INTERNATIONAL JOURNAL OF MARITIME TECHNOLOGY
IJMT Vol.15/ Winter 2021 (93-105)
Available online at: www. ijmt.ir

AUV Path Planning in Dynamic Cluttered Environment through the Randomized Kinodynamic Sampling-based method

Ehsan Taheri[1*], Ali Adeli[2]

[1] *PhD of Control Engineering, Electrical Engineering Department, Malek Ashtar University of Technology, 15875-1774, Tehran, Iran; taheri.ehsan@mut-es.ac.ir.*
[2] *M.Sc. of Mechanical Engineering, Department of Mechanical Engineering, Sharif University of Technology, 11155-9567, Tehran, Iran; aliadeli198222@gmail.com.*

ARTICLE INFO

Article History:
Received: 12 Jun. 2021
Accepted: 12 Oct. 2021

Keywords:
Autonomous underwater vehicle
Kinodynamic constraints
Sampling-based path planning
Collision avoidance

ABSTRACT

Considering both kinematic and dynamic constraints (kinodynamic constraints) of an autonomous underwater vehicle in a Kinodynamic path planning algorithm in a dynamic large-scale workspace is an NP-Hard problem. Computational and time complexity of the kinodynamic path planning problem increase in the order O (n^2) by increasing numbers of moving obstacles, AUV Kinodynamic constraints, degrees of freedoms, and workspace dimensions. This paper proposes a Randomized Kinodynamic Sub-optimal Planning (RKSP) algorithm for a man-portable class AUV. The proposed algorithm solves the path planning problem by applying a randomized sampling-based method to exploring and expanding in the workspace. RKSP re-plans the path to avoid collision with moving obstacles in a cluttered environment through a behavior-based method. RKSP consists of three main components that tightly coupled together. The first component is a Randomized kinodynamic Planning (RKP) module that generates the random offspring waypoints and plans a feasible path by considering the AUV kinodynamic constraints. The second component is a Numerical Path Optimization (NPO) module that prunes the inappropriate edges of the path and reduces the computational complexity. The third component is a Local-Reactive kinodynamic (LRK) module that re-plans the local path through the neighborhood waypoints to avoid collision with moving obstacles in an unknown environment. RKSP path planning method is evaluated through the three different scenarios in a narrow passage, maze-like space and complex space. Results demonstrate the planned path by the proposed method is feasible and the AUV tracks the path appropriately and avoids collision with moving obstacles. Also, the total numbers of waypoints reduce in comparison to the conventional randomized methods and the planned path is near to the optimal.

1. Introduction

According to the last roadmap in the Unmanned underwater vehicles (UUV) field, Autonomous underwater vehicles (AUV) have been received many attentions to carry out specific mission that no other vehicles can perform [1].

However, applying AUVs in long-term marine applications have several challenges due to the technology readiness level. Autonomy is one of the main challenges in this field. AUV should be able to plan a feasible and near to optimal path to perform specific function. Path planning in a dynamic large-scale environment is an NP-Hard (Non-deterministic polynomial-time hardness) problem for these autonomous vehicles, over the past two decades. Computational and time complexity of path planning problem increases in the order O(n2) by increasing numbers of moving obstacles, AUV Kinodynamic constraints, non-holonomic constraints, degrees of freedoms, and workspace dimensions. There are various categories of path planning algorithms. First category is graph search algorithms that have two types: non-heuristic and heuristic. Non-heuristic types only consider cost to start point [1]. Whereas cost to goal point (in addition to cost to start point) is used in heuristic types and they need many computational time (e.g. A*, SMA*, IDA*, KA*, L*, FM, AP Theta*, and D* [2-5]. Detailed literature reviews were

[Downloaded from ijmt.ir on 2026-09-10]

accomplished to explain the current state of the art of path planning and obstacles avoidance for the AUV in [6, 7]. The path planning algorithm based on behavioral decision-making for the AUV is proposed in [8]. The evolution approach is used in this research to the optimization of energy. The global optimal path planning algorithm is introduced in [9] based on the water wave optimization theory. Performance of this algorithm is increased through the elite opposition-based learning method. Second category of path planning algorithms is evolutionary algorithms that they are inconsistent and incompleteness (e.g. PSO, ACO, GA, MA, SFLA, and PSO-LPM [2-4]). There are other evolutionary algorithms that are trapped in local optimums (ABC, BFO, BINN, and FFA [5]). Third category of path planning algorithms is randomized algorithms that they have two main types: Probabilistic roadmaps (PRM) and Rapidly-exploring Random Tree (RRT) [6]. Also, another extensions of these algorithms are exist (e.g. RRT*, SRRT*, LBT-RRT, RRTX, FG-RR, Multi-RRT, and skilled-RRT [7-13]) that they improve main algorithms. RRT and PRM algorithms are better in online path planning and highly structured static environments, respectively [5]. Moreover, the tree-based planners same as RRT best can overcome kinodynamic constraints [14]. Kinodynamic constraints are various dependent to used robot and may be simplified in the motion planning. Donald and et al. [15] only considered bounds of velocity and acceleration as constraints. Dubin's car kinematic and double integral dynamic are used in a sample-based algorithm in [16]. Kinodynamic constraints of a two-wheeled robot is considered using RRT algorithm [17]. If feasible path from the nearest node of the tree to the random point applying kinodynamic constraints doesn't be possible, other node of the tree is tried. Double integral dynamic of a car-like is also used in a sample-based method [18]. Bordalba applies dynamic constraints of cable-suspended parallel robots to connection of the tree and generated random point [19]. The extend function, in growing the tree, achieves kinodynamic motion planning. Pham [20] tries the K-nearest nodes of the tree to connect the tree and generated random point applying dynamic of a manipulator. Closed loop method as the extend function is used considering kinematic constraints of a wheeled robot in [21]. Spline and B-spline fourth-order are used as steering in expansion of RRT* method in [22] and [23], respectively. Bera [24] applies control inputs to dynamic model in kinodynamic planning. These control inputs can be selected base of optimality performance. Two methods are presented for steering in [25] that include designing of controller, to connect the nearest node of the tree and random point, and trying a number of admissible controllers. The control input randomly is sampled and the tree is grown integrating forward dynamic in a short time [26, 27]. A kinodynamic motion planner sample-based method KPIECE is used in [28] designed for systems with complex dynamic.

In this paper, Randomized Kinodynamic Sub-optimal Planning (RKSP) in Unknown and Dynamic environment is proposed. RKSP consists of three main components. The first component is RKP that a kinodynamic path plans considering kinematic and dynamic constraints of AUV. RKSP plans a kinodynamic path through the randomized sampling-based method. The second component is a NPO that prunes the inappropriate edges of path and reduces the computational complexity. The third component is a LRK that re-plans the local path through the neighborhood waypoints to avoid collision with moving obstacles in an unknown environment. The main contributions of this article can be summarized as follows:

- In the proposed Randomized Kinodynamic Sub-optimal Planning method, the reachability of each offspring vertex is evaluated through the low-level controller for the AUV, hence, both kinematic and dynamic (kinodynamic) constraints are applied in the planned path and the designed path is tracked through the AUV appropriately. It is notable that in the conventional randomized sampling-based path planning method only kinematic constraints are considered in the problem.
- Computational and time complexity of the kinodynamic path planning problem increase in the order O(n2) by increasing numbers of moving obstacles. In the proposed path planning algorithm competency of each offspring vertex is assessed through the sub-optimal module, and inappropriate vertices are eliminated. Hence, the total number of vertices and also the complexity of the proposed path planning method is reduced. So the proposed method is implemented in a real-time manner and AUV re-plans the path to avoided collision with moving obstacles.

