Optimal Path Planning using RRT* based Approaches: A Survey and Future Directions

Iram Noreen
Department of Computer Science
COMSATS Institute of Information Technology
Lahore, Pakistan

Amna Khan
Department of Computer Science
COMSATS Institute of Information Technology
Lahore, Pakistan

Zulfiqar Habib
Department of Computer Science
COMSATS Institute of Information Technology
Lahore, Pakistan

***Abstract*—Optimal path planning refers to find the collision free, shortest, and smooth route between start and goal positions. This task is essential in many robotic applications such as autonomous car, surveillance operations, agricultural robots, planetary and space exploration missions. Rapidly-exploring Random Tree Star (RRT*) is a renowned sampling based planning approach. It has gained immense popularity due to its support for high dimensional complex problems. A significant body of research has addressed the problem of optimal path planning for mobile robots using RRT* based approaches. However, no updated survey on RRT* based approaches is available. Considering the rapid pace of development in this field, this paper presents a comprehensive review of RRT* based path planning approaches. Current issues relevant to noticeable advancements in the field are investigated and whole discussion is concluded with challenges and future research directions.**

Keywords—optimal path; mobile robots; RRT*; sampling based planning; survey; future directions

I. Introduction

The term path planning refers to collision free path generation from an initial state to a specified goal state with optimal or near optimal cost. Considering different applications and constraints of robots, optimal criteria could be based on one or more conditions such as shortest physical distance, smoothness, low risk, less fuel requirements, maximum area coverage, and low energy consumption. Hence, in perspective of path planning for mobile robots optimal path refers to find a feasible plan with optimized performance according to application specified criterion [1]. Optimal path planning is also influenced by the holonomic and non-holonomic constraints. According to LaValle, the term non-holonomic refers to the differential constraints (restrictions on permissible velocities) that are not completely integrable, such as car-like robots and the others are holonomic constraints such as robotic arm [1].

Path planning algorithms are of vital importance for motion planning of mobile robots due to their numerous applications in autonomous cars [2], Unmanned Aerial Vehicles (UAVs) [3], forklifts [4], surveillance operations [5], medical [6], planetary and space missions [1, 7]. Initial complete practical planners such as Road Map (RM), Potential Fields, and Cell Decomposition (CD) techniques are unable to deal with dynamic and complex high dimension problems [1, 7-10]. Computational complexity of complete planners limits their applications to low dimensional problems [11].

Grid based algorithms such as Dijkstra [12], wavefront [13] , A* [14], D* [15], and Phi* [16] are resolution-complete and are computationally expensive for high dimensional complex problems. Evolutionary algorithms such as Particle Swarm Optimization (PSO) [17-19], Ant Colony Optimization (ACO) [20] and Genetic Algorithm (GA) [21] are suitable for multi-objective problems. Many other evolutionary algorithms such as Artificial Bee Colony (ABC) [22], Bacterial Foraging Optimization (BFO) [23], Bio Inspired Neural Networks [24, 25], and Fire Fly algorithm [26] are often trapped in local optimum, and bear high computational cost. Moreover, they are highly sensitive to search space size and data representation scheme of problem [27, 28].

Sampling Based Planning (SBP) approaches are the most influential advancement in path planning [7, 8]. Major advantages of Sampling Based Planning (SBP) are low computational cost, applicability to high dimensional problems and better success rate for complex problems [8, 29]. SBPs are probabilistic complete, i.e., it finds a solution, if one exists, provided with infinite run time [4, 8]. Most popular SBP algorithms are Probabilistic Roadmap (PRM) [7, 8, 30], Rapidly-exploring Random Tree (RRT) [11, 31] and Rapidly-exploring Random Tree Star (RRT*) [7]. PRM based methods [7, 32] are mostly used in highly structured static environment such as factory floors [11, 29, 33]. They are well suited for holonomic robots but could be extended for non-holonomic as well [31]. On the other hand RRT and RRT* based approaches [7] naturally extend non-holonomic constraints [11] and support dynamic environment as well.

Introduced by Karaman and Frazzoli [7], RRT* was a major breakthrough in optimal path planning for high dimensional problems. RRT* has proven asymptotically optimal property, i.e., RRT* always converges to an optimal solution, if adequate run time is provided. RRT* has gained tremendous success in solving high dimensional complex problems with numerous successful applications. A survey on Sampling Based Planning (SBP) approaches for mobile robots was presented in [8]. However, considerable body of research has specifically addressed the problem of optimal path planning focusing RRT* in recent years as compared to other SBP approaches. Rapid pace of development in optimal path planning using RRT* based approaches has grown it into a family of algorithms [2, 3, 28, 33-50]. To the best of our knowledge, no updated survey exists on RRT* based approaches. This paper is an effort to review the major breakthroughs in RRT* based approaches providing link to the most successful works in the field. Moreover, current state of the art is surveyed to explore recent contributions and future directions in optimal path planning.

This paper has been organized in such a way that a discussion on RRT* methodology is presented in next section. Section III categorizes RRT* based path planning approaches in recent years. State of the art techniques are summarized in Section IV. Section V presents challenges followed by conclusion along with future recommendations in Section VI.

II. RRT* Methodology

This section introduces important path planning concepts related to RRT* in order to provide a better understanding of this study. It is essential to introduce the basic operations of RRT* prior to describe its variant approaches. These procedures are found in all RRT* variants, but their implementation may differ in different planners and applications.

A. Problem Formulation

RRT* based approaches operate in the configuration space. This configuration space is a set of all possible transformations which are applicable to the robot. [1, 51]. Let the given configuration space be denoted by a set $Z \subset \mathbf{R}^n, n \in \mathbf{N}$ where n represents the dimension of the given space and $\mathbf{N}$ is a set of positive integers. Configuration space occupied by obstacles is denoted by $Z_{obs} \subset Z$ and obstacle-free region is denoted by $Z_{free} = Z / Z_{obs}$. $z_{goal} \subset Z_{free}$ is the goal and $z_{init} \subset Z_{free}$ is the starting point. z_{init} and z_{goal} are provided to planner as input. The problem is to find an optimal collision free path between initial z_{init} and goal z_{goal} states in Z_{free}, with minimum path cost in the least possible time $t \in \mathbf{R}$, where $\mathbf{R}$ is the set of real numbers.

B. Tree Expansion in RRT*

RRT* constructs multiple short paths randomly organized as tree instead of one long path. It originates tree from initial state z_{init} to find a path towards goal state z_{goal}. The tree gradually improves with iterations. In each iteration, a sampling process selects a random state say z_{rand} from configuration space Z. The random sample z_{rand} is rejected if it lies in Z_{obs}. However, if it lies in Z_{free} then a nearest node say $z_{nearest}$ is searched in tree T according to a defined metric ρ . If z_{rand} lies in Z_{free} and is also accessible to $z_{nearest}$ according to predefined step size, then a local planner inserts it in tree by connecting z_{rand} and $z_{nearest}$. Otherwise, planner returns a new node z_{new} by using a steering function and adds it in tree by connecting it with $z_{nearest}$. This property of RRT*, to explore region in Z_{free} is called Voronoi bias. A collision checking process is performed to ensure collision free connection between z_{new} and $z_{nearest}$ .The Node expansion process is illustrated in Fig. 1.

