

# Decentralized Multi-Robot Collision Avoidance

Subjects: Computer Science, Artificial Intelligence

Contributor: Abdul Hadi Abd Rahman, Mehak Raibail, Mohammad Faizul Nasrudin, Mohd Shahrul Mohd Nadzir, Nor Mohd Razif Noraini, Siok Yee Tan

When deploying a multi-robot system, it is ensured that the hardware parts do not collide with each other or the surroundings, especially in symmetric environments. Two types of methods are used for collision avoidance: centralized and decentralized. The decentralized approach has mainly been used in recent times, as it is computationally less expensive.

Keywords: decentralized ; multi-robot ; collision avoidance

---

## 1. Introduction

The use of autonomous mobile robots has increased in various fields of life in recent times. Mobile robots are being used in industry, medicine, search and rescue, and other applications. Most of these applications require a team of robots to fulfill the task efficiently and in less time than their human counterparts <sup>[1]</sup>. However, they face some issues hindering them from optimal path planning due to the symmetrical shape of the environment. Multiple robots are expected to explore more areas in less time while solving robot localization and collision-avoidance issues. In a scenario where a team of mobile robots is operating, it is necessary to keep them safe from collisions and the surroundings. Collision avoidance for a multi-robot system forms the focus of many current studies. When deploying a multi-robot system, it is ensured that the hardware parts do not collide with each other or the surroundings, especially in symmetric environment. There are two main approaches used to address collision avoidance in multi-robot systems: centralized and decentralized <sup>[2]</sup>. The centralized approach is efficient only for small groups of robots, and for large groups, the decentralized approach is more effective, as it is less expensive computationally. A decentralized method that uses a triangular grid pattern was introduced by Yang <sup>[3]</sup>, using previously explored maps and information from neighbors to avoid collisions.

The agents need a communication system to inform them about their current position and velocity in a decentralized approach. Various types of communication setups have been introduced in recent years. The Alternating Direction Method of Multipliers (ADMM) used by Rey et al. <sup>[4]</sup> works so that each agent communicates with the neighboring agents, creating a local coordinate system. The setup can manage transmission from all the agents simultaneously. Being a decentralized network, it can manage frequent setup changes and failures. Rodríguez et al. <sup>[5]</sup> used the velocity information of each agent to create a cooperative communication setup. The relative velocity of agents was then used to decrease an individual agent's detection region, resulting in collision avoidance. A scheme suggesting the use of a shared memory-driven device for coordination and communication was presented by Pesce et al. <sup>[6]</sup>. Each agent utilizes the shared device to learn from the collective private observations and share relevant information with other agents. Deep reinforcement learning and Voronoi cell setup were proposed in Wang et al. <sup>[7]</sup> and Nguyen et al. <sup>[8]</sup> to ensure cooperative behavior among the agents. Another approach towards collision avoidance is by using a local or global path planner. These approaches use path-planning strategies such as GMapping <sup>[9]</sup> to avoid collisions <sup>[10][11]</sup>. Socially aware path-planning strategies have gained popularity recently. Avoidance of robot-to-pedestrian collisions, human-like speed of motion, and navigation through dense environments require carefully planned trajectories <sup>[12][13]</sup>. When planning a trajectory, the structure of the environment also plays a vital role in collision avoidance. It is essential to explore the environment carefully to maintain a low time and energy cost while reaching the target. Some task planner techniques have been presented to address this issue <sup>[14]</sup>. However, very few studies have discussed cluttered and confined environments <sup>[15][16]</sup>.

