

Research Article

Aircraft Parking Trajectory Planning in Semistructured Environment Based on Kinodynamic Safety RRT*

Xianglei Meng , Nengjian Wang, and Qinhui Liu 

College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Qinhui Liu; liuqinhui@hrbeu.edu.cn

Received 16 August 2021; Revised 6 September 2021; Accepted 9 September 2021; Published 23 September 2021

Academic Editor: Gengxin Sun

Copyright © 2021 Xianglei Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To improve the safety and effectiveness of autonomous towing aircraft aboard the carrier deck, this study proposes a velocity-restricted path planner algorithm named as kinodynamic safety optimal rapidly exploring random tree (KS-RRT*) to plan a near time-optimal path. First, a speed map is introduced to assign different maximum allowable velocity for the sampling points in the workspace, and the traverse time is calculated along the kinodynamic connection of two sampling points. Then the near time-optimal path in the tree-structured search map can be obtained by the rewiring procedures, instead of a distance-optimal path in the original RRT* algorithm. In order to enhance the planner's performance, goal biasing scheme and fast collision checking technique are adopted in the algorithm. Since the sampling-based methods are sensitive to their parameters, simulation experiments are first conducted to determine the optimal input settings for the specific problem. The effectiveness of the proposed algorithm is validated in several common aircraft parking scenarios. Comparing with standard RRT* and human heuristic driving, KS-RRT* demonstrates a higher success rate, as well as shorter computation and trajectory time. In conclusion, KS-RRT* algorithm is suitable to generate a near time-optimal safe path for autonomous high density parking in semistructured environment.

1. Introduction

Aircraft parking aboard the carrier deck is a carefully planned procedure in naval operations standardization [1], and its safety and effectiveness are essential in determining the sortie generation capacity of air wing [2]. Currently, the aircraft movement afloat mainly relied on manually operated towing tractor with extra personnel to ensure the safety of parking route. However, the aircraft parking operation is usually manpower intensive and time-consuming evolution with low reliability, which creates a more hazard situation for the already congested deck environment leading to mishaps [3]. Therefore, it is imperative to introduce an autonomous path planner that can speed up the aircraft parking operations with major improvements in safety and reduction in total manpower.

Path planning is both a kinematic and geometric problem that specifies a set of configurations from one place to another and meanwhile avoids obstacles. And it is proved

to be PSPACE-hard problem [4], especially difficult for vehicles under nonholonomic and dynamic constraints. Considerable efforts are devoted to solve this problem. The algorithms of graph searching methods [5–7] give a path solution by discretizing the state of workspace and visiting different states, while the state space is always so large that evaluating every potential solution is computational costly. The interpolating curve-based planners [8–12] are more suitable for local planning, but they consume too much time when managing obstacles in real time. The function-based optimization methods [13, 14] can easily account for different constraints, but the function optimization happens in each state of motion that usually consumes too much time. Therefore, these methods are not suitable for quickly generating optimal path in real-time application. In contrast, the sampling-based methods are incremental sampling and random searching approaches by covering the workspace of tree structure that yield good performance in practice, which could quickly generate feasible solution in high dimensional

state space [15]. Rapidly random tree (RRT) [16] algorithm is a powerful sampling-based path planner, and it has been proved to be probabilistically complete to generate a feasible path but in no sense of being optimal. Subsequently, a variant of RRT, called RRT* [17], is proposed to achieve being asymptotically optimal by rewiring the random search tree structure for lower cost.

Now that the RRT* algorithms have been widely used in the autonomous path planning [18–23], its effectiveness in specific problem still needs to be explored. Figure 1 shows a common aircraft parking operation in a cluttered environment. The traditional sampling-based algorithm prefers the dotted straight line between the start and goal spot across the obstacles (black polygons). However, with consideration for safety, following such distance-optimal path will require much too slow speed to be safe. In this example, the true optimal path is that the aircraft can be towed around the obstacles (solid line) at its maximum allowable speed, which results in a minimum parking time.

Therefore, this study presents a variant of RRT*, called Kinodynamic Safety RRT* (KS-RRT*), which adopts a speed map to assign different reference safety velocities in the searching space and introduces a time-based metric to determine an optimal path for aircraft parking operation. In addition, to quickly generate a near optimal trajectory as the time critical nature of deck operations requires, the primitives of the RRT* algorithm are also modified, including goal bias sampling and expansion-based fast collision checking strategies. The effectiveness of KS-RRT* is estimated in multiple common aircraft parking scenarios and compared with standard RRT* and a human driving heuristic.

The remainder of this study is organized as follows. Section 2 introduces occupancy cost map combined with a speed map, describes the development of the tractor-aircraft dynamics, and proposes the modified RRT* algorithm. Section 3 evaluates the algorithm through a set of scenarios and interprets the simulation results. Section 4 concludes the paper.

2. Methods

2.1. Speed Combined Cost Map. The autonomous tractor is expected to steer the aircraft to an available parking spot. Figure 2(a) shows the operating environment as a 2D occupancy grid, including available and occupied parking spots, deck markings, and obstacles such as other aircraft (black polygons). This paper assumes such map is fully known by a flight deck monitor system [24].

