Proceedings of the 32nd Chinese Control Conference
July 26-28, 2013, Xi'an, China

Multi-Robot Path Planning Based on the Developed RRT* Algorithm

LI Yang[1], CUI Rongxin[1], YANG Chenguang[2], XU Demin[1]

1. School of Marine Engineering, Northwestern Polytechnical University, Xi’an 710072, P.R. China
Email: rongxin.cui@gmail.com

2. School of Computing and Mathematics, University of Plymouth, Plymouth PL4 8AA, UK

Abstract: In this paper, we investigate the multi-robot path planning problem, with both the robot motion constrains and the conflictions among the robots taken into account. We have extended the RRT* algorithm by considering the motion constraints and the sensor ranges of the robots and calculate the path of the robots in real time. Simulation studies are performed to show the effectiveness of our algorithm.

Key Words: Multi-Robot, Developed RRT*, Target bias, Path planning

1 Introduction

Multiple autonomous robots, including unmanned ariel vehicles (UAVs), automatic ground vehicles (AGVs), unmanned underwater vehicles (UUVs), play an important role in military and civil fields [1–5]. A common scenario is when a group of robots are dispatched to perform tasks, each robot needs to avoid other dispatched robots while moving towards its respective destination. In this scenario, the motion coordination problem for multiple robots should be considered, which is beyond the scope of the single robot path planning problem. The inter collision/confliction between robots and the motion of several robots need to be taken into account simultaneously.

Several approaches have been proposed to solve the path planning problem for multiple robots, mainly on the collision avoidance between each pair of robots. The methods can be categorized into centralized and decentralized. In the centralized algorithms, the multi-robot systems are always being treated as a single complex system. Each robot is considered as a part of component of this complex system. Thus the multi-robot path planning problem can be solved as planning the motions of different components in the complex system. It can be addressed by searching the configuration space of the complex system [6]. A pareto-optimal coordination approach for the multi-robot path planning with safety guarantees was proposed in [7], where it divides the problem into two steps; planning the path for each robot individual followed by a velocity scheduling of robots to avoid the collisions. Because the centralized approach takes the information of all the robots, it is suitable for the off-line computation. It uses the dynamic programming to solve a global optimization problem and always can achieve an optimal solution. The disadvantage is that it is not always easy to gather all information at a central location in the practical applications. In the decentralized methods [8, 9], such as the potential field approach, each robot is treated as an individual and its motion is computed independently by treating other robots as moving obstacles. A cost based negotiation based planning algorithm was proposed for the multi-robot path planning in [10], of which the basic algorithm was to solve the planning of multiple way-points. A decentralized receding horizon approach based on the mixed integer linear programming (MILP) was proposed for the multi-aircraft path planning in [11], where a dynamically feasible trajectory for each aircraft that terminates in a loiter pattern was presented. Each aircraft takes into account the latest trajectory and loiter pattern of the other aircrafts to avoid the collisions. The decentralized algorithms are always easy to implement and can be used for the real-time computation based on the information gathered by the onboard sensors of robots. The disadvantage is that it always cannot achieve the optimal solution.

Rapidly-exploring Random Tree (RRT) is a sampling based approach solving the path planning problem, which is widely employed in [12–17]. It is always easy to find the trajectories for the multi-robot system that avoid the collision risk. If a path calculated by RRT is given, we can deduce the velocity field to make the real robot avoid the obstacles [18]. This method is suitable for the high dimensional planning problems for the practical usage. In general, RRT is an efficiency way to solve the single robot path planning problem [19–21]. A formal analysis of the complexity and performance of the RRT algorithm was recently investigated in [22]. It is proved that, under some technical conditions, the cost of the best path in the RRT converges almost surely to a non-optimal value. In order to address this drawback, a RRT* based algorithm, which preserves the asymptotic optimality of Rapidly exploring Random Graph (RRG) while maintaining a tree structure like RRT, was proposed in [22]. Motivated by [22], we propose a developed RRT* algorithm for the multi-robot path planning problem, with both the robot motion constraints and the collisions between robots taken into account. The main procedure of our method can be divided into three steps as follows.