The structure of this paper as follows: a brief background of randomized sampling-based path planning algorithm and AUV dynamic model is presented in Section 2. The proposed Randomized Kinodynamic Sub-optimal Planning (RKSP) algorithm consists of three main components that are presented in Section 3. These components are: 1-Randomized Kinodynamic Planning (RKP) module, 2-Numerical Path Optimization (NPO) module, and 3-Local-Reactive kinodynamic (LRK) module. Performance and effectiveness of the proposed path planning algorithm are evaluated through the three different

[Downloaded from ijmt.ir on 2026-09-10]

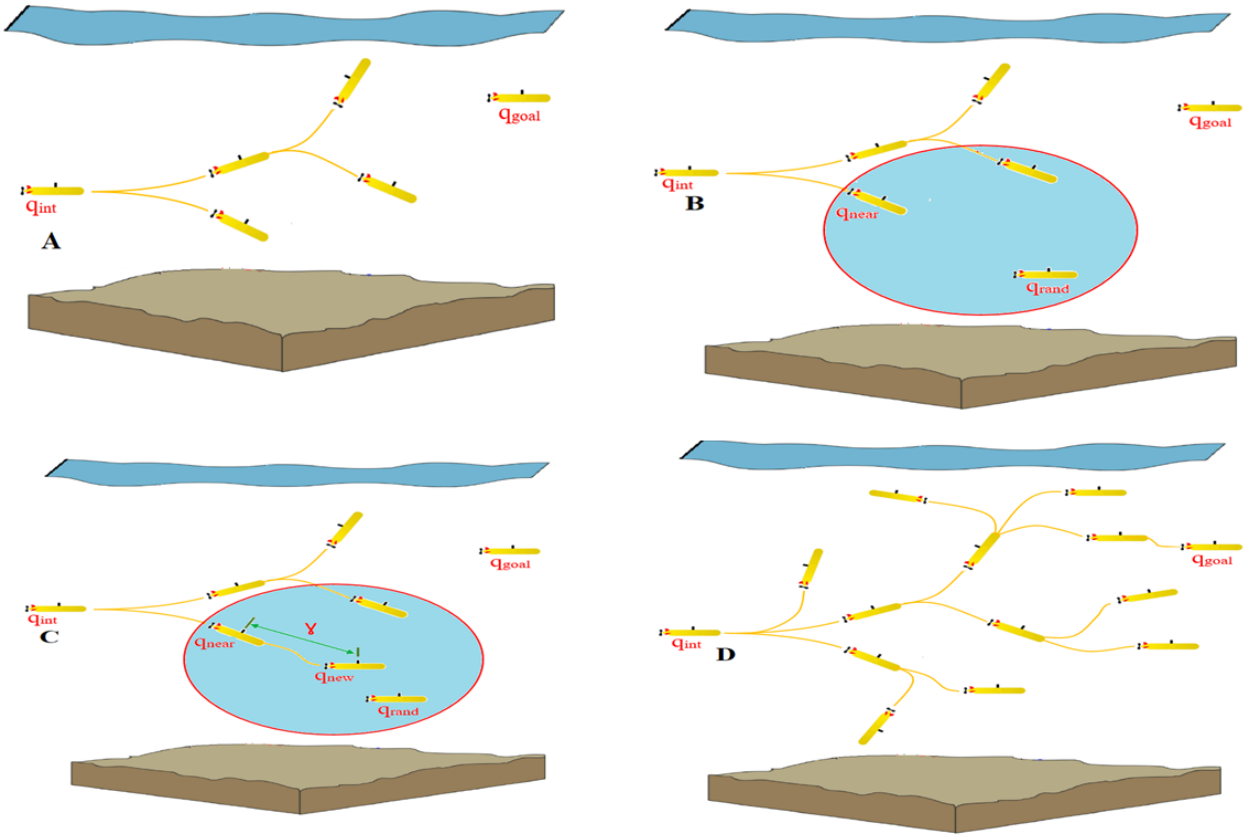

Figure 1. The sampling-based path planning algorithm concept is illustrated in the configuration space, Fig. A. New random offspring vertex is generated, and the nearest vertices are selected to evaluate as a parent vertex, Fig. B. Parent vertex is determined through the metric function and the new vertex is added to the vertices set, Fig. C. Random offspring vertices and related branch are expanded in the configuration space, Fig. D.

scenarios in Section 4, and finally the manuscript is concluded in Section 5.

2. Preliminary

In this article randomized sampling-based algorithm and autonomous underwater vehicle are considered as path planning method and robot, respectively. Briefly these two issue are introduced in the following subsections.

2.1. RRT Randomized Planning

RRT algorithm is a randomized and tree-based planner in which a tree $\tau(\square, E)$ is made in the configuration space χ with set of nodes and set of edges $\square$ and E, respectively. The tree is made using random generating of nodes and if exist a solution and enough samples, the tree includes the path from the start to the goal. Most popular random number generator algorithm is Mersenne Twister [29] used in software packages. The recursive relation of this random generator is as follows [30].

$$x_{k+n} = x_{k+m} \oplus (x_k^u \mid x_{k+1}^l) A, \quad k = 0, 1, \ldots \qquad (1)$$

Where integer n is degree of recurrence, integer m that $1 \le m \le n$, a constant $a \times a$ matrix A, x_k^u upper $a-b$ bits of x_k , x_{k+1}^l lower b bits of x_{k+1} , b an integer $0 \le b \le a-1$, and $\oplus$ is bitwise addition. In relation (1), x_n can be generated using initial seeds $x_0, x_1, \ldots, x_{n-1}$. Each random node $q_{rand} \in \chi$ is generated, using random number generator, and is added to the tree if distance of the tree and the random node q_{rand} be less than the step size Δq . If distance of the tree and the random node q_{rand} be larger than the step size Δq , a new node q_{new} is created using steering function with corresponding distance of the step size Δq to the tree. The nearest node $q_{nearest}$ of the tree to the random node is found using a metric function. The metric function is as follows.

$$\gamma = (\sum_{i=1}^{n} \left| q_i - q_j \right|^{\lambda})^{1/\lambda} \qquad (2)$$

That $\lambda = 1$ and $\lambda = 2$ give Manhattan and Euclidian distance, respectively. Because of indirect path, Manhattan distance may be proposed in kinodynamic constrained problems. However, in randomized

[Downloaded from ijmt.ir on 2026-09-10]

methods (e.g. RRT) robot heading anticipation is not possible. Moreover, numerical path optimization changes heading of robot in various section of path. So, Euclidian distance is simple metric function in randomized methods.

Remark 1: The conventional RRT path planning algorithm consists of five main components: 1) metric, 2) random sampling, 3)steer branch, 4) nearest neighbors and 5) collision detection functions [20]. The concept of these five functions is shown in Figure 1.

2.2. AUV Dynamic

In this paper, a man-portable class AUV is considered that its main parameters are tabulated in Table 1. Behaviors of this AUV in a 3D space are described through a six-DoF dynamic model and two coordinate systems (body and inertia) which are illustrated in Figure 2, [14]. Relation between these two coordinate systems are described through the following transformation matrix, [31, 32].

$$\dot{\eta} = J(\eta)V \leftrightarrow \begin{bmatrix}\dot{\eta}_1\\ \dot{\eta}_2\end{bmatrix} = \begin{bmatrix}J_1(\eta_2) & 0_{3\times3}\\ 0_{3\times3} & J_2(\eta_2)\end{bmatrix}\begin{bmatrix}\nu_1\\ \nu_2\end{bmatrix},$$
$$\eta = [\eta_1^T, \eta_2^T]^T,\ \eta_1 = [x\ y\ z]^T,\ \eta_2 = [\phi\ \theta\ \psi]^T, \quad (3)$$
$$V = [\nu_1^T, \nu_2^T]^T,\ \nu_1 = [u\ v\ w]^T,\ \nu_2 = [p\ q\ r]^T$$

Where, $\eta_1 \& \eta_2 \in R^3$ presents the position and orientation

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = J_1(\eta_2)\begin{bmatrix}u\\ v\\ w\end{bmatrix},$$
$$J_1(\eta_2) = \begin{bmatrix}\cos\psi\cos\theta & -\sin\psi\cos\phi+\cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi+\cos\psi\sin\theta\cos\phi\\ \sin\psi\cos\theta & \cos\psi\cos\phi+\sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi+\sin\psi\sin\theta\cos\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi\end{bmatrix} \quad (4)$$

Table 1. Main parameters that are considered for the AUV.

