SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

# An Autonomous Parking System using the Hybridization of the Rapidly-Exploring Random Trees Star and Ant Colony System

Muhammad Aria[1,*], Jana Utama[1]

[1] Electrical Engineering Departement, Universitas Komputer Indonesia, Jl. Dipatiukur 102-116, Bandung 40132, Indonesia

**ARTICLE INFO**

***Article history:***
Received 23 January 2023
Received in revised form 30 May 2023
Accepted 6 June 2023
Available online 22 June 2023

***Keywords:***
Autonomous system; RRT-ACS; parking

**ABSTRACT**

One key autonomous driving car application that can be utilized to address the problem of parking spaces is autonomous parking. The goal of this study is to develop a path planning algorithm that can swiftly create a path for autonomous parking in a variety of car parking settings. The method employed is a mix of Rapidly Exploring Random Tree Star and Ant Colony Systems (RRT-ACS). The RRT-ACS method can quickly provide an ideal path. The reed sheep planner technique is also used to provide a smooth curving path that non-holonomic vehicles can follow. The performance of the RRT-ACS algorithm-based autonomous parking system is compared to that of the RRT*-connect and informed RRT*-connect algorithms. The simulation tests were performed in common parking settings such as parallel parking and vertical parking. The test results reveal that the suggested autonomous parking system outperforms other comparative algorithms. It is possible to infer that a route planning algorithm capable of rapidly designing a path for automatic parking in a variety of car parking circumstances has been successfully built. These findings may have ramifications for self-driving cars equipped with advanced driver support systems.

## 1. Introduction

The number of cars on the road is increasing and is expected to reach 2.5 billion by 2050 [1]. In densely populated areas, such as cities, the availability of parking spaces is often less than the availability of vehicles, resulting in a parking space shortage [2]. The increasing scarcity of parking spaces has increased the stress accumulated at work, lowering one's quality of life [3]. Drivers hunting for parking places contribute for around 30% of traffic backup in a typical downtown location, according to reports [4]. The anticipated rise in vehicle numbers indicates more new and unskilled drivers, resulting in increased road congestion and the waste of important time and resources [5]. As a result, an immediate parking solution, such as an automatic parking system, is required [6]. Using environmental data acquired by sensors, the autonomous parking system can intelligently manage the steering wheel to execute parking tasks [7]. The path planning algorithm is vital in automatic parking systems because they often function in a confined operational space [8,9]. Aside from the

* *Corresponding author.*
*E-mail address: muhammad.aria@email.unikom.ac.id*

https://doi.org/10.37934/araset.31.1.291297

path planning algorithm, the automatic parking system requires localization, perception, and decision methods [10,11]. Most research domains, however, already have existing methods for dealing with the three methods [9]. The path planning algorithm for autonomous parking should be able to find a collision-free path around the parking spot [12], the algorithm should take ride comfort and path smoothness into account [13]. Even in complex parking situations, the path should be obtained in a short planning time.

Several path planning algorithms have been proposed in the various research to generate paths for automated parking in a short period of time. Among existing motion planning strategies in parking scenarios, sampling-based planners such as Rapidly Exploring Random Tree (RRT)-related methods are commonly employed [9,12,14]. Shin *et al.,* presented Desired orientation RRT (Do-RRT) for the operation of autonomous vehicle restricted locations and locations with frequent traffic jams [15]. For autonomous parking, Wang *et al.,* suggested a two-stage RRT path planner [16]. Manav and Lazoglu suggested a cascade RRT algorithm for self-parking truck trailers [17]. But, in a tight parking spot or on a small road, planning time can be extended much more. Various strategies for improving the planning time of RRT algorithms in a constrained context have also been proposed. Klemm *et al.,* [18] a dual-tree variation of RRT* aptly named “RRT*-connect” that discovers solutions quicker than RRT*, especially in tight passageways. In contrast, RRT*-connect must search all states for the optimum path. Gammel *et al.,* [19] proposed the informed RRT* algorithm that performs informed sampling on RRT* after the discovery of a first solution. This algorithm can locate the optimal solution faster than the RRT* algorithm; nevertheless, it takes longer to reach the destination when the target node is hidden behind a narrow passage. Mashayekhi *et al.,* [21] suggested a dual-tree variation of informed-RRT*, which is the informed RRT*-connect algorithm. However, Pohan *et al.,* [22] presented numerous environmental conditions in which the ellipsoid sampling area of the informed approach can be excessively big. As a result, the ellipsoid area is ineffective for limiting the search area. As a result, RRT-ACS method was created by mixing Ant Colony System (ACS)and RRT algorithm. In the nonparametric test by Friedman, Pohan found that the RRT-ACS algorithm surpassed the RRT* algorithms, informed RRT*, informed RRT*-connect, and RRT*-connect. Currently, there are no studies have used the RRT-ACS algorithm in an automated parking system. Another disadvantage of the reference's autonomous parking method is that it only addresses one or two sorts of parking issues even though there are three broad categories of parking methods namely vertical, parallel, and oblique parking [23]. The difference between this study and other studies is that it uses a path planning algorithm that is faster than the RRT algorithm for automatic parking in oblique parking, vertical parking, and parallel parking.

