طراحی شده می‌نماید. از جمله سایر روش‌های طراحی مسیر مبتنی بر قیدهای سینماتیکی می‌توان به: الگوریتم‌های ابتکاری مشتمل بر بیوگرافی^۸، حشره شب‌تاب^۹ و دیفرانسیل تکاملی^{۱۰} [۵]، درخت جستجو سریع تصادفی بهینه چند بعدی^{۱۱} [۶]، روش لژاندر و مونت کارلو^{۱۲} [۷]، الگوریتم گرده گل^{۱۳} [۸]، الگوریتم گروه گرگ‌های اصلاح شده^{۱۴} [۹]، درخت جستجوی سریع تصادفی زیربهینه قابل تنظیم^{۱۵} [۱۰]، دیاگرام ورونی [۱۱]، توابع هموتوپیک^{۱۶} [۱۲]، فیلتر ذرات [۱۳]، توابع پتانسیل [۱۴] و ... لذا به دلیل عدم

⁷ Fuzzy Greedy RRT (FG-RRT)

⁸ Biogeography-Based Optimization (BBO)

⁹ Firefly Algorithm (FA)

¹⁰ Differential Evolution (DE)

¹¹ Multidimensional RRT*

¹² Legendre pseudospectral method (LPM) and Monte Carlo

¹³ Flower Pollination Algorithm (FPA)

¹⁴ Modified Wolf Pack Algorithm (WPA)

¹⁵ RRT*-Adjustable Bounds (RRT*-AB)

¹⁶ Homotopic Method

¹ Intelligence, Surveillance, and Reconnaissance (ISR)

² Mine Countermeasures (MCM)

³ Payload Delivery

⁴ Inspection/Identification (In/Id)

⁵ Anti-Submarine Warfare (ASW)

⁶ Technology Readiness Level (TRL)

بطور خلاصه مهم‌ترین دستاوردهای این پژوهش عبارتند از:
- گرہ‌ها و شاخه‌های الگوریتم طراحی مسیر پیشنهادی در فضای پیکربندی از موقعیت و سرعت ابتدایی ربات به موقعیت و سرعت نهایی مد نظر ماموریت شروع به گسترش و جستجو می‌نمایند، بنحوی که امکان پذیر بودن دستیابی به آنها توسط ربات از منظر قیدهای سینودینامیکی توسط کنترل‌کننده سطح پایین و مدل دینامیکی غیرخطی بررسی می‌گردد. همچنین، چنانچه ربات در حین رهگیری مسیر طراحی شده با مانعی از پیش نامعلوم مواجه شود، LRPPM از موقعیت فعلی ربات به نزدیک‌ترین نقطه مسیر بعنوان هدف موقت یک درخت محلی جهت اجتناب از برخورد با آن مانع توسعه می‌دهد.

- الگوریتم طراحی مسیر پیشنهادی بر روی یک کامپیوتر صنعتی تک بورد بصورت زمان حقیقی در محیط xPC-Target متلب پیاده‌سازی و طی یک سناریو مشخص مورد آزمون پردازنده در حلقه قرار خواهد گرفت. بدین ترتیب پیش از ورود ربات به آزمون‌های میدانی، الگوریتم طراحی مسیر اعتبارسنجی می‌گردد و از هزینه‌های آزمون میدانی و ریسک‌های مربوطه کاسته خواهد شد.

روند ارائه مطالب در ادامه بدین صورت خواهد بود که، در بخش دوم مدل غیرخطی شش درجه آزادی ربات به همراه کنترل‌کننده سطح پایین سمت و عمق ارائه می‌گردد. در بخش سوم ابتدا الگوریتم طراحی مسیر پیشنهادی معرفی شده و سپس بر روی کامپیوتر صنعتی تک بورد مدل Axiomtek 84710 از طریق xPC-Target متلب بصورت زمان حقیقی پیاده‌سازی و توسط یک سناریو آزمون مشخص عملکرد آن ارزیابی خواهد شد. نتایج آزمون پردازنده در حلقه در بخش چهارم نشان خواهد داد که الگوریتم پیشنهادی با ماهیت تصادفی خود توانایی طراحی مسیر سریع مجدد جهت اجتناب از برخورد با موانع از پیش نامعلوم را دارد. بخش پنجم به جمع‌بندی و نتیجه‌گیری موضوع اختصاص خواهد یافت.

۲- بیان مسئله

۲-۱- معادلات سینماتیک و دینامیک حرکت ربات خودمختار زیرسطحی بدون سرنشین

به منظور تعیین موقعیت و وضعیت ربات خودمختار زیرسطحی در فضای کاری سه بعدی نیاز به شش درجه

لحاظ نمودن محدودیت نیروها، گشتاورها و سایر قیدهای دینامیکی ربات، اجرایی بودن مسیر طراحی شده توسط ربات تضمین نمی‌شود.

الگوریتم‌های طراحی مسیر سینودینامیکی^۱ با در نظر گرفتن همزمان قیدهای سینماتیک و دینامیکی در فرآیند طراحی مسیر به منظور ایجاد قابلیت اجرایی بودن مسیر طراحی شده توسط ربات برای اولین بار در [۱۵] ارائه گردید. از جمله الگوریتم‌های طراحی مسیر سینودینامیکی می‌توان به: طراحی مسیر سینودینامیکی ابداعی [۱۶]، طراحی مسیر در فضای هزینه [۱۷]، درخت زمان معکوس [۱۸] و ... اشاره داشت. یکی از مشکلات مطرح در این کلاس از الگوریتم‌های طراحی مسیر، افزایش نمایی پیچیدگی محاسباتی و زمان مورد نیاز برای طراحی مسیر با افزایش درجات آزادی ربات و بعد فضای پیکربندی است. به منظور غلبه بر این موضوع، الگوریتم‌های طراحی مسیر تصادفی^۲ مبتنی بر نمونه‌برداری افزایشی^۳ در فضای پیکربندی ارائه شده‌اند. این کلاس از الگوریتم‌های طراحی مسیر کامل احتمالاتی^۴ می‌باشند. از جمله مهم‌ترین مزایای این کلاس می‌توان به: طراحی مسیر در فضای حالت بجای فضای پیکربندی و لذا امکان لحاظ نمودن نرخ تغییر متغیرها در فرآیند طراحی مسیر، امکان اعمال قیدهای سینماتیک و دینامیکی در فرآیند نمونه برداری، عدم نیاز به نقشه دقیق از فضای کاری و عبور از موانع بصورت جعبه سیاه و ... اشاره داشت. از جمله الگوریتم‌های مطرح در این حوزه می‌توان از: روش نقشه راه احتمالاتی^۵ [۱۹] درخت جستجو سریع متراکم^۶ [۲۰]، درخت جستجو سریع تصادفی [۲۱]، درخت گسترده در فضای^۷ [۲۲] نامبرد.

در این پژوهش، مزایای هر دو کلاس الگوریتم طراحی مسیر سینودینامیکی و تصادفی به منظور طراحی سریع یک مسیر قابل اجرا توسط ربات در یک فضای کاری با موانع از پیش نامعلوم با یکدیگر ترکیب شده و الگوریتم درخت جستجو سریع تصادفی محلی مبتنی بر نمونه برداری افزایشی معرفی می‌گردد.

¹ Kinodynamic Path Planning Algorithms

² Randomized Path Planning Algorithms

³ Incremental Sampling-Based

⁴ Probabilistically Complete

⁵ Probabilistic Roadmap (PRM)

⁶ Rapidly Exploring Dense Tree (RDT)

⁷ Expansive-Spaces Tree (EST)

ماتریس توزیع^۷، $u = [\delta_R, \delta_E]^T \in R^2$ ورودی‌های کنترلی^۸، که δ_R, δ_E سیگنال کنترلی اعمالی بر سطوح کنترل افقی و عمودی است. حرکت انتقالی و دورانی ربات در راستای هر یک از محورهای طولی، عرضی و عمودی توسط شش معادله ذیل بیان می‌شود. حرکت انتقالی ربات (Surge) در راستای محور طولی^۹:

$$m(\ddot{u} - vr + wq - x_g(q^2 + r^2) + z_g(pr + \dot{q})) = X_{HS} + X_u|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr + X_{rr}rr + X_{prop} \quad (3)$$

حرکت انتقالی ربات در راستای محور عرضی^{۱۰} (Sway):

$$m(\ddot{v} - wp + ur + z_g(qr - \dot{p}) + x_g(pq + \dot{r})) = Y_{HS} + Y_{|v|}v|v| + Y_{r|r}r|r| + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur + Y_{wp}wp + Y_{pq}pq + Y_{uv}uv + Y_{uu\delta_r}u^2\delta_r \quad (4)$$

حرکت انتقالی ربات در راستای محور عمودی^{۱۱} (Heave):

$$m(\ddot{w} - uq + vp - z_g(q^2 + p^2) + x_g(rp - \dot{q})) = Z_{HS} + Z_{|w|}w|w| + Z_{q|q}q|q| + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp + Z_{pr}pr + Z_{uw}uw + Z_{uu\delta_s}u^2\delta_s \quad (5)$$

