



DESIGN AND TESTING OF AUTOMATIC CAR PARKING USING PUZZLE LOGIC

¹Miss. Gaikwad Tanuja Kalyan, ²Miss. Gaikwad Ankita Dattatraya,
³Miss. Ware Aakanksha Vasant, ⁴Prof. Gawali Ghanshyam D.

¹UG Scholar, Electrical Engg., Dept. HSBPVT'S GOI Faculty of Engineering, Kashti

²UG Scholar, Electrical Engg., Dept. HSBPVT'S GOI Faculty of Engineering, Kashti

³UG Scholar, Electrical Engg., Dept. HSBPVT'S GOI Faculty of Engineering, Kashti

⁴Asst. Prof., Electrical Engg., Dept. HSBPVT'S GOI Faculty of Engineering, Kashti

aakankshaware2004@gmail.com

ABSTRACT - Parking shortage in residential and commercial locations has become a major issue as urbanization rises, necessitating creative, space-saving solutions. In order to maximize car stacking in smaller areas, this study suggests an intelligent, puzzle-based parking system that makes use of a fuzzy Logic Controller (FLC). In contrast to traditional parking, the suggested solution uses a modular 2D/3D matrix layout in which automated lifts are used to rearrange automobiles in order to optimize maximum density. Retrieval times are decreased by using the FLC to control slot allocation based on factors like vehicle dimensions and entry proximity. In order to address the requirement for high-density parking, this study outlines the design and development of an automated multilevel puzzle parking system.

To reduce parking and urban traffic congestion, intelligent parking technologies are crucial. shortages. This study presents a smart puzzle-based parking system that uses fuzzy logic control to maximize parking space use. The development of intelligent parking systems with improved space use and operating efficiency is required due to the fast increase in urban vehicle density. The design and implementation of an autonomous parking system based on puzzle logic for the best possible vehicle layout in confined spaces are presented in this study. For real-time vehicle identification, positioning, and movement control, proximity sensors are interfaced with a microcontroller-based control architecture.



I. INTRODUCTION

During the prototype's flowchart. To start the system, flip the switch on. After the system is turned on, the vehicle advances until it finds a parking spot. The system is prepared to reverse and then reverse into the parking place if it has located a suitable spot. An impediment can be detected by the car's rear infrared sensor.

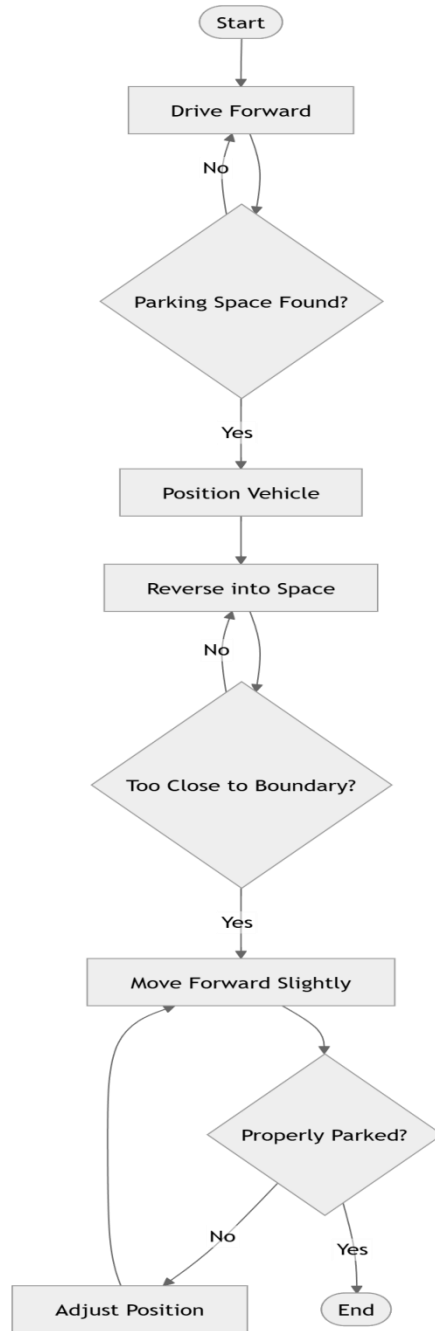


Fig. 1. Flowchart of the paper

II. FUZZY LOGIC TOOLS

Fuzzy logic toolbox is used to build a membership function of the Fuzzy Inference System (FIS). Membership function of input; which are front sensor 4 (S4) (0, 1), rear sensor 2 (S2) (0, 1), two right-sided sensors for the distance and decision input with distance (near: N, middle: M, far: F), and decision (front, rear), while two output variables are steering (left: L, center: C, right: R) and direction (reverse: RV, stop: S, forward: FW), are shown as in Table I.