This research aims to implement a path planning algorithm that can quickly plan a path for automatic parking in various car parking scenarios. The method is based on Rapidly-exploring Random Tree star and Ant Colony Systems (RRT-ACS) hybrid. The RRT-ACS method can quickly provide an ideal path. The reed sheep planner technique is also used to provide a smooth curving path that non-holonomic vehicles can follow. The algorithm’s performance is compared to the performance of the RRT* and informed RRT*-connect algorithms used in the autonomous parking system. Simulation was used to run the tests in common parking settings such as parallel and vertical parking. The results of the tests show that the suggested autonomous parking system outperforms existing comparative algorithms. The results suggest that the proposed autonomous parking system is appropriate for an advanced driver-assistance system in a self-driving car.

## 2. Methodology

This section provides a detailed explanation of the recommended algorithm. In Figure 1, a Reeds-Shepp curves illustrates the RRT-ACS algorithm. Generation of a random value as a variable at each iteration decides whether the exploration or exploitation process should be employed (Line 10). A random state, which includes both position and direction information, is created. When the exploration process is selected, the random sample is used to establish the random state (Line 16). In addition, the exploitation process is chosen if the pheromone dispersion data is utilized to determine the random state (Lines 12-14). The GrowTree function employs a collision-free Reeds-Shepp curve to link the parent node and the random state (Line 20 and Line 32).

Bidirectional RRTs are used in the proposed algorithm to improve the efficiency of environment exploration, particularly in narrow environments. In Figure 1, it can be seen that there are two search trees used, namely T_A and T_B. This algorithm simultaneously starts the search from the start and goal points. These two trees are attempting to connect in an aggressive manner (Line 26 and Line 38). Tree T_B has a better chance of growing out of narrow parking spaces. Once the tree T_B has grown out of the parking slot, numerous solutions to improve exploring efficiency can be found quickly.

The proposed approach ensures the quality of the end result by utilizing pheromone information to compute the new path based on the idea of learning from experience. An approach for updating the pheromone data has been established (Lines 46-48). At each iteration, only ants that have discovered the best path will add more pheromones to reinforce the good paths. The local search function (Line 45) optimizes the current branches of the RRT to enhance the quality of the discovered paths.