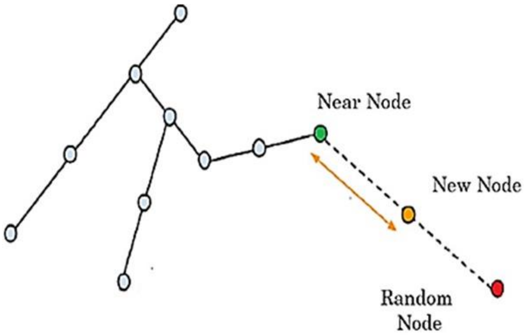


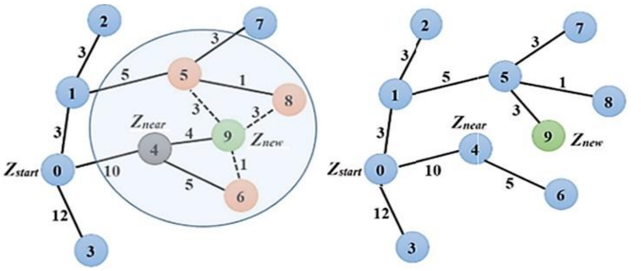
Fig. 1. RRT* Tree expansion process [52]

If z_{new} is found collision free then near neighbors of z_{new} are searched within the area of a ball of radius defined by

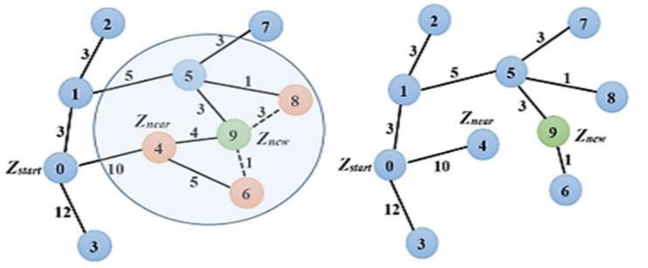
$$k = \gamma(\log(n)/n)^{1/d} \text{ [7] ,} \tag{1}$$

where d is the configuration space dimension and γ is the planning constant based on environment. Within the area defined by (1), neighbor z_{min} with least cost is selected to be parent of z_{new}. Procedure of near neighbor search is similar to k-near neighbor problem to find out the best parent node z_{min} of new node z_{new} before its insertion in tree. New node z_{new} is inserted as child of z_{min} in tree. Further, the cost of near neighbor's parent node is also compared with the cost of z_{new} . If z_{new} gives less cost as parent, then rewiring process rebuilds the tree for minimum parent cost within the area identified by (1) [53]. This process is shown in Fig. 2.

The process of selecting least cost parent and rewiring tree are two most promising features of RRT* and contribute to asymptotic optimal property of RRT* [7]. Though best parent selection and rewiring of tree improve the path quality. However, these features have an efficiency trade-off with path quality and make convergence slow as number of nodes in the tree increase. When z_{goal} is found, a path connecting z_{init} and z_{goal} is established. This path is improved as planner continues until a predefined number of iterations are executed or given time expires. The RRT* Algorithm is described as Algorithm 1.



(a) Find near neighbors (b) Select best parent



(c) Check cost for rewiring (d) Rewired tree

Fig. 2. Near neighbor search and rewiring operations in RRT* [52]

```
Algorithm 1.  T = (V, E) ← RRT*( $z_{init}$)
T ← InitializeTree();
T ← InsertNode(Ø, $z_{init}$, $T$);
for $i$=0 to $i$=$N$ do
    $z_{rand}$ ← Sample($i$);
    $z_{nearest}$ ← Nearest($T$, $z_{rand}$);
    ($z_{new}$, $U_{new}$) ← Steer ($z_{nearest}$, $z_{rand}$);
    if Obstaclefree($z_{new}$) then
        $z_{near}$ ← Near($T$, $z_{new}$, $|V|$);
        $z_{min}$ ← Chooseparent ($z_{near}$, $z_{nearest}$, $z_{new}$);
        $T$ ← InsertNode($z_{min}$, $z_{new}$, $T$);
        $T$ ← Rewire ($T$, $z_{near}$, $z_{min}$, $z_{new}$);
return $T$
```

III. Methodologies based on RRT* Algorithm

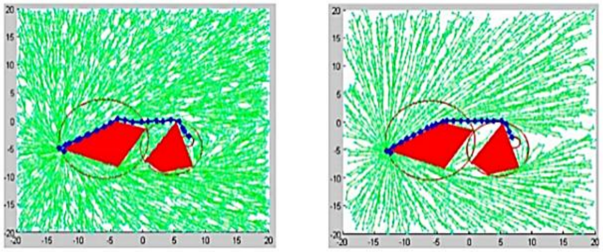
This section provides review of optimal path planning using RRT* in recent six years with major breakthroughs in the field. To provide a better understanding of research body, we have classified these approaches based on the similar concepts such as type of environment information available, the structure of the tree and the constraints managed by approach. RRT* have been used in online mode or offline mode depending upon availability of environment information. If environment parameters are unknown or highly uncertain then local planning is performed, also called *online* (sensor based, or reactive). Whereas, a known environment requires global planning, also called *offline* (map based) [23, 54]. Further, RRT* variants based on bidirectional trees also exist in literature, which generate two trees simultaneously from start and goal states. Further, RRT* based approaches considering *non-holonomic* constraints also exist.

A. *Single Directional Holonomic RRT* Approaches*

This section presents RRT* based approaches which generate path for holonomic robots and construct a single tree originating from initial state z_{init} towards goal state z_{goal} to find path in search space. Both online and offline approaches are discussed in this subsection. S. Karaman et al. [4] presented an online Anytime variant of RRT*. Basic idea of Anytime RRT* is to deal with the issue of large computational time by executing the planner for a predefined planning time. Once an initial path is obtained and stored, then rest of the time is used to improve initial solution [55, 56]. Anytime RRT* introduced two key features called committed trajectories and branch-and-bound adaptation. Strategy of committed trajectory originates robot's movement to follow first segment of initially planned path while improving remaining segments of the path using iterative strategy. Whereas Branch-and-bound optimizes the tree for optimal cost. Anytime RRT* improved trajectory and computational efficiency in simulation and in real-time implementation using forklift robot.

Despite success stories, RRT* was suffering from high memory consumption due to large expansion of its search space. Adiyatov and Varol [33] introduced memory efficient version of RRT*, called RRT* Fixed Nodes (RRT*FN). RRT*FN allows limited number of nodes in tree. When tree is expanded to a preset fixed number of nodes then new node can only be inserted by deleting the old node.

The old nodes are deleted according to a defined node removal policy. RRT*FN implies a global node removal procedure and a local node removal procedure for this purpose. Local node removal procedure deletes nodes with single child from near neighbors during rewiring operation, if new node has better cumulative path cost for their child node as parent. In case when no such nodes are identified during rewiring operation then a global node removal procedure is used, which searches entire tree to find nodes without children and deletes them. When both local and global schemes could not found such nodes then new node is not inserted [33]. Hence, search space of RRT*FN consumes less memory by forcing a fixed number of nodes in the tree. Difference of tree density between RRT* and RRT*FN is evident from Fig. 3 (a) and Fig. 3 (b). Such memory efficient versions of RRT* are useful in robots and embedded systems with limited memory [33].