حرکت دورانی ربات در راستای محور طولی (Roll):

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr + m[-z_g(\dot{v} - wp + ur)] = K_{HS} + K_{p|p}p|p| + K_{\dot{p}}\dot{p} + K_{prop} \quad (6)$$

حرکت دورانی ربات در راستای محور عرضی (Pitch):

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M_{HS} + M_{(w|w)}w|w| + M_{(q|q)}q|q| + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uq}uq + M_{vp}vp + M_{rp}rp + M_{uw}uw + M_{(uu\delta_s)}u^2\delta_s \quad (7)$$

حرکت دورانی ربات در راستای محور عمودی (Yaw):

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})qp + m[x_g(\dot{v} - wp + ur)] = N_{HS} + N_{|v|}v|v| + N_{r|r}r|r| + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{ur}ur + N_{wp}wp + N_{qp}qp + N_{uv}uv + N_{uu\delta_r}u^2\delta_r \quad (8)$$

رفتار مدل دینامیکی شش درجه آزادی غیرخطی ربات بصورت حلقه باز به کمک مقادیر عددی ضمیمه (۱) توسط

آزادی می‌باشد [۲۳]. سه مولفه اول و نرخ تغییرات آنها توسط ماتریس انتقال موقعیت و سرعت خطی ربات و سه مولفه دوم و نرخ تغییرات آنها توسط ماتریس انتقال وضعیت و سرعت زاویه‌ای ربات نشان داده می‌شود. ارتباط بین دو دستگاه مختصات بدنی و اینرسی توسط معادلات سینماتیک حرکت ربات بصورت ماتریسی زیر بیان می‌شود.

$$\dot{\eta} = J(\eta)v \leftrightarrow \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \eta = [\eta_1^T, \eta_2^T]^T, \eta_1 = [x \ y \ z]^T, \eta_2 = [\phi \ \theta \ \psi]^T, \quad (1)$$

$$v = [v_1^T, v_2^T]^T, v_1 = [u \ v \ w]^T, v_2 = [p \ q \ r]^T$$

که در آن بردار $\eta \in R^6$ در دستگاه مختصات اینرسی نشان‌دهنده موقعیت و وضعیت ربات، بردار $v \in R^6$ در دستگاه مختصات بدنی نشان‌دهنده سرعت خطی و زاویه‌ای ربات و $J: R^6 \rightarrow R^{6 \times 6}$ ماتریس انتقال بین دو دستگاه مختصات می‌باشد. قابل ذکر است که مستقیماً نمی‌توان از بردار سرعت‌های انتقالی و دورانی به بردارهای موقعیت و وضعیت در دستگاه اینرسی رسید، لذا توسط ماتریس انتقال $J_1(\eta_2) = C_{z,\psi}^T C_{y,\theta}^T C_{x,\phi}^T$ ارتباط بین موقعیت و سرعت انتقالی ربات بیان می‌شود. همچنین ارتباط بین متغیرهای وضعیت و سرعت دورانی ربات توسط $J_2(\eta_2)$ برقرار است. معادلات شش درجه آزادی حاکم بر دینامیک حرکت ربات به صورت رابطه کلی (۲) بیان می‌شود:

$$\underbrace{M_{RB}v + C_{RB}(v)v}_{\text{rigid body terms}} + \underbrace{M_{AV} + C_A(v)v + D(v)v}_{\text{hydrodynamic terms}} + \underbrace{g(\eta)}_{\text{hydrostatic terms}} = \tau_d + \tau_u = \tau_d + Bu$$

external disturbances forces and moments

که در آن $M_{RB} + M_A = M \in R^{6 \times 6}$ ماتریس اینرسی و جرم افزوده^۱ $C_{RB} + C_A = C: R^6 \rightarrow R^{6 \times 6}$ ماتریس شتاب کوریولیس و جانب مرکز^۲ $D: R^6 \rightarrow R^{6 \times 6}$ ماتریس میرایی^۳ و $g: R^6 \rightarrow R^6$ بردار نیروها و گشتاورهای^۴ ناشی از جاذبه زمین است. $\tau_d \in R^6$ بردار اغتشاشات خارجی^۵، $\tau_u \in R^6$ بردار نیروها و گشتاورهای اعمالی به ربات^۶، $B \in R^{6 \times 2}$

⁷ Distribution Matrix

⁸ Control Input

⁹ Longitudinal Axis

¹⁰ Lateral Axis

¹¹ Vertical Axis

¹ Inertia Matrix (including added mass)

² Coriolis and Centripetal Terms

³ Hydrodynamic Damping Matrix

⁴ Gravitational Forces and Moments

⁵ Vector of the External Disturbance

⁶ Vector of Forces and Moments Acting on the AUV

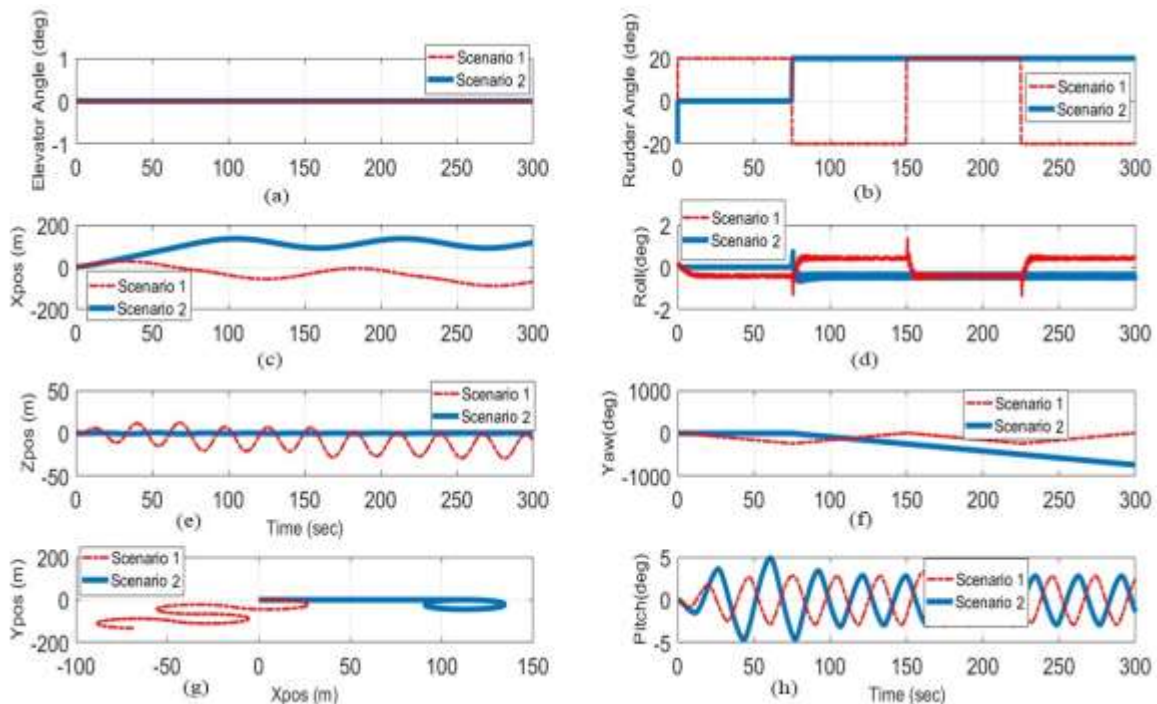
سناریو آزمون در کانال طولی رفتار نوسانی دارد، شکل‌های (h,e-۱) خط چین قرمز رنگ. شایان ذکر است که رول وسیله در راستای محور طولی به کمک فاصله عمودی بین مرکز جرم و مرکز شناوری خنثی می‌شود. در سناریو آزمون شماره دو ربات ابتدا مسیر مستقیمی را طی می‌نماید و پس از آن شروع به چرخش در صفحه افقی می‌نماید.

رفتار ربات در این سناریو آزمون نیز در صفحه افقی رضایت بخش، شکل‌های (g-f,c-۱) خط ممتد آبی رنگ و در صفحه عمودی همراه با نوسان است، شکل‌های (h,e-۱) خط ممتد آبی رنگ. با توجه به ارزیابی‌های صورت گرفته از رفتار مدل شش درجه آزادی غیرخطی ربات در سناریوهای آزمون تعریف شده می‌توان نتیجه گرفت که ربات به صورت حلقه باز در کانال عرضی پایدار و در کانال طولی رفتار نوسانی دارد. با توجه به نتایج آزمون حلقه باز، کنترل کننده سطح پایین در دو صفحه طولی و عرضی در ادامه ارائه می‌گردد. در بخش سوم از مدل دینامیکی شش درجه آزادی غیرخطی ربات و کنترل کننده‌های سطح پایین ارائه شده جهت اعمال قیدهای سینودینامیکی ربات بر رشد گره‌ها و شاخه‌های درخت طراحی مسیر پیشنهادی L-RRT استفاده خواهیم نمود.