(i) We adopt the RRT* algorithm as the basic way to sampling a random state and generate an optimal extended state in contrast of the traditional RRT.

All correspondences should be addressed to Rongxin Cui, E-mail: rongxin.cui@gmail.com. This work was supported by the National Natural Science Foundation of China under grant 51209174, the China-UK NSFC-RS International Exchange Programme under grant 51311130137 and grant RS IE121414, the State Key Laboratory of Robotics and System (HIT) under grant SKLRS-2012-MS-04, and the Fundamental Research Program of Northwestern Polytechnical University (NPU) under grant J-CY20130113.

(ii) A target biased approach was involved in the sampling state generation. When sampling the random state, we generate a random coefficient first, and determine whether the state will be extended or take the goal state. Therefore, the “Random Tree” will grow up towards the destination.

(iii) Considering the nonholonomic constraints of the robots, a region where the robots may reach in a time-interval is generated. Then, the paths consist of both position and velocity that leading the multi-robot arrive the goal states is calculated.

The remainder of the paper is organized as follows. In Section 2, some preliminaries are presented, including the motion constrains of the robots. Then, the developed RRT* algorithm is presented in detail in Section 3. In section 4, simulation results are provided to show the effectiveness of the proposed approach. Concluding remarks are given in Section 5.

2 Preliminaries

In this work, we consider that the obstacles in the work place are described by some rectangles. We use a set $S = \{p, x, y\}$, where p represents the center point of a rectangle, x represents the length of the rectangle in x-axis and y represents the width of the rectangle in y-axis to describe the obstacles.

Two constraints need to be considered in the multi-robot path planning: (i) motion constraints including the maximum velocity and acceleration, the minimum turning radius [23], and (ii) confliction among the robots [24].

In this work, we consider that the robots moves in a 2D space with boundaries. Following assumptions are made for our problem.

Assumption 1 *The robots are points in the workspace and are numbered with $P_i,\ i = 1, \ldots, n$. The collision happens if the distance between any pair of robots is less than a predefined value.*

We define the cost of robot movement starting from an initial point p_1 to the target point p_2 as $C(p_1, p_2) = L(p_1, p_2)$, where $L(p_1, p_2)$ is the Euclid distance between p_1 and p_2.

3 Developed RRT* Algorithm

3.1 The Fundamental of Developed RRT*

RRT* is one of the most widely-used sampling-based motion planning algorithms [22]. It continuously generates sample states to make the ”Path Tree” develop until any of the leaf nodes arrives the goal. This procedure can be viewed as a search on a metric space, S, for a continuous path from a given initial state $x_{init} \in S$ towards the goal area $X_{goal} \subseteq S$ or the goal state $x_{goal} \in S$ [15].

We define $X_{obs} \subset S$ as the set of the obstacles, and $X_{fre} = (S - X_{obs})$ as the free region. The Rapidly-exploring Random Tree will be constructed when all vertices belong to X_{fre}.

The tree expansion algorithm, which attempts to add one or more nodes to the tree, is described as follows. We take a random $p \in [0, 1]$ from a random number generation. If $p \geq p',\ p' \in [0, 1]$, we chose $x_{rand} = x_{goal}$, otherwise x_{rand} is selected from X, where X is bounded. We take x_{near} as the closet vertex to x_{rand} in terms of ρ, where ρ denotes the distance metric on the state space. Given input $\subset U$, we can get next state x_{new} through integrating the motion dynamic functions $F(x) : \dot{x} = f(x, u)$ over a fixed time interval, $\triangle t$. To simplify the function $F(x)$, we take $x_t = x_{t_0} + f(x, u)\triangle t$. Then we take the vertex set X'_{near} including all leaf vertexes in a given circle O, where x_{new} is the center point. We select a best vertex as the parent node of x_{new} according the robot’s dynamics and cost function. The constructions of traditional RRT and the Developed RRT* are described in Algorithm 1. In order to evaluate the complexity of the developed RRT* algorithm, we run the program with one million iterations and there are no obstacles in the environment. As we expected, the run time is less than $20s$ with one million iterations, less than $35s$ with two million iterations. It is acceptable for the practical usage.