Parameter	Value	Unit
[m,L]	[31,1.65]	$[Kg, m]$
$[X_g, Y_g, Z_g]$	[0.0, 0.0, 25]	mm
$[X_b, Y_b, Z_b]$	[0.0, 0.0, 0.0]	mm
$[I_{XX}, I_{YY}, I_{ZZ}]$	[4.232, 29.452, 29.452]	$kg.m^2$
$[X_{prop}, K_{prop}]$	[5.12, 0]	$[N, N.m]$
Rate of the pitch and yaw angle change	±3	deg/ sec
Max/min command pitch angle	±12	deg
Max/min command yaw angle	±30	deg
Surge speed	3	knot
Rudder and Elevator surface deflection	±15	deg

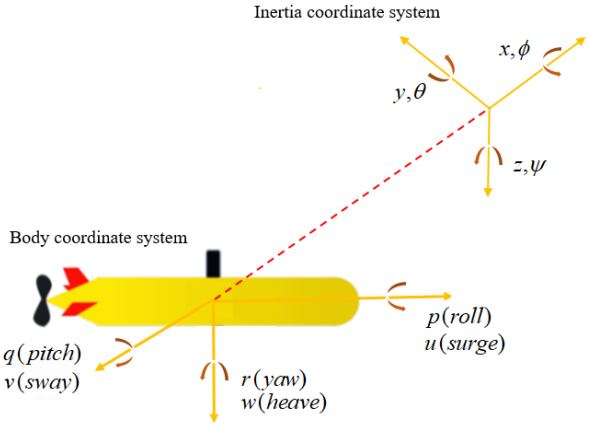

Figure 2. Body and inertia coordinate systems.

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = J_2(\eta_2)\begin{bmatrix}p\\ q\\ r\end{bmatrix},$$
$$J_2(\eta_2) = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta\end{bmatrix} \quad (5)$$

of the AUV in the inertia coordinate system, respectively. Vectors $\nu_1 \& \nu_2 \in R^3$ presents the linear and angular velocities in the body coordinate system. $J_1(\eta_2)$ explains the connection between the position and linear velocities and $J_2(\eta_2)$ describes relation between the rate of changes in orientation angles and angular velocities.

Remark 2: We will stand the following assumptions in the AUV dynamic model. AUV is symmetric about the x–z and x-y planes. Also, the pitch angle of the AUV is rarelly greater or less than ±45° in practical terms.

Differential equations of AUV is as following:

$$M\dot{V} + C(V)V + D(V)V + g = \tau_{PG} \quad (6)$$

Where $V = [u\ \ v\ \ w\ \ p\ \ q\ \ r]^T$ is a state vector that includes translational and rotational speeds that are explained before. Global mass matrix M is:

$$M = \begin{bmatrix}m-X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g\\ 0 & m-Y_{\dot{v}} & 0 & -mz_g & 0 & -mx_g-Y_{\dot{r}}\\ 0 & 0 & m-Z_{\dot{w}} & my_g & -mx_g-Z_{\dot{q}} & 0\\ 0 & -mz_g & my_g & I_{xx}-k_{\dot{p}} & 0 & 0\\ mz_g & 0 & -mx_g-M & 0 & I_{yy}-M_{\dot{q}} & 0\\ -my_g & mx_g-N_{\dot{v}} & 0 & 0 & 0 & I_{zz}-N_{\dot{r}}\end{bmatrix} \quad (7)$$

Coriolis and centripetal matrix C is:

[Downloaded from ijmt.ir on 2026-09-10]

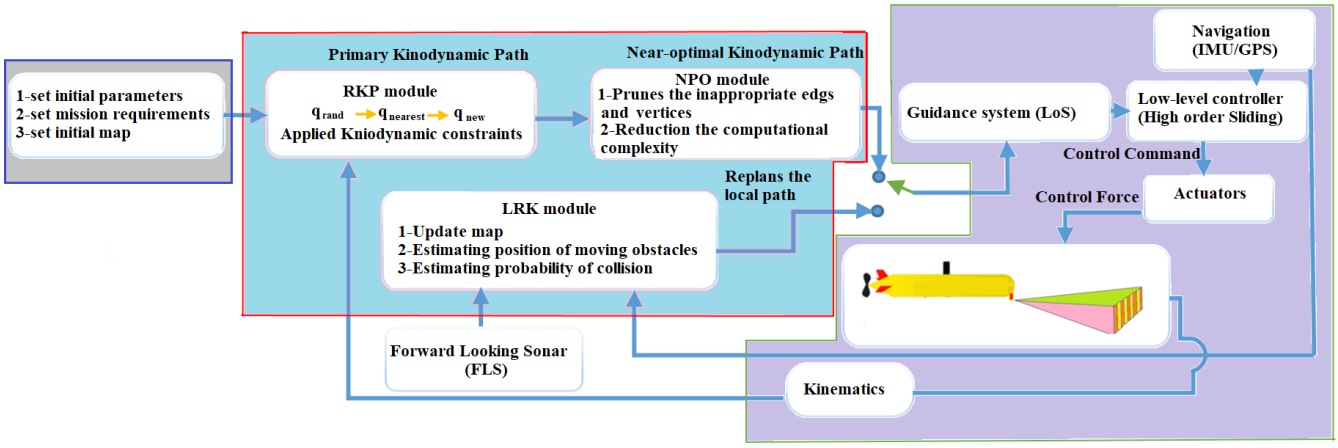

Figure 3. The schematic block diagram of the proposed Randomized Kinodynamic Sub-optimal Planning (RKSP) algorithm.

$$C=\begin{bmatrix} 0 & -mr & mq \\ mr & 0 & -mp \\ -mq & mp & 0 \\ -m(y_g q + z_g r) & my_g p & mz_g p \\ mx_g q & mx_g q & mz_g q \\ mx_g r & my_g r & -m(x_g p + y_g q) \end{bmatrix} \quad \begin{bmatrix} m(y_g q + z_g r) & -mx_g q & -mx_g r \\ -my_g p & m(z_g r + x_g p) & -my_g r \\ -mz_g p & -mz_g q & m(x_g p + y_g q) \\ 0 & I_{zz} r & -I_{yy} q \\ -I_{zz} r & 0 & I_{xx} p \\ I_{yy} q & -I_{xx} p & 0 \end{bmatrix} \tag{8}$$

Six equations of damping term $D(V)V$ are described as follow, [32]:

$$\begin{aligned} X_D &= -(X_{uu} u|u| + X_{wq} wq + X_{qq} q^2 + X_{vr} vr + X_{rr} r^2) \\ Y_D &= -(Y_{vv} v|v| + Y_{rr} r|r| + Y_{uv} uv + Y_{wp} wp + Y_{ur} ur + Y_{pq} pq) \\ Z_D &= -(Z_{ww} w|w| + Z_{qq} q|q| + Z_{uw} uw + Z_{uq} uq + Z_{vp} vp + Z_{rp} rp) \\ K_D &= -K_{pp} p|p| \\ M_D &= -(M_{ww} w|w| + M_{qq} q|q| + M_{rp} rp + M_{uq} uq + M_{uw} uw + M_{vp} vp) \\ N_D &= -(N_{vv} v|v| + N_{rr} r|r| + N_{uv} uv + N_{pq} pq + N_{wp} wp + N_{ur} ur) \end{aligned} \tag{9}$$

