Letter

Collision and Deadlock Avoidance in Multi-Robot Systems Based on Glued Nodes

Zichao Xing, Xinyu Chen, Xingkai Wang, Weimin Wu,
Senior Member, IEEE, and Ruifen Hu

Dear editor,

In this letter, we would like to discuss a method to avoid collisions and deadlocks in multi-robot systems based on a new concept of glued nodes.

In terms of collision and deadlock avoidance, many methods are based on zone control which has two disadvantages. First, unless all nodes are collision-free, the roadmap must be divided into disjoint zones, which increases the difficulty of applying the methods. Moreover, each zone should be able to accommodate a robot, which leads to imprecision and waste of space. This letter proposes the concept of glued nodes, which can dynamically determine the mutual influence between nodes based on the real-time sizes and paths of the robots. Based on the glued nodes, this letter proposes a collision and deadlock avoidance algorithm, which can be applied to multi-robot systems with variable-sized robots and roadmaps with any structure. The experimental results indicate that the method proposed in this letter is effective and efficient.

Introduction: Multi-robot systems have been studied to apply in many areas to help people perform dangerous and tedious tasks, such as environmental monitoring, disaster rescue, and minefield mapping. As a main content of motion coordination, collision and deadlock avoidance is receiving increasing attention.

In multi-robot systems, collisions damage the robots and cause losses, while deadlocks cause the system stagnation until the deadlocks are solved manually or automatically. In industrial environments like workshops and container terminals, the roads on which robots can travel are usually planned in advance. These roads make up a roadmap, wherein an edge represents a road or part of a road, and a node represents an intersection or a point on the road. However, the compact design of some roadmaps means that two nodes are not necessarily collision-free for two robots. Especially when robots are variable-sized, i.e., the sizes of the robots are different before and after loading, two nodes are collision-free for two robots when the robots are not loaded, and they are not collision-free when both robots are loaded. That is, whether two nodes are collision-free for two robots depends on the structure of the roadmap and the size of the robots.

The arbitrariness of the roadmap structure and the variable-sized robots bring difficulties to collision and deadlock avoidance in multi-robot systems. However, the dynamic nature of the glue nodes can solve this problem. Besides, the dynamic nature makes the collision and deadlock avoidance method proposed in this latter more adaptable and more accurate than those methods based on zone control [1]−[5].

Related work: Significant studies have been devoted to addressing

the collision and deadlock avoidance problem in recent years. The essence of collision avoidance is to prevent several robots from appearing in the same space at the same time. There are generally two ways to avoid collisions, one is the coupling method, which statically or dynamically plans collision-free paths or trajectories for robots, such as state lattice, reciprocal collision avoidance [6], etc. The other is the decoupling method, which decouples path planning and motion coordination. The method first plans paths for the robots with a certain goal (such as the shortest distance, least congestion, etc.), and then solves the collision problem based on the existing paths. Wang *et al.* [7] set different initial delays for different robots to avoid collisions. However, this type of method is time sensitive and not robust. In [8], the authors propose a petri-net controller to avoid collisions for automated guided vehicles based on zone control.

It is not enough that robots can move without collisions in multi-robot systems because deadlocks will cause partial or complete stagnation of the system. There are three major approaches for deadlock handling in multi-robot systems: deadlock detection and resolution, deadlock prevention and deadlock avoidance [9].

As an online method, deadlock avoidance has been the focus of numerous studies. In [1], the authors give a limit that robots can only stop on arcs, each arc represents a zone. If a robot can find an intermediate position that is not occupied by the paths of other robots, it can move to this intermediate position to avoid deadlocks. Zhou *et al.* [2] model robot motion through labeled transition systems, and design a distributed algorithm to avoid deadlocks. The method is based on fixed path scenarios, where all the paths of robots are fixed. In [3], the authors divide the transport road network into non-overlapping zones and introduce a structural online control policy which guarantees that the execution of an elementary transport operation does not lead to deadlock. Malopolski [4] proposes a deadlock avoidance method based on chains of reservations, which is suitable for transportation systems with square structures. In [5], the authors propose a spare zone based hierarchical motion coordination algorithm to avoid deadlocks by locally adjusting the paths of robots.

Although these methods are effective in some scenarios, they explicitly or implicitly divide the environment into several disjoint zones, which reduces the adaptability of the methods.

Contributions: 1) This letter proposes the concept of glued nodes, based on which, a novel method is proposed to avoid collisions and deadlocks in multi-robot systems. 2) The proposed method can avoid collisions and deadlocks for heterogeneous robots and variable-sized robots in a roadmap with any structure.