دو سناریو آزمون ذیل مورد ارزیابی قرار گرفت. در سناریو آزمون شماره یک (زیگزاگ)، زاویه سطوح کنترل افقی در صفر درجه ثابت می‌شود و زاویه سطوح کنترل عمودی از $+20$ درجه به -20 درجه و بالعکس تغییر می‌کند. سرعت پیشروی در راستای محور طولی در ۳ نات ثابت شده است، شکل (a-۱) و شکل (b-۱) خط چین قرمز رنگ.

در سناریو آزمون شماره دو (دایروی)، زاویه سطوح کنترل عمودی قبل از 75 ثانیه صفر و بعد از آن در حداکثر مقدار خود ($+20$) قرار می‌گیرد، شکل (a-۱) و شکل (b-۱) خط ممتد آبی رنگ. با توجه به سناریوهای آزمون یک و دو و پیش از طراحی کنترل کننده‌های سمت و عمق مناسب، جهت بررسی رفتار شش درجه آزادی حلقه باز ربات در کانال‌های طولی و عرضی معادلات حاکم بر سینماتیک و دینامیک (روابط یک الی هشت) در محیط سیمولینک نرم افزار متلب پیاده‌سازی شد. موقعیت و وضعیت ربات در سناریو آزمون شماره یک و دو در شکل‌های (c-۱ الی h) به ترتیب نشان داده شده است.

در سناریو آزمون شماره یک ربات اگرچه مقداری رول دارد، شکل (d-۱) خط چین قرمز رنگ، ولیکن رفتار تقریباً مطلوبی در کانال عرضی از خود نمایش می‌دهد، شکل‌های (g-f,c-۱) خط چین قرمز رنگ. با این وجود ربات در این



شکل (۱) رفتار حلقه باز ربات در سناریوهای آزمون شماره یک و دو (زیگزاگ و دایره).

۲-۳- کنترل کننده سطح پایین در صفحه طولی

بردار خطا به منظور کنترل عمق ربات در صفحه طولی بصورت رابطه ذیل تعریف می‌گردد:

$$\tilde{x} = x - x_d = \begin{bmatrix} w \\ q \\ z \\ \theta \end{bmatrix} - \begin{bmatrix} w_d \\ q_d \\ z_d \\ \theta_d \end{bmatrix} \quad (16)$$

سطح لغزش بصورت جمع وزن دار متغیرهای بردار خطا تعریف می‌گردد.

با توجه به مدل صفحه طولی AUV ارائه شده در [۲۹] سیگنال کنترلی اعمالی به سطوح کنترلی افقی همانند بخش قبل طراحی می‌گردد. عملکرد کنترل کننده سطح پایین پیش از استفاده در الگوریتم طراحی مسیر پیشنهادی L-RRT در حضور اغتشاشات محیطی و نامعینی در ضرایب هیدرودینامیکی توسط دو سناریو ارزیابی می‌گردد. در سناریو شماره یک آزمون: ۱۰ درصد نامعینی در ممان‌های اینرسی و جرم ربات به همراه اغتشاشات محیطی بصورت رابطه ذیل اعمال می‌گردد:

$$M\dot{v} + C(v)v + D(v)v + G(\eta) = \tau + d$$

$$d_{surge} = \begin{cases} 7+5*\sin(0.1*t)-3*\cos(0.1*t) & 60 < t < 70 \\ 5-3*\sin(0.1*t)-3*\cos(0.1*t) & 130 < t < 150 \end{cases} \quad (17)$$

$$d_{sway} = \begin{cases} 7-5*\sin(0.1*t)-3*\cos(0.1*t) & 60 < t < 70 \\ -7+5*\sin(0.1*t)+3*\cos(0.1*t) & 130 < t < 150 \end{cases}$$

در سناریو شماره دو آزمون کنترل کننده سطح پایین: ۳۰ درصد نامعینی در ممان‌های اینرسی و جرم ربات به همراه اغتشاشات محیطی بصورت رابطه ذیل اعمال می‌گردد:

$$M\dot{v} + C(v)v + D(v)v + G(\eta) = \tau + d \quad (18)$$

$$d_{surge} = \begin{cases} -7+10*\text{randn}(1,1)+5*\sin(0.1*t)-3*\cos(0.1*t) & 50 < t < 60 \\ 6-10*\text{randn}(1,1)+3*\sin(0.1*t)-3*\cos(0.1*t) & 115 < t < 135 \end{cases}$$

$$d_{sway} = \begin{cases} 8+10*\text{randn}(1,1)-4*\sin(0.1*t)-3*\cos(0.1*t) & 50 < t < 60 \\ 8-10*\text{randn}(1,1)+5*\sin(0.1*t)+3*\cos(0.1*t) & 115 < t < 135 \end{cases}$$

در رابطه فوق بخش تابع سینوسی، تابع تصادفی و قسمت ثابت به ترتیب نشان دهنده موج، اغتشاشات محیطی تصادفی و جریانات زیرسطحی می‌باشد. ربات در سناریو شماره یک با وجود ۱۰ درصد نامعینی و اغتشاشات مطابق رابطه (۱۷) بدون خطای ماندگار مسیر مرجع را رهگیری می‌نماید.

۲-۲- کنترل کننده سطح پایین در صفحه عرضی

بردار خطا به منظور کنترل سمت ربات در صفحه عرضی بصورت رابطه ذیل تعریف می‌گردد [۲۶, ۲۷]:

$$\tilde{x} = x - x_d = \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} - \begin{bmatrix} v_d \\ r_d \\ \psi_d \end{bmatrix} \quad (9)$$

سطح لغزش بصورت جمع وزن دار متغیرهای بردار خطا در رابطه زیر تعریف می‌گردد:

$$\sigma = S^T \tilde{x} = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} \begin{bmatrix} x_1 - x_{1d} \\ x_2 - x_{2d} \\ x_3 - x_{3d} \end{bmatrix} = s_1(v - v_d) + s_2(r - r_d) + s_3(\psi - \psi_d) = 0 \quad (10)$$

با توجه به صفر بودن مقدار مطلوب $v_d = r_d = 0$ در صفحه عرضی، سیگنال کنترلی اعمالی به سطوح کنترلی عمودی بصورت رابطه زیر محاسبه خواهد شد:

$$\hat{\delta}_R = \hat{\delta}_R + \bar{\delta}_R = \frac{-\eta \text{satsgn}\left(\frac{\sigma}{\phi}\right) - (s_1 a_{11} + s_2 a_{21})v - (s_1 a_{12} + s_2 a_{22} + s_3)r}{s_1 b_1 + s_2 b_2} \quad (11)$$

در رابطه (۱۲) قسمت خطی سیگنال کنترلی اعمالی به سطوح کنترلی عمودی (۱۱) و $\bar{\delta}_R$ در رابطه (۱۳) قسمت غیرخطی سیگنال کنترلی اعمالی به سطوح کنترلی عمودی (۱۱) می‌باشد، که وظیفه جبران سازی عدم قطعیت‌های غیرساختاری که سبب خروج متغیرهای حالت از سطح لغزش می‌شوند را بر عهده دارد.

$$\hat{\delta}_R = \frac{-[(s_3 + s_1 a_{12} + s_2 a_{22})r + (s_1 a_{11} + s_2 a_{21})v]}{s_1 b_1 + s_2 b_2} = \quad (12)$$

که در آن K بهره بردار بازخورد می‌باشد.

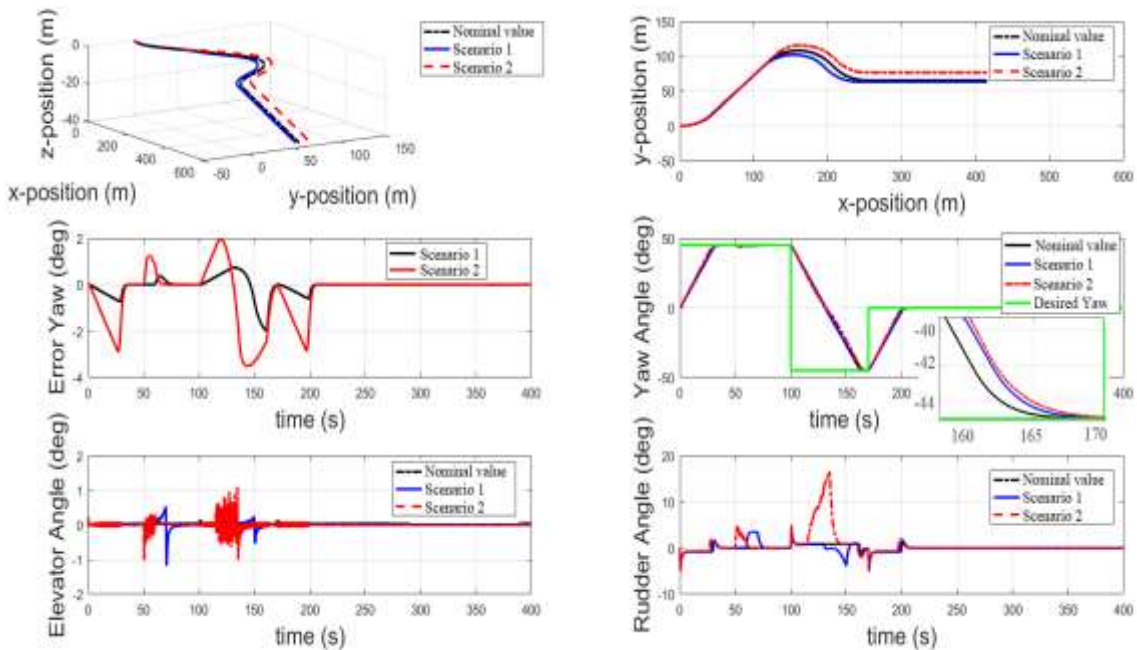
$$\bar{\delta}_R = \frac{-\eta \text{satsgn}\left(\frac{\sigma}{\phi}\right)}{s_1 b_1 + s_2 b_2} = -(S^T B)^{-1} \eta \text{satsgn}\left(\frac{\sigma}{\phi}\right) \quad (13)$$