For considering the safety velocity restrictions and maintaining safe clearance for the tractor-aircraft maneuvering in close-quarters, a speed map is generated for assigning different maximum allowable velocity in the environment (Figure 2(b)). According to the naval operations standards [2], we confine different velocities in the obstacle areas (v_{obs}), the safety velocity (v_{safe}) in the buffer contour d around the obstacle (making sure the towed aircraft can be slowed to an immediate stop), and the maximum velocity (v_{max}) in the sparseness areas that there are no obstacles in the vicinity.

2.2. Tractor-Aircraft Kinematics. Figure 3 shows a non-rod tractor that directly attaches to the front wheel of an aircraft, creating a tractor-aircraft articulated system. The kinematics functions of such two-body system [25] can be written as follows:

$$\left\{ \begin{array}{l} \dot{x}_a = v \cos \theta_a, \\ \dot{y}_a = v \sin \theta_a, \\ \dot{\theta}_a = \frac{v}{l_a} \tan \varphi, \\ \dot{x}_b = v \cos \theta_b \cos(\theta_a - \theta_b), \\ \dot{y}_b = v \sin \theta_b \cos(\theta_a - \theta_b), \\ \dot{\theta}_b = \frac{v}{l_b} \sin(\theta_a - \theta_b), \end{array} \right. \quad (1)$$

where (x_a, y_a) is the tractor's mass position that lies in the middle of its rear wheel axle and θ_a is the orientation of the tractor; v denotes the linear velocity of the tractor; \dot{x}_a and \dot{y}_a are the horizontal and vertical linear velocity, respectively; and $\dot{\theta}_a$ is the angular velocity of the tractor. Similarly, the aircraft's state is presented as (x_b, y_b, θ_b) . And l_a and l_b are the wheelbases of the tractor and aircraft; φ is the tractor's steering angle, and it cannot exceed its maximum rotation angle $|\varphi_{max}|$, which leads to a minimum turning radius r_{min} . Notice that the moving speed is limited by safety velocity v_{max} , and the inertial effect that brings the slipping of wheels can be ignored.

2.3. KS-RRT* Algorithm. In order to improve the safety and effectiveness, we propose a velocity-restricted KS-RRT* algorithm to plan a near time-optimal path in the semi-structured cluttered environment.

The primitives of KS-RRT* algorithm include goal bias sampling, kinodynamic local steering, expansion-based fast collision checking, and time-based rewiring procedures. Algorithm 1 outlines the KS-RRT* algorithm. At first, a single vertex is initialized in the tree graph as $G(V, E)$. For each iteration of generating random sample nodes, a node x_{rand} in free configuration space is sampled. Then, the x_{rand} chooses the nearest vertex in the tree. Along the edge $E(x_{nearest}, x_{rand})$ by local steering curve with a step size, a potential new node x_{new} is generated. Now a collision check is performed to examine whether the x_{new} is obstructed. If not, the newly generated edge $E(x_{nearest}, x_{new})$ is added to the tree. Next, the rewiring procedure works as an optimization process, by connecting from x_{new} to each potential parent nodes set X_{near} within a radius r . Comparing different routes from start position q_{start} to x_{new} , the path with minimum time cost is selected as the newly added edge. Finally, the iteration stops when a new node has reached the goal region q_{goal} . The details and modifications of the algorithm are described below.

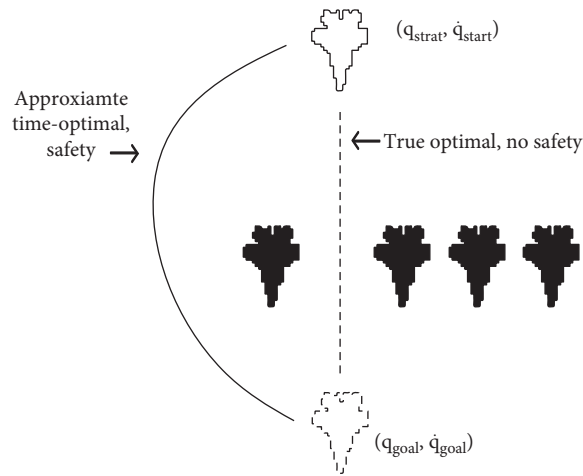


FIGURE 1: An illustration of aircraft parking scenario. The dotted line is an optimal path without considering safety. The solid line is an approximate time-optimal solution with respect of safety.