In the process of exploring the databases for this SLR study, it was observed that most of the studies on collision avoidance are conducted for aerial vehicles <sup>[17][18][19]</sup>, surface vehicles <sup>[20][21][22]</sup>, ships <sup>[23][24][25][26][27]</sup> and underwater vehicles <sup>[15][28][29]</sup>. However, the focus of this SLR is only on ground vehicles. By carefully scanning the titles and abstracts of the studies, any study using other than ground vehicles was excluded. In this SLR, 17 studies from 2015 to date are selected after systematic screening. All these studies address the collision-avoidance issue for multi-robots (ground

vehicles) directly or indirectly. These studies propose new algorithms such as CADRL or innovate classic techniques such as ORCA to avoid collisions. Although various studies were conducted before 2015, they rarely addressed the issue of collision avoidance in a decentralized manner for confined spaces. Many studies attempted to resolve the said issue for no more than two agents. In this SLR, researchers select only those studies which use decentralized approaches for a large group of agents in confined space scenarios. Some studies experimented with up to a hundred agents in different scenarios [2], while others considered techniques such as ECBS for a group of agents [30].

## 2. Characteristics of Collision Avoidance in Multi-Robot System Techniques

This section discusses RQ2, involving the main characteristics of various collision-avoidance techniques using reinforcement learning (RL) and non-reinforcement learning (non-RL) for a multi-robot system. It was observed that only a few studies address the said issue directly, while others propose a path-planning algorithm that avoids collisions. Each study considered the hardware specifications of the agent to implement the proposed model effectively. Different real-world scenarios were evaluated to narrow the simulation and real transition gap. Following is a summary of each related study. The summary of the technique experimental design, benchmark, strength, and weakness on all selected studies are summarized in **Table 1**.

**Table 1.** Synthesis analysis of decentralized collision avoidance.

References	Technique	No of Agents	Experimental Design	Benchmark	Strength	Weakness
<b>Non-RL approaches</b>						
[31]	Continuous Best-Response Approach	4	Indoor but open space maps	ORCA	Shorter paths	Prolongation issue
[30]	Push and rotate with bounded-suboptimal ECBS	4	Open Space	ORCA	Higher success rate	N/A
[32]	Voronoi cells superimposed with RVO cones	4, 10, 25, 70	Open space	ORCA	Passive collision avoidance guarantees	Not optimized
[33]	Safe Zone Discrete-ORCA-MPC	4	Open space	PID, PB-NC-DMPC	Non-holonomic robots	Not suitable for the unknown scenario
[34]	Alturistic coordination	8	Indoor and confined spaces	PID	Local adjustments capability	Ignore low-level control parameters
[35]	Modified MR-DFS strategy	2	Open space	MR-DFS	Reduces edge traversals	Only static obstacles
[5]	Velocity-based detection regions	Up to 20	Open space	Constant detection radii	Yield faster and less conservative trajectories	N/A
[36]	Spot Auction-based Robotic Collision Avoidance Scheme (SPARCAS)	Up to 500	Open space	M*	Prioritization and dynamic handling	Not optimized
[37]	Independent virtual center points	2, 6, 8	Open space	Monte Carlo	An arbitrary number of agents	N/A
<b>RL approaches</b>						
[13]	CADRL	2, 4, 6, 8	Open space	ORCA	Higher path quality	Stuck in a dense setting
[38]	RL-ORCA	Up to 42	Open space	ORCA	Congestion and deadlock capability	N/A
[2]	Hybrid-RL	Up to 100	Open space with obstacles	NH-ORCA, RL	High success rate and generalization capability	Not socially compliant

References	Technique	No of Agents	Experimental Design	Benchmark	Strength	Weakness
[12]	Multi-sensor Fusion and DRL	4	Open and confined space	A*, D*	Dense environment	Oscillatory behavior in a highly spacious environment
[39]	Deep Q learning—CNN	4	Open and confined space	A*, D*		Unnecessary movement occurs
[40]	End-to-end DRL policy	Team of 3	Open space	Ind-PPO, MAPPO	Navigate through a dense environment	Unable to address deadlock situation
[41]	CBF-MARL	2	Open space	MADDPG	Safety guarantees	Only static obstacles