رفتار حلقه بسته سیستم هنگامی که متغیرها بر روی سطح لغزش قرار دارند بصورت رابطه زیر خواهد شد:

$$\begin{aligned} \dot{x} &= Ax + B \hat{\delta}_R, \\ \dot{x} &= [A - BK]x, \\ \dot{x} &= [A_{cl}]x \end{aligned} \quad (14)$$

که A_{cl} ماتریس حالت حلقه بسته می‌باشد. ضرایب سطح لغزش از رابطه ذیل محاسبه خواهد شد [۲۷, ۲۸]:

$$S^T = [s_1 \ s_2 \ s_3] = \text{Eigenvector}(A_{cl}^T)_{\text{Eigenvalue} = \text{Null}} \quad (15)$$



شکل (۲) رفتار مطلوب و واقعی ربات در سناریوهای آزمون شماره یک و دو کنترل کننده سطح پایین.

پیش نامعلوم عبور می نماید و به کمک روش هر لحظه^۱ قیدهای بهینه سازی در طول مسیر اعمال می گردد. الگوریتم طراحی مسیر درخت جستجو سریع تصادفی در ادامه با $T(V,E)$ نمایش داده خواهد شد، که T معرف درخت جستجو، V مجموعه گره های درخت و E مجموعه شاخه های درخت را نشان می دهد. ساختار $T(V,E)$ از پنج تابع اصلی ذیل تشکیل می گردد.

۱- تابع اندازه گیری^۲: انتخاب مناسب این تابع بر عملکرد الگوریتم های طراحی مسیر مبتنی بر نمونه برداری^۳ تاثیر بسزایی خواهند داشت. تابع اندازه گیری از فضای توپولوژی به فضای متریک توسط رابطه زیر تعریف می گردد.

$$\rho: X \times X \rightarrow R \quad a, b, c \in X, \quad (19)$$

$$\rho(x, x') = \left(\sum_{i=1}^n |x_i - x'_i|^p \right)^{1/p}$$

که در آن (ρ) تابع اندازه گیری است. چنانچه p در (ρ) برابر با ۱، ۲ و ∞ باشد، تابع اندازه گیری به ترتیب منتهن^۴، اقلیدسی^۵ و تخته شطرنج نامیده می شود. در ادامه از این تابع به منظور انتخاب گره والد اصلح (q_{parent}) برای گره فرزند تصادفی (q_{rand}) تولید شده استفاده خواهیم نمود.

این امر در حالی است که، در سناریو شماره دو با افزایش دامنه اغتشاشات (۱۸) و وجود ۳۰ درصد نامعینی، خطای ماندگار در رهگیری مسیر مرجع مشاهده می شود. دامنه سیگنال کنترلی در سناریو دو افزایش می یابد ولیکن قابل قبول است. همچنین خطای گذار در زاویه سمت ربات وجود دارد، که با گذشت زمان میرا می شود، شکل (۲).

مسئله ۱: طراحی مسیر روی خط برای ربات خودمختار زیرسطحی در یک محیط متشکل از موانع از پیش معلوم و نامعلوم با لحاظ نمودن همزمان قیدهای سینماتیکی و دینامیکی (سینودینامیکی) در فرآیند طراحی مسیر به منظور حصول مسیری با قابلیت اجرایی توسط ربات.

۳- الگوریتم طراحی مسیر پیشنهادی ۳-۱- درخت جستجو سریع تصادفی

الگوریتم درخت جستجو سریع تصادفی (RRT) در کلاس روش های هیوریستیکی می باشد که به دلیل ماهیت تصادفی، نیاز به حجم محاسبات محدودی برای طراحی مسیر در فضاهای پیکربندی چند بعدی برای ربات های با درجات آزادی بالا خواهد داشت. لذا از روش مذکور جهت طراحی مسیر بصورت زمان حقیقی در طیف وسیعی از ربات های خودمختار استفاده می گردد. در این روش ابتدا یک مسیر اولیه به سرعت بصورت تصادفی طراحی شده و سپس توسط درخت های جستجو محلی ربات از موانع از

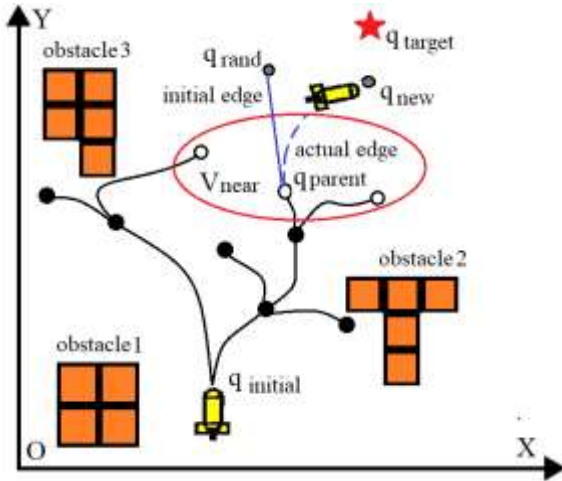
¹ Any-Time RRT

² Metric Function (MF)

³ Sampling-Based Algorithm

⁴ Manhattan (Taxicab)

⁵ Euclidean



شکل (۳) توصیف مفهوم الگوریتم طراحی مسیر درخت جستجو سریع تصادفی.

Algorithm 1 conventional RRT path planning algorithm

1. **Begin** CONVENTIONAL RRT
2. $q_{initial}, q_{target} \leftarrow$ initial and target vertices
3. $c \leftarrow$ configuration space
4. $N \leftarrow$ maximum iteration for path planning
5. $T_C \leftarrow L^2$ norm (q_{new}, q_{target})
6. $E \leftarrow [\emptyset]$
7. $V \leftarrow [q_{initial}, q_{target}]$
8. $T(V, E) \leftarrow$ search tree
9. $\beta \leftarrow$ threshold of q_{target}
10. **while** $i \leq N$ and $T_C > \beta$ **do**
11. $q_{rand} \leftarrow$ RSF (c_{free}) ▷ by Fun. [2]
12. $V_{near} \leftarrow$ NNF (q_{rand}, V) ▷ by Fun. [3]
13. $q_{parent} \leftarrow$ MF (V_{near}) ▷ by Fun. [1]
14. $[q_{new}, \gamma] \leftarrow$ SBF (q_{rand}, q_{parent}) ▷ by Fun. [4]
15. **If** CDF ($q_{rand}, q_{new}, \gamma$) in c_{free} **then**
▷ by Fun. [5]
16. $E \leftarrow E \cup \gamma$
17. $V \leftarrow V \cup q_{new}$
18. Update (T)
19. Check termination condition
20. **End**

گره والد طراحی می‌شود. سپس با توجه به محدودیت‌های ربات مسیر واقعی (خط چین آبی) و گره جدید (q_{new}) به $T(V, E)$ افزوده می‌شود (خط ۱۴ الگوریتم ۱). قرار گرفتن مسیر واقعی (شاخه درخت جستجو)، گره جدید و گره فرزند تصادفی در فضای آزاد پیکربندی توسط تابع تشخیص مانع بررسی می‌گردد. در صورت عدم برخورد

۲- تابع نمونه‌برداری تصادفی^۱: از این تابع به منظور تولید گره فرزند تصادفی (q_{rand}) و اضافه نمودن آن به مجموعه گره (V) درخت جستجو سریع تصادفی $T(V, E)$ جهت گسترش درخت در فضای پیکربندی استفاده می‌گردد.

۳- تابع نزدیک‌ترین همسایه^۲: این تابع به کمک تابع اندازه‌گیری (ρ) مجموعه‌ای از گره‌های درخت جستجو ($V_{near} \subset V$) را که صلاحیت تبدیل شدن به گره والد (q_{parent}) برای گره فرزند تصادفی (q_{rand}) را دارند، مشخص می‌نماید.