(a) RRT* (b) RRT*FN

Fig. 3. Effect of fixed nodes after 5000 iterations in tree

Nasir et al. [35] presented an offline variant of RRT* called RRT*-Smart to address the issue of slow convergence. RRT*-Smart introduced two major features called intelligent sampling and path optimization. Initial path finding procedure in RRT*-Smart is similar to RRT*. However, once a path is found, it is optimized based on triangular inequality principle to remove redundant nodes [35]. Optimization task generates beacon nodes to further improve path cost. After optimization, it uses both intelligent and uniform sampling strategies alternatively according to defined *Biasing Ratio* for the rest of the iterations using

$$Biasing\ Ratio=(n\ /Z_{free}\ )*B \quad [35], \qquad (2)$$

where B is a programming constant and n is total number of nodes in tree. Intelligent sampling is biased towards beacon nodes. Each time it gets a new path with shorter cost, it optimizes the path again and identifies new beacon nodes. This process is based on a Biasing Radius to set radius for intelligent exploration around selected beacons. The proposed approach accelerated the convergence rate with improved path cost and time efficiency. However, intelligent sampling has a trade-off between rate of convergence and rate of exploration. Therefore, frequency of intelligent sampling needs careful adjustment according to different environment types. An experimental comparison for performance evaluation is also provided in [52] for RRT, RRT* and RRT*-Smart.

Another offline approach called Informed RRT* was presented by Gammell et al. [34] for optimal path planning in narrow passages. They proposed a direct subset sampling technique for configuration space exploration. Once an initial path is found, it further explores configuration space within a limited elliptical area defined by an ellipsoidal informed subset. As area of the ellipse decreases, it also improves the rate of convergence and path quality.

Arslan and Tsiotras [50] proposed RRT* variant called RRT$^{\#}$ (RRT "sharp") to address the issue of slow convergence. RRT$^{\#}$ used two processes during each iteration namely exploration and exploitation. Exploration performs the extension process whereas exploitation uses a global re-planning procedure to keep track of promising nodes of tree. Promising nodes are the ones which are good candidate to contribute in the final path with lowest cost. During each iteration, RRT$^{\#}$ updates information about promising nodes and prioritize them for re-planning in next iteration. Hence, it makes fast convergence by expanding promising nodes towards goal region and exploiting available node information to the highest degree at each iteration.

B. Bidirectional Holonomic RRT* Approaches

All approaches discussed above build single tree in configuration space. This section gives insight on bidirectional approaches. Bidirectional approaches generate two trees simultaneously from start and goal states directing towards each other, as shown in Fig. 4 (a) and 4 (b). Use of bidirectional tree was initially proposed by Kuffner and LaValle in RRT-Connect [11]. They used it initially for motion planning of 7-DOF arm of animated characters used in 3D virtual world. Hence, it was specifically designed for path planning problems with no differential constraints [11].

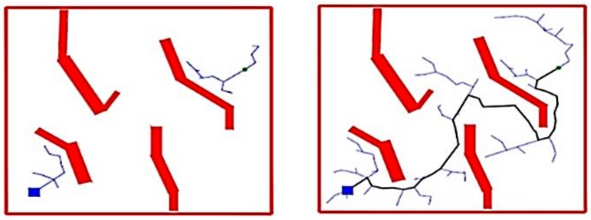

(a) Two trees growing from start and goal (b) Joined trees

Fig. 4. RRT-Connect, Growing two trees towards each other [11]

Moreover, RRT-connect is not asymptotically optimal like RRT*. Applying bidirectional trees to asymptotically optimal RRT* requires neighborhood rewiring in two trees resulting in high computational cost. Though, bidirectional approaches execute faster for holonomic robots. However, when used for non-holonomic problems, they made very slow convergence. This is due to the fact that managing non-holonomic constraints using connect heuristic of bidirectional tree does not guarantee the connection of both trees [57]. Therefore bidirectional variants of RRT* are considered suitable only for holonomic robots [57, 58].

An asymptotic optimal variant of RRT* [4] and RRT-connect [11] called Bidirectional RRT* (B-RRT*) was proposed by Akgun and Stilman [58]. It showed empirical results indicating fast convergence and path refinement using sample rejection with an admissible heuristic. Though, this procedure selects only promising nodes but also affects space exploration. Moreover, attempt to connect both trees in each iteration incurred computational overhead.

Another bidirectional RRT* was presented by Jordan and Perez [36] for optimal path planning also called Optimal B-RRT*. It was provably asymptotically optimal bidirectional approach with improved convergence rate using a number of heuristic techniques [36]. However, use of multiple heuristics also increased computational overload. Moreover, these biased heuristics interfered with the algorithm characteristics (such as exploration, node rejection, cost function) and limited its application.

Another approach called Intelligent Bidirectional RRT* (IB-RRT*) [37] was proposed by Qureshi and Ayaz for complex cluttered environment. It used an intelligent sample insertion heuristic technique. Simulation results of IB-RRT* showed fast convergence towards optimal path using less memory resources in comparison with RRT* and Bi-RRT.

Recently, Yi et al. [45] presented Homotopy-Aware RRT* (HARRT*) based on bidirectional RRT* [58]. HARRT* is inspired with the idea of homotopy, i.e., to plan path from one topological space to another by human intervention. This approach addresses the planning problems of human-robot team interactions in search and rescue, police, and military operations. Effectiveness of proposed approach is theoretically proved using case studies. However, further investigation of the proposed approach remains to explore using simulations and real world experiments.

C. Non-holonomic and Kinodynamic RRT* Approaches

Non-holonomic robots are car-like robots which have to perform complex motions to achieve a particular direction. This phenomenon also restricts geometry of the path [1]. A car-like robot needs to change its position coordinates in order to rotate around its axis (see Fig. 5). It is under-actuated due to the non-holonomic constraints imposed by the wheels. Therefore, non-holonomic path planning requires satisfying both internal constraints (physical limitations of robot) and external constraints (obstacles in environment) [1, 38, 59]. Further, if kinodynamic constraint arises, then it also affects path planning mechanism.

Kinodynamic planning [60] refers to motion planning problems for which velocity and acceleration bounds must be satisfied. Precisely, kinodynamic is an umbrella term used to deal with kinematics (position, bounds on velocity and acceleration) and dynamic constraints (force) simultaneously [1, 60]. Non-holonomic planning deals with either both (i.e., kinodynamic) constraints or kinematic constraints only [1, 60].

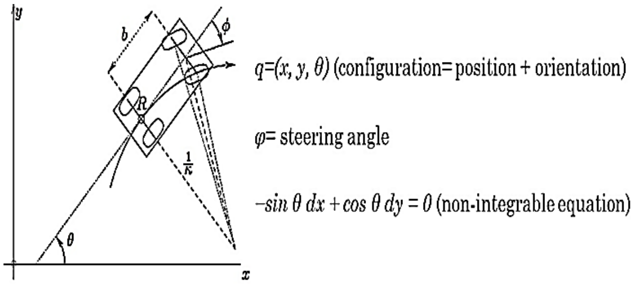

y
b
ϕ
R
$\frac{1}{\kappa}$
θ
x
$q=(x, y, \theta)$ (configuration= position + orientation)
φ= steering angle
$-\sin\theta\, dx + \cos\theta\, dy = 0$ (non-integrable equation)

Fig. 5. Non-holonomic constraints of car-like robot [61]

As RRT* connects pair of states using straight lines, which is not feasible for kinodynamic systems due to the differential constraints. Prior kinodynamic extensions of RRT* such as Kinodynamic-RRT* [62] and LQR-RRT* [63] only satisfy bounded sub optimality and require RRT* to re-propagate the tree partially during each iteration. Thus, making these approaches computationally expensive.