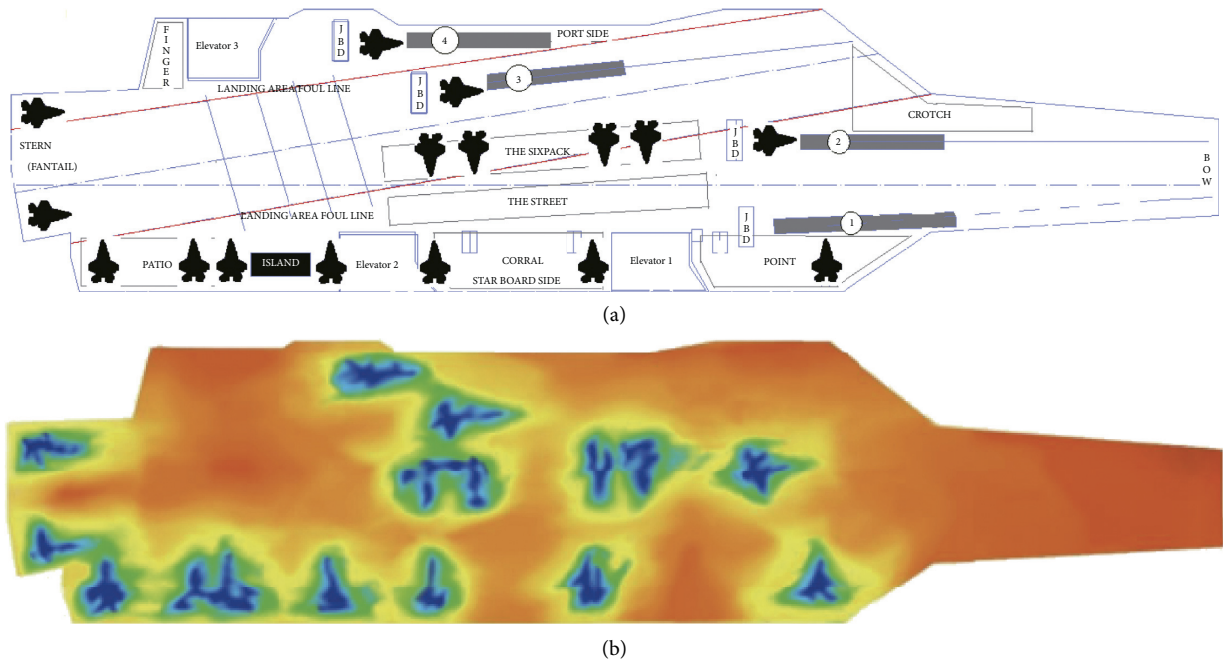


FIGURE 2: A notional aircraft carrier flight deck layout (a) and its corresponding speed map (b). Blue colour represents the velocity in the obstacle area; green colour represents the safety velocity in the buffer contour around the obstacle area; warm toned colour (yellow to orange) represents the maximum velocity in the sparseness area.

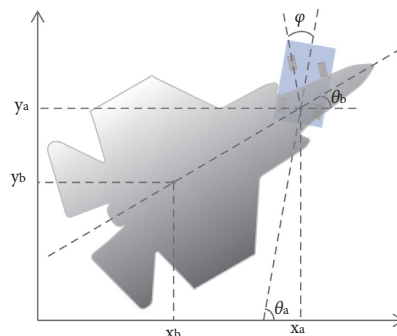


FIGURE 3: Two-body tractor-aircraft towing system. The motion of the tractor-aircraft towing system is described by equation (1).

```

Input: occupancy cost map, start and goal configuration
Output: time-optimal path connecting  $q_{start}$  and  $q_{goal}$ 
(1) Vertex  $\leftarrow \{x_{initial}\}$ ; Edge  $\leftarrow \emptyset$ ;
(2) for  $i = 1$  to  $n$  do
(3)    $x_{rand} \leftarrow$  Goalbias_Sampling;
(4)    $x_{nearest} \leftarrow$  Nearest_Connect ( $G = (V, E), x_{rand}$ );
(5)    $x_{new} \leftarrow$  Kinodynamic_Steering ( $x_{nearest}, x_{new}$ );
(6)   if Collision_free ( $x_{new}, x_{nearest}$ ) then
(7)      $Tree.add\_vertex(x_{new})$ 
(8)      $Tree.add\_edge(x_{new}, x_{nearest})$ 
(9)      $x_{near} \leftarrow$  Nearest_Neighbors ( $Tree, x_{new}, r_i$ )
(10)    for all ( $x_{near}, X_{near}$ ) do
(11)      Time-based Rewiring ( $x_{near}, x_{new}$ )
(12) return  $G = (V, E)$ ;

```

ALGORITHM 1: Kinodynamic safety RRT*.

2.3.1. Goal Bias Sampling. Sampling is the first step in exploring the workspace of the tree structure. The sampling-based algorithm originally adopts uniform sampling scheme that evenly extends towards outside. However, uniform sampling scheme may result in lower sampling chances in narrow regions of cluttered environment. Thus, we adopt a goal biasing scheme to explore the state space. A probabilistic threshold $P_{goal} \in (0, 1)$ is set, and each time before sampling, a random number $P_{rand} \in (0, 1)$ is generated. If $P_{rand} < P_{goal}$, the goal region is set to be the next sample point; else if $P_{rand} \geq P_{goal}$, a random sampling is made in the free space. Such sampling scheme has been proved to have better performance than the dense distribution [26].

2.3.2. Kinodynamic Local Steering. In the original algorithm, the subsequent way points are connected by straight lines, which may bring trouble for nonholonomic vehicles performing stationary turns due to the vehicle's turning radius constraints. Since the tractor-aircraft system can move both forwards and backwards, Reeds-Shepp curves [8] is chosen as the local connection between two sampling poses, as it has been proven to be kinodynamically feasible withholding the small-time local controllability property to reach any place in the map. However, it would generate 46 types of curves for each connection, resulting in a high cost to test every possible connection at each iteration Step [5]. Thus, we put a limit on the curvature of steering (as shown in Figure 4), where the initial and final curvature of the steering is set as zero. Such method can eliminate many connection possibilities meanwhile guaranteeing the continuous smoothness of successive curve connection. This becomes beneficial especially considering real-time implementation, as the computational complexity is significantly reduced.