TABLE I. MEMBERSHIP FUNCTION ANALYSIS

	1st Input	2nd Input	3rd Input	4th Input	Output	
	S4	S2	Distance	Decision	Steering	Direction
1	1	1	N	X	C	FW
2	0	1	N	X	C	FW
3	0	0	N	X	C	FW
4	X	X	F	Rear	L	RV
5	X	X	M	Rear	R	RV
6	X	X	N	Rear	C	FW
7	X	X	N	Front	C	S

However, the following step shows the essential rules for each condition of the forward, stop, or reverse parking.

- If (S4 is 1) and (S2 is 1) then (STEERING is Center)(DIRECTION is FORWARD)(1)
- If (S4 is 0) and (S2 is 1) then (STEERING is Center)(DIRECTION is FORWARD)(1)
- If (S4 is 0) and (S2 is 0) then (STEERING is Center)(DIRECTION is FORWARD)(1)
- If (DISTANCE is FAR) and (DECISION is REAR) then (STEERING is LEFT)(DIRECTION is REVERSE)(1)
- If (DISTANCE is MIDDLE) and (DECISION is REAR) then (STEERING is RIGHT)(DIRECTION is REVERSE)(1)



f. If (DISTANCE is NEAR) and (DECISION is REAR) then (STEERING is CENTER)(DIRECTION is FORWARD)(1)

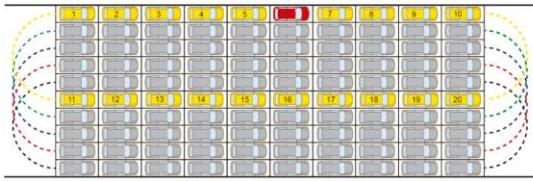
g. If (DISTANCE is NEAR) and (DECISION is FRONT) then (STEERING is CENTER)(DIRECTION is STOP)(1)

1. A puzzle-based parking lot

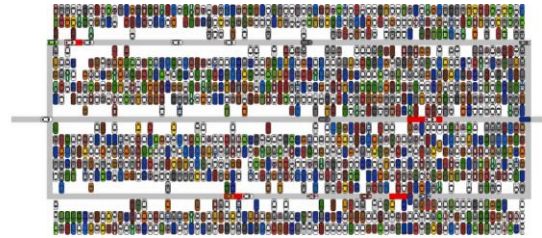
Autonomous passenger cars provide a number of intriguing possibilities to enhance urban mobility. In terms of parking, the ability to operate automobiles remotely would enable, for instance, cars to be parked closer together as doors wouldn't need to open in order for passengers to get out (Hayes, 2011). According to Ferreira et al. (2014), there is a 20% chance that parking density will increase. Another idea is that autonomous vehicles don't require parking in crowded cities. Instead, in low-cost suburban regions, a distant parking zone can be established (Zakharenko, 2016). Another widely held belief is that private car ownership will be eliminated by autonomous vehicles. Shared mobility will be made possible by autonomous taxis. In this regard, Bosch (2019) and Bosch (2020) have built an automated valet parking lot for driverless vehicles in partnership with Mercedes and Ford. The human rider departs the vehicle at a drop-off location after arriving at the parking lot. Smart sensor systems in the parking lot direct cars to a designated location. A smartphone app may be used to get the vehicle whenever the rider wants it, and it will arrive at the pick-up location. Interestingly, no human is required to operate the vehicles, and no new civil infrastructure has to be constructed.

Driving lanes can be removed in such an automated parking lot to increase parking density. In a shared parking area blocked for one simple reason: a blocked car might need to leave before the car that blocks it. However, an automated valet parking lot creates an opportunity to overcome this problem, and to achieve higher parking density. Therefore, we already have the required technology to achieve higher density, all we need is efficient facility design and planning.