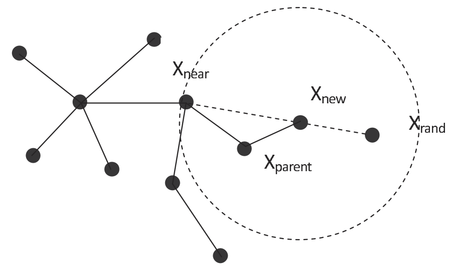

Fig. 1: Constructure of the Expanding Tree.

Algorithm 1 Traditional RRT Algorithm. [22] [25]

Require: Tree T and Iterations K
for $i = 1...K$ **do**
  Generate random x_{rand}
  Find the nearest node x_{near} from x_{rand}
  Calculate $f(x, u)$ according to x_{rand} and x_{near}
  $x_{new} = x_{near} + f(x, u)$
  Extend the leaf node $x_{near} \Rightarrow x_{new}$
end for

Algorithm 2 Developed RRT* Algorithm

Require: Tree T and Iterations K and Probability Coefficient $p' \in [0, 1]$
for $i = 1...K$ **do**
  Generate a $p \in [0, 1]$
  if $p \geq p'$ **then**
    $x_{rand} = x_{goal}$
  else
    x_{rand} Generated by random function
  end if
  Find the nearest node x_{near} from x_{rand}
  Calculate $f(x, u)$ according to x_{rand} and x_{near}
  $x_{new} = x_{near} + f(x, u)$
  Calculate the radius and generate the circle
  Pick the nearest node x'_{near} in the circle from x_{new}
  Extend the leaf node $x'_{near} \Rightarrow x_{new}$
end for

3.2 Collision Avoidance

There are two kinds of collisions, one is the static obstacles in the environment, and the other is the conflictions among the robots. As mentioned in Section 2, we divide the

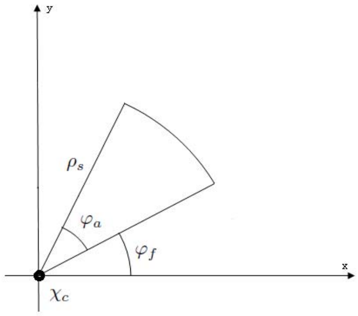

Fig. 2: The Factors of Sector.

obstacles into some rectangles, each is described by a set $S = \{p, x, y\}$, where p is the center, x and y are the length and width of the obstacle, respectively. The detection of the static obstacles is described in Algorithm 3.

Algorithm 3 Detection of Static Obstacles.

Require: Workplace boundary $x_{max}, x_{min}, y_{max}, y_{min}$ and X_{obs} and current point x_{cur}

Test $x_{cur} \in [x_{min}, x_{max}]$ and $y_{cur} \in [y_{min}, y_{max}]$
for $i = 1...n$, where n is the number of obstacles **do**
  if $x - x_{obs}.x \leq x_{obs.length}$ and $y - y_{obs.y} \leq y_{obs.width}$ **then**
    return TRUE
  end if
end for
return FALSE

To avoid the conflictions with other robots, we make a sector region to record the area where other robots might arrive in. The region is described as a set $Sector$: $\{\chi_c, \rho_s, \varphi_a, \varphi_f\}$, where χ_c represents the position of the obstacle center, ρ_s represents the radius of the sector, φ_a is the angle between the sector first edge with the x-axis anti-clockwise and φ_f is the angle of the sector. Thus, a specific sector can be defined by these four parameters. In this work, each robot take other robots as the moving obstacles in the environment. The detection of the dynamical obstacles is described as Algorithm 4.

Algorithm 4 Detection of the Dynamic Obstacles.

