AUTOMATIC ROBOT CAR PARALLEL PARKING SYSTEM USING ARTIFICIAL NEURAL NETWORK

SARA GHATTA[1], SORAYA ZENHARI & AMIRHASSAN MONADJEMI[2]
[1]Institut für Software Engineering (ISTE), Germany
[2]Department of Computer Science School of Computing, National University of Singapore, Singapore

ABSTRACT
In today's busy world, parking a car in crowded cities is time-consuming and challenging. Parallel parking systems are a valuable innovation, especially for disabled and inexperienced drivers, as they can prevent collisions. Despite their common use, artificial neural networks (ANNs) and backpropagation algorithms (BP) have issues, such as difficulty in estimating network configurations and low accuracy. While BP is traditionally used to train ANNs, it is not effective for parking a car. To achieve accurate vehicle control, a self-organising map (SOM) clusters data from various parking scenarios and uses this information to classify the data. According to simulations conducted using MATLAB, SOM provides better accuracy and reliability for parking cars compared to BP.
Keywords: self-organising map algorithm, automated vehicle, car-like robot parallel parking, artificial neural network (ANN), robot navigation, data scaling.

1 INTRODUCTION

In today's hectic and stressful life, parking a car accurately is a challenge for many people There are many problems associated with parallel parking in cities because of their crowded streets and limited parking spaces, which often leads to vehicle damage. This task is particularly difficult for new drivers and can contribute to traffic congestion. To address these problems, automatic car parallel parking systems have emerged, offering significant benefits, especially for disabled and inexperienced drivers [1]. Automatic Parking processes can be categorised into three parts, including environment perception, path planning, and tracking using a control strategy Perception of the environment defines a parking space as the space formed by two obstacles or lines painted on the road. For a parking space formed by obstacles, methods based on range sensors have been developed [2]. Ultrasonic sensors and light detection and ranging (LiDAR) have been used to determine the shape of each obstacle. Simultaneous localisation and mapping (SLAM) have been established during the parking process, to locate the vehicle at the cm level. Path planning is considered in finding a collision-free geometric path from an initial position to a final position while traveling in the route. Trajectory planning is in connection with real-time planning, which is parameterised by time as well as velocity and acceleration [3]. It can exist many feasible paths, but we often require the shortest path in length.

1.1 Background and preliminaries

Automatic path planning and path tracking technology have attracted many researchers in recent years. Parking path planning approaches are mainly characterised into four sets: geometric curve approaches [4], graph search approaches [5], random sampling approaches [6], and intelligent optimisation approaches [7]. In the geometric curve technique, Vorobieva et al. [8] examined tiny parking places characterised by the clothoid curve and considered parking pathways appropriate for tiny parking places. But the clothoid curvature design was moderately complex demanding a large number of mathematical operation and calculation. In the graph search technique, Ren et al. [5] planned an automatic parking path planning

www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/BE470051

system applying the Hybrid A* procedure and RS curves, and then optimised the pathway by means of Bezier curves and gradient descent method. The study enhanced path search skill and flexibility in complex situations and resolved the disjointed curvature's issues, but the pathway was not sufficiently flat and it causes weakness in the tracking result. In the random sampling technique, Zhang et al. [6] proposed the series of principles for taking the reference points and updating route nodes that matched to the kinematic limitations, enhancing the discovering random tree algorithm, speedily. The results exposed that the new process was able to carefully control the vehicle to perform the parking works, but the tracking was problematic. In the intelligent optimisation technique, Chen et al. [9] suggested a training way utilising the syllabus learning with the aid of the position of rear axle's centre and the steering angle of front wheel as state variables, reached an arrangement success percentage of 91% though sufficient comfort based on deep reinforcement learning. Yet, it's process needed large numbers of data.