**Algorithm 1:** $X^{bs}$ ←RRT-ACS($map$)

1: **% ========== Initialization**
2: $T_A \leftarrow InitializeTree()$
3: $T_B \leftarrow InitializeTree()$
4: $T_A \leftarrow InsertNode(\emptyset, q_{init}, T_A)$
5: $T_B \leftarrow InsertNode(\emptyset, q_{goal}, T_B)$
6: $s \leftarrow 0$
7: **while** $termination\ condition\ not\ met$ **do**
8: **for** $k = 1$ to $m$ **do**
9: **while** $s = 0$ **do**
10: $q \leftarrow randvar[0,1]$ **and** $\tau \neq \emptyset$
11: **if** $q \leq q_o$
12: $q_{samp} \leftarrow RandomSampleFrom10LastNode(T)$
13: $\tau_{near} \leftarrow Near(\tau, q_{samp})$
14: $q_{randomstate} \leftarrow RouletteWhell(\tau_{near})$
15: **else**
16: $q_{randomstate} \leftarrow RandomSample(k)$
17: **end if**
18: $Q_{\boldsymbol{near}} \leftarrow Near(T_A, q_{randomstate})$
19: $q_{nearest} \leftarrow NearestNeighbr(q_{randomstate}, Q_{near}, T_A)$
20: $q_{new} \leftarrow GrowTree\ (q_{nearest}, q_{rand}omstate, \Delta q)$
21: **if** $Obstaclefree(q_{nearest}, q_{new})$ **then**
22: $Q_{\boldsymbol{near}} \leftarrow Near(T_A, q_{new})$
23: $q_{min} \leftarrow ChoosPrnt(q_{new}, Q_{near}, q_{nearest})$
24: $T_A \leftarrow InsertNode(q_{min,}q_{new}, T_A)$
25: $T_A \leftarrow Rewire(T, Q_{near}, q_{min})$
26: **if** $CanConnected(q_{new}, T_B)$ **then**

**Algorithm 1:** $X^{bs}$ ←RRT-ACS$(map)$

$X^{bs} \leftarrow UpdateBestPath(T_A, T_B)$
**end if**
**end if**
$Q_{\boldsymbol{near}} \leftarrow Near(T_A, q_{rand})$
$q_{nearest} \leftarrow NearestNeighbr(q_{rand}, Q_{near}, T_A)$
$q_{new} \leftarrow GrowTree\ (q_{nearest}, q_{rand}, \Delta q)$
**if** $Obstaclefree(q_{nearest}, q_{new})$ **then**
$Q_{\boldsymbol{near}} \leftarrow Near(T_B, q_{new})$
$q_{min} \leftarrow ChoosPrnt(q_{new}, Q_{near}, q_{nearest})$
$T_B \leftarrow InsertNode\left(q_{min,}q_{new}, T_B\right)$
$T_B \leftarrow Rewire(T, Q_{near}, q_{min})$
**if** $CanConnected(q_{new}, T_A)$ **then**
$X^{bs} \leftarrow UpdateBestPath(T_A, T_B)$
**end if**
**end if**
**end while**
$X(k) \leftarrow MakePath$ from $T$
**end for**
$X^{bs} \leftarrow LocalSearch(X^{bs}, map)$
$X^{w}$ ← Select $w$ best $Path(X)$
$\tau$ ← Add $Pheromone\ Node$ on the $w$ best $Path(X^{w}, \tau)$
$\tau$ ← Evaporate $Pheromone\ Node(\tau)$
**end while**

**Fig. 1.** RRT-ACS algorithm with Reed-Shepp curves

## 3. Results

To validate the performance of the autonomous parking system using the RRT-ACS algorithm, the algorithm was running through various parking scenarios and observed the algorithm's adaptability to different parking spaces and lane feasibility. During the parking process, the vehicle's steering angle is recorded. The PC utilized for the simulation tests had a 3.20 GHz Core i5 CPU and 4 GB RAM, and it was operating on Windows 10 64-bit. LabVIEW 7.1 was used as the compilation environment. By following the tests conducted by Zheng and Liu [6], Jhang and Lian [9], and Jhang *et al.,* [12], then the test was carried out on three parking scenarios namely Common Oblique, Vertical, and Parallel Parking Scenario.

### *3.1 Common Parallel Parking Scenario*

The simulations process for parallel parking is shown in Figure 2. The starting point is [9, 6, 0] and the target point is [6, 2.5, 0]. The proposed method grows RRT-ACS trees to explore the environment as shown in Figure 2. The final feasible path is shown in Figure 3. The RRT-ACS algorithm successfully constructs a path for the Common Parallel Parking scenario, as shown in Figure 2. As reported by Zheng and Liu, the RRT algorithm also successfully constructs a path for the Common Parallel Parking scenario [6]. However, the RRT algorithm takes longer than the RRT-ACS algorithm.