In recent years, many RRT* based planners were focused to solve the optimal path planning problem for car-like robots dealing with non-holonomic or kinodynamic constraints. Webb and Berg [49, 64] presented a kinodynamic extension of RRT* called Adapted RRT* to overcome the above limitations. The proposed approach found asymptotically optimal trajectories for a car-like robot with a 5D state space and an aerial vehicle with a 10D state space.

Recently, Lee et al. [44] proposed Spline-based RRT* (SRRT*) for non-holonomic path planning of fixed-wing Unmanned Aerial Vehicles (UAVs) in three-dimensional environment. The proposed algorithm expands the tree by using a cubic Bézier spline curve. Use of Bézier spline parameterization in SRRT* as local planner replaced time and input discretization. Moreover, it performed dynamic feasibility and geometric collision checks as part of the tree extension. This phenomenon enabled SRRT* to produce smooth and cost-optimal path in 3D simulations using constraint model of UAV.

Alejo et al. [3] presented RRT**i* for efficient motion planning of UAVs. They compared the proposed approach with genetic algorithm, RRT and RRT*. Simulations results and real world experiments using UAV proved that RRT**i* computes more predictable and smoother trajectories as compared to aforementioned techniques. Initially, RRT**i* also works similar to RRT*. However, as soon as an initial trajectory is found, it adapts local sampling with Gaussian distribution and node rejection technique. Using this sampling technique it refines tree in vicinity of initial path solution, which leads to rapid convergence with less jagged path segments.

Recently, Csorvasi et al. [43] have proposed RTR+C*CS which uses a global planner and a local planner for car-like robots. First, global planner called RTR (Rotate-Translate-Rotate) generates path. Then local planner called C*CS makes it feasible for car-like robots using circular arcs and straight segments. However, its local planner requires large number of iterations to find less sharp turns.

Moon et al. [39] presented a kinodynamic variant of RRT* called Dual-Tree RRT (DT-RRT). DT-RRT manages two trees called state tree and workspace tree. At first, workspace tree explores targeted environment without considering any physical constraints. Then, state tree generates trajectories from workspace tree nodes using kinematic and dynamic constraints. It also offers reconnect-tree scheme in contrast to rewire operation of RRT* [7]. The reconnect-tree scheme [39] maintains child nodes with reduced computational cost, in order to guarantee asymptotic optimality. The dual tree scheme of DT-RRT [39] approach showed high success rate for node extension because it reduced the node rejection chances by applying kinodynamic conditions. Moreover, this approach is compatible with advanced near-neighbor search schemes, for example *k-d* trees with reduced space dimensionality. The proposed approach showed better computation time in simulation results using two wheeled mobile robot for high speed navigation. However, this scheme is only useful for known or partially known environment.

Another variant of RRT* proposed by Lan and Cairano [2] was experimented under Mitsubishi Electric Research Laboratories for autonomous driving vehicle. The proposed approach used two-stage sampling strategy similar to SRRT [40] and a weighted cost function tailored to drive semi-autonomous vehicle. The proposed approach could also manage driving lanes beside collision avoidance. A local re-planning procedure enables algorithm to react with dynamic obstacles. Further, path pruning and G^2 continuous curvature smoothing techniques are applied as post processing.

J. Suh et al. [65] presented an offline Cost Aware RRT* (CARRT*) for energy efficient optimal path planning in high dimensional space for humanoid robot. The approach used two trees to address the dense sampling issue in RRT*. First tree is a standard RRT* tree to determine nearest node for newly sampled random point whereas second tree contains first tree to extend additional long branches. They also proposed a cross entropy function based on CE path planning [66] and a cost function based on mechanical work (MW) [67] to measure energy consumption along a path for humanoid robot. Limitation of large memory requirements is also addressed by another variant called Potential Function RRT* (PRRT*) presented by Qureshi, and Ayaz [48]. PRRT* (PRRT*) is the extension of two other variants PGD-RRT* and APGD-RRT* [68]. As compared to these two approaches, PRRT* efficiently integrates artificial potential field with RRT* for guided exploration of search space with less memory needs and improved convergence.

Devaurs et al. [46] proposed Transition based RRT* (T-RRT*) which focuses on optimal path planning for continuous configuration-cost space. Their approach integrated transition test based functions used in T-RRT [67] to address the issue of path quality with respect to a given criterion.

Recently, Choudhury et al. [47] have proposed RABIT* to address the problem of planning in high dimensions. The proposed approach focuses homotopy classes which are difficult to sample for example narrow passages. RABIT* was extended by an informed global technique called BIT* [69] by using a local optimization module to improve an initial sub-optimal path towards a local optimum. Thus, the proposed approach preserved almost-sure global optimal convergence.

IV. State of the Art (2011-2016)

The most relevant papers reviewed in this article, along with the research contributions and limitations are summarized below in Table 1. Different attributes of the state of the art approaches are also listed in Table 2.

TABLE I. State of the Art (2011-2016)

Sr#	Author, year	Approach	Research Contributions	Limitations / Future Recommendations
1.	Karaman and Frazzoli [7], 2011	RRT*	• Proved asymptotically optimal property for RRT*. • Introduced new key features of near neighbor search and rewiring operations. • Visibly refined path quality than original RRT.	• New features had an efficiency trade-off. Insertion of good candidate node with best parent selection improved tree cost but on the other hand it also slowed down convergence rate of RRT*. • Jagged, suboptimal paths and slow convergence • Large memory requirements.
2.	Karaman et al. [4], 2011	Anytime RRT*	• It introduced two key features called committed trajectories and branch-and-bound adaptation. • It improved trajectory and computational efficiency by gradually removing nodes from tree which are unable to improve current solution path.	• Jagged and suboptimal paths. • Could overestimate and may cause unnecessary node removal during initial expansion of tree when it is not mature.
3.	Akgun and Stilman [58], 2011	B-RRT*	• Improved convergence speed and path refinement using sample rejection with an admissible heuristic.	• Attempt to connect both trees in each iteration incurred computational overhead. • Jagged and suboptimal paths. • Large memory requirements.
4.	Adiyatov and Varol [33], 2013	RRT*FN	• Memory efficient version of RRT* by forcing a fixed number of nodes in the tree.	• Jagged, suboptimal path. • Rate of convergence to optimal path is lower than base RRT*. • Worked only for static known environment.
5.	Nasir et al. [35], 2013	RRT*-Smart	• Two new features intelligent sampling and path optimization were introduced. • It accelerated the convergence rate with improved efficiency with respect to both time and cost	• It is dependent upon a heuristic called Biasing Ratio which has a trade-off between convergence rate and exploration of space. • Heuristic used by this approach are not automated and require programmer dependent value for different environments.
6.	Jordan and Perez [36], 2013	Optimal B-RRT*	• Introduced multiple heuristics, based on different conditions to increase convergence rate of bidirectional RRT*.	• Use of multiple heuristics caused computational overload. • Biased heuristics interfere with the algorithm characteristics (such as exploration, node rejection, cost function) and limited its application also.
7.	O. Arslan and P. Tsiotras [50], 2013	RRT# (RRT "sharp")	• Introduced a global replanning scheme to maintain promising nodes of tree to make fast convergence towards optimal path with low cost.	• Real time experiments and applications would be beneficial to investigate and improve the efficiency of approach.
8.	Webb and Berg [49, 64], 2012, 2013	Adapted RRT*	• Presented kinodynamic extension of RRT* with ensured asymptotic optimality with controllable linear dynamics, in multi dimension state space.	• Post processing steps for path smoothness and real world quadrotors experiments are planned for future work. However, such experiment would require controller stabilization for final trajectory.
9.	Lee et al. [44], 2014	SRRT*	• Proposed spline based RRT* based on a cubic Bézier curve for fixed-wing UAVs. • Presented geometric collision and dynamic feasibility function checking constraints during tree expansion. • Produced feasible smooth and cost-optimal path in 3D simulations.	• Online planner with real time application would be beneficial to further investigate and improve the efficiency of approach.
10.	Gammell et al. [34], 2014	Informed RRT*	• Proposed direct sampling technique based on ellipsoidal informed subset, which showed improved convergence than RRT*.	• Heuristic used to shrink planning problem is highly dependent upon initial solution cost which makes it effective only under certain conditions.
11.	Qureshi and Ayaz [37], 2015	IB-RRT*	• Introduced intelligent sample insertion heuristic with minimal memory requirements, which improved the path quality as compared to RRT* and B-RRT*.	• Its application needs further investigation for online planning.
12.	Moon and Chung [39], 2015	DT-RRT	• Addressed kinodynamic planning for high speed mobile robot and produced practically feasible trajectories. • Instead of using rewire operation, it introduced reconnect-tree scheme to maintain child nodes with reduced computational cost.	• Approach need advancement for dynamic and unknown scenarios as its reconnect-tree scheme is only beneficial in known environments.