Six equations of hydrostatic forces (g) are also as follows:

$$\begin{aligned} X_g &= (W-B)\sin(\theta) \\ Y_g &= -(W-B)\sin(\varphi)\cos(\theta) \\ Z_g &= -(W-B)\cos(\varphi)\cos(\theta) \\ K_g &= (y_g W - y_b B)\cos(\theta)\cos(\varphi) + (z_g W - z_b B)\cos(\theta)\sin(\varphi) \\ M_g &= (z_g W - z_b B)\sin(\theta) + (x_g W - x_b B)\cos(\theta)\cos(\varphi) \\ N_g &= (x_g W - x_b B)\cos(\theta)\sin(\varphi) + (y_g W - y_b B)\sin(\theta) \end{aligned} \tag{10}$$

The main concern here is AUV avoiding collision by moving obstacles through the kinodynamic path planning algorithm. The dynamic model of AUV is described briefly in this subsection and interested readership about the AUV dynamic model are referred to these references, [33, 34]. Propulsion and control forces (τ_{PG}) are:

$$\begin{aligned} X_\tau &= X_{prop} \\ Y_\tau &= Y_{uud_r} u^2 \delta_r \\ Z_\tau &= Z_{uud_s} u^2 \delta_e \\ K_\tau &= K_{prop} + K_{uud_a} u^2 \delta_d \\ M_\tau &= M_{uud_s} u^2 \delta_e \\ N_\tau &= N_{uud_r} u^2 \delta_r \end{aligned} \tag{11}$$

Behavior of this six-DoF nonlinear model in the 3D space is evaluated through the numerical values from Table 1 and Appendix A.

2. Randomized Kinodynamic Sub-optimal Planning

Randomized Kinodynamic Sub-optimal planning (RKSP) includes Randomized Kinodynamic Planning (RKP) module, Numerical Path Optimization (NPO) module and Local-Reactive Kinodynamic (LRK) module. In the RKP, a kinodynamic path is generated through a kinodynamic tree. The NPO optimizes the generated kinodynamic path with the RKP. The LRK re-plans the path through the behavioral-based method to avoid collision with moving obstacles. These three modules are presented in this section. Block diagram of the proposed Randomized Kinodynamic Sub-optimal path planning algorithm is shown in Figure 3.

3.1. Randomized Kinodynamic Planning (RKP) module

RKP is randomized kinodynamic planning in which path is planned using randomized method RRT. In the RRT, steering function expands the tree from the nearest node $q_{nearest}$ to the generated random point q_{rand}. Steering function of a kinodynamic planning consists of kinematic and dynamic constraints of AUV. The AUV with complex kinodynamic constraints lead to a complex steering function. The AUV has six DoF and three actuators (propeller, rudder, and elevator). In this work, steering function carries kinodynamic constraints of the AUV. Steering function must

[Downloaded from ijmt.ir on 2026-09-10]

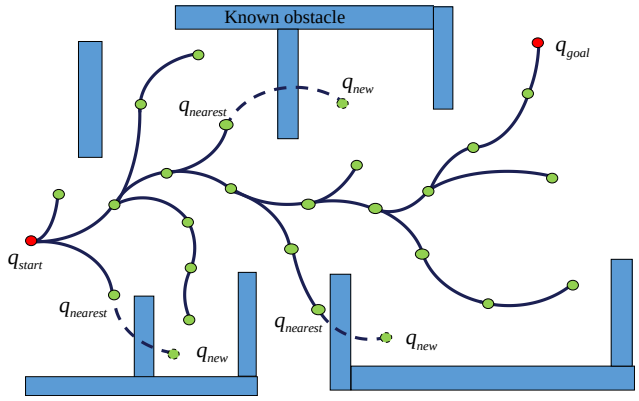

Figure 4. Randomized kinodynamic planning through RKP in known environment

satisfies moving constraints that are tabulated in Table 1 and dynamic equations (6) which can be written in the following form.

$$\dot{q}(t) = f(q(t), u(t))$$

Where $q \in Q \subset R^n$, Q is state space, and $u(t) \in U \subset R^m$ is control action.

Steering function is applied the action $u(t)$ to the equations (12) in order to steer the path from the nearest node $q_{nearest}$ to generated random point q_{rand}. The action $u(t)$ is applied and the node q_{new} is achieved with maximum Euclidean distance Δq from the node $q_{nearest}$. Kinodynamic path from the node $q_{nearest}$ to the node q_{new} is accepted if each generated state of equations 0(12) be collision-free in known environment. The tree of RKP is shown in Figure 4 through which kinodynamic path from the start q_{start} to the goal q_{goal} is obtained.

The dashed curves show collision avoidance in known environment with the blue static obstacles. Pseudocode of kinodynamic RRT is shown in Algorithm 1 in which τ is the tree. First, the start point q_{init} is added to the tree in line 1. Second, random points are generated with $random_state(\chi)$ function from configuration space χ with iteration K (lines 2 and 3). Then, nearest node of the tree to the random point q_{rand} is found through nearest_neighbor(τ, q_{rand}) function (line 4). The Steer($q_{nearest}, q_{rand}, \Delta q$) function generates the curve q_{curve} from the nearest node $q_{nearest}$ to the random point q_{rand} carrying AUV constraints with maximum distance Δq from $q_{nearest}$ (line 5). The new configuration q_{new} is obtained from the end point of the curve q_{curve} in line 6.

Algorithm 1: Kinodynamic RRT(q_{init}, Q_{goal}, χ)

1. $\tau.init(q_{init})$;
2. for k=1,…,K do
3. $q_{rand} \leftarrow random_state(\chi)$;
4. $q_{nearest} \leftarrow$nearest_neighbor(τ, q_{rand});
5. $q_{curve} \leftarrow$Steer($q_{nearest}, q_{rand}, \Delta q$);
6. $q_{new} \leftarrow$ New_config(q_{curve});
7. if CollisionFree(q_{curve}) then
8. $\tau.add_edge(q_{curve})$;
9. $\tau.add_vertex(q_{new})$;
10. if $q_{new} \in Q_{goal}$ then
11. Return Reached;
12. end if
13. end if
14. end for
15. Return(τ);

If the curve q_{curve} be collision-free, this curve and the new configuration are added to the tree as edge and node, respectively (lines 7 to 9). Kinodynamic path is founded if the new configuration reaches to the region Q_{goal} around the goal. The steer function is shown in Algorithm 2. The AUV_Model function generates dynamic model of AUV in line 3. The Control function generates control action *u* corresponding the dynamic model of AUV (line 4). The AUV_Simulator function simulates the dynamic model of AUV using control action *u* and generates the state s_i (line 6). If this state be collision-free, it is added to the curve q_{curve} (lines 7 and 8). This process is continued until the distance d_i between s_i and the nearest node $q_{nearest}$ be less than Δq.

3.2. Numerical Path Optimization (NPO) module

The generated offline kinodynamic path through RKP has many curvatures that can increase computational and time complexity. In a generated path through basic RRT, path can be optimized eliminating redundant nodes (considering obstacles avoidance). Generated path of RKP is formed from many curvature that each waypoint has its heading. In this subsection, two heuristic stage optimization is proposed for kinodynamic paths.

In first stage, three conditions must be prepared to optimize the path in each waypoint. First, arrow shown AUV heading collides with the path. Second, heading

Algorithm 2: Steer($q_{nearest}, q_{rand}, \Delta q$)

[Downloaded from ijmt.ir on 2026-09-10]

1. $d_i \leftarrow 0$;
2. $q_{curve} \leftarrow \{q_{nearest}\}$;
3. $M \leftarrow$ AUV_Model
4. $u \leftarrow$ Control function (M)
5. while $d_i < \Delta q$ do
6. $s_i \leftarrow$ AUV_Simulator(M, u);
7. if CollisionFree(s_i) then
8. $q_{curve} \leftarrow q_{curve} \bigcup \{s_i\}$;
9. $d_i \leftarrow \|q_{nearest} - s_i\|$;
10. end if
11. end while
12. Return (q_{curve});

of collided point be same as waypoint heading. Third, optimized path be collision-free. Corresponding to Figure 5, arrow in waypoint A collides with path in C (first condition) and heading ψ_A is equal with heading ψ_C (second condition). Steering function generates the red dashed kinodynamic path carrying kinodynamic constraints (12). This red dashed path has collision with obstacles, therefore it doesn't have third condition. The waypoints B and E have all three conditions and the blue dashed optimized paths, through steering function, to the waypoints D and F, respectively, are accepted.