۴- تابع رشد شاخه^۳: خروجی تابع نزدیک‌ترین همسایه (q_{parent}) و نمونه‌بردار تصادفی (q_{rand}) بعنوان ورودی توسط این تابع گرفته و ابتدا شاخه اولیه بصورت خط مستقیم و سپس شاخه واقعی (γ) بر اساس محدودیت‌های حرکتی و زمانی در فضای آزاد پیکربندی (c_{free}) طراحی می‌شود. با توجه به محدودیت‌های ذکر شده امکان عدم تطبیق گره انتهایی شاخه با گره فرزند تصادفی می‌باشد. در این صورت (q_{rand}) حذف و گره تصادفی طراحی شده تحت عنوان گره جدید (q_{new}) به مجموعه گره‌های درخت افزوده می‌شود.

۵- تابع تشخیص مانع^۴: تضمین عدم برخورد خروجی توابع نمونه برداری تصادفی (q_{rand})، تابع رشد شاخه (γ) و (q_{new}) با موانع توسط این تابع صورت می‌پذیرد
 $(q_{rand}(x), q_{new}(x), \gamma(x)) \forall x \in c_{free}$

بر اساس پنج تابع معرفی شده، توصیف مفهوم $T(V, E)$ در شکل (۳) و شبه کد آن در الگوریتم ۱ معرفی می‌گردد. در شکل (۳) در مرحله اول گره فرزند تصادفی (q_{rand}) توسط تابع نمونه‌برداری تصادفی که با دایره کوچک خاکستری نمایش داده شده است (خط ۱۱ الگوریتم ۱) تولید می‌گردد. در مرحله دوم مجموعه‌ای از گره‌های دارای صلاحیت (V_{near}) توسط تابع اندازه‌گیری (ρ) و تابع نزدیک‌ترین همسایه به منظور تعیین گره والد اصلح (q_{parent}) که با دایره بزرگ قرمز رنگ نمایش داده شده‌اند (خطوط ۱۲ و ۱۳ الگوریتم ۱) انتخاب می‌شوند.

در مرحله سوم ابتدا توسط تابع رشد شاخه، مسیر مطلوب بصورت خط مستقیم آبی رنگ بین گره فرزند تصادفی و

¹ Random Sampling Function (RSF)

² Nearest Neighbors Function (NNF)

³ Steer Branches Function (SBF)

⁴ Collision Detection Function (CDF)

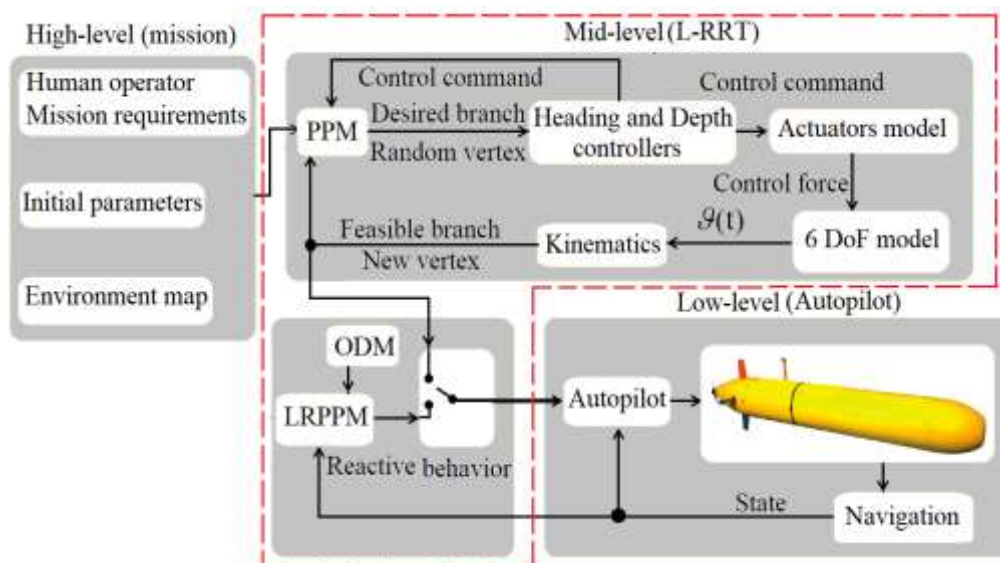
معرفی می‌گردد. ماژول طراحی مسیر (PPM) به کمک الگوریتم درخت جستجو سریع تصادفی با دریافت نیازمندی‌های ماموریت و نقشه فضای پیکربندی اولیه، شروع به طراحی گره فرزند تصادفی (خط ۱۱ الگوریتم ۲)، گره والد اصلح (خطوط ۱۲ و ۱۳ الگوریتم ۲) و گره جدید (خط ۱۴ الگوریتم ۲) در هر تکرار می‌نماید. گره فرزند تصادفی بعنوان نقطه مسیر مطلوب به همراه گره والد انتخابی وارد کنترل‌کننده سمت و عمق می‌شوند. کنترل‌کننده سطح پایین سیگنال‌های کنترلی عمودی و افقی را جهت رسیدن ربات به گره فرزند تصادفی طراحی می‌نماید (خط ۱۵ الگوریتم ۲).

سیگنال‌های کنترلی به مدل دینامیک شش درجه آزادی غیر خطی AUV اعمال شده و گره و شاخه جدید قابل دسترس $[q_F, \gamma_F]$ (اجرای توسط AUV) طراحی می‌گردند (خط ۱۶ الگوریتم ۲). بدین ترتیب گره‌ها و شاخه‌های درخت جستجو با لحاظ نمودن مدل دینامیکی وسیله و به عبارت دیگر با لحاظ نمودن قیده‌های سینودینامیکی ربات در فضای پیکربندی رشد و توسعه می‌یابد و در نهایت از مجموعه‌ای از شاخه‌ها و گره‌ها با ضمانت اجرایی توسط AUV برخوردار خواهیم بود. پس از اطمینان از عدم برخورد گره و شاخه قابل دسترس $[q_F, \gamma_F]$ با موانع (خط ۱۷ الگوریتم ۲)، سیگنال‌های کنترلی مربوطه و همچنین گره و شاخه طراحی شده جهت طراحی مسیر نهایی توسط ماژول طراحی مسیر ذخیره می‌گردند (خط ۲۰ الگوریتم ۲).

موارد ذکر شده با فضای اشغال شده توسط موانع، گره جدید و مسیر واقعی به مجموعه گره‌ها و شاخه‌های درخت افزوده می‌شود (خطوط ۱۵-۱۸ الگوریتم ۱). این فرآیند تا رسیدن اولین گره به شعاع تعریف شده توسط کاربر به نزدیکی گره هدف ادامه می‌یابد (خط ۱۹ الگوریتم ۱).

۲-۳- درخت جستجو سریع تصادفی محلی

به منظور افزایش سطح خودمختاری ربات حداقل به همکاری چهار زیرسیستم هدایت، کنترل، ناوبری و طراحی مسیر نیاز می‌باشد. در اینجا بمنظور طراحی سریع یک مسیر با قابلیت اجرای توسط AUV با در نظر گرفتن همزمان قیده‌های سینماتیکی و قیده‌های دینامیکی در یک فضای کاری با موانع از پیش نامعلوم، الگوریتم طراحی مسیر درخت جستجو سریع تصادفی محلی که L-RRT نامیده می‌شود، معرفی می‌گردد. الگوریتم L-RRT پیشنهادی ضمن برخورداری از توانایی اجتناب از برخورد با موانع از پیش نامعلوم، بطور همزمان از مزایای هر دو کلاس الگوریتم‌های طراحی مسیر مبتنی بر نمونه‌برداری تصادفی (سرعت بالا در طراحی مسیر اولیه) و الگوریتم‌های سینودینامیکی (اعمال قیده‌های سینماتیکی و دینامیکی در فرآیند نمونه برداری) برخوردار می‌باشد. الگوریتم طراحی مسیر پیشنهادی (L-RRT) از سه جزء اصلی: ماژول طراحی مسیر، ماژول طراحی مسیر محلی بلادرنگ و ماژول تشخیص موانع تشکیل می‌گردد. بلوک دیاگرام L-RRT پیشنهادی در شکل (۴) و شبه کد آن در الگوریتم ۲



شکل (۴) بلوک دیاگرام الگوریتم طراحی مسیر درخت جستجو سریع تصادفی محلی پیشنهادی.