For non-RL approaches, eight different studies are selected with various agents in both open and confined spaces. Most studies used ORCA as their benchmark evaluation. Cap et al. [31] address the problem of finding a collision-free trajectory for an agent in a dynamic environment. The setup considered is an infrastructure with agents already performing tasks when a new task is assigned to an individual agent. The proposed algorithm implements a token system and plans a global trajectory considering all the agents. Meanwhile, Dergachev et al. [30] suggest coordinating sub-groups of the agents that appear to be deadlocked, using locally confined multi-agent pathfinding (MAPF) solvers. The limitation of this model is its assumption that each agent has prior knowledge of the environment and its failure to perform in uncertain situations. Arul et al. [32] used buffered Voronoi cells (BVC) and reciprocal velocity obstacles (RVO) to develop a collision-avoidance method for dense environments. To calculate a local collision-free path for each agent, first, a suitable direction is computed by superimposing BVC and RVO cones together. However, this method does not guarantee deadlock resolution, similar to earlier decentralized methods—more studies focus on alternative approaches to avoid the need for global communication among robots [31][34][42].

Mao et al. [33] presented a collision-avoidance approach by considering the non-holonomic constraints of the agents. The proposed method is cheaper than PB-NC-DMPC, as it does not use central coordination or rely on communication among the robots. Another study by Wei et al. [34] proposed altruistic coordination where each robot is ready to make concessions whenever in congested situations. It is demonstrated that when robots face a congested situation, they can implement waiting, moving forward, dodging, retreating, and turning-head strategies to make local adjustments. Another approach using robot graph exploration is proposed by Nagavarapu et al. [35], in which no direct communication is needed to avoid collisions. A data structure is proposed to provide efficient information exchange. Modified Multi-Robot Depth First Search (MR-DFS) strategy is used to achieve better execution than other tree strategies. Zhang et al. [37], suggest a technique using two cooperative strategies to decrease the effective detection regions of the vehicles, for a random number (large number) of agents, using velocity information. Another approach using prioritization is presented by Das et al. [36], where agents intentionally disclose their information in order to become prioritized. Competitive robots take part in spot auctions, where they show their willingness to pay the price to obtain access to the desired location. The results show that the proposed method can manage dynamic arrival without compromising the path-length optimality too much. Zhang et al. [37] propose an obstacle-avoidance method that incorporates virtual center points, implemented in a distributed manner, which is set based on the current state of the nearby robots and the agent itself. The stability of the system is proved using a Lyapunov function. Two control modes—the obstacle-free mode and obstacle avoidance mode—are used for robots, which are switched carefully using a direct signal.

Researchers have applied several reinforcement learning approaches for decentralized collision avoidance. Chen et al. [13] developed an innovative method that applies deep reinforcement learning to offload the online computation to an offline learning procedure to predict interaction patterns. A value network that uses each agent's joint configuration is developed to estimate the time to the goal. This value network also finds a collision-free velocity vector by admitting the efficient queries while considering other agents' motion uncertainty. However, some robots become stuck when the obstacle field is dense such that various traps and dead ends are formed. One effective method to resolve the dead-end issue is presented by [40]. Meanwhile, Li et al. [38] presented a continuous action space-based algorithm. In this method, only the positions and velocities of nearby agents are observed by each agent. A solution of simple convex optimization with safety constraints from ORCA is implemented to resolve the multi-robot collision-avoidance problem. The training process of the proposed approach is much faster than other RL-based collision-avoidance algorithms. Fan et al. [2] designed a decentralized sensor-level collision-avoidance policy. The movement velocity for an agent's steering commands is directly drawn from raw sensor measurements. The technique used here is policy-gradient-based reinforcement learning. The said technique is integrated into a hybrid control framework to improve the policy's robustness

and effectiveness. Liang et al. [12] used a depth camera with 2D LIDAR as multiple perception sensors to detect nearby dynamic agents and calculate collision-free velocities. The previously unseen virtual and dense real-world environment is directly transferable from the navigation model learned by the agents. However, in the case of glass or nonplanar surfaces, the sensors fail to perform accurately.