2.3.3. Expansion-Based Fast Collision Checking. Collision checking is usually considered as the bottleneck of sampling-based planners [27]. Here we use the expansion-based fast collision checking method [28]. In such context, the vehicle is usually approximated as a circumscribed circle, while the

obstacles are expanded by a buffer, checking whether the circle lies on the expanded grid. However, such approximation leaves too much redundancy for path planning in cluttered environment; thus, an approximation algorithm [29] is implemented by covering the tractor-aircraft with multiple overlapping circles. Then the collision checking is simply performed by checking if these circles lie in the expanded grid. As shown in Figure 5, the shape of the aircraft towing system is approximate by 6 equal overlapping solid circles, and the obstacles (dark blue) are expanded (light blue) by a safety clearance δ .

2.3.4. Time-Based Rewiring. Before the rewiring procedure begins, the algorithm searches X_{near} , which is a vertices set within a ball of radius from x_{rand} , to find its potential time-optimal parent node. Once the path from $x_{initial}$ to x_{rand} with the least time cost is assured, x_{rand} is added to the tree. If the steering from x_{rand} has lower cost than current path, then its parent x_{near} is substituted by x_{rand} . Algorithm 2 shows the pseudocode of rewiring procedure.

The path traversing time in the tree is calculated as follows.

Figure 6 shows one of the path segments in the tree-structured map, where s is the arc length connecting the two nodes and v is the maximum allowable speed with respect of the speed map.

The k th arc length is s_k , with initial and final velocity of v_k and v_{k+1} . Assuming the movement from configuration q_k to configuration q_{k+1} is uniform acceleration (or deceleration) process, the traverse time along the path can be calculated based on the following equation:

$$s_k = \int_k^{k+1} v dt \Rightarrow t = \frac{2s_k}{v_k + v_{k+1}}, \Rightarrow t = \frac{2s_k}{v_k + v_{k+1}} \quad (2)$$

and the time-based cost function C_{time} from initial node to the k node is calculated by

$$C_{time} = \sum_0^{k-1} t_k. \quad (3)$$

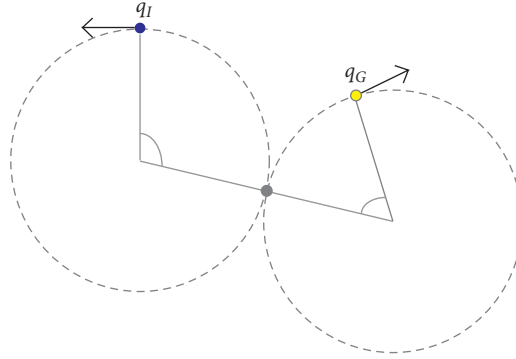


FIGURE 4: An illustration of a zero-angled curve path for one of a Reeds-Shepp family curves. Here, q_I is the initial pose and q_G is the final pose.

The benefit of the proposed algorithm is shown in the following example (Figure 7). The proposed algorithm generates a σ' path (solid line) while the standard RRT* generates a σ path (dashed line). Comparing the paths generated by the two planners, it can be noted that σ' path (solid line) is longer but safer as the aircraft does not enter the expanded contour δ region. Following σ' path with maximum allowable speed can reach the goal spot faster than that of σ path with much slower speed across the safety buffer contour.

3. Simulation and Results

In order to assess the effectiveness of KS-RRT*, the simulation experiments are analyzed through several scenarios representing common parking patterns aboard (as shown in Figure 8).

The tractor-aircraft parameters are listed in Table 1, which are defined referencing to the naval operations standardization [1].

A $300\text{ m} \times 80\text{ m}$ occupancy grid with a resolution of 0.5 m is used to represent the flight deck environment. Table 2 lists the parameters of the cost map and the reference speed map, referenced to the naval operations standardization [1].

Since the efficiency of sampling-based algorithm is heavily influenced by its resolution parameters, different parameter settings are first tested to find the suitable values that result in less computation time and better trajectory qualities. Then several parking scenarios are analyzed by comparing with standard RRT* and human driving heuristic. The simulation platform is a virtual carrier deck operations simulator we previously developed [30], as shown in Figure 9, whose input device is Logitech driving suite that can control tractor towing aircraft in the virtual environment. All the simulations are performed on a ThinkPad T450 computer (2.60 GHz Intel i7-5600 U core CPU, 4 GB RAM, Microsoft Windows 10 64 bit operating system).

3.1. KS-RRT* Parameters. Since sampling-based planners are sensitive to their embedded parameters, many parameters in the algorithm can be customized to improve their

performances. Figure 10 shows the distribution results. We give 30 independent runs for each testing to eliminate the randomized causes of sampling-based technique.

KS-RRT* is an asymptotically optimal algorithm; thus, the convergence property of the algorithm is first investigated. For each scenario, the planner was run from 1000 to 20,000 iterations, and the results were averaged for analysis. Figure 10(a) shows as expected that in all three scenarios, the trajectory time will converge to optimal value as the iteration increases, but with increased computation time. By pairwise comparison of iterations 5000 and 20000, the computation time expenses increase more than five times with less than 1% improvements in trajectory cost. Hence a run for the planner of 5000 iterations is compromised to result in a near optimal solution with least computation cost.