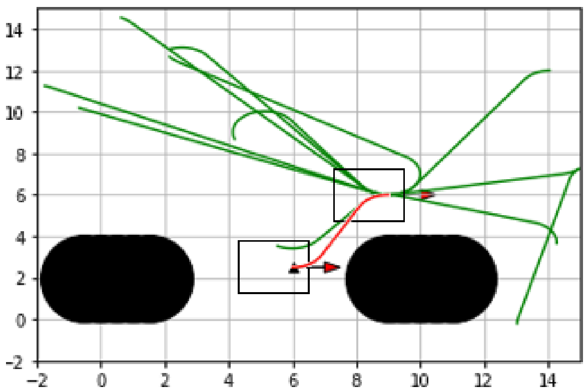

**Fig. 2.** RRT-ACS tree when a feasible path is found

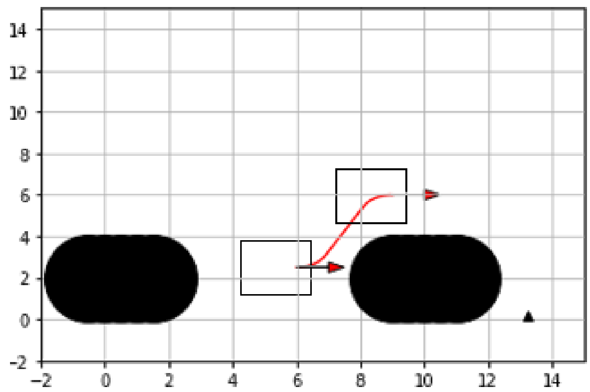

**Fig. 3.** A feasible path for the parallel parking scenario

### *3.3 Common Oblique Parking Scenario*

The simulation process for oblique parking is depicted in Figure 4. The starting point is [2.5, 12, 0] while the target point is [6, 2, PI/4]. The proposed method, as shown in Figure 4, grows RRT-ACS trees to explore the environment. Figure 5 depicts the final feasible path. The RRT-ACS algorithm successfully constructs a path for the Common Vertical Parking scenario, as illustrated in Figure 4. As reported by Zheng and Liu, the RRT algorithm also successfully constructs a path for the Oblique Parallel Parking scenario [6]. However, RRT-ACS algorithm constructs faster than the RRT algorithm.

Table 1 presents a comparison between the Informed RRT*-Connect algorithms, RRT*-Connect, RRT-ACS algorithm and regarding their performance in a Common Vertical Parking Scenario, with best, worst, and mean path length being the performance metrics. The results show that the RRT-ACS algorithm surpasses the others in all three metrics. These findings align with the findings of Pohan [22].

**Table 1**
Comparison of path length performance generated by tested algorithms

| Path length performance | RRT-ACS | Informed RRT*-Connect | RRT*-Connect |
|---|---|---|---|
| Best | 14.40 | 15.41 | 15.71 |
| Mean | 14.36 | 15.36 | 15.68 |
| Worst | 14.31 | 15.34 | 15.54 |

The RRT-ACS algorithm has been tested in three different common parking scenarios, namely the Parallel, Vertical, and Oblique Parking Scenario, and it has been shown to outperform other algorithms in terms of best, worst, and mean path length. This outcome is consistent with Pohan's study [22], which found that the RRT-ACS algorithm surpasses the informed RRT*-connect and RRT*-connect algorithms in Friedman's nonparametric test. Furthermore, Mashayekhi [21] has stated that the informed RRT*-connect algorithm outperforms the RRT*-connect algorithm in all three metrics as well.