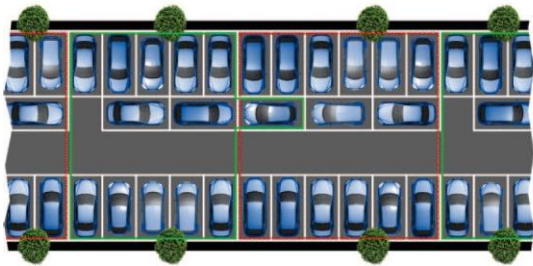
Some authors have envisioned parking schemes to improve the density of parked cars, or equivalently, the capacity of the parking lot. Ferreira et al. (2014) were first to propose high density parking for autonomous vehicles (Banzhaf et al., 2017). In their design, cars are parked by blocking each other in parallel lanes, similar to the deep lane storage systems use



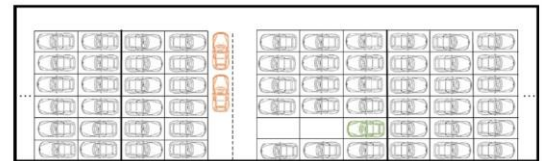
(a) From Ferreira et al. (2014).



(b) From Timpner et al. (2015).



(c) From Banzhaf et al. (2017).



(d) From Nourinejad et al. (2018).

Fig. 1. High density parking designs from other authors. (a) Ferreira et al. (2014) propose a design to park cars by blocking each other in parallel lanes. In order to allow automobiles to make the required adjustments for relocations, they set aside a buffer area at the facility's entry and departure. (b) To lessen the number of automobiles that interfere, Timpner et al. (2015) place perpendicular alleys between the obstructed cars' lanes. Blocking cars move to the aisle to make room for the leaving vehicle. (c) Banzhaf et al. (2017) modify existing parking lots by adding an aisle of blocking cars. (d) In the design of Nourinejad et al. (2018), the width of the aisle varies with the depth of the lane to help accommodate relocating cars.

They believe that the movement of vehicles within the parking lot is managed by an automated parking controller. This study has been expanded by the same team of researchers to take into account a variety of operational factors, including the number of movements, journey distance, retrieval time, and parking space per automobile. Parking capacity is increased by 50% because to their design.

By lowering the lane depth and permitting vehicles to exit from either end of a lane, Timpner et al. (2015) altered the design of Ferreira et al., 2014 to lessen the number of interfering automobiles. The authors calculate a 33% increase in density. Banzhaf et al. (2017) suggest altering current layouts by permitting automobiles to obstruct existing aisles in adjacent lanes. They claim that the density has increased by 25%. the aisle varies with the depth of the lane to help accommodate relocating cars. The authors develop a mixed-integer non-linear program to minimize the

expected number of vehicle relocations. However, all these researchers are planning for large parking lots which are more appropriate for suburban areas.

We've learnt from the literature that an autonomous vehicle won't stand still in a parking lot. It will be utilized to carry another passenger after a passenger has been dropped off. However, there will be some idle time when there is no journey because of the gap between supply and demand, therefore the automobile needs to park someplace. These vehicles won't be able to answer consumer calls if they are parked in suburban regions. As a result, autonomous cars will prefer to drive about the city center to pass the time rather than park outside, which will cause traffic in the city (Millard-Ball, 2019). As a result, urban parking cannot be totally eliminated. Instead, cheaper. To do this, we can offer small and dense parking lots for urban areas. Such parking lots can be compared to *micro-fulfillment centers* in the retail industry. Micro-fulfillment centers are small warehouses located near customers which are designed for on-demand e-commerce fulfillment (Kang (2019)). Whenever an autonomous vehicle is idle, it can be parked in these small lots. As these lots will be located in urban areas, whenever a car receives a trip request it can serve the request quickly. As a valued-added service: vehicle charging, cleaning or similar services can also be introduced in such parking lots.

In this paper, we study high-density layout design of a parking lot for autonomous vehicles. Our work differs from existing literature in two important ways:

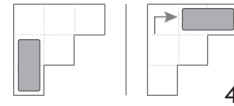
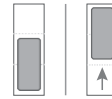
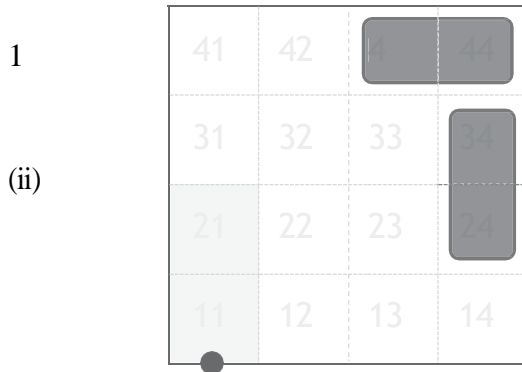
1. We develop optimal designs for small parking lots, which one might expect to find in urban settings.
2. All existing research *presupposes* a layout and then investigates optimal parameters and performance with respect to that layout. We ask and answer the more basic question of how many cars can be parked in a lot, *without* assuming any particular layout or structure. As we will show, this relaxation leads to designs with maximum density without the traditional “row-and-aisle” parking structure.