			• Reduced the node rejection chances by using kinodynamic conditions.	
13.	Alejo et al. [3], 2015	RRT**i*	• Predictable and practical trajectory generation for UAVs. • Smoother trajectories as compared to RRT and RRT* with improved path quality. • Introduced new local sampling technique.	• Could produce unexpected collisions in multi-UAV applications with increased uncertainty.
14.	Csorvasi et al. [43], 2015	RTR+CS*	• Used a local planner to generate feasible path for car-like robots using circular arcs and straight segments.	• Generated paths are not curvature continuous and natural. • It is still under experimental process to improve computational performance.
15.	Lan and Di Cairano [2], 2015	Mitsubishi RRT*	• Addressed the problem of G^2 continuous curvature smooth path for autonomous driving vehicle with the capability of lane management on roads. • Introduced a local replanning procedure to safely avoid and re-plan due to dynamic obstacles.	• In context of autonomous vehicle driving, uncertainty of dynamic environment is not managed.
16.	J. Suh et al. [65] , 2015	CARRT*	• Addressed the problem of large number of samples by using two trees and cross entropy in RRT*. • Produced energy efficient path in high dimensional space.	• It is limited to address the planning problems for humanoid robots only.
17.	Qureshi and Ayaz [48], 2016	PRRT*	• Addressed the problem of high memory consumption and slow convergence by incorporating artificial potential field characteristic in RRT*.	• Considering optimal results with fast convergence than RRT*, approach could be employed for online planning in future.
18.	Yi et al. [45], 2016	HARRT*	• Presented a human-robot interactive planner using RRT* and homotopy algorithm.	• Trade-off between computational efficiency and path quality. • Natural language processing or graphical user interface could be adapted for compatibility, usability and workload of human and robot interactions.
19.	Devaurs et al. [46], 2016	T-RRT*	• Integrated transition tests with RRT* for efficient extension of tree in a cost space.	• Performance analysis with RRT* in different problems would be of interest to show its beneficial problem class.
20.	Choudhury et al. [47], 2016	RABIT*	• Used an informed global technique BIT* with RRT* to find optimal path for narrow passages in high dimensions.	• Local optimizer suggested by approach needs further investigation for optimization of path.

TABLE II. ATTRIBUTE SUMMARY OF THE STATE OF THE ART (2011-2016)

Approaches	Constraints	Planning Mode	Kinematic Model	Sampling Strategy	Metric
1. RRT* [7]	Holonomic	Offline	Point	Uniform	Euclidean
2. Anytime RRT* [4]	Non-holonomic	Online	Dubin Car	Uniform	Euclidean + Velocity
3. B-RRT* [58]	Holonomic	Offline	Rigid Body	Local bias	Goal biased
4. RRT*FN [33]	Holonomic	Offline	Robotic Arm	Uniform	Cumulative Euclidean
5. RRT*-Smart [35]	Holonomic	Offline	Point	Intelligent	Euclidean
6. Optimal B-RRT* [36]	Holonomic	Offline	Point	Uniform	Euclidean
7. RRT# [50]	Holonomic	Offline	Point	Uniform	Euclidean
8. Adapted RRT* [64], [49]	Non-holonomic	Offline	Car-like and UAV	Uniform	A* Heuristic
9. SRRT* [44]	Non-holonomic	Offline	UAV	Uniform	Geometric + dynamic constraint
10. Informed RRT* [34]	Holonomic	Offline	Point	Direct Sampling	Euclidean
11. IB-RRT* [37]	Holonomic	Offline	Point	Intelligent	Greedy + Euclidean
12. DT-RRT [39]	Non-holonomic	Offline	Car-like	Hybrid	Angular + Euclidean
13. RRT*i [3]	Non-holonomic	Online	UAV	Local Sampling	A* Heuristic
14. RTR+CS* [43]	Non-holonomic	Offline	Car-like	Uniform + Local Planning	Angular + Euclidean
15. Mitsubishi RRT* [2]	Non-holonomic	Online	Autonomous Car	Two-stage sampling	Weighted Euclidean
16. CARRT* [65]	Non-holonomic	Online	Humanoid	Uniform	MW Energy Cost
17. PRRT* [48]	Non-holonomic	Offline	P3-DX	Uniform	Euclidean

18. HARRT* [45]	Holonomic	Offline	Point	Uniform	Homotopy check
19. T-RRT* [46]	Non-holonomic	Offline	Quadrotor	Uniform	Transition test
20. RABIT* [47]	Non-holonomic	Offline	Autonomous helicopter	Uniform	A* Heuristic

V. Limitations and Prospect Challenges

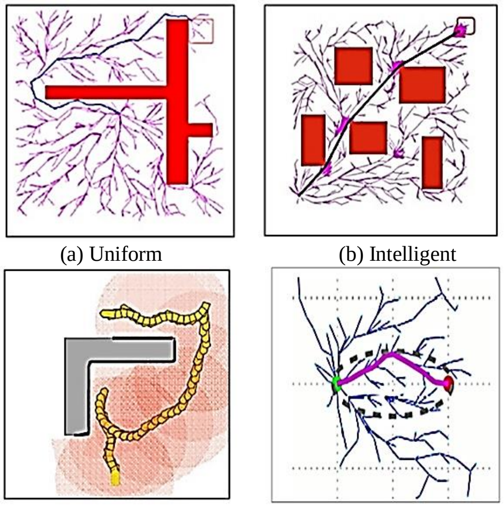
The existing state of the art requires improvement particularly in terms of accuracy, efficiency, robustness, and path optimization. Optimal path planning is a challenging problem and for online planning applications convergence to optimal path is even more important. This section describes the limitations addressed by variants of RRT* in recent years and also highlights the incessant future challenges.