Problem statement: In a multi-robot system based on a roadmap, motion coordination is the process of nodes allocation in the roadmap. There is a control center responsible for the assignment of nodes, and each robot applies to the control center for nodes in real time. Only nodes which do not cause collisions and deadlocks can be authorized to the robot and become its occupied nodes.

Let $\mathbb{N} = \{1, 2, \ldots, N\}$ be the index of robots, N is the number of the robots. $R_{\mathbb{N}}$ denotes all robots in the system and $R_i, i \in \mathbb{N}$ denotes a robot in $R_{\mathbb{N}}$. The roadmap in the system is denoted as $G = (V, E)$. $V_m, m \in \mathbb{M}$ denotes a node where a robot can reach and stop, $\mathbb{M} = \{1, 2, \ldots, M\}$ is the index of nodes. $E_{m,n} = (V_m, V_n)$ denotes an edge between V_m and V_n, $E_{m,n}$ and $E_{n,m}$ are not the same edge. These edges can be unidirectional or bidirectional.

Problem: Given a multi-robot system with a roadmap, find an online control method to avoid collisions and deadlocks during the movement of robots.

Glued nodes: This section gives definitions related to the glued nodes.

The path of R_i is denoted as $P_i = \{V_1, E_{1,2}, \ldots, E_{k-1,k}, V_k\}$. As shown in Fig. 1, $P_1 = \{V_1, E_{1,2}, V_2, E_{2,3}, V_3, E_{3,4}, V_4, E_{4,8}, V_8\}$ denotes the path of R_1. Although the paths of the two robots are not intersect at V_2, they may collide with each other when R_1 reach V_2 and R_2
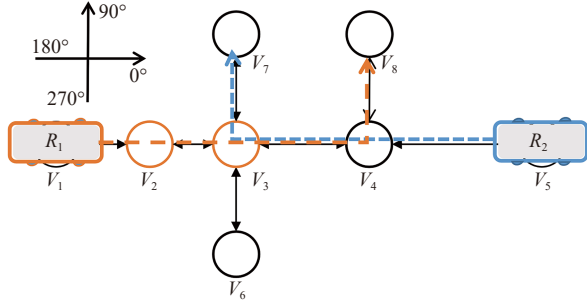
Fig. 1. An example of the robots and their paths, V_2 is authorized to R_1.

reach V_3.

Definition 1 (node action): A node action of R_i is a four tuple $\Gamma_i^m = \langle \,^\bullet E_m, \alpha_i^m, V_m, \beta_i^m \rangle$, where

1) $\,^\bullet E_m$ is the preceding edge of V_m in P_i.

2) α_i^m is the angle when R_i reaches V_m along $\,^\bullet E_m$ or the angle of R_i if V_m is the first node of P_i.

3) V_m is a node where R_i is currently located or a node on its path.

4) β_i^m is the angle when R_i leaves V_m or the angle is equal to α_i^m if V_m is the last node of P_i.

Definition 2 (action area): An action area of R_i, denoted as Θ_i^m, is the area swept by the robot when performing a node action Γ_i^m.

Definition 3 (glued nodes): $\forall m, n \in \mathbb{M}_M$, $\forall i, j \in \mathbb{N}_N$, $V_m \in P_i$, $V_n \in P_j$, if $\Theta_i^m \cap \Theta_j^n \neq \emptyset$, then V_m and V_n are a pair of glued nodes for R_i and R_j, denoted as $GN_{m,n}^{i,j} = 1$. Otherwise, $GN_{m,n}^{i,j} = 0$.

As shown in Fig. 1, R_2 will enter V_3 at angle $180°$ and then rotates $90°$ clockwise, there is $\Gamma_2^3 = \langle E_{4,3}, 180°, V_3, 90° \rangle$. Similarly, $\Gamma_1^2 = \langle E_{1,2}, 0°, V_2, 0° \rangle$. When R_1 and R_2 perform Γ_1^2 and Γ_2^3 respectively, because V_2 and V_3 are close to each other, $\Theta_1^2 \cap \Theta_2^3 \neq \emptyset$, such that $GN_{2,3}^{1,2} = 1$.

Proposed algorithm: Based on the concept of glued nodes, this section presents the collision and deadlock avoidance algorithm.

The applying nodes and occupied nodes of a robot R_i are respectively denoted as AV_i and OV_i. Once a node is occupied by a robot, the robot can move to this node along the path. Therefore, when a robot applies to the control center for a node, it needs to determine whether the authorization of the node to the robot will cause collisions or deadlocks.