Then we test different step size values that influence the quality of generated trajectory (Figure 10(b)). The planner starts with a step size of 1 m . The average computation time reduces as the step size enlarges. This is because larger steps are capable of covering more free configuration space. However, the path quality will be lowered as the step size increases. A step size of 5 meters is suitable as it compensates the solution quality with exploration speed.

Goal biasing ratio takes the similar analysis. It can be seen from Figure 10(c), for scenario 1 and 3, increasing the goal biasing ratio accelerates reaching the goal, but for scenario 2 it takes much more computation time. This is because scenario 2 creates a high density of narrow passage; thus, the larger goal biasing ratio will behave like greedy algorithm that falls into circumnavigate obstacles. Increasing biasing ratio from 10% to 90% provides hardly no improvements in trajectory time. Hence, bias ratio of 10% is considered as a better biasing scheme.

3.2. Numerical Experiments. For comparing the performance of the KS-RRT* with both standard RRT* and human driving heuristic, the algorithm is implemented in the virtual deck operations simulator [30] as noted above. A human expert steers the tractor in the simulation to reach goal spot quickly meanwhile maintain the same safety protocols as human driving heuristic. Once successfully parked the aircraft to the designate spot, the total

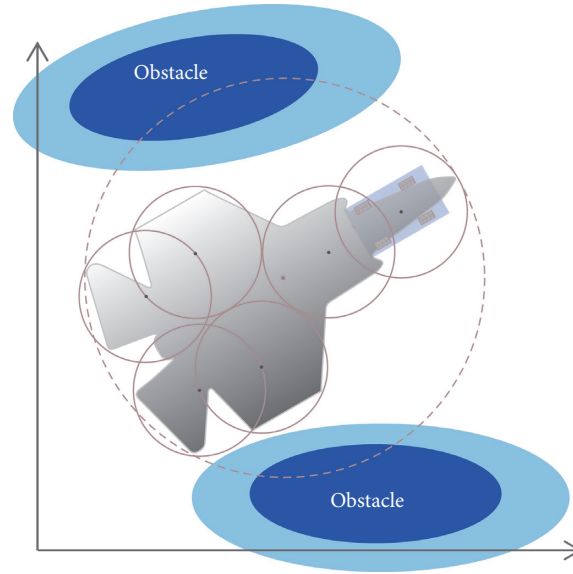


FIGURE 5: Collision checking model. The obstacle is shown in dark blue and the expanded contour is shown in light blue. The shape of the tractor-aircraft system is approximated as a circumscribed circle (dotted circle) and multiple overlapping circles (solid circles).

```

(1) for each  $x_{near} \in X_{near}$ , do
(2)    $x_{new} \leftarrow \text{Kinodynamic\_Steering}(x_{rand}, x_{near});$ 
(3)   if  $\text{Time\_Cost}(x_{rand}) + \text{Time\_Cost}(x_{new}) < \text{Cost}(x_{near})$  then
(4)      $x_{parent} \leftarrow \text{Parent}(x_{near});$ 
(5)      $E \leftarrow (E \setminus \{(x_{parent}, x_{near})\}) \cup \{(x_{rand}, x_{new})\};$ 
(6)   return  $G = (V, E);$ 

```

ALGORITHM 2: Time-based rewiring procedure.



FIGURE 6: An illustration of the k th segment connecting configuration q_k and q_{k+1} . s is the arc length and v is the velocity.

driving time from start will be recorded as the trajectory time. Driving the same scenario for 30 times will get the average trajectory time.

First, the planner is executed for 5000 iterations with a step size of 5 meters and 10% goal biasing ratio, which is the optimal parameters of previous section. Every setting runs for 30 times. The results are compared with the standard RRT* and human driving heuristic in all three scenarios.

Scene 1 is considered as easy case of sparseness environment. Aircraft starts at the bow area and is towed aft to the fantail. The average trajectory time is 156 seconds. Figure 11 shows one of the 30 running results. As shown in Table 3, all the three algorithms get similar results, this is because the aircraft can be towed at its maximum speed towards goal spot without obstacles.

Since narrow passages can cause difficulty for sampling-based path planners, Scenario 2 aims to test the algorithm in a much more congested environment of hangar deck spotting. As shown in Figure 12, the start position is at the left top and the goal position is in the

right bottom. One resulting path is given in Figure 12. The aircraft first moves slowly through the narrow passage, until it nears the final spot, and it gets back into an interim spot and then finally moves forward into the goal spot. As shown in Table 4, the human driving takes longer time due to maneuvering in close-quarters with less sensing accuracy.

Scene 3 represents the prelaunch status with many aircraft waiting to take off on the flight deck, where an aircraft tries to get to a launch spot of catapult 1 through congested street area. Figure 13 shows the result of two different homotopy types of trajectories. The KS-RRT* finds the safety time-optimal trajectory that bypasses the six-pack congested area (Figure 13(a)). In contrast, the standard RRT* prefers the shortest distance passing through the six-pack area (Figure 13(b)), without considering safety speed limits. Statistical results (Table 5) show that such a path generated by the standard RRT* would require unacceptably slow speeds to be safe, consuming more time than that generated by the KS-RRT*.