A. Slow Convergence and Large Memory Requirements

RRT* requires large number of iterations and samples to avoid local minima consequently increasing memory requirements [48, 65]. Pure exploration also expands search space exponentially to find global optimum [47]. RRT* was proposed initially using uniform sampling strategy which was unable to effectively capture the connectivity of environment [7]. Further it also expands tree in the areas of configuration space that are far away from the final solution. Hence, a large number of nodes in tree are not good enough to contribute in the optimal path. These large number of nodes increase tree density by adding non-promising branches in tree. This phenomenon increases computational time and reduces convergence rate. Hence, slow convergence is also linked with search space exploration criteria and sampling strategy used by planner.

It is evident from discussion in Section III that dense sampling, large memory requirements and slow convergence are proven issues in RRT*. Recent RRT* based planners have addressed these issues by exploiting search space using different sampling strategies such as *direct sampling* [34], *goal biased sampling* [11, 56, 58], *intelligent sampling* [28, 35, 70], *two-stage sampling* [40], and *disc based sampling* [39, 42]. Few of them are shown in Fig. 6. The sampling strategies shown in Fig. 6 use different space explorations criteria. They try to limit search space using different heuristics to grow only promising branches and nodes in the tree. Another strategy reported in [48] performs guided exploration using artificial potential field to solve these issues. Moreover, different node deletion [33] or node rejection techniques [37] are also used to limit the tree cost by maintaining promising nodes according to a defined criterion.

However, these solutions especially guided exploration based solutions also require a careful balance of exploration and exploitation in search space. Moreover, sampling strategies based solutions use manual heuristics, which need specific tuning according to application or environment type for better performance. Therefore, all the strategies discussed above need further improvement regarding robustness and automation of heuristics parameters, specifically in the context of online planning.



(a) Uniform (b) Intelligent

(c) Disc biased (d) Direct

Fig. 6. Different sampling strategies to overcome slow convergence [35, 42]

B. Dealing with Narrow Passages

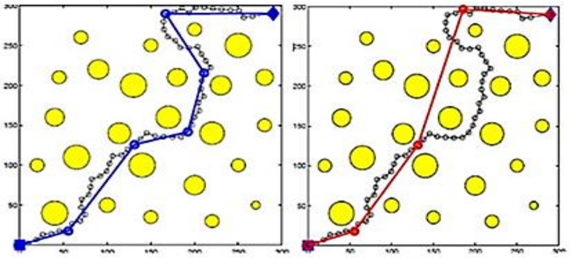
Conventional uniform sampling of RRT* reduces the probability of nodes selection from narrow passages. Very few approaches are reported to specifically address the problem of optimal path in narrow passages. These approaches either use heuristics dependent upon initial solution such as Informed RRT* [34] or they are under theoretical assumptions such as HARRT* [45]. There is a need to investigate the potential of these approaches to achieve optimization and reliability in narrow passages using real world experiments. Moreover, in context of kinodynamic constraints, problem of narrow passages is still an open research issue.

C. Efficiency of Nearest Neighbor Search

Computational complexity of near neighbor search in each iteration also grows as tree expands exponentially. Therefore, it is considered a bottleneck for efficiency and convergence. Adiyatov and Varol [33] maintained the efficiency by fixing maximum possible nodes in tree. Other strategies that most of the researchers have adopted are to use smarter search techniques such as Box approach [71] or smarter data structure such as *k-d* tree and quad trees [71]. Yershova and LaValle [72] proposed *k-d* trees based near neighbor search algorithm for Euclidean spaces. However, alternative techniques to promote least cost connections in tree or faster search algorithms could be helpful to further improve the efficiency.

D. Post Processing Requirements

As RRT* based approaches generate sub-optimal path therefore, post processing techniques are adopted to further optimize the path. Two post processing techniques usually adopted for path refinement are pruning and smoothing. Path pruning reduces the path length by removing redundant nodes [56]. Two types of pruning, local pruning and global pruning are shown in Fig. 7 (a) and Fig. 7 (b) respectively. Local pruning is based on Line of Sight (LOS) principle whereas, global pruning removes the myopic behavior of local pruning by pruning the nodes of entire path [56].



(a) Local pruning (b) Global pruning

Fig. 7. Post processing schemes [40]

Even after pruning, generated linear piece-wise path is not feasible for UAVs and car-like robots. Curvature discontinuities in path make controller unstable and overshoot energy [41]. This phenomenon can cause mechanical aging, localization errors and high energy requirements. The situation becomes more complicated when planning application involves heavy machinery such as industrial, defense or agricultural robots.

To resolve the issues stated above, different levels of smoothing such as path smoothing, continuous smoothing, and continuous curvature smoothing [41] have been applied as post processing steps. Approaches used for this purpose are classified as graphical methods (lines, arcs, circles, and clothoids) and functional methods (Bézier, B-spline and polynomial interpolation). Recent RRT* variants have used Bézier and B-spline to meet the challenges of kinodynamic planning for non-holonomic vehicles effectively in [40], see Fig. 7(c).

However, these approaches also have the limitation of maintaining continuity using suitable degree of curve. Clamped B-spline is more robust for path smoothing than Bézier and B-spline due to its ability of maintaining continuity and order for dynamic re-planning [41]. Recently Elbanhawi et al. [41] have proposed a C^2 continuous path smoothing approach using clamped B-spline for continuous steering of car-like robots. Their approach mimics human steering with high accuracy using a threshold angle and segment insertion technique. Their smoothing approach could be applied with recent RRT* variants for improved performance.

However, dealing with non-holonomic and kinodynamic constraints after trajectory generation as post processing step increases the complexity of the planning and search space. Moreover, it is also computationally expensive due to frequent updates considering real world applications and online planners.

E. *Dealing with Kinodynamic Complexities*

Non-holonomic constraints require addition of robot's orientation to state vector. Thus, increased dimension increases complexity of configuration space exploration. Moreover, they involve solving differential equations and has more complicated state transition equation. Thus, planning for non-holonomic robots with kinodynamic constraints is more difficult and challenging than holonomic robots [1, 73]. Therefore, RRT* application with non-holonomic motion requires more iterations to converge than a holonomic version [57]. Usually path planners generate linear piece-wise path ignoring all kinodynamic constraints and path smoothing is applied later as post processing step as discussed above in section 4.4. Recent state of the art [41] is inclined to apply kinematic constraint model while searching the path to avoid complexities of post processing [8, 60].

RRT* based approaches have a steering function that connects configurations using a straight line from $z_{nearest}$ to z_{rand} to generate z_{new}. However, such a steering function is not feasible for non-holonomic robots. To resolve these issue kinematic model of non-holonomic robot is used, which involves numerical integration [40]. However, there is always a trade-off between computational efficiency and accuracy when using numerical integration. Recently, Lee et al. used spline in SRRT* [44] to resolve these issues by encoding non-holonomic constraints in spline generation during path searching process of planner. This process also reduced the kinodynamic planning to lower dimensional space.