Bae et al. [39] suggested a combination of Deep q-learning and CNN algorithm. This combination enhances the learning algorithm and analyzes the situation more efficiently. Depending on the given situation, the agents can act independently or collaboratively. The memory regeneration technique is used to reuse experience data and reduce correlations between samples to improve data efficiency. The presented method uses image-processing techniques such as object recognition [43][44] to obtain the robot's location. However, an unnecessary movement occurs in an environment where the generated path is simple or without obstacles. Lin et al. [40] propose a method with a geometric centroid of the robot team, which avoids collisions while maintaining connectivity using Deep RL. The proposed model can sometimes fail to predict the dead-end scenario effectively, which can cause the agent formation to take extra time to reach the goal. Cai et al. [41] suggest a combination of Multi-robot Reinforcement Learning MARL and decentralized Control Barrier Function (CBF) shields based on available local information. They developed a Multi-robot Deep Deterministic Policy Gradient (MADDPG) to Multi-robot Deep Deterministic Policy Gradient with decentralized multiple Control Barrier Functions (MADDPGCBF). Each agent has its unique CBFs according to the proposed approach. These CBFs involve cooperative CBFs and non-cooperative CBFs, which deal with the respective types of agents.

### **3. Collision-Avoidance Techniques to Solve Coordination Issues**

To answer RQ3, researcher analyzed the primary works presenting collision-avoidance techniques for multi-robot systems while considering the coordination issues. The techniques studied in this entry are focused on presenting solutions to the said problem for a group of more than two agents. Many studies on collision avoidance in a multi-robot system made a list when first searched. Most of these studies addressed manipulators or warehouse scenarios, which were unsuitable for this entry. Studies that addressed the type of agents other than ground robots were excluded, which resulted in around 121 studies. These studies were further screened by excluding those that focused on aspects of the multi-robot system other than collision avoidance, such as trajectory optimization or navigation issues [10][45]. Studies that addressed only the coordination issue without considering collision avoidance were excluded [7]. The final screening resulted in 17 carefully selected studies for relevance to the topic, each presenting a unique collision-avoidance strategy.

Collision avoidance is an essential part to be considered when dealing with multi-robot systems. It includes collisions with obstacles and among the agents. Two types of methods most often used are centralized and decentralized approaches. The decentralized approach is computationally inexpensive and enables the agent to be more independent. Another division is classical and reactive approaches [46]. However, researchers provide information about effective decentralized techniques, classical or reactive. This entry is designed according to the guidelines provided by Okoli [47], which resulted in the final selection of 17 papers. All the studies included are focused on finding and developing an effective strategy to avoid collisions in a multi-robot system. Several methods, including deep reinforcement learning, fuzzy logic, and supervised learning, are used as a base to validate or apply the proposed strategies. Most of the studies are focused on application in an open-space environment with static or dynamic obstacles, while confined-space scenarios are not often studied. Environments such as AC ducts and sewers need to be explored even more, as these environments offer different types of hurdles for a multi-robot system compared to open-space environments. Only 2 of the 17 studies addressed the deadlock situation [2][5], which can appear when agents need to swap their positions or cross a narrow entrance, while others fail to perform in such scenarios [42]. Some of the crucial criteria for decentralized multi-robot collision avoidance are summarized as the following:

*Coordination strategy:* Several efficient coordination systems have produced successful collision avoidance for multi-robot systems. The velocity obstacle method allows robots to transmit and receive each other's states and intentions via an altruistic coordination network. A token-passing technique based on a synchronized shared memory holds all robots' current trajectories, which learns a value function that completely encodes collaborative behaviors. By utilizing the processed data from the LIDAR sensor, agents coordinate with others via a robot team that works as a centroid or beacon to exchange information among the robots. In the decentralized control barrier function, local information received through the agent's sensors can be shared by other robots nearby.

*Traversable region detection region:* A unique application of the locally confined multi-robot pathfinding (MAPF) solvers is suggested by Dergachev et al. [30]. This approach presents a way to build a grid-based MAPF instance to avoid deadlocks. Through the learned policies, the robots use local observations of each robot and traversable detection region to collaboratively plan the move to accomplish the team's navigation task. A proper data-structure-based technique is

essential for providing efficient information exchange, as suggested by Nagavarapu et al. [35]. Combining 2D multiple perception sensors using LIDAR and depth cameras enabled the agents to sense the dynamic agents in the surroundings and compute collision-free velocities. An approach using Voronoi cells and RVO cones provides an efficient calculation of collision-free direction for each agent. On the other hand, exploiting the velocity information in the proposed technique resulted in less complicated and collision-free trajectories by the traversable region-detection region. Furthermore, incorporating virtual center points implemented in a distributed manner should be set based on the current state of nearby robots and the agent itself.