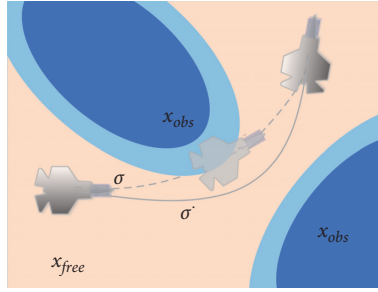


FIGURE 7: An illustration of the advantage of inducing velocity constraint in the algorithm. Our proposed algorithm generates a σ' path (solid line), while standard RRT* generates a σ path (dashed line).

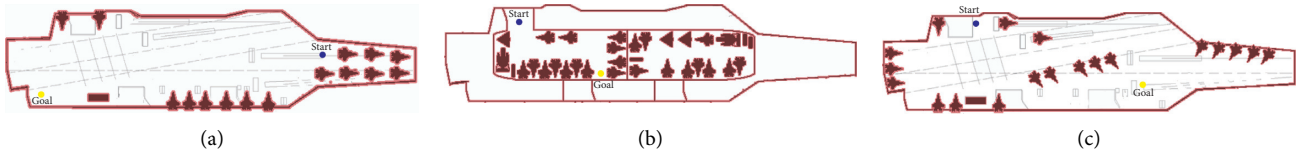


FIGURE 8: Path planning in three autonomous parking scenarios, where each scenario highlights a challenging parking situation within the cyclic flight operations afloat in congested environment. (a) Postflight parking. (b) Hangar deck parking. (c) Preflight parking.

TABLE 1: Tractor-aircraft parameters.

Parameter	Value
Minimum turning radius (r)	12 m
Length (l)	15.3 m
Wheel base (L)	8 m
Width (w)	6 m
Maximum velocity (v)	1.5 m/s
Maximum acceleration (a)	2 m/s ²
Maximum steering angle (φ_{\max})	$\pi/4$

TABLE 2: The map parameters.

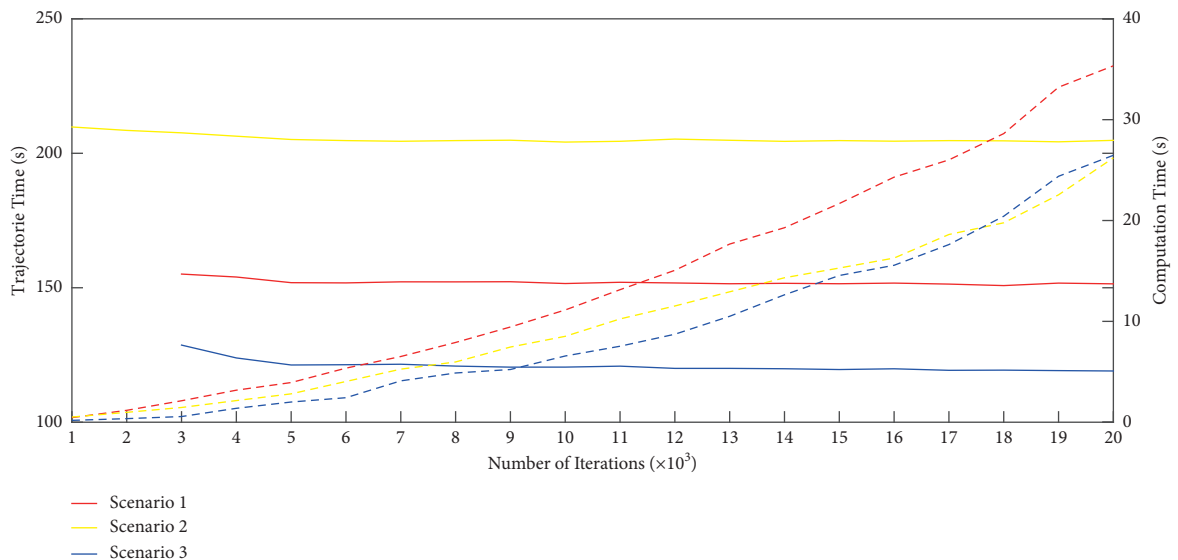
Parameter	Value
Map size	300 m \times 80 m
Cell resolution	0.5 m
Goal tolerance	0.5 m
Hard obstacle margin	0.2 m
Safety buffer δ	1.5 m
$v_{free}/v_{safe}/v_{obs}$	1.5/0.2/0 m/s

Throughout the experiments, the KS-RRT* shows its ability of generating a robust safety time-optimal trajectory, despite the complexity of the environment. Compared to the standard RRT*, the KS-RRT* has faster computation and trajectory time and a higher success rate in a congested environment without

the need for further speed profile generation. Besides, it also outperforms the human driving heuristic in terms of safety trajectory time with a dramatic reduction in total manpower. Hence, the KS-RRT* is more suitable for automated aircraft towing in the carrier deck environment.