Fig. 2. The Rush Hour puzzle.

The goal of the game is to move the red car to the exit directly on the opposite side by moving other cars out of the way.

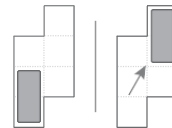
(i)



(iii)

(a)

4



(b)

Fig. 3. (a) An example of a 4×4 Parking lot with two cars, one car is parked horizontally on cells (43, 44), and another is parked vertically on cells (23, 34), and cells (11, 21) are the I/O point. (b) Three valid moves inside a parking lot. Each of these moves is assigned a weight according to number of cells travelled.

2. Designing a large parking lot

We can easily calculate the number of cars in a small parking lot using the exact graph method shown in Section 2. However, as the size of the parking lot increases, the state space explodes, e.g., a 5×5 parking lot graph has 2.6 million nodes. Exploring such a large state space is computationally expensive. Thus, we need an efficient method to overcome this difficulty.

2.1. Limited egress design

From the allowable moves inside a parking lot, we observe: *In an $m \times n$ parking lot, if we bypass two or more of the initial columns, we can park m cars horizontally on the following two columns. Similarly, if we bypass three or more initial rows, we can park n cars*

vertically on the following two rows (See Fig. 10). Using this observation, we develop a parking algorithm, which we call PUZZLE-BASED PARKING ALGORITHM.

1. In an $m \times n$ parking lot, park m cars horizontally, bypassing the initial n_b columns, where $n_b \geq 2$, on the $(n_b + 1)^{th}$ and $(n_b + 2)^{th}$ columns. Then, move these m cars to the far most empty columns. Repeat this process until only the n_b columns remain. This area can be designated as the *horizontal parking zone*.
2. Next, park n_b cars vertically, bypassing the initial m_b rows, where $m_b \geq 3$, on $(m_b + 1)^{th}$ and $(m_b + 2)^{th}$ rows. Then, move n_b cars to the up most empty rows. Repeat this process until only m_b rows remain. This area can be designated as the *vertical parking zone*.
3. Finally, park cars on the $m_b \times n_b$ grid. This area can be designated as the *base grid*. We use limited egress algorithm to park maximum number of cars in the base grid.

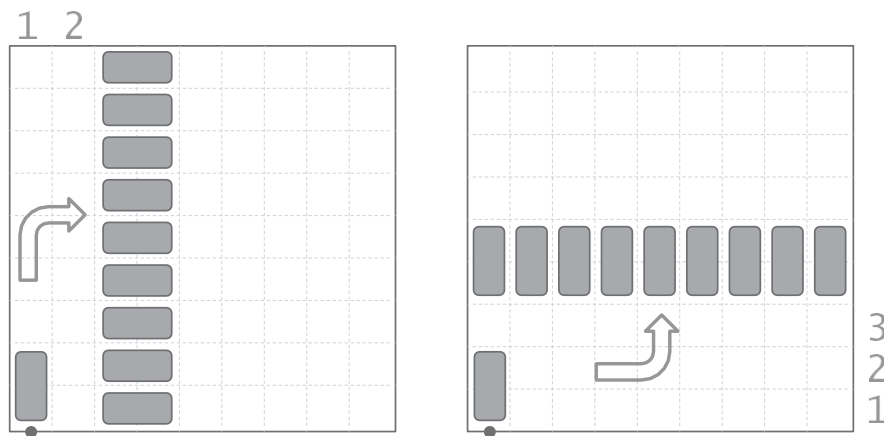


Fig. 4. In an $m \times n$ parking lot, if we bypass two or more of the initial columns, we can park m cars horizontally on the following two columns. Similarly, if we bypass three or more initial rows, we can park n cars vertically on the following two rows.