*Optimal multi-robot trajectories:* While ensuring optimal collision-free navigation, agents are also required to maintain a coordination link within their coordinated network to lower the communication overhead in decentralized systems. One of the essential factors is detecting nearby dynamic agents and calculating collision-free velocities. Early detection and trajectory prediction will result in a higher success rate with less time and trajectory length taken to the goal. Detection of the obstacle's direction and assuming it to be the leader, the agent can be trained to maintain a predefined distance using formation control.

*Adaptability:* One of the essential criteria for successful application is adaptability, especially in deadlock situations that often occur in multi-robot systems. Few studies faced a problem where agents become stuck when deployed in a dense environment. In some studies, the problem of failing to navigate in an uncertain environment also arose. In some scenarios, only static obstacles are considered while training the agents, making it challenging to address inter-agent collisions. Overall, it is observed that not a single study was able to address all the issues faced when designing a collision-avoidance algorithm. However, deadlock and inter-agent collisions are two main issues that need to be addressed to develop an efficient collision-avoidance model.

---

## References

1. Stączek, P.; Pizoń, J.; Danilczuk, W.; Gola, A. A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment—A case study. *Sensors* 2021, 21, 7830.
2. Fan, T.; Long, P.; Liu, W.; Pan, J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot. Res.* 2020, 39, 856–892.
3. Yang, X. A decentralized algorithm for collision-free search tasks by multiple robots in 3D areas. In *Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017, Macau, Macao, 5–8 December 2017*; pp. 2502–2507.
4. Rey, F.; Pan, Z.; Hauswirth, A.; Lygeros, J. Fully Decentralized ADMM for Coordination and Collision Avoidance. In *Proceedings of the 2018 European Control Conference, ECC 2018, Limassol, Cyprus, 12–15 June 2018*; pp. 825–830.
5. Rodriguez-Seda, E.J.; Stipanovic, D.M. Cooperative avoidance control with velocity-based detection regions. *IEEE Control. Syst. Lett.* 2020, 4, 432–437.
6. Pesce, E.; Montana, G. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. *Mach. Learn.* 2020, 109, 1727–1747.
7. Wang, D.; Deng, H.; Pan, Z. MRCDRL: Multi-robot coordination with deep reinforcement learning. *Neurocomputing* 2020, 406, 68–76.
8. Nguyen, M.T.; Maniu, C.S.; Olaru, S. Decentralized constructive collision avoidance for multi-agent dynamical systems. In *Proceedings of the 2016 European Control Conference, ECC 2016, Aalborg, Denmark, 29 June–1 July 2016*; pp. 1526–1531.
9. Lee, W.C.; Salam, A.S.A.; Ibrahim, M.F.; Rahni, A.A.A.; Mohamed, A.Z. Autonomous industrial tank floor inspection robot. In *Proceedings of the IEEE 2015 International Conference on Signal and Image Processing Applications, ICSIPA 2015-Proceedings, Kuala Lumpur, Malaysia, 19–21 October 2016*; pp. 473–475.
10. Wang, Y.; Wang, D.; Mihankhah, E. Navigation of multiple mobile robots in unknown environments using a new decentralized navigation function. In *Proceedings of the 2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016, Phuket, Thailand, 13–15 November 2016*; pp. 13–15.
11. Xiong, X.; Wang, J.; Zhang, F.; Li, K. Combining Deep Reinforcement Learning and Safety Based Control for Autonomous Driving. 2016. Available online: <https://arxiv.org/abs/1612.00147> (accessed on 14 February 2022).
12. Liang, J.; Patel, U.; Sathyamoorthy, A.J.; Manocha, D. Real-time Collision Avoidance for Mobile Robots in Dense Crowds using Implicit Multi-sensor Fusion and Deep Reinforcement Learning. 2020. Available online: <http://arxiv.org/abs/2004.03089> (accessed on 14 February 2022).