Algorithm 1 Collision Avoidance: $CA(R_i, AV_i)$

Input: Robot R_i and its applying nodes AV_i;
Output: AN; /*Nodes without collisions.*/
1   Initialization: $AN = \emptyset$;
2   **foreach** V_m in AV_i **do**
3   **if** $O_m \neq \emptyset$ **then**
4   **return** AN;
5   **foreach** R_j in $R_{\mathbb{N}}$ **do**
6   **if** $R_j \neq R_i$ **then**
7   **foreach** V_n in OV_j **do**
8   **if** $GN_{m,n}^{i,j} = 1$ **then**
9   **return** AN;
10   add V_m to AN;
11   **return** AN;

Theorem 1: For $\forall V_m \in OV_i$ and $\forall V_n \in OV_j$, if $OV_i \cap OV_j = \emptyset$ and $GN_{m,n}^{i,j} = 0$, then R_i and R_j are collision-free.

Proof: There are only two types of collision in a roadmap: collision on the same node, collision not on the same node. $OV_i \cap OV_j = \emptyset$ means a node will not be occupied by R_i and R_j at the same time, so it is impossible for two robots to appear at the same node at the same time and collide with each other. If $GN_{m,n}^{i,j} = 0$, there is no overlap in the areas swept by R_i and R_j when performing Γ_i^m and Γ_j^n, so R_i will not collide with R_j. ∎

$O_m = R_i$ denotes V_m is authorized to R_i. Based on Theorem 1, the collision avoidance algorithm is shown in Algorithm 1. AV_i is the nodes applied by R_i, and each node is checked in turn. If a node is already occupied by other robots, the node cannot be authorized (Lines 3 and 4). Furthermore, if a node is a pair of glued nodes with another node that have been authorized to other robots, it cannot be authorized to the robot (Lines 5−9). Line 11 returns the nodes that can be authorized to R_i. The complexity of the algorithm is $O(NHK)$, in which N is the number of robots, H and K are the maximum number of nodes in the applying nodes and occupied nodes of the robots, respectively.

However, these collision-free nodes are not necessarily deadlock-free. A deadlock avoidance method needs to be designed to avoid deadlocks among robots.

Definition 4 (conflict node): $\forall m, n \in \mathbb{M}_M$, $\forall i, j \in \mathbb{N}_N$, $V_m \in P_i$, $V_n \in P_j$, if $V_m = V_n$ or $GN_{m,n}^{i,j} = 1$, then V_m and V_n are conflict nodes of R_i and R_j.

Definition 5 (conflict area): $\forall i, j \in \mathbb{N}_N$, a conflict area $\Phi_{i,j}$ is a collection of all conflict nodes of R_i and R_j.

Definition 6 (area occupation): If $\{\exists V_m | V_m \in OV_i, V_m \in \Phi_{i,j}\}$, then $\Phi_{i,j}$ is occupied by R_i, which is denoted as $R_i \to \Phi_{i,j}$.

As shown in Fig. 1, $\Phi_{1,2} = \{V_2, V_3, V_4\}$, V_2 is authorized to R_1, then there is $R_1 \to \Phi_{1,2}$.

Definition 7 (conflict circle): A conflict circle is a sequence of conflict occupation like $(R_1 \to \Phi_{1,2}, R_2 \to \Phi_{2,3}, \ldots, R_n \to \Phi_{n,1})$.

Theorem 2: There is no deadlock if there is no conflict circle.

Proof: According to Definitions 4 and 5, a conflict area is actually a common space swept by two robots. Once a robot occupies a conflict area, it may induce another robot to avoid collisions and stop. A conflict circle means a circular avoidance may take place. If multiple robots do not form a conflict circle, then at least one of these robots can move without being blocked, and a deadlock can not occur. ∎

When the nodes occupied by the robots change, the conflict areas and area occupations will be automatically updated. The collision and deadlock avoidance algorithm is shown in Algorithm 2.

Algorithm 2 Collision and Deadlock Avoidance: $CDA(R_i, AV_i)$

Input: Robot R_i and its applying nodes AV_i.
Output: DAN; /*Nodes without collisions and deadlocks.*/
1   Initialization: $DAN = \emptyset$;
2    $AN = CA(R_i, AV_i)$;
3    $V_s = \emptyset$;
4    $flag = $ true;
5   **foreach** V_m in AN **do**
6   **if** V_m not in any conflict areas **then**
7        $V_s = V_m$;
8        $flag = $ false;
9   **foreach** V_m in AN **do**
10   **if** $flag = $ false **then**
11   **if** $V_m = V_s$ **then**
12          $flag = $ true;
13   add V_m to DAN;
14   **continue;**
15   add V_m to OV_i;
16   **if** $R_i \to \Phi_{i,j}$ **then**
17   **if** a conflict circle containing $R_i \to \Phi_{i,j}$ is generated **then**
18   **break;**
19   **else**
20   add V_m to DAN;
21   **else** /* V_m is not in any conflict areas.*/
22   add V_m to DAN;
23   **return** DAN;