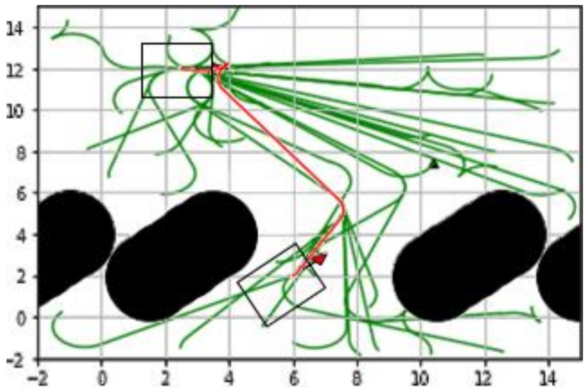

**Fig. 4.** RRT-ACS tree when a feasible path is found

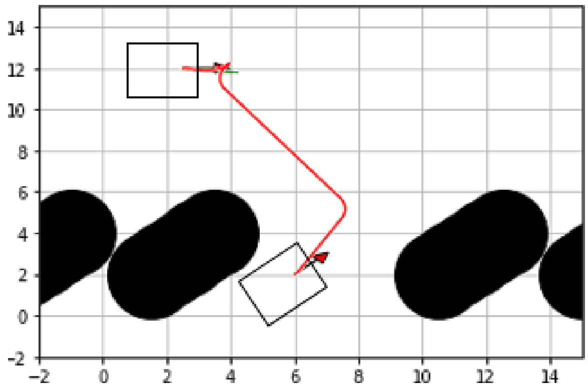

**Fig. 5.** A feasible path for the oblique parking scenario

## 4. Conclusions

Using the RRT-ACS algorithm, a path planning algorithm was implemented for autonomous parking in various car parking scenarios. The performance of RRT-ACS algorithm was compared to that of the informed RRT*-connect algorithms and RRT*-connect in different parking scenarios. The test results show that the proposed autonomous parking system surpasses the RRT*-connect and informed RRT*-connect algorithms, indicating its suitability for advanced driver-assistance systems in self-driving cars.

### Acknowledgement
Our gratitude goes to UNIKOM for their support of this study. There is no financial support such as grant in the process of this research and its writing.

### References
[1] Kim, Oanh Tran Thi, Nguyen H. Tran, Chuan Pham, Tuan LeAnh, My T. Thai, and Choong Seon Hong. "Parking assignment: Minimizing parking expenses and balancing parking demand among multiple parking lots." *IEEE Transactions on Automation Science and Engineering* 17, no. 3 (2019): 1320-1331. https://doi.org/10.1109/TASE.2019.2948200

[2] Paidi, Vijay, Hasan Fleyeh, Johan Håkansson, and Roger G. Nyberg. "Smart parking tools suitability for open parking lots: A review." In *4th International Conference on Vehicle Technology and Intelligent Transport Systems, March 16-18 2018, Funchal, Madeira*, pp. 600-609. 2018. https://doi.org/10.5220/0006812006000609

[3] Bajçinovci, Bujar, and Mejreme Bajçinovci. "Anxiety and Urban Stress for Parking Spots." *Journal of Science, Humanities and Arts* 6 (2019). https://doi.org/10.17160/josha.6.1.525

[4] Nawaz, Sarfraz, Christos Efstratiou, and Cecilia Mascolo. "Parksense: A smartphone based sensing system for on-street parking." In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 75-86. 2013. https://doi.org/10.1145/2500423.2500438

[5] Wu, Mingkang, Haobin Jiang, and Chin-An Tan. "Automated parking space allocation during transition with both human-operated and autonomous vehicles." *Applied Sciences* 11, no. 2 (2021): 855. https://doi.org/10.3390/app11020855

[6] Zheng, Kaiyu, and Shan Liu. "RRT based path planning for autonomous parking of vehicle." In *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 627-632. IEEE, 2018. https://doi.org/10.1109/DDCLS.2018.8515940

[7] Undale, Akash, Mendgule, Shruti, Metkari, Nisha, Savairam, Darshana, and Kharat, J.P. "Self Parking Vehicle," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* 8, no. 6 (2020): 415-422. https://doi.org/10.22214/ijraset.2020.6063

[8] Petrov, Plamen, and Veska Georgieva. "Geometric path planning and tracking control with bounded steering angle for the parking problem of automatic vehicles." In *AIP Conference Proceedings*, vol. 2048, no. 1, p. 060017. AIP Publishing LLC, 2018. https://doi.org/10.1063/1.5082132

[9] Yusof, Ahmad Anas, Mohd Khairi Mohamed Nor, Shamsul Anuar Shamsudin, Mohd Rizal, Mohd Shahrieel Mohd Aras Alkahari, Mohamad Riduwan Md Nawawi, Mohd Zaidi Mohd, and Mohammad'Afif Kasno Tumari. "Lessons learned from UTeM Autonomous Underwater Vehicle Competition Initiatives." *challenge* 2, no. 1 (2018): 10-26.