13. Chen, Y.F.; Liu, M.; Everett, M.; How, J.P. Decentralized non-communicating multi-agent collision avoidance with deep reinforcement learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 285–292.
14. Ibrahim, M.F.; Huddin, A.B.; Hussain, A.; Zaman, M.H.M. Frontier Strategy with GA based Task Scheduler for Autonomous Robotic Exploration Systems. *Adv. Nat. Appl. Sci.* 2020, 14, 259–265.
15. Bechlioulis, C.P.; Giagkas, F.; Karras, G.C.; Kyriakopoulos, K.J. Robust Formation Control for Multiple Underwater Vehicles. *Front. Robot. AI* 2019, 6, 90.
16. Hua, X.; Wang, G.; Xu, J.; Chen, K. Reinforcement learning-based collision-free path planner for redundant robot in narrow duct. *J. Intell. Manuf.* 2021, 32, 471–482.
17. Dai, X.; Mao, Y.; Huang, T.; Qin, N.; Huang, D.; Li, Y. Automatic obstacle avoidance of quadrotor UAV via CNN-based learning. *Neurocomputing* 2020, 402, 346–358.
18. Han, D.; Yang, Q.; Wang, R. Three-dimensional obstacle avoidance for UAV based on reinforcement learning and RealSense. *J. Eng.* 2020, 2020, 540–544.
19. Back, S.; Cho, G.; Oh, J.; Tran, X.T.; Oh, H. Autonomous UAV Trail Navigation with Obstacle Avoidance Using Deep Neural Networks. *J. Intell. Robot. Syst. Theory Appl.* 2020, 100, 1195–1211.
20. Woo, J.; Kim, N. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. *Ocean. Eng.* 2020, 199, 107001.
21. Meyer, E.; Robinson, H.; Rasheed, A.; San, O. Taming an Autonomous Surface Vehicle for Path following and Collision Avoidance Using Deep Reinforcement Learning. *IEEE Access* 2020, 8, 41466–41481.
22. Xu, X.; Lu, Y.; Liu, X.; Zhang, W. Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs. *Ocean. Eng.* 2020, 217, 107704.
23. Wang, C.; Zhang, X.; Cong, L.; Li, J.; Zhang, J. Research on intelligent collision avoidance decision-making of unmanned ship in unknown environments. *Evol. Syst.* 2019, 10, 649–658.
24. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* 2021, 26, 509–524.
25. Zhao, L.; Roh, M.i. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean. Eng.* 2019, 191, 106436.
26. Xie, S.; Garofano, V.; Chu, X.; Negenborn, R.R. Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks. *Ocean. Eng.* 2019, 193, 106609.
27. Xie, S.; Chu, X.; Zheng, M.; Liu, C. A composite learning method for multi-ship collision avoidance based on reinforcement learning and inverse control. *Neurocomputing* 2020, 411, 375–392.
28. Havenstrøm, S.T.; Rasheed, A.; San, O. Deep Reinforcement Learning Controller for 3D Path Following and Collision Avoidance by Autonomous Underwater Vehicles. *Front. Robot. AI* 2021, 7, 211.
29. Kim, J. While Preserving Collision Avoidance. *IEEE Trans. Cybern.* 2016, 47, 4038–4048.
30. Dergachev, S.; Yakovlev, K. Distributed Multi-Agent Navigation Based on Reciprocal Collision Avoidance and Locally Confined Multi-Agent Path Finding. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Lyon, France, 23–27 August 2021; pp. 1489–1494.
31. Čáp, M.; Vokřínek, J.; Kleiner, A. Complete decentralized method for on-line multi-robot trajectory planning in well-formed infrastructures. In Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS, Jerusalem, Israel, 7–11 June 2015; pp. 324–332.
32. Arul, S.H.; Manocha, D. V-RVO: Decentralized Multi-Agent Collision Avoidance using Voronoi Diagrams and Reciprocal Velocity Obstacles. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 8097–8104.
33. Mao, R.; Gao, H.; Guo, L. A Novel Collision-Free Navigation Approach for Multiple Nonholonomic Robots Based on ORCA and Linear MPC. *Math. Probl. Eng.* 2020, 2020, 4183427.
34. Wei, C.; Hindriks, K.v.; Jonker, C.M. Altruistic coordination for multi-robot cooperative pathfinding. *Appl. Intell.* 2016, 44, 269–281.
35. Nagavarapu, S.C.; Vachhani, L.; Sinha, A. Multi-Robot Graph Exploration and Map Building with Collision Avoidance: A Decentralized Approach. *J. Intell. Robot. Syst.* 2016, 83, 503–523.
36. Das, S.; Nath, S.; Saha, I. SPARCAS: A Decentralized, Truthful Multi-Agent Collision-free Path Finding Mechanism. *arXiv* 2019, arXiv:1909.08290.