Line 2 calls Algorithm 1 to obtain collision-free nodes, and then the algorithm judges whether these nodes are deadlock-free. Lines 3−14 search for the farthest node among these nodes that is not in conflict areas, and the robot can move to this node without deadlock. Lines 16−22 judge whether the authorized nodes form a conflict circle, and only those nodes that do not form a conflict circle can be

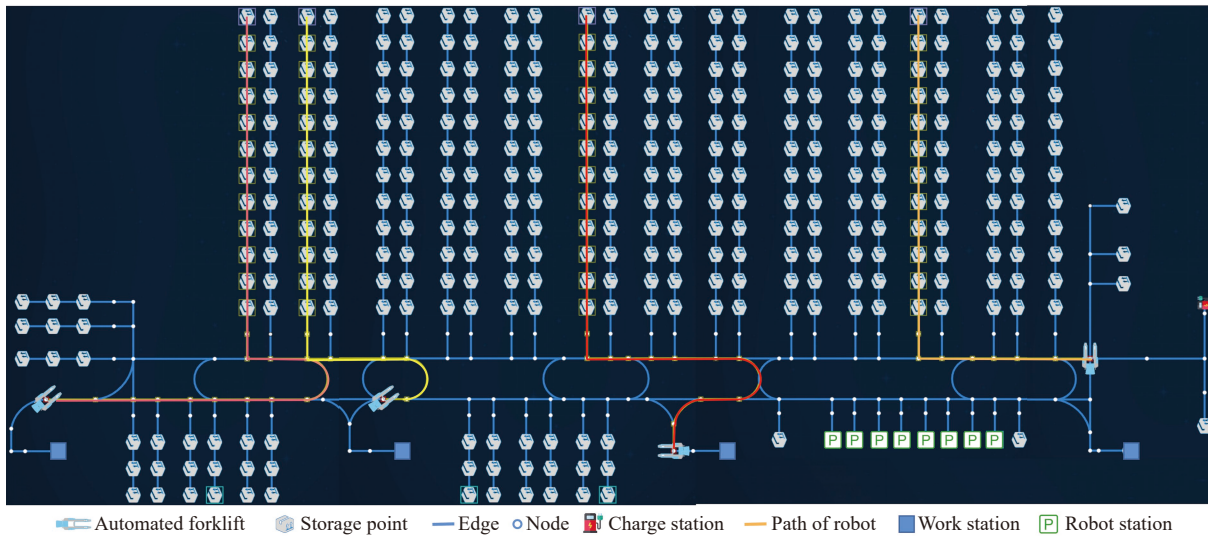Automated forklift　Storage point　— Edge　○ Node　Charge station　— Path of robot　Work station　P Robot station

Fig. 2. Simulation in an automated warehousing scenario.

occupied by the robot. In the worst case, all nodes are in conflict areas. Therefore, in Lines 6 and 16, the complexity of judging whether a node is in a conflict area is $O(M)$, in which M is the number of nodes in G. The occupation relationship of the conflict areas among robots can form a directed graph, and the complexity of finding the circles in the directed graph is $O(V+E)$ [10], V and E are the number of nodes and edges in the graph, respectively (Lines 17). Therefore, the complexity of the algorithm is $O(HM(V+E))$.

Experiments: Our experiments run on a desktop running Windows 10, equipped with an Intel(R) Xeon(R) Platinum 8280L 2.6 GHz and 64 GB of RAM.

To evaluate the performance of the proposed algorithm, this letter conducts experiments based on a real automated warehousing scenario. As shown in Fig. 2, automated forklifts perform transport tasks between work stations and storage points, the parameters of robots are shown in Table 1.

Table 1. Experimental Parameters of Robots

Parameter	Meaning	Value
$(ul \times ud)$	The size of a robot when it is not loaded.	(1.64 m×1.2 m)
$(ll \times ld)$	The size of a robot when it is loaded.	(1.8 m×1.4 m)
L	Total number of nodes in AV and OV for each robot.	5
Spd	Maximum moving speed of the robots.	1.8 m/s
Acc	Acceleration of the robots.	1.2 m/s^2
Dec	Deceleration of the robots.	−0.8 m/s^2

The proposed method based on glue nodes (GN) is compared with the following methods: Structural on-line control policy (SOCP) [3], spare zone based hierarchical motion coordination (SZH) [5] and method based on chains of reservations (COR) [4]. Each set of experiments was set up with 50 outbound tasks (the robots transport cargos from storage points to work stations) and 50 inbound tasks (the robots transport cargos from work stations to storage points). Tasks are always allocated to nearest idle robots. Each task consists of four subtasks: 1) move to a work station (or a storage point); 2) load cargos; 3) move to a storage point (or a work station) with cargos; 4) unload cargos.