In second stage of optimization, kinodynamic curves are generated through steering function from variants points of path to forward points of path with large steps (that is named optimization local goal). If each curve has three conditions, it is replaced with corresponding piece-path of main kinodynamic path. First, heading of optimization local goal and end point of the curve be same. Second, distance of the curve be less than corresponding piece-path of main kinodynamic path. Third, the curve be collision-free. Corresponding to Figure 6, the kinodynamic curve from A to B (the green dashed curve) has three conditions (heading of this curve in B is same as heading of main path, distance of this curve is less than corresponding piece-path of main path, and this curve is collision-free). So, this curve is accepted and replaced with corresponding piece-path of the main path. Also, the kinodynamic curve from C to D has same conditions.

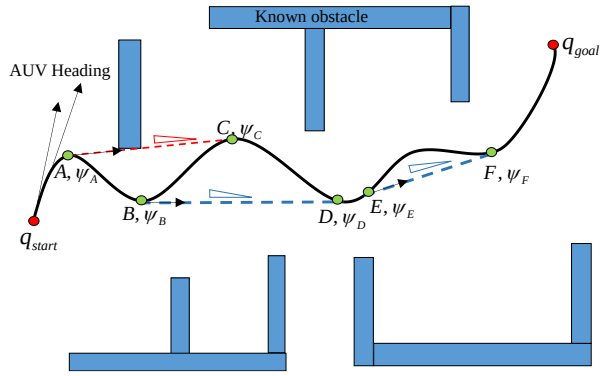

Figure 5. First stage Optimization of the kinodynamic path

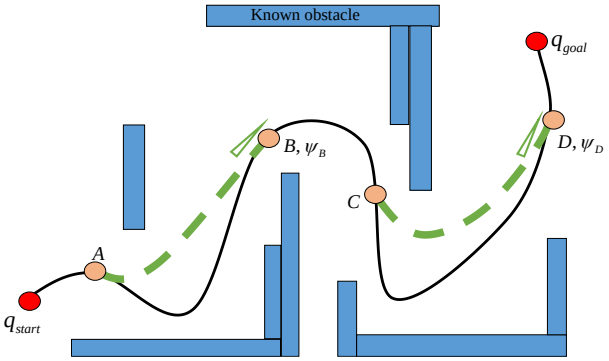

Figure 6. Second stage optimization of the kinodynamic path

3.3. Local-Reactive Kinodynamic Re-planning (LRK) module in Known and Unknown Environment

LRK is Local and Reactive kinodynamic re-planning in Unknown Environment in which kinodynamic re-planning is implemented to avoid static and moving obstacles. Four methods are considered to avoid obstacles in kinodynamic re-planning are called: 1. Local kinodynamic planning, 2. Local tree path extraction, 3. Reactive kinodynamic planning, 4. Reactive-local tree path extraction.

First, collision points CP1 and CP2 is detected and the local start LS and the local goal LG are determined corresponding to Figure 7 Then, the LRUE is called and the local kinodynamic tree is generated. Kinodynamic path (the blue dashed curve) is found from the LS to the LG through the local tree (first method). If kinodynamic path doesn't found, the kinodynamic path is extracted to the frontier local goal LG2 through the local tree path extraction method (the blue curve corresponding to Figure 8). If kinodynamic path cannot be extracted to the LG2, the reactive kinodynamic planning method generates the kinodynamic edge (the blue dashed curve corresponding to Figure 9) and the local kinodynamic

[Downloaded from ijmt.ir on 2026-09-10]

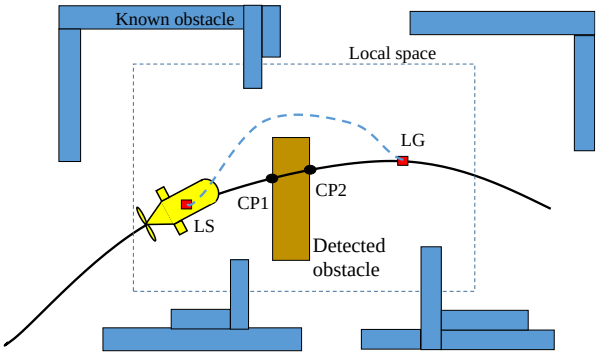

Figure 7. Local kinodynamic path re-planning from the LS to the LG to avoid collision by an unknown static obstacle.

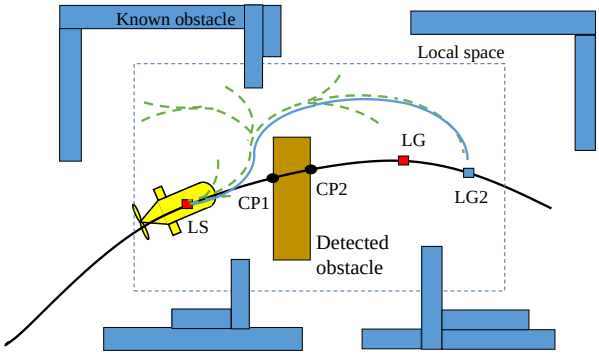

Figure 8. Extending and exploring the random kinodynamic offspring vertices to avoid collision by an unknown static obstacle.

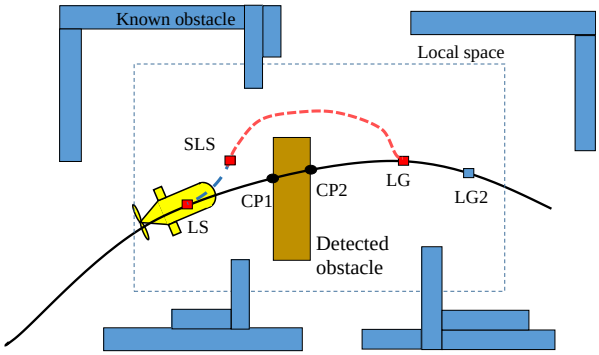

Figure 9. Plan the local path to avoid collision by an unknown static obstacle

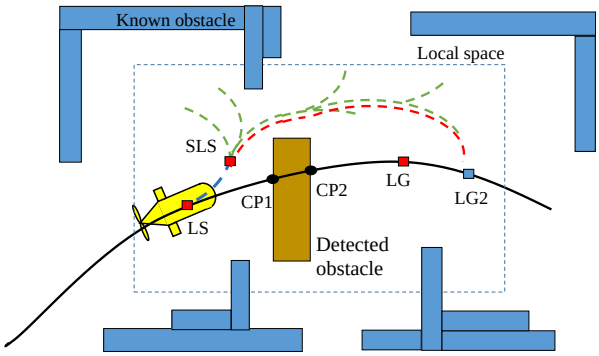

Figure 10. Re-plan the local kinodynamic path to the next local-goal (LG2)

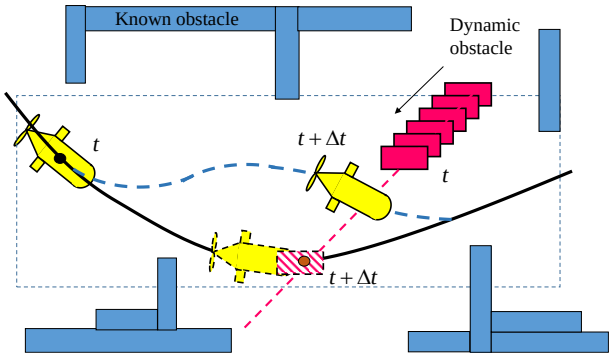

Figure 11. Estimating future collision point with moving obstacle and re-planning the local path through the LRK module

path is generated from the second local start SLS to the LG (the red dashed curve). If the local kinodynamic path doesn't found, the kinodynamic path is extracted to the LG2 through the Reactive-local tree path extraction method with the root SLS (the red dashed curve corresponding to Figure 10).