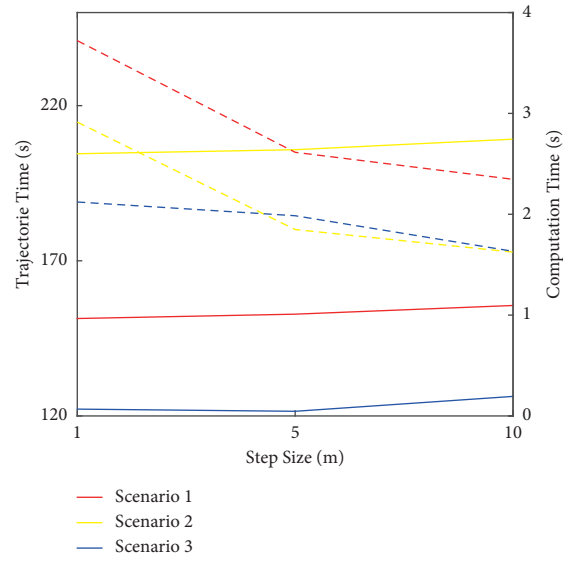


FIGURE 9: The carrier deck operation simulator environment. (a) The Logitech driving input device. (b) The simulated 3D tractor towing aircraft. (c) The tractor-aircraft movement and path generated (green line).

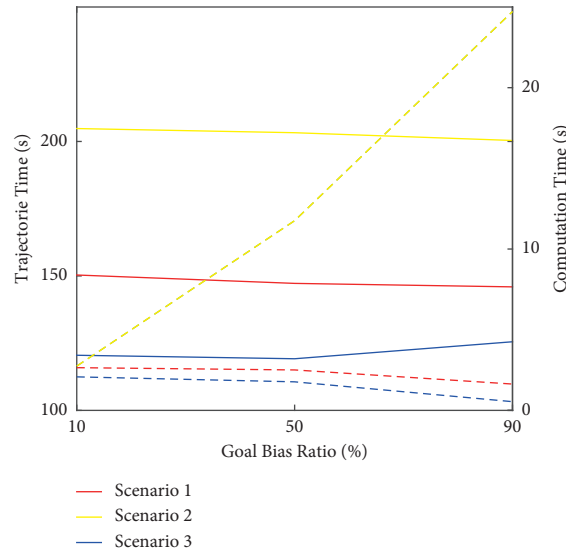


(a)

FIGURE 10: Continued.



(b)



(c)

FIGURE 10: Simulation results of parameter comparison in different scenarios. (a) Convergence test. (b) Step size test. (c) Goal bias test. The solid lines represent the cost of trajectory time (left vertical axis) and the dashed lines represent the computation time (right vertical axis).

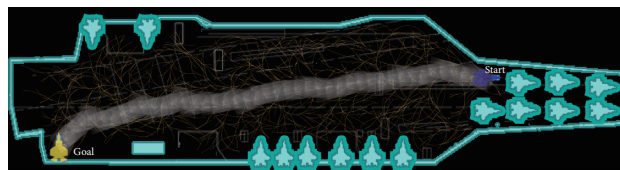


FIGURE 11: One sample result of aircraft-tractor foot print with respect to the planned route in Scenario 1. Start and goal spots are shown as blue and yellow colour, respectively.

TABLE 3: Simulation statistical results of Scenario 1.

Simulation	Success rate (%)	Average computation time (s)	Average trajectory time (s)
KS-RRT*	100	2.6	156
RRT*	100	4.2	165
Human driving	100	—	160

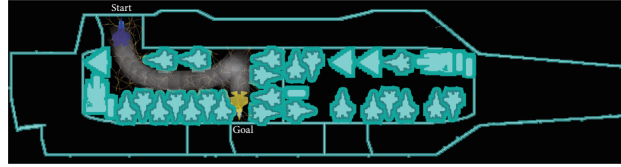
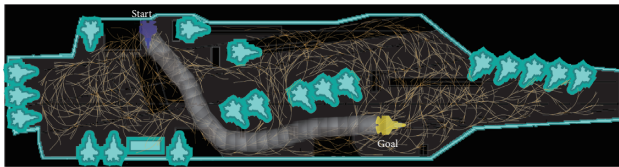


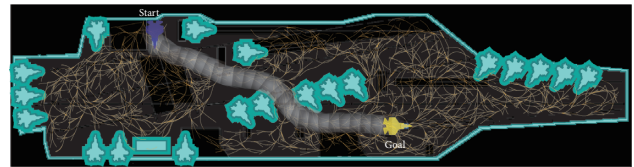
FIGURE 12: One sample result of Scenario 2. Start and goal spots are shown as blue and yellow colour, respectively.

TABLE 4: Simulation statistical results of Scenario 2.

Simulation	Success rate (%)	Average computation time (s)	Average trajectory time (s)
KS-RRT*	100	2.3	213
RRT*	84	7	231
Human driving	95	-	255



(a)



(b)

FIGURE 13: One simulation result of Scenario 3. (a) A near time-optimal trajectory generated by the KS-RRT. (b) A solution found by the standard RRT. Start and goal spots are shown as blue and yellow colour, respectively.

TABLE 5: Simulation statistical results of Scenario 3.