[10] Jo, Kichun, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungho Sunwoo. "Development of autonomous car—Part I: Distributed system architecture and development process." *IEEE Transactions on Industrial Electronics* 61, no. 12 (2014): 7131-7140. https://doi.org/10.1109/TIE.2014.2321342

[11] Hamami, G., M. Mazni, M. S. Amir, F. Sukarman, and M. Katon. "A Review on Local Minimum and Multiple Minimum Avoidance Techniques in Local Path Planning." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 4, no. 1 (2016): 58-71.

[12] Jhang, Jyun-Hao, Feng-Li Lian, and Yu-Hsiang Hao. "Forward and Backward Motion Planning for Autonomous Parking Using Smooth-feedback Bidirectional Rapidly-exploring Random Trees with Pattern Cost Penalty." In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 260-265. IEEE, 2020. https://doi.org/10.1109/CASE48305.2020.9216968

[13] Li, Chenxu, Haobin Jiang, Shidian Ma, Shaokang Jiang, and Yue Li. "Automatic parking path planning and tracking control research for intelligent vehicles." *Applied sciences* 10, no. 24 (2020): 9100. https://doi.org/10.3390/app10249100

[14] Jhang, Jyun-Hao, Feng-Li Lian, and Yu-Hsiang Hao. "Human-like motion planning for autonomous parking based on revised bidirectional rapidly-exploring random tree* with Reeds-Shepp curve." *Asian Journal of Control* 23, no. 3 (2021): 1146-1160. https://doi.org/10.1002/asjc.2439

[15] Shin, Seho, Joonwoo Ahn, and Jaeheung Park. "Desired orientation rrt (do-rrt) for autonomous vehicle in narrow cluttered spaces." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4736-4741. IEEE, 2016. https://doi.org/10.1109/IROS.2016.7759696

[16] Wang, Yebin, Devesh K. Jha, and Yukiyasu Akemi. "A two-stage RRT path planner for automated parking." In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 496-502. IEEE, 2017. https://doi.org/10.1109/COASE.2017.8256153

[17] Manav, Ahmet Canberk, and Ismail Lazoglu. "A novel cascade path planning algorithm for autonomous truck-trailer parking." *IEEE Transactions on Intelligent Transportation Systems* 23, no. 7 (2021): 6821-6835. https://doi.org/10.1109/TITS.2021.3062701

[18] Klemm, Sebastian, Jan Oberländer, Andreas Hermann, Arne Roennau, Thomas Schamm, J. Marius Zollner, and Rüdiger Dillmann. "Rrt-connect: Faster, asymptotically optimal motion planning." In *2015 IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 1670-1677. IEEE, 2015. https://doi.org/10.1109/ROBIO.2015.7419012

[19] Gammell, Jonathan D., Timothy D. Barfoot, and Siddhartha S. Srinivasa. "Informed asymptotically optimal anytime search." *Int. J. Robot. Res., submitted for publication* (2017).

[20] Wang, Jiankun, Charles X-T. Li, Wenzheng Chi, and Max Q-H. Meng. "Tropistic RRT: An Efficient Planning Algorithm via Adaptive Restricted Sampling Space." In *2018 IEEE International Conference on Information and Automation (ICIA)*, pp. 1639-1646. IEEE, 2018. https://doi.org/10.1109/ICInfA.2018.8812520

[21] Mashayekhi, Reza, Mohd Yamani Idna Idris, Mohammad Hossein Anisi, Ismail Ahmedy, and Ihsan Ali. "Informed RRT*-connect: An asymptotically optimal single-query path planning method." *IEEE Access* 8 (2020): 19842-19852. https://doi.org/10.1109/ACCESS.2020.2969316

[22] Pohan, Muhammad Aria Rajasa, Bambang Riyanto Trilaksono, Sigit Puji Santosa, and Arief Syaichu Rohman. "Path Planning Algorithm Using the Hybridization of the Rapidly-Exploring Random Tree and Ant Colony Systems." *IEEE Access* 9 (2021): 153599-153615. https://doi.org/10.1109/ACCESS.2021.3127635

[23] Yan, Wenxu, Jing Deng, and Dezhi Xu. "Data-driven automatic parking constrained control for four-wheeled mobile vehicles." *International Journal of Advanced Robotic Systems* 13, no. 6 (2016): 1729881416663667. https://doi.org/10.1177/1729881416663667