Algorithm 2 L-RRT path planning algorithm

```

1. Begin L-RRT
2.  $q_{initial}, q_{target} \leftarrow$  initial and target vertices
3.  $c \leftarrow$  configuration space
4.  $N \leftarrow$  maximum iteration for path planning
5.  $T_C \leftarrow L^2$  norm ( $q_{new}, q_{target}$ )
6.  $E \leftarrow [\emptyset]$ 
7.  $V \leftarrow [q_{initial}, q_{target}]$ 
8.  $T(V, E) \leftarrow$  search tree
9.  $\beta \leftarrow$  threshold of  $q_{target}$ 
10. While  $i \leq N$  and  $T_C > \beta$  do
11.  $q_{rand} \leftarrow$  RSF ( $c_{free}$ )  $\triangleright$  by Fun.[2]
12.  $V_{near} \leftarrow$  NNF ( $q_{rand}, V$ )  $\triangleright$  by Fun.[3]
13.  $q_{parent} \leftarrow$  MF ( $V_{near}$ )  $\triangleright$  by Fun.[1]
14.  $[q_{new}, \gamma] \leftarrow$  SBF ( $q_{rand}, q_{parent}$ )  $\triangleright$  by Fun.[4]
15.  $[\delta_R, \delta_E] \leftarrow$  SMC ( $q_{new}, q_{parent}$ )  $\triangleright$  by Sec.[2-3,4]
16.  $[q_F, \gamma_F] \leftarrow$  SNM ( $\delta_R, \delta_E$ )  $\triangleright$  by Equ.[2]
17. If CDF ( $q_{rand}, q_F, \gamma_F$ ) in  $c_{free}$  then  $\triangleright$  by Fun.[5]
18.  $E \leftarrow E \cup \gamma_F$ 
19.  $V \leftarrow V \cup q_F$ 
20.  $U_i \leftarrow [q_{parent}, q_F, \gamma_F, \delta_R, \delta_E]$ 
21. Update ( $T$ )
22. Check termination condition
23. While  $i \leq$  Number of ( $q_F$ )
24. If ODM==1 then
25.  $[q_{new}, \gamma] \leftarrow$  LRPPM (AUV location,  $U_i$ )
26. AUV  $\leftarrow$  Autopilot ( $q_{new}, \gamma$ )
27. else
28. AUV  $\leftarrow$  Autopilot ( $U_i$ )
29. Check termination condition
30. End

```

رایانه میزبان، رایانه هدف و محیط xPC-Target نرم افزار متلب که در شکل (۵) نشان داده شده‌اند. رایانه میزبان یک لپ‌تاپ با ۸ گیگابایت RAM و پردازنده Intel Core i7-4700HQ و فرکانس ۲,۴ GHz می‌باشد. رایانه هدف یک SBC مدل Axiomtek 84710 بوده و xPC-Target وظیفه تبدیل، انتقال و پیاده‌سازی کدهای اجرایی L-RRT را توسط پورت ارتباطی TCP/IP با سرعت ۱۰ مگابیت بر ثانیه بصورت زمان حقیقی بر روی رایانه هدف را برعهده دارد.

این فرآیند تا رسیدن اولین گره به شعاع تعریف شده توسط کاربر به نزدیکی گره هدف ادامه می‌یابد. چنانچه ماژول تشخیص مانع (ODM) به کمک سنسورهای مرتبط در حین رهگیری هریک از گره‌ها و شاخه‌های طراحی شده (نقطه مسیرها با قابلیت اجرایی توسط ربات)، مانعی از پیش نامعلوم را بر روی مسیر حرکتی طراحی شده توسط PPM را تشخیص دهد (خط ۲۴ الگوریتم ۲)، نقشه فضای کاری اولیه به کمک روش تجزیه سلولی به روزرسانی شده و ماژول طراحی مسیر محلی بلادرنگ (LRPPM)، درخت جستجو سریع تصادفی را از موقعیت فعلی ربات به نزدیک‌ترین نقطه مسیر توسعه خواهد داد (خط ۲۵ الگوریتم ۲). بمنظور اجتناب ربات از برخورد با موانع از قبل نامعلوم در معماری رفتار واکنشی^۱ تدوین شده، نقطه مسیرهای طراحی شده توسط LRPPM از اولویت بالاتری نسبت به نقطه مسیرهای طراحی شده توسط PPM برخوردار می‌باشند.

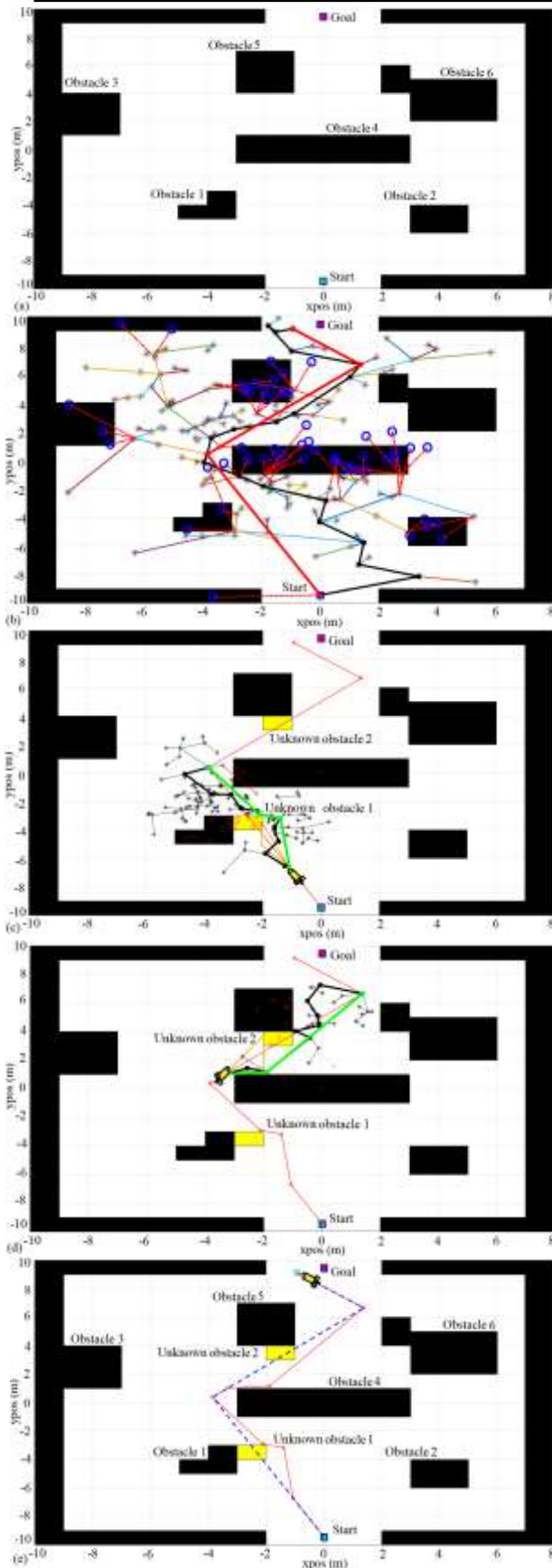
۳-۳- پیاده‌سازی بر روی کامپیوتر صنعتی تک‌برد و آزمون پردازنده در حلقه

الگوریتم طراحی مسیر درخت جستجو سریع تصادفی محلی پیشنهادی به منظور طراحی سریع یک مسیر قابل اجرا توسط AUV در یک محیط کاری با موانع از پیش نامعلوم با لحاظ نمودن همزمان قیدهای سینماتیکی و دینامیکی (سینودینامیکی) توسط آزمون پردازنده در حلقه مورد ارزیابی قرار خواهد گرفت.

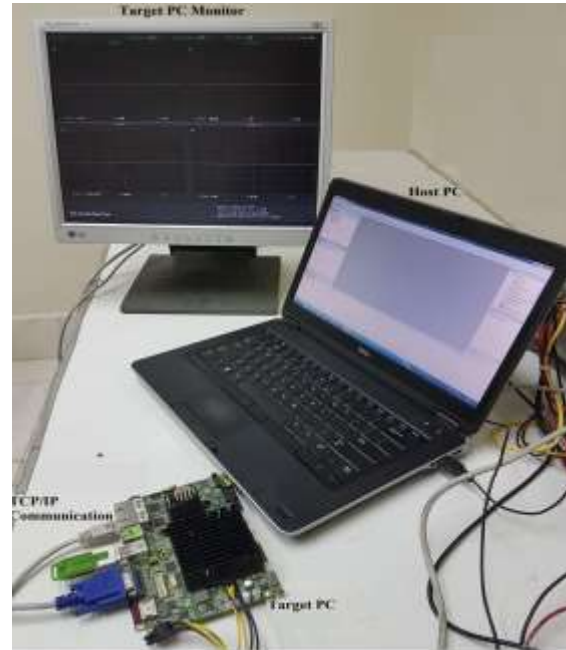
هدف از این آزمون افزایش اعتبار و اشکال‌زدایی از کدهای تدوین شده قبل از انجام آزمون سخت افزار در حلقه و بعد از آزمون نرم‌افزار در حلقه، به کمک پردازنده مرکزی واقعی مورد استفاده در AUV و در محیط xPC-Target نرم افزار متلب می‌باشد. به کمک انجام این آزمون، تیم طراحی می‌تواند بدون درگیر نمودن AUV واقعی و پیش از اعمال هزینه‌های آزمون‌های کارخانه‌ای، حوضچه و در نهایت میدانی^۲ به پروژه از صحت عملکرد روش پیشنهادی فراتر از سطح شبیه‌سازی اطمینان حاصل [۳۰] نماید. ساختار اصلی آزمون PIL از سه جزء تشکیل می‌گردد:

¹ Reactive Behavior

² Factory Acceptance Tests (FAT), Harbour Acceptance Tests (HAT) and Sea Acceptance Tests (SAT).