Require: Current point x_{cur} and the set of the other robot's sector $Sector$

Calculate the distance d between x_{cur} and $Sector.center$
if $d \leq radius$ and x_{cur} is among the edges of the sector **then**
  return TRUE
else
  return FALSE
end if

4 Simulation Results

In this section, two simulations are provided to show the effectiveness of the Developed RRT* algorithm.

Case 1: In the first simulation, we consider that there are two robots in the environment and the initial conditions of the robots are shown in Table 1.

First we consider that there is only one obstacle in the environment. The simulation result is shown in Fig. 3.

Table 1: Case1: Initial Conditions.

	Initial Point	Goal	Direction	Velocity
Robot A	$(-20, -20)$	$(20, 20)$	$0°$	0.1
Robot B	$(20, -20)$	$(-20, 20)$	$180°$	0.1

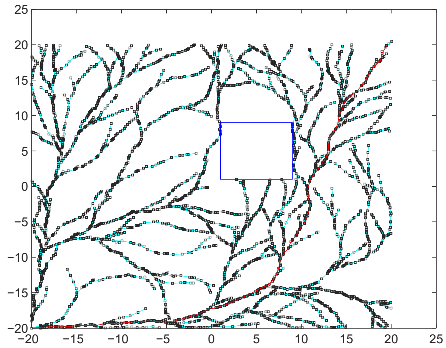

Fig. 3: Robot path: single obstacle case.

Case 2: In the second simulation, we consider that there are two or three obstacles in the environment. The result is shown in Fig. 4.

The distances between the robots along the paths are shown in Fig. 5. We can find that the minimum distance between the robots is larger than 3.5, it follows that there the collisions will not happen between the robots.

To clearly show the motion of the robots, we provide the snapshots of the simulation in Fig. 6–Fig. 12.

From the simulation results, we can find that the robots will not experience a collision with the obstacles or other robots.

5 Conclusion

In this work, a developed Rapidly-exploring Random Tree has been proposed to solve the multi-robot motion planning problem. The collisions with both the static obstacles and other robots are taken into account. A target biased algorithm has been proposed to reduce the time to find the goal states. The motion constrains of the robots have been considered in extending the tree and a sector has been provided to address the dynamic obstacles, therefore the constructing of the robots velocity field can be avoided in our algorithm. One of the future research directions is that considering more

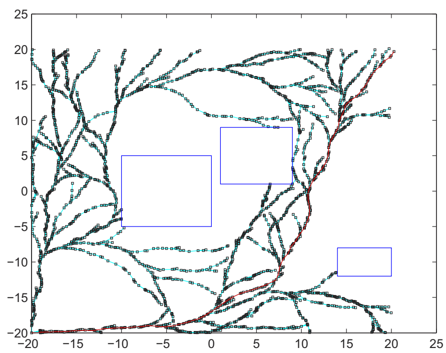

Fig. 4: Robot path: multiple obstacles case.

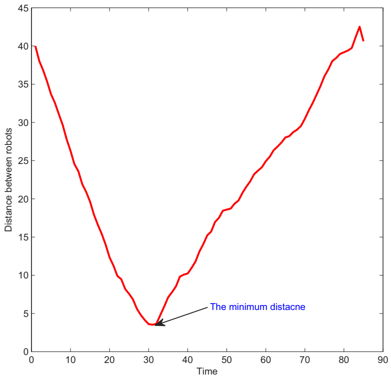

Fig. 5: The distance between two robots.

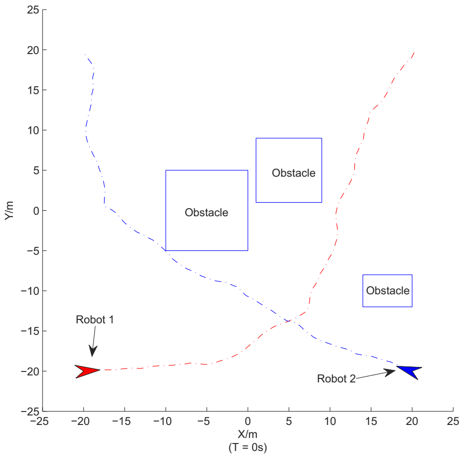

Fig. 6: Initial positions of robots: t=0s.