3.4. Local-Reactive Kinodynamic Re-Planning (LRK) module in Dynamic Environment

As same as previous subsection LRK carries dynamic obstacles. However, a dynamic obstacle can moves to the AUV. Then, LRK first estimates future collisions of kinodynamic path and pathway of the moving obstacle.

If collision time of the AUV and the dynamic obstacle approximately be equal, LRK re-plans. Corresponding to Figure 11 the red rectangular dynamic obstacle moves to the AUV path and future collision is detected in same time $t + \Delta t$. LRK re-planes and the blue dashed kinodynamic path is generated. Figure 3 shows the framework of RRT kinodynamic planning that a kinodynamic path is generated considering known obstacles and optimization stages are applied. In this phase kinodynamic constraints of AUV are carried and generated path are feasible for AUV. This path is executed by AUV and the path is re-planned updating area of static and dynamic obstacles.

4. Simulation Results

To evaluate the path planning abilities of the proposed Randomized Kinodynamic Sub-optimal Planning (RKSP) algorithm in designing a feasible path for the AUV, three scenarios with different complexities are designed through MATLAB R2019a.

[Downloaded from ijmt.ir on 2026-09-10]

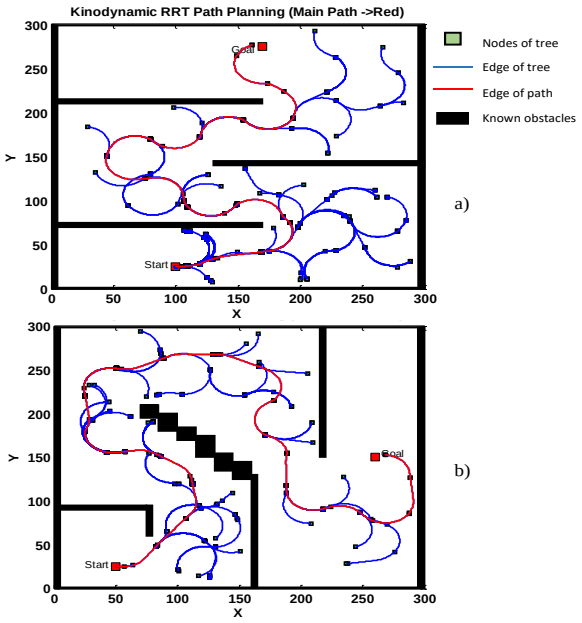

Figure 12. RRT kinodynamic planning in known environment a) narrow passage, b) maze-like space

Scenario-1 (Randomized kinodynamic path planning in a narrow passage and maze-like space with static known environment): In the first scenario the RKSP path planning algorithm is performed for the AUV in a static known environment. In this scenario two type of static known environment with different complexities are considered: 1-narrow passage with 68 static obstacle, and 2-maze-like space with 79 static obstacle.

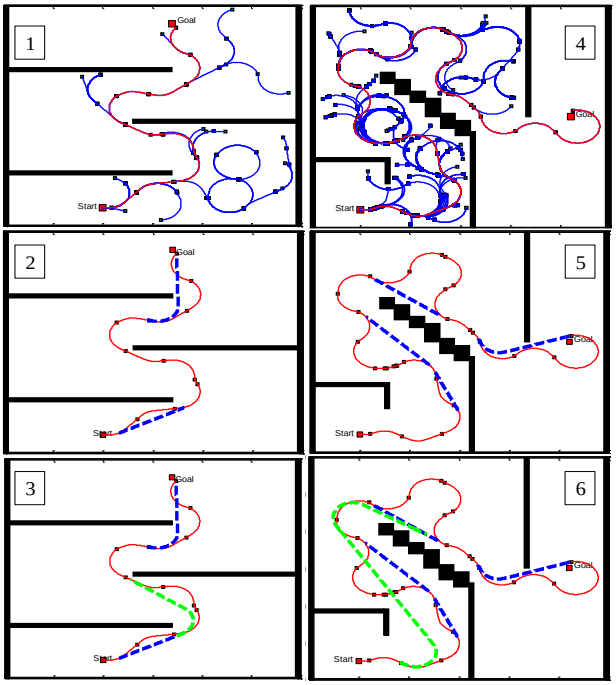

Figure 13. First heuristic stage of kinodynamic path optimization (the blue dashed curves) and second heuristic stage of kinodynamic path optimization (the green dashed curves) in narrow space and maze-like space.

Performance of RKSP algorithm in designing a feasible Kinodynamic path for the AUV in the narrow passage and maze-like space is shown in Figure 12 and tabulated in Table 2. In this scenario the RKP module plans the primary randomized kinodynamic path and the NPO module optimizes the planed path by pruning the inappropriate edges. Hence, the total number of vertices and also the complexity of the proposed RKSP path planning algorithm is reduced.

The primary randomized kinodynamic path and the near-optimal path in the narrow passage and maze-like space are shown in Figure 13 through the red and green line respectively.

Scenario-2 (Randomized kinodynamic path planning in a static unknown environment): In the second scenario the ability of the RKSP path planning algorithm is assessed in a static unknown environment. In this scenario, workspace consist of 94 known and 3 unknown static obstacles. The primary randomized kinodynamic path and the near-optimal path are shown in Figure 14 through the red and blue line respectively. In this scenario the kinodynamic path is planned through the RKP module and is optimized through the NPO module. The LRK module is applied to re-plan the local path to avoid collision with unknown static obstacles that are detected with the forward looking sonar. The re-planed local paths are shown in Figure 15 with green line.

Scenario-3 (Randomized kinodynamic path planning in a dynamic unknown environment): In the third scenario the ability of the RKSP path planning algorithm is assessed in a dynamic unknown

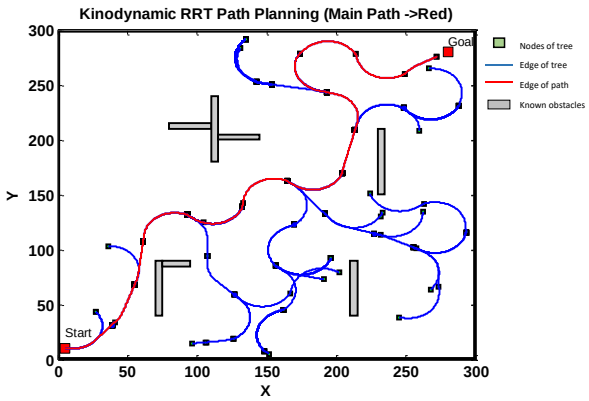

Figure 14. RRT kinodynamic planning in presence of known static obstacles

[Downloaded from ijmt.ir on 2026-09-10]

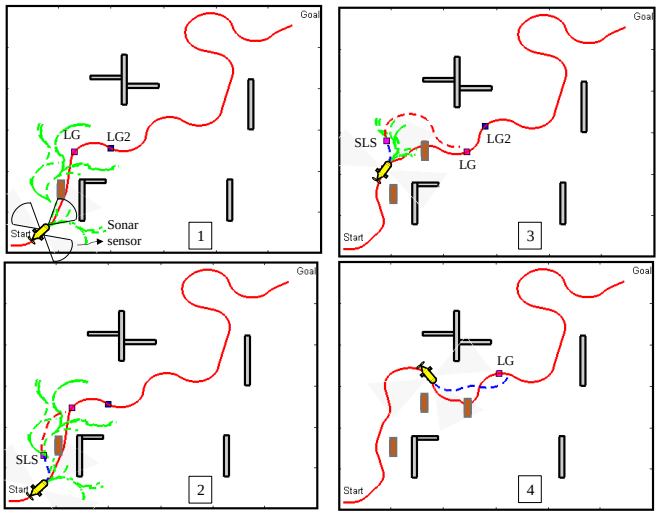

Figure 15. Path planning and path re-planning to avoid collision with unknown static obstacles

environment. In this scenario, workspace consist of 109 known and 5 unknown moving obstacles. The primary randomized kinodynamic path is planned through the RKP module and is optimized through the NPO module. AUV avoid to collision with moving obstacles by re-planning the local paths in the LRK module, Figure 16.