RRT* defines a metric in configuration space to identify the nearest neighbors of a given configuration. Since robots have to meet the challenges of un-certain dynamic environment, actuator constraints, localization errors and computation limitations. There is no silver bullet managing all these factors and it is difficult to distinctively define such metric. In context of non-holonomic planning, simple Euclidean distance metric is not proficient to capture cost of node in configuration space. Tree nodes need to qualify feasibility test for non-holonomic constraints, which is quite a challenging task. Lee et al. [44] have used a spline based metric in SRRT* which performs dynamic feasibility and geometric collision checks. However, the proposed approach takes several minutes in simulation and in real-time it is even more delayed [73].

Poor selection of metric for non-holonomic planning could also limit robot motion in narrow passage and slow down the planner [62]. Therefore, metric should be a true representative of the effort, or time-to-go between two configurations, otherwise highly sub-optimal paths are produced [8]. Future work in metric design solving kinodynamic problem with real time computational efficiency is an important element to improve performance of RRT*. Moreover, path smoothness could be further improved using other Computer Aided and Graphic Design (CAGD) tools such as NURBS or spiral segments. Defining a suitable weight exploitation criteria for NURBS can control local change in path segments while performing path smoothness. Cubic Bézier spiral segments reduce the number of curvature extrema in path [74]. Therefore, exploiting curvature continuous spiral transition curves could be useful to produce smooth path for high speed moving objects [75] such as UAVs.

VI. Conclusion and Future Directions

Research over the past decade has revealed that traditional path planning methods are not feasible for non-holonomic, cluttered and high dimensional problems. RRT* have proved its worth for dealing with such complex problems. This paper presents review of major contributions in optimal path planning using RRT* planning algorithm and its extended variants in recent six years. RRT* based approaches have revolutionized the state of the art in path planning. These recent variants have mostly addressed the issues of sub-optimal paths, slow convergence and high memory requirements. However, generalization and reliability in context of online planners and non-holonomic constraints are still open research issues. Though RRT* based approaches addressing kinodynamic and non-holonomic constraints are also in progress, however kinodynamic planning still confronts issues of high dimension, narrow passages and trade-off between accuracy and computational efficiency.

In recent planners, use of spline with RRT* has opened new horizon of research for path planning of non-holonomic robots. Curvature continuous path for high speed vehicles while considering non-holonomic constraints, uncertainty of dynamic environment, and preserving computational efficiency is another thriving area of future research. Use of other CAGD tools such as clamped B-spline, NURBS and spiral segments with RRT* variants for path planning of non-holonomic robots would be an interesting research endeavor in future.

REFERENCES

[1] S. M. Lavalle, Planning algorithms: Cambridge University Press, 2006.

[2] X. Lan, and S. Di Cairano, "Continuous curvature path planning for autonomous vehicle maneuvers using RRT*", presented at the European Control Conference (ECC), 2015.

[3] Alejo, J. A. Cobano, G. Heredia, J. R. Martínez-De Dios, and A. Ollero, "Efficient trajectory planning for wsn data collection with multiple UAVs", in Cooperative robots and sensor networks. vol. 604, ed: Springer International Publishing, 2015, pp. 53-75.

[4] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*", presented at the IEEE International Conference on Robotics and Automation (ICRA) 2011.

[5] *Lau, and H. H. T. Liu, "Real-time path planning algorithm for autonomous border patrol: Design, simulation, and experimentation", J Intell Robot Syst, vol. 75, pp. 517-539, 2013.*

[6] X. Kong, X. Duan, and Y. Wang, "An integrated system for planning, navigation and robotic assistance for mandible reconstruction surgery", INTEL SERV ROBOT, vol. 9, pp. 113-121, 2015.

[7] S. Karaman, and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", Int J Rob Res, vol. 30, pp. 846-894, 2011.

[8] M. Elbanhawi, and M. Simic, "Sampling-based robot motion planning: A review survey", IEEE Access, vol. 2, pp. 56-77, 2014.

[9] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance", J INTELL ROBOT SYST, vol. 57, pp. 65-100, 2009.

[10] M. Nosrati, R. Karimi, and H. A. Hasanvand, "Investigation of the * (star) search algorithms characteristics methods and approaches", World appl program, vol. 2, pp. 251-256, April 2012.

[11] J. J. Kuffner, and S. M. Lavalle, "RRT-connect: An efficient approach to single-query path planning", in Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2000, pp. 1-7.

[12] W. Dijkstra, "A note on two problems in connexion with graphs", NUMER MATH, vol. 1, pp. 269-271, 1959.

[13] Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot", in Proceedings of International Conference on Advanced Robotics (ICAR), 1993.

[14] W. Zhan, W. Wang, N. Chen, and C. Wang, "Efficient UAV path planning with multiconstraints in a 3d large battlefield environment", MATH PROBL ENG, vol. 2014, pp. 1-12, 2014.

[15] Stentz, "Optimal and efficient path planning for partially-known environments", presented at the Proceedings of the International Conference on Robotics and Automation, 1994.

[16] Nash, S. Koenig, and M. Likhachev, "Incremental phi*: Incremental any-angle path planning on grids", in International Joint Conference on Aritificial Intelligence, 2009, pp. 1824–1830.

[17] K. Sameshima, K. Nakano, T. Funato, and S. Hosokawa, "StRRT-based path planning with pso-tuned parameters for robocup soccer", AROB, vol. 19, pp. 388-393, 2014.

[18] P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", APPL SOFT COMPUT, vol. 9, pp. 1102-1110, 2009.

[19] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization", in IEEE International Conference on Neural Networks (ICNN), 1995.

[20] M. Dorigo, Ant colony optimization: Cambridge University, Massachusetts: MIT Press, 1992.

[21] T. Arora, Y. Gigras, and V. Arora, "Robotic path planning using genetic algorithm in dynamic environment", INT J COMPUT APPL, vol. 89, pp. 8-12, 2014.

[22] J. H. Liang, and C. H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm", ADV ENG SOFTW, vol. 79, pp. 47-56, 2015.

[23] M. A. Hossain, and I. Ferdous, "Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique", ROBOT AUTON SYST, vol. 64, pp. 137-141, 2015.

[24] D. Q. Zhu, W. C. Li, M. Z. Yan, and S. X. Yang, "The path planning of auv based on d-s information fusion map building and bio-inspired neural network in unknown dynamic environment", International Journal of Advanced Robotic Systems, vol. 11, March 6 2014.

[25] X. Wang, Z. Hou, F. Lv, M. Tan, and Y. Wang, "Mobile robots′ modular navigation controller using spiking neural networks", Neurocomputing, vol. 134, pp. 230-238, 2014.

[26] Fister, I. Fister, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms", Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.

[27] Kenedy, Eberhart, and Shi, Swarm intelligence: Moragan Kaufmann Division of Academic Press, 2001.

[28] *Lin, and C. Yang, "2d-span resampling of bi-RRT in dynamic path planning", Int J Autom Smart Technol, vol. 4, pp. 39-48, 2015.*

[29] I. Tsianos, I. A. Sucan, and L. E. Kavraki, "Sampling-based robot motion planning: Towards realistic applications", Comput Sci Rev, vol. 1, pp. 2-11, 2007.

[30] E. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces ", IEEE Transactions on Robotics and Automation, vol. 12, pp. 566-580, 1996.