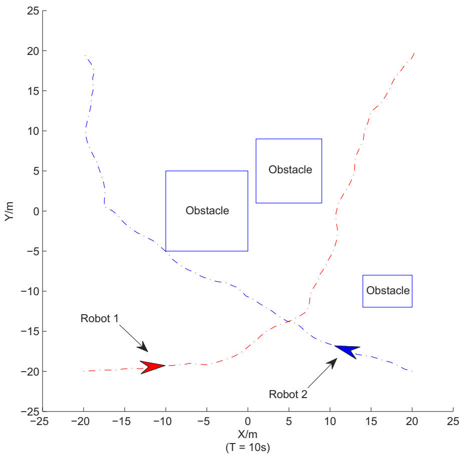

Fig. 7: Positions of robots: t=10s.

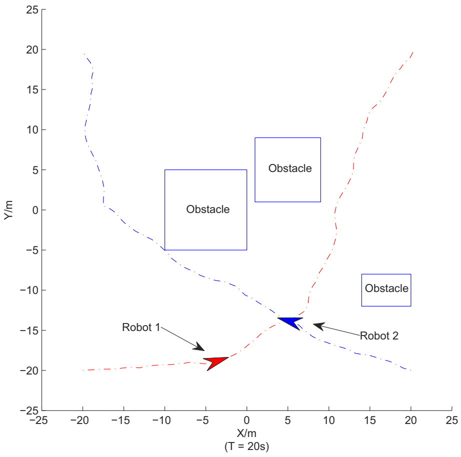

Fig. 8: Positions of robots: t=20s.

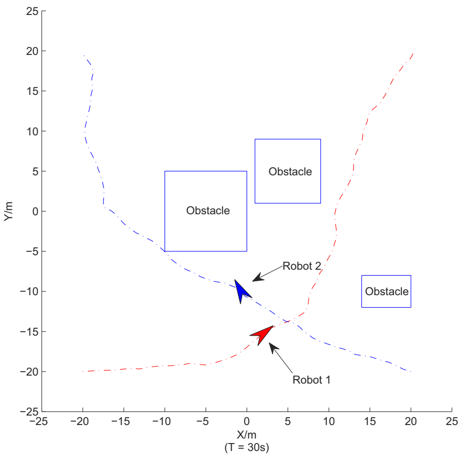

Fig. 9: Positions of robots: t=30s.

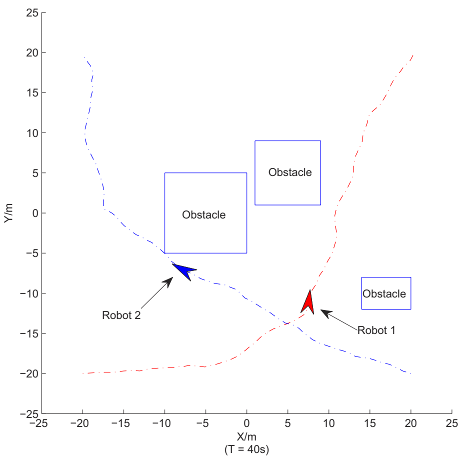

Fig. 10: Positions of robots: t=40s.

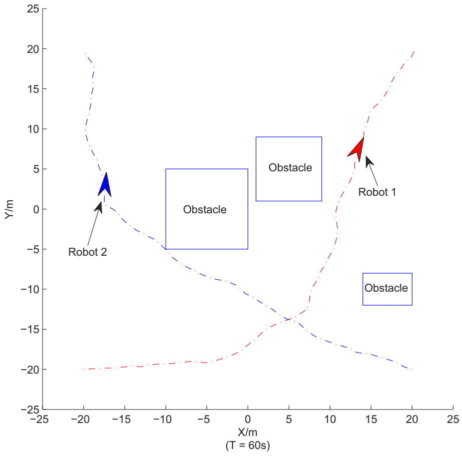

Fig. 11: Positions of robots: t=60s.