شکل (۶) پیاده سازی L-RRT در فضای کاری مدنظر.



شکل (۵) ساختار انجام آزمون پردازنده در حلقه از طریق xPC-Target

۴- نتایج آزمون پردازنده در حلقه

به منظور ارزیابی عملکرد الگوریتم طراحی مسیر L-RRT پیشنهادی در اجتناب از برخورد ربات با موانع از پیش نامعلوم، فضای کاری شکل (۶-ا) متشکل از ۶ مانع ثابت از پیش معلوم و ۲ مانع از پیش نامعلوم توسط نرم افزار متلب 2016a طراحی گردید. ماژول طراحی مسیر (PPM) جهت طراحی مسیر اولیه به کمک الگوریتم درخت جستجو سریع تصادفی شروع به گسترش در فضای کاری تدوین شده می نماید. گره های معتبر درخت جستجو با ستاره های مشکی رنگ، گره های نامعتبر با دایره های توخالی آبی رنگ، شاخه های معتبر توسط خطوط نازک ممتد و شاخه های نامعتبر درخت جستجو توسط خط چین های نازک قرمز رنگ در شکل (۶-ب) نمایش داده شده اند. مسیر اولیه طراحی شده با متوسط ۲۵۰ گره در هر بار اجرای برنامه با خط ضخیم مشکی رنگ و پس از اعمال بهینه سازی با خط ضخیم قرمز رنگ در شکل (۶-ب) ترسیم شده است. پس از طراحی مسیر، دو مانع از پیش نامعلوم که با مستطیل های زرد رنگ در شکل (۶-ج) و (۶-د) نشان داده شده اند بر روی مسیر طراحی شده قرار می گیرد. ربات تا زمانی که مسیر بهینه طراحی شده با ماژول تشخیص مانع (ODM) عاری از برخورد تشخیص داده شود آن را رهگیری می نماید.

موقعیت فعلی ربات به نزدیک‌ترین گره تولید شده توسط RRT و ۳-تولید خروجی مناسب جهت اعمال به کنترل‌کننده سطح پایین. مدت زمان مورد نیاز جهت انجام روی خط این ۳ پروسه در L-RRRT برابر با ۵ تا ۱۰ درصد کوچکترین ثابت زمانی ربات زیرسطحی خودمختار بدون سرنشین است.

با توجه به این نکته که کوچکترین ثابت زمانی ربات مد نظر برابر ۵۰ میلی ثانیه است و طراحی مسیر جهت اجتناب از برخورد با هر یک از دو مانع از پیش نامعلوم معرفی شده در شکل (c-۶) کمتر از ۱۰ میلی ثانیه زمان نیاز خواهد داشت.

جدول (۱) مقایسه نتایج الگوریتم L-RRRT پیشنهادی با G-RRT, RRT و FG-RRT پیشنهاد شده در مقاله [۲].

الگوریتم طراحی	کل فضای پیکربندی	تعداد از عبور از مانع از پیش گره درخت	قابلیت عبور از مانع از پیش گره درخت نامعلوم	زمان
				اجراء
Conventional RRT	CCS	۸۸۴۹	ندارد	۱۹,۳
	NPCS	Max	ندارد	INF
Greedy RRT	CCS	۹۸۳	ندارد	۱,۸۷
	NPCS	۱۳۸۲	ندارد	۳,۴۸
FG-RRT	CCS	۱۵۱	ندارد	۰,۶
	NPCS	۱۱۳۸	ندارد	۲,۵۳
L-RRT	CCS	۱۰۴۲	دارد	۲,۰۱
	NPCS	۱۵۲۱	دارد	۳,۶۵



شکل (۷) a-فضای پیکربندی بهم‌ریخته Cluttered Configuration Space (CCS). b-گذرگاه‌های باریک Narrow Passages Configuration Space (NPCS).

چنانچه سنسور ربات که به صورت شماتیک با مثلث نارنجی رنگ در جلو ربات در شکل (c-۶) و شکل (d-۶) نشان داده شده، احتمال برخورد با مانع از پیش نامعلوم را تشخیص دهد، ربات به منظور اجتناب از آن رفتار واکنشی از خود نشان خواهد داد. برای این منظور، ماژول طراحی مسیر محلی بلادرنگ (LRPPM) درخت جستجوی سریع تصادفی را از موقعیت فعلی ربات به نزدیک‌ترین نقطه مسیر توسعه می‌دهد. گره‌های درخت جستجوی محلی توسعه داده شده جهت عبور از مانع از پیش نامعلوم ۱ با ستاره‌های مشکی رنگ، شاخه‌های آن با خطوط نازک ممتد مشکی، مسیر اولیه طراحی شده با خط مشکی ضخیم و مسیر بهینه محلی با خط ضخیم سبز رنگ در شکل (c-۶) نمایش داده شده‌اند. جهت عبور ربات از مانع از پیش نامعلوم ۲، درخت جستجو محلی در شکل (d-۶) گسترش می‌یابد. مسیر اولیه طراحی شده توسط RRT متداول با خط‌چین آبی رنگ و مسیر طراحی شده توسط L-RRRT با قابلیت اجتناب از برخورد با مانع از پیش نامعلوم بصورت با خط ممتد قرمز رنگ در شکل (e-۶) ترسیم شده است. جهت مقایسه نتایج الگوریتم طراحی مسیر پیشنهادی در این تحقیق با نتایج الگوریتم طراحی مسیر FG-RRT طراحی شده در مقاله [۲]، دو فضاهای پیکربندی مورد استفاده در آن مقاله بار دیگر در اینجا شبیه‌سازی شد، شکل (۷). نتایج مقایسه عملکرد الگوریتم طراحی مسیر درخت جستجوی سریع تصادفی محلی (L-RRT) پیشنهاد شده در این مقاله با سه الگوریتم RRT استاندارد، RRT حریم (G-RRT) و RRT حریم فازی (FG-RRT) در جدول (۱) آورده شده است.

از جدول (۱) مشاهده می‌شود که، L-RRRT پیشنهاد شده در این مقاله از منظر حجم محاسبات مورد نیاز و زمان اجرا از RRT استاندارد و RRT حریم (G-RRT) عملکرد بهتری داشته است. بعنوان مثال در فضای پیکربندی بهم ریخته (CCS) معرفی شده در شکل (a-۷) الگوریتم RRT با استفاده از ۸۸۴۹ گره در ۱۹,۳ ثانیه موفق به طراحی مسیر اولیه شده است، در حالی که L-RRRT پیشنهادی با استفاده از ۱۰۴۲ گره در ۲,۰۱ ثانیه مسیر اولیه را طراحی نموده است. ویژگی بارز دیگر L-RRRT پیشنهادی نسبت به FG-RRT قابلیت عبور از مانع از پیش نامعلوم است. L-RRT در زمان عبور از هر مانع از پیش نامعلوم پروسه تقریباً یکسانی را انجام می‌دهد: ۱- خواندن اطلاعات از واحد ODM، ۲- انجام محاسبات لازم جهت رشد درخت از

۶- پیوست

۶-۱- برخی از ضرایب مدل دینامیکی AUV مد نظر

توضیحات	واحد	مقدار	پارامتر
مرکز جرم در راستای محور طولی	mm	0.0	X_g
مرکز جرم در راستای محور عرضی	mm	0.0	Y_g
مرکز جرم در راستای محور عمودی	mm	40	Z_g
مرکز شناوری در راستای محور طولی	mm	0.0	X_b
مرکز شناوری در راستای محور عرضی	mm	0.0	Y_b
مرکز شناوری در راستای محور عمودی	mm	0.0	Z_b
جرم ربات	Kg	31	m
ممان اینرسی	$Kg.m^2$	35.82	I_{XX}
ممان اینرسی	$Kg.m^2$	43280	I_{YY}
ممان اینرسی	$Kg.m^2$	43280	I_{ZZ}
سرعت جلوروی (سرچ)	m/sec	1.534	u
تراست پروانه	N	3.85	X_{prop}

۶-۲- قیدهای دینامیکی AUV مد نظر

محدوده	قید
$\pm 20 \text{deg}$	حداقل و حداکثر زاویه پیچ مطلوب AUV
$\pm 3 \text{deg/sec}$	نرخ تغییر زاویه پیچ AUV
$\pm 60 \text{deg}$	حداقل و حداکثر زاویه سمت مطلوب AUV
$\pm 3 \text{deg/sec}$	نرخ تغییر زاویه سمت AUV
$\pm 20 \text{deg}$	محدوده تغییر زاویه سطوح کنترلی عمودی
$\pm 20 \text{deg}$	محدوده تغییر زاویه سطوح کنترلی افقی