[31] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning", 1998.

[32] Snchez, L. Zapata, J. Abraham, and A. B., "Motion planning for car-like robots using lazy probabilistic roadmap method", in Proceedings of the Second Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, 2002, pp. 1-10.

[33] O. Adiyatov, and H. A. Varol, "Rapidly-exploring random tree based memory efficient motion planning", presented at the IEEE International Conference of Mechatronics and Automation (ICMA), 2013.

[34] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic", in IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, 2014, pp. 2997-3004.

[35] J. Nasir et al., "RRT*-smart: A rapid convergence implementation of RRT*", International Journal of Advanced Robotic Systems, vol. 10, pp. 1-12, 2013.

[36] Jordan, and A. Perez, "Optimal bidirectional rapidly-exploring random trees", Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep. MIT-CSAIL-TR-2013-021, August 2013.

[37] H. Qureshi, and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments", ROBOT AUTON SYST, vol. 68, pp. 1-11, 2015.

[38] K. Yang, S. K. Gan, and S. Sukkarieh, "A gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV", ADV ROBOTICS, vol. 27, pp. 431-443, 2013.

[39] Moon, and W. Chung, "Kinodynamic planner dual-tree RRT (dt-RRT) for two-wheeled mobile robots using the rapidly exploring random tree", IEEE T IND ELECTRON, vol. 62, February 2015.

[40] K. Yang et al., "Spline-based RRT path planner for non-holonomic robots", J Intell Robot Syst, vol. 73 pp. 763-782, 2014.

[41] Elbanhawi, M. Simic, and R. Jazar, "Continuous path smoothing for car-like robots using b-spline curves", J Intell Robot Syst, pp. 1-34, 2015.

[42] Masehian, and H. Kakahaji, "Nrr: A nonholonomic random replanner for navigation of car-like robots in unknown environments", Robotica, vol. 32, pp. 1101-1123, 2014.

[43] Csorvasi, A. Nagy, and D. Kiss, "Rtr+c*cs: An effective geometric planner for car-like robots", presented at the 16th International Carpathian Control Conference (ICCC), 2015.

[44] Lee, H. Song, and D. H. Shim, "Optimal path planning based on spline-RRT* for fixed-wing UAVs operating in three-dimensional environments", presented at the 14th International Conference on Control, Automation and Systems (ICCAS 2014), Korea, 2014.

[45] Yi, M. A. Goodrich, and K. D. Seppi, "Homotopy-aware RRT* : Toward human-robot topological path-planning", presented at the The 11th ACM/IEEE International Conference on Human Robot Interaction (HRI), Christchurch, New Zealand, 2016.

[46] Devaurs, T. Simeon, and J. Cortes, "Optimal path planning in complex cost spaces with sampling-based algorithms", IEEE T AUTOM SCI ENG, vol. 13, pp. 415-424, Apr 2016.

[47] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. Scherer, "Regionally accelerated batch informed trees (rabit*): A framework to integrate local information into optimal path planning", presented at the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016.

[48] H. Qureshi, and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning", AUTON ROBOT, vol. 40, pp. 1079-1093, 2016.

[49] D. J. Webb, and J. V. D. Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics", presented at the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013

[50] Arslan, and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal path planning", presented at the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013.

[51] Y. K. Hwang, "Gross motion planning-a survey", ACM COMPUT SURV, vol. 24, pp. 219-291, 1992.

[52] Noreen, A. Khan, and Z. Habib, "A comparison of RRT, RRT* and RRT*-smart path planning algorithms", IJCSNS, vol. 16, pp. 20-27, 2016.

[53] W. Loeve, "Finding time-optimal trajectories for the resonating arm using the RRT* algorithm", Master of Science, Faculty Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, 2012.

[54] *S. Yoon, and T. H. Park, "Motion planning of autonomous mobile robots by iterative dynamic programming", INTEL SERV ROBOT, vol. 8, pp. 165-174, 2015.*

[55] D. Ferguson, and A. Stentz, "Anytime, dynamic planning in high-dimensional search spaces", presented at the International Conference on Robotics and Automation (ICRA), Italy, 2007.

[56] Yang, "Anytime synchronized-biased-greedy rapidly-exploring random tree path planning in two dimensional complex environments", INT J CONTROL AUTOM, vol. 9, pp. 750-758, 2011.

[57] R. Nguyen. (2010). Project adv - autonomous driving vehicle. Available: http://webpages.uncc.edu/nhnguye1/ADV.html

[58] B. Akgun, and M. Stilman, "Sampling heuristics for optimal motion planning in high dimension", presented at the International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011.

[59] N. Ahmidi, G. D. Hager, L. Ishii, G. L. Gallia, and M. Ishii, "Robotic path planning for surgeon skill evaluation in minimally-invasive sinus surgery", presented at the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2012.

[60] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning", J Acm, vol. 40, pp. 1048-1066, Nov 1993.

[61] T. Fraicharda, and A. Scheuerb, "From reeds and shepp's to continuous-curvature paths", IEEE T ROBOTIC AUTOM, vol. 20, 2004.

[62] S. Karaman, and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods", presented at the 49th International Conference on Decision and Control (CDC), 2010.

[63] Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Lqr-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics", in IEEE International Conference on Robotics and Automation, 2012, pp. 2537-2542.

[64] D. J. Webb, and J. V. D. Berg, "Kinodynamic RRT*: Optimal motion planning for systems with linear differential constraints", arXiv:1205.5088v1, 2012.

[65] Suh, J. Gong, and S. Oh, "Energy efficient high dimensional motion planning for humanoids using stochastic optimization", presented at the 15th International Conference on Humanoid Robots, Seoul, Korea, 2015.

[66] Kobilarov, "Cross-entropy randomized motion planning", in Proceedings of Robotics: Science and Systems, Los Angeles, CA, USA, 2011, pp. 1-8.

[67] Jaillet et al., "Sampling-based path planning on configuration-space costmaps", IEEE T ROBOT, vol. 26, pp. 635-646, 2010.

[68] H. Qureshi et al., "Potential guided directional-RRT* for accelerated motion planning in cluttered environments", in IEEE International Conference on Mechatronics and Automation, 2013, pp. 519-524.

[69] *D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs", presented at the IEEE International Conference on Robotics and Automation, 2015.*

[70] Islam, J. Nasir, U. Malik, Y. Ayaz, and O. Hasan, "RRT*-smart: Rapid convergence implementation of RRT* towards optimal solution", presented at the International Conference on Mechatronics and Automation (ICMA), Chengdu, 2012.

[71] Svenstrup, T. Bak, and H. J. Andersen, "Minimising computational complexity of the RRT algorithm a practical approach", presented at the IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.

[72] Yershova, and S. M. Lavalle, "Improving motion planning algorithms by efficient nearest-neighbor searching", IEEE T ROBOTIC AUTOM, vol. 23, pp. 151-157, 2007.

[73] S. M. Lavalle, and J. J. Kuffner, "Randomized kinodynamic planning", Int J Rob Res, vol. 20, pp. 378-400, May 2001.

[74] Z. Habib, and M. Sakai, "Fairing arc spline and designing by using cubic Bézier spiral segments", MATH MODEL ANAL, vol. 17, pp. 141-160, 2012.

[75] Z. Habib, Spiral function and its applications in CAGD: VDM, 2010.