Three different indicators are used to evaluate the performance: average task execution time, average waiting time of the robots and the total mileages when robots execute tasks. The experimental results are shown in Table 2.

In terms of average task execution time, GN performs much better than other methods. When the number of robots is 8, GN consume 41.01%, 34.09% and 28.06% less time than COR, SOCP and SZH, respectively. This is because other methods cause more waiting time for the robots to avoid collisions and deadlocks than GN. Since the number of tasks remains the same in each experiment, the average task execution time reflects that GN is more efficient than other methods. In terms of total mileage, when using GN, the robot traveled less mileage (less energy consumption) than using other methods.

Both the average task execution time and the average waiting time increase with the number of robots, because more robots mean more congested traffic, and robots are more likely to wait to avoid collisions and deadlocks.

Conclusion: This letter proposes the concept of glued nodes in roadmaps. Based on glued nodes, a collision and deadlock avoidance method in multi-robot systems is proposed. The method can be applied to any roadmap without dividing the environment like the methods based on zone control. Experimental results show that the proposed method is more effective than several other methods. Anymore, the method proposed in this letter has been applied to many industrial projects, like manufacturing, warehousing and container terminal. For a detailed explanation of some concepts in this letter, please see the appendix. In the future, we will study the efficient method of calculating glued nodes and apply the method to

Table 2. Simulation Results

#Robots	Average task execution time (s)				Average waiting time (s)				Total mileage (m)			
	GN	COR	SOCP	SZH	GN	COR	SOCP	SZH	GN	COR	SOCP	SZH
2	**84.69**	103.21	100.46	88.65	**4.19**	24.77	19.21	8.63	**8580.42**	10041.1	10034.68	8597.26
4	**98.54**	135.91	121.17	114.67	**16.21**	52.09	38.77	31.07	**7965.21**	9322.29	9337.53	8023.47
6	**121.36**	195.68	178.82	168.98	**40.75**	114.73	100.02	88.08	**7601.92**	8894.24	8872.78	7782.35
8	**153.07**	259.5	232.25	212.76	**73.48**	181.06	150.65	131.04	**7492.51**	8746.23	8759.16	7710.24

large-scale scenarios.

References

[1] L. Kalinovcic, T. Petrovic, S. Bogdan, and V. Bobanac, "Modified banker's algorithm for scheduling in multi-AGV systems," in *Proc. IEEE Int. Conf. Automation Science and Engineering*, 2011, pp. 351–356.

[2] Y. Zhou, H. Hu, Y. Liu, S.-W. Lin, and Z. Ding, "A distributed method to avoid higher-order deadlocks in multi-robot systems," *Automatica*, vol. 112, p. 108706, 2020. DOI: 10.1016/j.automatica.2019.108706.

[3] J. Zajac and W. Malopolski, " Structural on-line control policy for collision and deadlock resolution in multi-AGV systems," *J. Manufacturing Systems*, vol. 60, pp. 80–92, 2021.

[4] W. Malopolski, "A sustainable and conflict-free operation of AGVs in a square topology," *Computers & Industrial Engineering*, vol. 126, pp. 472–481, 2018.

[5] Y. Zhao, X. Liu, S. Wu, and W. Gang, "Spare zone based hierarchical motion coordination for multi-AGV systems," *Simulation Modelling Practice and Theory*, vol. 109, p. 102294, 2021. DOI: 10.1016/j.simpat. 2021.102294.

[6] C. Zu, C. Yang, J. Wang, W. Gao, D. Cao, and F.-Y. Wang, "Simulation and field testing of multiple vehicles collision avoidance algorithms," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1045–1063, 2020. DOI: 10.1109/JAS.2020.1003246.

[7] X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, "Collision avoidance of mobile robots by using initial time delays," in *Proc. 54th IEEE Conf. Decision and Control*, 2015, pp. 324–329.

[8] J. Luo, Y. Wan, W. Wu, and Z. Li, " Optimal petri-net controller for avoiding collisions in a class of automated guided vehicle systems," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4526–4537, 2020.

[9] E. Coffman, M. Elphick, and A. Shoshani, "System deadlocks," *ACM Computing Surveys (CSUR)*, vol. 3, no. 2, pp. 67–78, 1971.

[10] R. Tarjan, " Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.