Simulation	Success rate (%)	Average computation time (s)	Average trajectory time (s)
KS-RRT*	100	1.5	122
RRT*	97	2.2	166.7
Human driving	100	-	145

4. Conclusions

In this paper, a modified version of RRT* algorithm was proposed to quickly generate a near time-optimal safe path for tractor-aircraft parking in cluttered onboard environments. The proposed method integrates a safety speed map in the configuration space, which allows for rewiring the tree structure in the algorithm by calculating minimum travelling time cost. The proposed KS-RRT* path planner is validated through numerical experiments. The experimental results also show that the KS-RRT* algorithm performs better than the standard RRT* and human driving heuristic. However, the spotting operations aboard might be operated concurrently in the real-world situations, which will be multirobot motion planning problem that leads to new problems such as interference of different moving body, towing sequence, arising more serious safety issues. In the near future, our planner can be fit from single query to more

complicated scenarios, such as the multirobot collaboration problem that makes them cooperate to optimize the overall traffic.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the “National Ministries and Commissions Foundation of China, grant no. 614200301030217.

References

- [1] R. L. Mahr, "Cvn flight/hangar deck natops manual," 2010, <https://info.publicintelligence.net/USNavy-CVN-FlightDeck.pdf>.
- [2] A. Jewell, *Sortie Generation Capacity of Embarked Airwings*, Center For Naval Analyses Alexandria Va, Arlington, VA, USA, 1998.
- [3] K. H. Roberts, G. I. Rochlin, and T. R. La Porte, "The self-designing high reliability organization: aircraft carrier flight operations at sea," *Naval War College Review*, vol. Autumn, pp. 1–16, 1987.
- [4] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proceedings of the IEEE Symposium on Foundations of Computer Science*, Rome, Italy, October 1979.
- [5] S. M. Lavalle, *Planning Algorithms*, Cambridge University Press, Cambridge, UK, 2006.
- [6] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.
- [7] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Munich, Germany, September 2005.
- [8] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [9] J. Horst and A. Barbera, "Trajectory generation for an on-road autonomous vehicle," *Unmanned systems technology VIII*, vol. 6230, p. 62302J, 2006.
- [10] M. Brezak and I. Petrović, "Real-time approximation of clothoids with bounded error for path planning applications," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 507–515, 2013.
- [11] A. Simon and J. C. Becker, "Vehicle guidance for an autonomous vehicle," in *Proceedings of the 1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 429–434, Tokyo Japan, October 1999.
- [12] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G/sup 2/-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.
- [13] J. Jing Ren, K. A. McIsaac, and R. V. Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384–391, 2006.
- [14] N. P. Singh and V. P. Singh, "Efficient segmentation and registration of retinal image using gumble probability distribution and brisk feature," *Traitement du Signal*, vol. 37, no. 5, pp. 855–864, 2020.
- [15] H. M. Choset, K. M. Lynch, S. Hutchinson et al., *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, Cambridge, MA, USA, 2005.
- [16] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [17] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Proceedings of the 49th IEEE conference on decision and control (CDC)*, pp. 7681–7687, Atlanta, GA, USA, December 2010.
- [18] O. Adiyatov and H. A. Varol, "A novel RRT*-based algorithm for motion planning in dynamic environments," in *Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1416–1421, Takamatsu, Japan, August 2017.
- [19] Y. Du, Y. Wang, X. Zhang, and Z. Nie, "Automatic separation management between multiple unmanned aircraft vehicles in uncertain dynamic airspace based on trajectory prediction," *Revue d'Intelligence Artificielle*, vol. 33, no. 3, pp. 171–180, 2019.
- [20] W. Wu, W. Liu, F. N. Zhang, and V. Dixit, "A new flexible parking reservation scheme for the morning commute under limited parking supplies," *Networks and Spatial Economics*, 2021, in press.
- [21] J. Tavooosi, "A new type-2 fuzzy sliding mode control for longitudinal aerodynamic parameters of a commercial aircraft," *Journal Européen des Systèmes Automatisés*, vol. 53, no. 4, pp. 479–485, 2020.
- [22] Z. Tang, B. Chen, and R. Lan, "Vector field guided rrt* based on motion primitives for quadrotor kinodynamic planning," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 3, pp. 1325–1339, 2020.
- [23] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: an improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, Article ID 113425, 2020.
- [24] A. S. Michels and T. Thate, *Requirements for Digitized Aircraft Spotting (Ouija) Board for Use on US Navy Aircraft Carriers*, Naval Postgraduate School, Monterey CA, USA, 2002.
- [25] J. S. Johnston and E. D. Swenson, "Feasibility study of global-positioning-system-based aircraft-carrier flight-deck persistent monitoring system," *Journal of Aircraft*, vol. 47, no. 5, pp. 1624–1635, 2010.
- [26] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for rapid replanning in dynamic environments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1603–1609, Roma, Italy, April 2007.
- [27] J. Ziegler, M. Werling, and J. Schroder, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*, pp. 787–791, Eindhoven, Netherlands, June 2008.
- [28] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium*, pp. 518–522, La Jolla, CA, USA, June 2008.
- [29] G. K. Das, S. Das, S. C. Nandy, and B. P. Sinha, "Efficient algorithm for placing a given number of base stations to cover a convex region," *Journal of Parallel and Distributed Computing*, vol. 66, no. 11, pp. 1353–1358, 2006.
- [30] N. Wang, X. Meng, Q. Liu, and J. Li, "High level architecture based simulation for aircraft carrier deck operations," in *Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 765–769, Xi'an, China, October 2016.