۷- مراجع

- [1] Fletcher, B. UUV Master Plan: A Vision for Navy UUV Development. in OCEANS MTS/IEEE Conference and Exhibition, 2000.
- [2] Taheri, E., et al., "Fuzzy Greedy RRT Path Planning Algorithm in a Complex Configuration Space", Vol.16, No.6, 2018.
- [3] Li, B., "Rendezvous Planning for Multiple AUVs With Mobile Charging Stations in Dynamic Currents", Vol.4, No.2, pp.1653-1660, 2019.
- [4] MahmoudZadeh, S., "Efficient AUV Path Planning in Time-Variant Underwater Environment Using Differential Evolution Algorithm", Vol.17, No.4, pp.585-591, 2018.
- [5] MahmoudZadeh, S., "Online Path Planning for AUV Rendezvous in Dynamic Cluttered Undersea Environment Using Evolutionary Algorithms". Applied Soft Computing, Vol.70, pp.929-945, 2018.
- [6] Cui, R., Y. Li, and W. Yan, "Mutual Information-based Multi-AUV Path Planning for Scalar Field Sampling using Multidimensional RRT". IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol.46, No.7, pp.993-1004, 2016.
- [7] Zhuang, Y., "Efficient Collision-Free Path Planning for Autonomous underwater Vehicles in Dynamic Environments with a Hybrid

لذا به کمک الگوریتم طراحی مسیر پیشنهادی امکان عبور از موانع از پیش نامعلوم برای ربات فراهم می‌گردد.

۵- نتیجه‌گیری

در این مقاله الگوریتم طراحی مسیر L-RRT به منظور طراحی مسیر قابل اجرا توسط AUV در یک فضای کاری تقریباً ناشناخته ارائه گردید. الگوریتم مذکور در کلاس الگوریتم‌های تصادفی مبتنی بر نمونه‌برداری افزایش است که از سه جزء به هم پیوسته: ماژول طراحی مسیر، ماژول طراحی مسیر محلی بلادرنگ و ماژول تشخیص مانع تشکیل می‌گردد. هریک از گره‌ها و شاخه‌های درخت جستجو طراحی شده توسط L-RRT ابتدا از منظر قیدهای سینماتیکی و دینامیکی (سینودینامیکی) ربات بررسی شده و در صورتی که برای ربات قابل اجرا باشند به مجموعه گره‌ها و شاخه‌های درخت افزوده می‌شود. به عبارت دیگر برخلاف عمده روش‌های طراحی مسیر مبتنی بر درخت‌های جستجو که صرفاً در آنها قیدهای سینماتیکی لحاظ می‌گردد، در L-RRT پیشنهاد شده در این تحقیق قیدهای سینودینامیکی با استفاده از مدل غیرخطی شش درجه آزادی و کنترل‌کننده سطح پایین لحاظ شده و لذا قابل رهگیری بودن هریک از گره‌های درخت و در نهایت مسیر طراحی شده برای ربات تضمین می‌گردد.

الگوریتم طراحی مسیر مذکور بر روی SBC مدل Axiomtek 84710 مورد استفاده در AUV از طریق xPC-Target پیاده‌سازی و مورد آزمون PIL قرار گرفت. نتایج نشان می‌دهد که هر دو روش RRT و L-RRT با بهره‌گیری از ماهیت تصادفی خود قادر به طراحی مسیر سریع برای AUV می‌باشند. مسیر طراحی شده به روش RRT به دلیل ملاحظه موانع صرفاً از پیش معلوم و ملاحظه صرفاً قیدهای سینماتیکی توسط AUV در یک محیط کاری ناشناخته قابل اجرا نمی‌باشد و همچنین قابل رهگیری بودن مسیر طراحی شده تضمین نمی‌گردد. این در حالی است که، مسیر طراحی شده برای AUV به روش L-RRT پیشنهادی قادر به طراحی مسیر مجدد محلی و همچنین قادر به اعمال همزمان قیدهای سینماتیکی و دینامیکی در فرایند طراحی مسیر بوده و لذا اجتناب از برخورد با موانع از پیش نامعلوم و قابل رهگیری بودن مسیر توسط ربات تضمین شده است.

- [20] Siciliano, B. and O. Khatib, Springer handbook of Robotics, Springer, 2016.
- [21] LaValle, S.M., Rapidly-Exploring Random Trees: A New Tool for Path Planning, 1998.
- [22] Hsu, D., J.-C. Latombe, and R. Motwani. Path Planning in Expansive Configuration Spaces. in Robotics and Automation, IEEE International Conference, pp.1-7, 1997.
- [۲۳] ا. علیزاده، م. ت. ثابت، م. ز. لاریمی، ح. نورمحمدی. شناسایی مدل دینامیکی یک رونده زیرآبی خودگردان با تخمین ضرایب هیدرودینامیکی، نشریه علمی - پژوهشی دریافنون، دوره ۵، شماره ۴ - شماره پیاپی ۱۴، زمستان ۱۳۹۷، صفحه ۵۰-۵۸.
- [24] Fossen, T.I., "Guidance and Control of Ocean Vehicles". Vol.199. Wiley New York, 1994.
- [25] Fossen, T.I., "Marine Control System-Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles". Marine Cybernetics, 2002.
- [۲۶] ی. عبدی، ج. قاسمی، کنترل عمق زیردریایی بدون سرنشین با استفاده از کنترل کننده مد لغزشی عصبی، نشریه علمی - پژوهشی دریافنون، دوره ۵، شماره ۲ - شماره پیاپی ۸۶، تابستان ۱۳۹۷، صفحه ۱-۱۳.
- [27] Vuilmet, C. High Order Sliding Mode Control Applied to a Heavyweight Torpedo". in Control Applications, Proceedings of IEEE Conference, pp.1-6, 2005.
- [28] Sarkar, M., "Modelling and Simulation of a Robust Energy Efficient AUV Controller. Mathematics and Computers in Simulation, Vol.121, pp.34-47, 2016.
- [29] Jun, B.-H., Development of the AUV 'ISiMI' and a Free Running Test in an Ocean Engineering Basin". Ocean Engineering, Vol.36, No.1, pp.2-14, 2009.
- Optimization Algorithm". Ocean Engineering, Vol.127, pp.190-199. 2016.
- [8] Zhou, Y. and R. Wang, "An Improved Flower Pollination Algorithm for Optimal Unmanned Undersea Vehicle Path Planning Problem". International Journal of Pattern Recognition and Artificial Intelligence, Vol.30, No.4, pp.1659010, 2016.
- [9] Zhang, L., "Three-Dimensional Underwater Path Planning based on Modified Wolf Pack Algorithm. IEEE Access, Vol.5, pp.22783-22795. 2017
- [10] Noreen, I., "Optimal Path Planning in Cluttered Environment using RRT*-AB". Intelligent Service Robotics, Vol.11, No.1, pp.41-52, 2018.
- [11] Candeloro, M., A.M. Lekkas, and A.J. Sørensen, "A Voronoi-diagram-based Dynamic Path-Planning System for Underactuated Marine Vessels". Control Engineering Practice, Vol.61, pp.41-54, 2017.
- [12] Hernandez, E., M. Carreras, and P. Ridao, "A Comparison of Homotopic Path Planning Algorithms for Robotic Applications". Robotics and Autonomous Systems, Vol.64, pp.44-58, 2015.
- [13] Novinzadeh, B., A. Reza, and M. Asadi Matak, "Design of Stable Nonlinear Guidance of an Underwater Vehicle in the Ship Wake via Estimated Path by Particle Filter". Modares Mechanical Engineering, Vol.17, No.5, pp.260-266, 2017.
- [14] Tabataba'i-Nasab, F.S. and A. Keymasi Khalaji, "Potentially Directed Robust Control of an Underwater Robot in the Presence of Obstacles". Modares Mechanical Engineering, Vol.18, No.3, pp.1-8, 2018.
- [15] Donald, B., "Kinodynamic Motion Planning". Journal of the ACM (JACM), Vol.40, No.5, pp. 1048-1066, 1993.
- [16] Paden, B., V. Varricchio, and E. Frazzoli, "Verification and Synthesis of Admissible Heuristics for Kinodynamic Motion Planning". IEEE Robotics and Automation Letters, Vol.2, No.2, pp.648-655, 2017.
- [17] Hauser, K. and Zhou, Y., "Asymptotically Optimal Planning by Feasible Kinodynamic Planning in a State-Cost Space". IEEE Transactions on Robotics, Vol.32, No.6, pp.1431-1443, 2016.
- [18] Kim, C.H. and S. Sugano, "Closed Loop Trajectory Optimization based on Reverse Time Tree". International Journal of Control, Automation and Systems, Vol.14, No.6, pp.1404-1412, 2016.
- [19] Svestka, P., J. Latombe, and L. Overmars Kavraki, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces". IEEE Transactions on Robotics and Automation, Vol.12, No.4, pp.566-580, 1996.