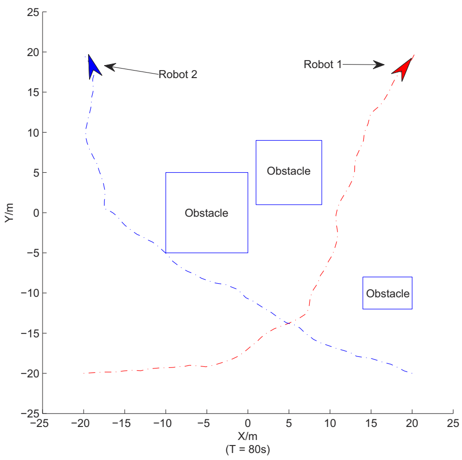

Fig. 12: Positions of robots: t=80s.

realistic constraints for the field robots, such as the wind disturbance, the energy limitation, etc.

References

[1] M. Chen and S. S. Ge, "Robust attitude control of helicopters with actuator dynamics using nerual networks," *IET Control Theory & Applications*, pp. 2837–2854, 2010.

[2] Z. Li, P. Y. Tao, S. S. Ge, M. Adams, and W. S. Wijesoma, "Robust adaptive control of cooperating mobile manipulators with relative motion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 103–116, 2009.

[3] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.

[4] R. Cui, J. Guo, and B. Gao, "Game theory-based negotiation for multiple robots task allocation," *Robotica*, 2013. [Online]. Available: http://dx.doi.org/10.1017/S0263574713000192

[5] Z. Li, X. Cao, and N. Ding, "Adaptive fuzzy control for synchronization of nonlinear teleoperators with stochastic time-varying communication delays," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 4, p. 745, 2011.

[6] J. Barraquand and J. Latombe, "Robot motion planning: A distributed representation approach," *International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.

[7] R. Cui, B. Gao, and J. Guo, "Pareto-optimal coordination of multiple robots with safety guarantees," *Autonomous Robots*, vol. 32, no. 3, pp. 189–205, 2012.

[8] M. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonomous Systems*, vol. 16, no. 2, pp. 321–332, 1995.

[9] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[10] O. Purwin, R. D'Andrea, and J. W. Lee, "Theory and implementation of path planning by negotiation for decentralized agents," *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 422–436, 2008.

[11] T. Schouwenaars, J. How, and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2004.

[12] D. Levine, "Information-rich path planning under general constraints using rapidly-exploring random trees," Ph.D. dissertation, Citeseer, 2010.

[13] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, pp. 1–23, 2012.

[14] R. Knepper and M. Mason, "Real-time informed path sampling for motion planning search," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1231–1250, 2012.

[15] S. LaValle, "Rapidly-exploring random trees a new tool for path planning," 1998.

[16] C. Urmson and R. Simmons, "Approaches for heuristically biasing rrt growth," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003*, vol. 2. IEEE, 2003, pp. 1178–1183.

[17] P. Cheng and S. LaValle, "Resolution complete rapidly-exploring random trees," in *IEEE International Conference on Robotics and Automation, ICRA'02.*, vol. 1. IEEE, 2002, pp. 267–272.

[18] M. Hoy, A. Matveev, and A. Savkin, "Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments," *Robotics and Autonomous Systems*, 2012.

[19] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2383–2388.

[20] N. Melchior and R. Simmons, "Particle rrt for path planning with uncertainty," in *IEEE International Conference on Robotics and Automation, ICRA2007*. IEEE, 2007, pp. 1617–1624.

[21] M. W. Otte, "Any-com multi-robot path planning," Ph.D. dissertation, University of Colorado of Boulder, 2012.

[22] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] J. R. Bruce, "Real-time motion planning and safe navigation in dynamic multi-robot environments," Ph.D. dissertation, Rice University, 2006.

[24] K. Savla, *Multi UAV Systems with Motion and Communication Constraints*. ProQuest, 2007.

[25] Z. W *et al.*, "Algorithm of path planning of rapidly-exploring random tree based on heuristics," *Machinery*, vol. 12, p. 003, 2007.