Path planning through the proposed RKSP in the two types of static known environment with different complexities: 1-narrow passage, and 2-maze-like space is considered in the first scenario and their results are shown in Figure 12 a&b. In both environments, the initially planned path is illustrated through the blue line

Table 2. Comparison of the LRKP results in the three scenarios with different complexity.

	Scenario1		Scenario2	Scenario3
	Maze-like	Narrow passage		
Number of static known obstacles	79	68	94	109
Number of static unknown obstacles	0	0	3	0
Number of unknown moving obstacles	0	0	0	5
Number of total offspring vertex	8849	6325	1257	2131
Number of re-planed local path	0	0	5	2
Initial path cost	48	32	75	73

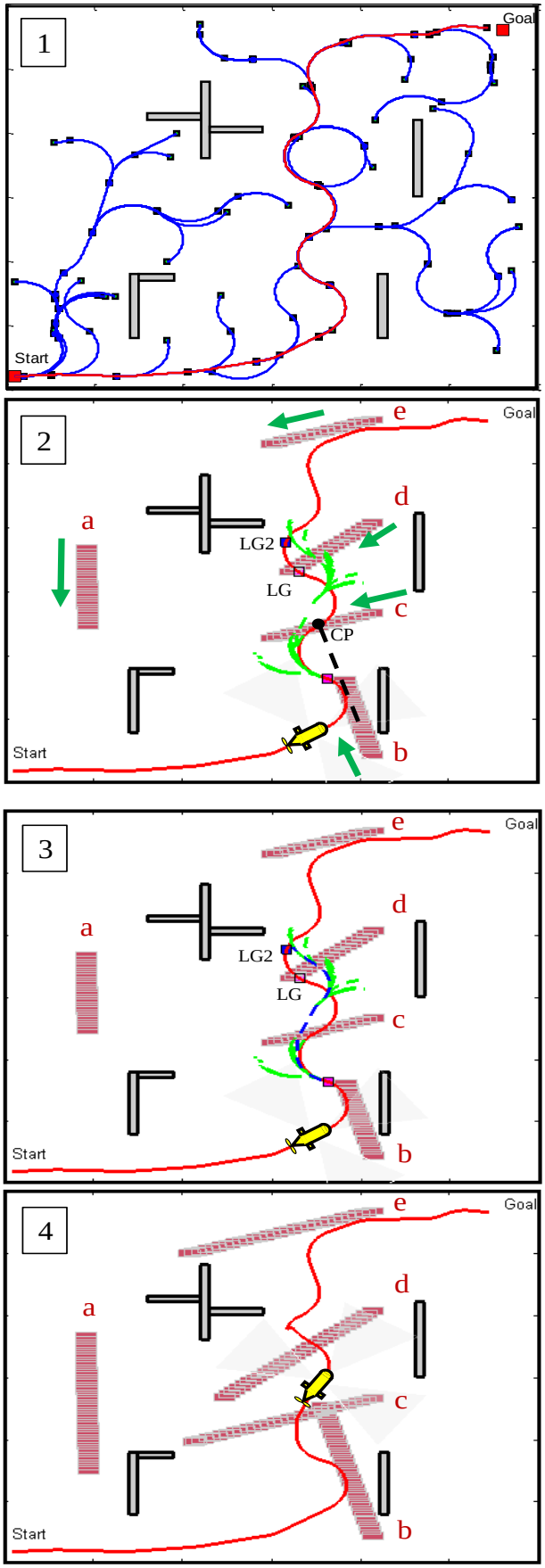

Figure 16. Kinodynamic path planning and local-path re-planning to avoid collision with unknown moving obstacles.

[Downloaded from ijmt.ir on 2026-09-10]

Table 3. Comparison of the standard RRT with LRKP.

	Type of Constraints	Type of Obstacles	Type of Workspace	Training and query
Proposed RKSP	Kinodynamic	Known-Static	Needs pre-define map	Offline-Online
Standard RRT [11]	Kinematic	Unknown-Moving	Needs exact pre-define map	Offline-Offline

and the optimized path with the triangular inequality is illustrated through the red line. The number of the vertices (waypoints) in the optimized path with the triangular inequality is reduced in comparison with the initial path while due to the randomized nature of the proposed RKSP the optimized path is sub-optimal. Path planning through the proposed RKSP in the unknown static environment is considered in the second scenario. Its related results are shown in Figure 15. The local search tree with green edges is shown in this scenario to avoid collision by the unknown static obstacles. In the third scenario, five unknown moving obstacles are considered in the simulations. Moving obstacles b and d have collided with the initially planned path. Hence, the initial path is re-planed in the local area that is shown by the green edges in Figure 16. Furthermore, the features of proposed RKSP are compared with the standard RRT in Table 3.

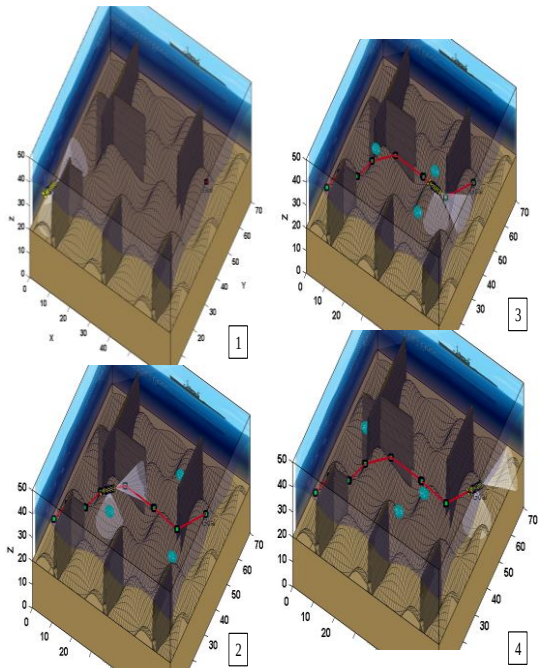

Figure 17. 3D view of AUV path planning.

Moreover, simulations of the second scenario are performed in the 3D environment which is shown in Figure 17. In Figure 17-1 initial kinodynamic path is planned, AUV tracks the initial path in Figure 17-2, AUV detects the unknown static obstacles and re-planes the path in Figure 17-3, and finally, AUV reaches the goal in Figure 17-4.