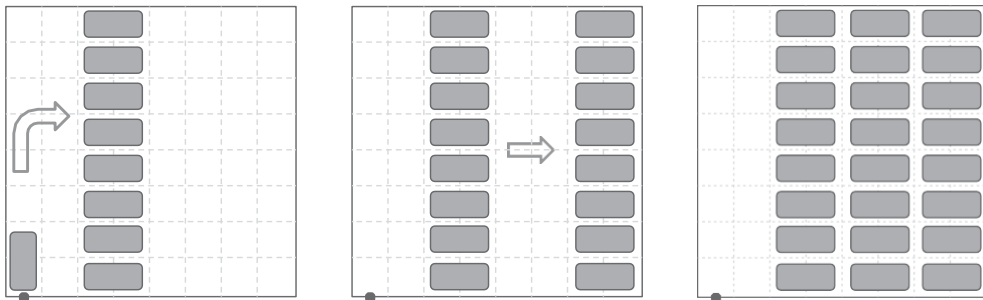
applying PUZZLE-BASED PARKING ALGORITHM to an 8×8 parking lot we park:

1. 24 cars in the horizontal parking zone;
2. 4 cars in the vertical parking zone;
3. and 3 cars in the base grid.

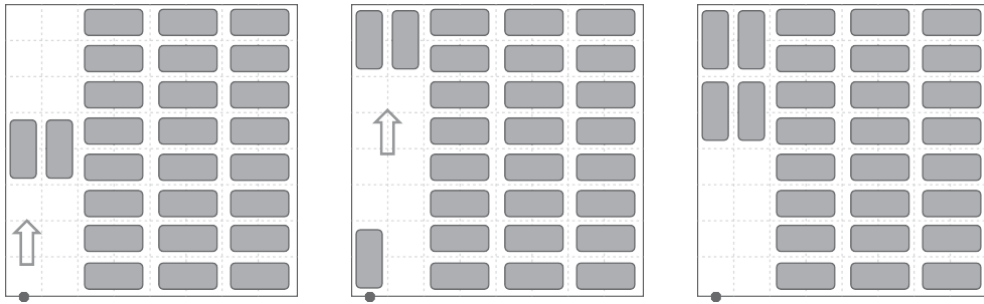
Therefore, we can park 31 cars in a 8×8 parking lot.

To identify appropriate sizes of base grids which maximize the number of cars in the parking lot, we apply the PUZZLE-BASED PARKING ALGORITHM to various grid sizes, as shown In an $m \times n$ parking lot, there can be four cases, (i) both m and n are odd, (ii) m even and n odd, (iii) m odd and n even, and iv) both m and n are even. For the first case, we observe two repeating patterns in base grid: 5×5 (when $m \geq 5$ and $n \geq 5$) and 3×3 (when either of m or n equals to 3). For the remaining three cases, base grids are $4 \times 3, 5 \times 2$, and 4×2 , respectively (when $m \geq 4$ and $n \geq 2$). However, we do not observe these patterns when $m < 3$ and $n < 2$, because cars cannot make all the moves inside these small grids. From this figure, we assert by observation that:

Step 1



Step 2



Step 3

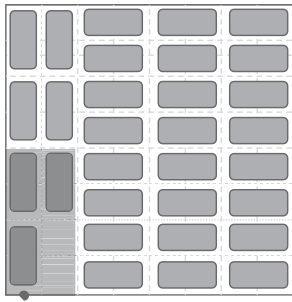


Fig. 5. An example

2.2. Parking density

While conventional parking lots can only reach a maximum density of 67%, large puzzle-based parking lots with both limited and complete egress can reach 100% density (Chodash, 1986). As a result, a big puzzle-based parking lot may hold 50% more vehicles than a conventional one. Large parking lots, however, are inappropriate for urban environments. In order to comprehend the effects of puzzle-based design, we examine modest parking lots, ranging in size from 4×4 to 8×8 . For conventional parking lots, we assume that automobiles should be positioned so that they do not obstruct one another.

Plots of parking density for 4×4 to 8×8 parking lots are displayed. We find that the density for puzzle-based parking lots achieves 90% density as the grid size grows. immediately. However, this growth is slow and gradual for the traditional parking lot, and it only reaches to 60%. For instance, in an 8×8 parking lot, we can park 31 cars using limited egress design, 29 cars using complete egress design, and 17 cars using the traditional design. We have also studied the percentage gain in density, and find that the limited egress design can accommodate 81% more cars and the complete egress design can accommodate 55% more cars, than the traditional design on average

reasonable.