37. Zhang, L.; Wang, J.; Lin, Z.; Lin, L.; Chen, Y.; He, B. Distributed Cooperative Obstacle Avoidance for Mobile Robots Using Independent Virtual Center Points. *J. Intell. Robot. Syst.* 2019, 98, 791–805.
38. Li, H.; Weng, B.; Gupta, A.; Pan, J.; Zhang, W. Reciprocal Collision Avoidance for General Nonlinear Agents using Reinforcement Learning. 2019. Available online: <http://arxiv.org/abs/1910.10887> (accessed on 13 February 2022).
39. Bae, H.; Kim, G.; Kim, J.; Qian, D.; Lee, S. Multi-robot path planning method using reinforcement learning. *Appl. Sci.* 2019, 9, 3057.
40. Lin, J.; Yang, X.; Zheng, P.; Cheng, H. End-to-end Decentralized Multi-robot Navigation in Unknown Complex Environments via Deep Reinforcement Learning. In *Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation, ICMA 2019, Tianjin, China, 4–7 August 2019*; pp. 2493–2500.
41. Cai, Z.; Cao, H.; Lu, W.; Zhang, L.; Xiong, H. Safe Multi-Agent Reinforcement Learning through Decentralized Multiple Control Barrier Functions. 2021. Available online: <http://arxiv.org/abs/2103.12553> (accessed on 13 February 2022).
42. Mehdipour, N.; Abdollahi, F.; Mirzaei, M. Consensus of multi-agent systems with double-integrator dynamics in the presence of moving obstacles. In *Proceedings of the 2015 IEEE Conference on Control and Applications, CCA 2015- Proceedings, Sydney, Australia, 21–23 September 2015*; pp. 1817–1822.
43. Salameh, M.O.; Abdullah, A.; Sahran, S. Ensemble of vector and binary descriptor for loop closure detection. *Adv. Intell. Syst. Comput.* 2017, 447, 329–340.
44. Dewi, D.A.; Sundararajan, E.; Prabuwo, A.S.; Cheng, L.M. Object detection without color feature: Case study Autonomous Robot. *Int. J. Mech. Eng. Robot. Res.* 2019, 8, 646–650.
45. Ramalho, G.M.; Carvalho, S.R.; Finardi, E.C.; Moreno, U.F. Trajectory Optimization Using Sequential Convex Programming with Collision Avoidance. *J. Control. Autom. Electr. Syst.* 2018, 29, 318–327.
46. Patle, B.K.; Babu L, G.; Pandey, A.; Parhi, D.R.K.; Jagadeesh, A. A review: On path planning strategies for navigation of mobile robot. *Def. Technol.* 2019, 15, 582–606.
47. Okoli, C. A guide to conducting a standalone systematic literature review. *Commun. Assoc. Inf. Syst.* 2015, 37, 879–910.

---

Retrieved from <https://encyclopedia.pub/entry/history/show/52495>