5. Conclusion

Plan a feasible path through the considering the kinodynamic constraints of AUV in a dynamic large-scale environment and avoid collision with unknown moving obstacles is an NP-Hard problem. Computational and time complexity of the kinodynamic path planning problem increase in the order $O(n^2)$ by increasing numbers of moving obstacles. This paper, the Randomized Kinodynamic Sub-optimal Planning (RKSP) algorithm is proposed to plan a feasible path. The main advantage of this RKSP is to obtain a set of feasible vertices for the AUV, and also the ability to re-plan the local paths in a real-time manner. In a nutshell, vertices in the RKSP algorithm are expanded in the large-scale environment through the randomized nature in a rapid manner, while, the kinodynamic constraints of AUV are considered in designing these vertices. Also, the RKSP prunes the inappropriate offspring vertices to plan a sub-optimal path. The RKSP consists of three main components that tightly coupled together. Kinodynamic constraints are applied to the planed path through the first component. The planed path is optimized through the second component. The local paths are re-planed through the third component to avoid collision with unknown moving obstacles. To assess the path planning abilities of the proposed RKSP algorithm in planning a feasible path for the AUV, three scenarios with different complexities are designed. In the first scenario, the RKSP plans near-optimal kinodynamic path in the both maze-like and narrow passage environment. In the second scenario, the RKSP re-plan the path to avoid collision with unknown static obstacles while considering kinodynamic constraints in path planning approach. In the third scenario, the RKSP re-plan the path to avoid collision with moving obstacles in real-time manner. The results show that the AUV tracks the planned vertices and the path is re-planned to avoid collision with unknown moving obstacles. In complex situations, the AUV can't reach the local-goal to re-plan the path due to the kinodynamic constraints of

[Downloaded from ijmt.ir on 2026-09-10]

AUV, however, the proposed path planning algorithm is able to design the next local-goal.

6. References

[1] W. Kazimierski, A. Sawczak, and N. Wawrzyniak, "Analysis of Graph Searching Algorithms for Route Planning in Inland Navigation," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 9, no. 2, pp. 281--286, 2015.

[2] M. P. Aghababa, "3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles," *Applied Ocean Research*, vol. 38, pp. 48-62, 2012.

[3] Y. Zhuang, S. Sharma, B. Subudhi, H. Huang, and J. Wan, "Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm," *Ocean Engineering*, vol. 127, pp. 190-199, 2016.

[4] Z. Zeng, A. Lammas, K. Sammut, F. He, and Y. Tang, "Shell space decomposition based path planning for AUVs operating in a variable environment," *Ocean Engineering*, vol. 91, pp. 181-195, 2014.

[5] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using RRT* based approaches: a survey and future directions," *Int. J. Adv. Comput. Sci. Appl*, vol. 7, no. 11, pp. 97-107, 2016.

[6] J. D. Hernández Vega, "Online path planning for autonomous underwater vehicles under motion constraints," 2017.

[7] M. Otte, and E. Frazzoli, "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797-822, 2016.

[8] S. Karaman, and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, 2010.

[9] R. Hess, R. Jerg, T. Lindeholz, D. Eck, and K. Schilling, "SRRT*-a probabilistic optimal trajectory planner for problematic area structures," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 331-336, 2016.

[10] O. Salzman, and D. Halperin, "Asymptotically near-optimal RRT for fast, high-quality motion planning," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 473-483, 2016.

[11] E. Taheri, M. H. Ferdowsi, and M. Danesh, "Fuzzy greedy RRT path planning algorithm in a complex configuration space," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 3026-3035, 2018.

[12] W. Wang, L. Zuo, and X. Xu, "A learning-based multi-RRT approach for robot path planning in narrow passages," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 1, pp. 81-100, 2018.

[13] Y. Dong, E. Camci, and E. Kayacan, "Faster RRT-based nonholonomic path planning in 2D building environments using skeleton-constrained path biasing," *Journal of Intelligent & Robotic Systems*, vol. 89, no. 3, pp. 387-401, 2018.

[14] E. Taheri, M. H. Ferdowsi, and M. Danesh, "Closed-loop randomized kinodynamic path planning for an autonomous underwater vehicle," *Applied Ocean Research*, vol. 83, pp. 48-64, 2019.

[15] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM (JACM)*, vol. 40, no. 5, pp. 1048-1066, 1993.

[16] S. Karaman, and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods." pp. 7681-7687.

[17] C.-b. Moon, and W. Chung, "Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Transactions on industrial electronics*, vol. 62, no. 2, pp. 1080-1090, 2014.

[18] D. J. Webb, and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics." pp. 5054-5061.

[19] R. Bordalba, J. M. Porta, and L. Ros, "Randomized kinodynamic planning for cable-suspended parallel robots," *Cable-Driven Parallel Robots*, pp. 195-206: Springer, 2018.

[20] Q.-C. Pham, S. Caron, and Y. Nakamura, "Kinodynamic Planning in the Configuration Space via Admissible Velocity Propagation."

[21] L. Palmieri, and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning." pp. 205-211.

[22] S. Yoon, D. Lee, J. Jung, and D. H. Shim, "Spline-based RRT∗ using piecewise continuous collision-checking algorithm for Car-like vehicles," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3, pp. 537-549, 2018.

[23] S. Stoneman, and R. Lampariello, "Embedding nonlinear optimization in RRT* for optimal kinodynamic planning." pp. 3737-3744.

[Downloaded from ijmt.ir on 2026-09-10]

[24] T. Bera, D. Ghose, and S. Suresh, "Asymptotic optimality of rapidly exploring random tree," *arXiv preprint arXiv:1707.03976*, 2017.

[25] B. Sakçak, "Optimal kinodynamic planning for autonomous vehicles," 2018.

[26] K. Hauser, and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state–cost space," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1431-1443, 2016.

[27] R. Bordalba, L. Ros, and J. M. Porta, "Kinodynamic planning on constraint manifolds," *arXiv preprint arXiv:1705.07637*, 2017.

[28] M. Moll, L. Kavraki, and J. Rosell, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712-719, 2017.

[29] M. Herrero-Collantes, and J. C. Garcia-Escartin, "Quantum random number generators," *Reviews of Modern Physics*, vol. 89, no. 1, pp. 015004, 2017.

[30] M. Matsumoto, and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3-30, 1998.

[31] B.-H. Jun, J.-Y. Park, F.-Y. Lee, P.-M. Lee, C.-M. Lee, K. Kim, Y.-K. Lim, and J.-H. Oh, "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.

[32] T. T. J. Prestero, "Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle," Massachusetts institute of technology, 2001.

[33] E. Kim, S. Fan, N. Bose, and H. Nguyen, "Current Estimation and Path Following for an Autonomous Underwater Vehicle (AUV) by Using a High-gain Observer Based on an AUV Dynamic Model," *International Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 478-490, 2021.

[34] A. Karmozdi, M. Hashemi, H. Salarieh, and A. Alasty, "INS-DVL navigation improvement using rotational motion dynamic model of AUV," *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14329-14336, 2020.

Appendix A

Definitions of the main parameters of the AUV model that are used in this paper.

No.	Parameter	Value	Unit
1	x_{uu}	-4.326	kg/m
2	x_{u}	-0.564.812	kg
3	x_{wq}	-42.048	kg/rad
4	x_{qq}	-45.005	kg*m/rad
5	x_{w}	-29.859	kg/rad
6	x_{rr}	-11.658	kg*m/rad
7	y_{w}	-50.232	kg/m
8	y_{rr}	-27.968	kg*m/rad^2
9	y_{uv}	-13.552	kg/m
10	y_{v}	29.859	kg
11	y_{r}	11.658	kg*m/rad
12	y_{ur}	0.564.812	kg/rad
13	y_{wp}	42.048	kg/rad
14	y_{pq}	45.005	kg*m/rad
15	y_{uudr}	40.109	kg/(m*rad)
16	z_{ww}	-50.232	kg/m
17	z_{qq}	27.968	kg*m/rad
18	z_{uw}	-13.552	kg/m
19	z_{w}	-42.048	kg
20	z_{q}	-45.005	kg*m/rad
21	z_{uq}	0.564.812	kg/rad
22	z_{vp}	0.564.812	kg/rad
23	z_{rp}	11.658	kg/rad
24	z_{uuds}	-40.109	kg/(m*rad)
25	M_{uud_s}	-17.357	kg/rad
26	N_{uud_r}	-17.357	kg/rad
27	K_{uud_a}	0.041235	kg/rad

[Downloaded from ijmt.ir on 2026-09-10]