III. RESULTS

The entire autonomous car parking system is designed using fuzzy logic, which combines an infrared sensor circuit, a microcontroller PIC 16F84A circuit, a motor driver circuit, and a pulse width modulation (PWM) circuit. The steering/direction output is -0.5 for left/reverse, 0.5 for center/stop, and 1.5 for right/forward. Fuzzy logic is effectively used in the creation of autonomous parking. Using an infrared sensor, this prototype can find a suitable parking spot and reverse the park on its own. The DC motor's speed is managed using PWM. Thus, the vehicle may smoothly enter the parking place and proceed forward or backward. However, this prototype can also turn left and right with success.

IV. CONCLUSION

We suggest puzzle-based parking, a high density parking lot design that makes use of autonomous cars' self-parking and sophisticated communication technologies. When we model the parking lot as a graph, we see that there are three different kinds of parking lot designs: limited egress, entire egress, and traditional. Cars in a restricted egress configuration can only exit in a last-in, first-out order and totally block one another. Cars obstruct one another in a complete egress design, yet there is still room for any automobile to exit by pushing other vehicles. Lastly, no vehicle blocks another in a conventional parking lot. We provide a heuristic technique known as the puzzle-based parking algorithm to address computational complexity in modeling bigger architectures. A large puzzle-based parking lot, both limited egress and complete egress, can park 50% more cars than the traditional lots. However, for large limited egress designs, mean waiting time to retrieve a car is longer. Whereas, in large complete egress designs, cars have to be relocated many times to retrieve a target car. Therefore, we recommend keeping puzzle-based designs small to reduce waiting time and number of relocations. We also show that in small parking lots, it is possible to achieve 80% higher density, on average, than a traditional parking lot.

We discover that a compact parking lot with puzzle elements works well in urban environments. Limited egress designs are an option available to transportation network firms. They can benefit greatly from the last-in, first-out operation sequence, in which the last vehicle in the parking lot departs to service whenever a trip request is made. As a result, other automobiles are unaffected by the time it takes to retrieve a car. To handle more vehicles in a crowded downtown location, commercial garages might use the full egress design. To reduce the frequency of relocations, we advise limiting density to 80% or fewer. This study might be expanded in a number of ways. Because the problem is intractable, creating effective retrieval planning in a comprehensive egress design is difficult. Only with

minor issues can we provide the best retrieval outcomes. Therefore, creating techniques to collect autos more effectively is a crucial future expansion of our research. Additionally, we have only taken into account one I/O point for a parking lot. Another intriguing development of this study will be the incorporation of various I/O points. Future study could also focus on including different automobile sizes.

REFERENCES

- [1]E. Cox, "Fuzzy Fundamentals", IEEE Spectrum, 1992, pp. 58-61.
- [2]Arjun Nagappan and Arjun Prakash (2009), "Autonomous Parallel Parking RC Car", Cornell University, New York
- [3]A., Kizil, A.C., Kutlucan, C., Dogan, S., Kocak, V., Sezer, "Design and implementation of autonomous parallel and vertical parking mobile vehicle," International Conference on Control Engineering and Information Technology, CEIT 2018.
- [4]M., Saleh, M., Ismail, N., Riman, "Enhanced algorithm for autonomous parallel parking of a car-like mobile robot," IEEE International Multidisciplinary Conference on Engineering Technology, 2016, pp.191- 195.
- [5]S.R., Ganesh, V., Sai Ganesh, M.F., Hamza, S., Ramwal, S., Ghosh, "Automatic parallel car parking system using sensors and arduino UNO," International Journal of Recent Technology and Engineering, 8(11), 2019, pp. 3531-3534.
- [6]Wulandari, Malada, A., Arrazi, M. H., Priandana, K., Wahjuni, S., Hardhienata, M., & Buono, A. (2019). Development of an autonomous smart-parallel-parking robot for the prototype of collaborative robots. Paper presented at the IOP Conference Series: Earth and Environmental Science, , 299(1)
- [7]B., Li, K., Wang, Z., Shao, "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. IEEE Transactions on Intelligent Transportation Systems, 17(11), 2016, pp. 3263-3274.
- [8]R.H., Nedamani, P.M., Khiabani, S., Azadi, "Intelligent parallel parking using adaptive neuro-fuzzy inference system based on fuzzy C-means clustering algorithm." SAE Technical Papers, 2017.
- [9]Osório, F., Heinen, F., & Fortes, L. "Autonomous vehicle parking using finite state automata learned by J-CC artificial neural nets." Proceedings - Brazilian Symposium on Neural Networks, 2002, p. 196 gent Vehicles Symposium (IV), IEEE, pp. 157–164.