Presently, furthermost studies on path planning for automatic parking can mollify the curving limitations and difficulty avoidance limits throughout the process. Current parking route tracking approaches contain pure pursuit (PP) method [10], model predictive control (MPC) [11], and fuzzy control method [12]. Horváth et al. [13] recommended pure pursuit method. The PP algorithm modified lateral deviation by picking diverse target points and dynamically adjusting the preview interval. Experimental confirmation displayed good viability in real-world applications and improved tracking routine. Wu et al. [11] recommended a parking pathway pursuing control system applying MPC, which tracked routes involving clothoid curves, semicircles, and straight lines. The results showed that the method had excellent precision. Nakrani and Joshi [14] established a parallel parking process applying hybrid fuzzy inference model, which figured out several parking circumstances, counting sudden difficulty entrance throughout parking, the experimental results indicated that this technique could effectively complete parking in various parking situations, but its correctness was moderately weak.

Daxwanger and Schmidt [15] investigated path planning and tracking algorithms for automatic parallel parking, combining neural networks and fuzzy logic to achieve automatic parallel parking. Scicluna et al. [16] proposed a hardware solution for vehicle parallel parking and regarded fuzzy logic as a fast approach that could be implemented with the hardware. Cheng et al. [17] proposed fault-based steering control algorithm for segmental path planning problem for automatic parallel parking assist system in narrow parking places. Vorobieva [18] investigated automatic parallel parking with geometric continuous-curvature path planning. A geometric path was created by some arc, and it was transformed to the continuous path. Likewise, automatic parallel parking in a tiny spot, utilising path planning and control was reported by Vorobieva et al. [19]. Naderi Samani et al. [20] suggested a p-domain path-tracking controller for parking a car like mobile robot (CLMR). A path planning approach composed of two circular arcs connected by a tangential point was offered by Ji et al. [21] based on preview BP neural network PID controller. Li et al. [22] considered interior-point method (IPM) based on the simultaneous dynamic optimisation approach to solve the problem of automatic vehicle parallel parking. This global optimum method took a long time to find the optimal parking manoeuvre. Wu et al. [23] designed an ultrasonic sensor and applied the smart wheeled mobile robot (SWMR) automatic algorithm to solve car parallel parking problem depending on the experience of the actual drives. Nedamani et al. [24] proposed a fifth-degree polynomial curve for autonomous parallel parking, based on interpolating algorithms. The proposed curve offers the advantages of low computational cost, comfort, and independence from global waypoints.

Many path planning and control methods have been proposed to produce the parking function. Sampling-based strategies (SBS) have largely been used in robotic applications. In some SBS, planning is based on randomly sampling configuration space and finding connectivity inside this space. Common algorithms of SBS, which is also applied in parking, are rapidly-exploring random tree (RRT) [25], Hybrid A* search [26] and optimisation-based path planning strategy [27]. RRT methods [28] can plot in continuous space using heuristic information. The A* search method [29] discretises the search space, which guaranteed completeness and optimality. However, choosing the mesh size requires specialised knowledge. In the optimisation methods, local optimisation [30] was developed to find the shortest path in a short time; although global optimality cannot be guaranteed.

For marching methods, linear quadratic regulator (LQR) [31], sliding function control [32] and Fuzzy controller [33] were used. Fuzzy Logic based methods have been applied in the numerous research. Chen and Feng [34] combined both fuzzy inference and self-organising map (SOM) neural networks to approach the intelligent vision-based car-like vehicle backing system. This study not only explains the computer simulation's results, but also proves the practical car parking achievements in a real situation. Joshi and Zaveri [35] proposed a neuro-fuzzy system for navigating mobile robots; it contained finding distances from an obstacle and training how to control movement with neural network. Marasigan et al. [36] assumed Neuro-fuzzy controller for data which gained from a fifth-degree polynomial path in backward manoeuvre. Utilising the genetic algorithm, Aye et al. [37] optimised the image-based fuzzy controller for an automatic parking system, based on digital image processing. With the genetic algorithm the system can be flexible to parking conditions. Nari et al.[38] designed an automatic parking system using trajectory control method with face-to-face feedback. They assumed that with the addition of a gyroscope sensor, the system can know the direction facing the actual car and adjust the direction towards the destination using fuzzy logic control. The use of gyro sensors and fuzzy logic control in automatic parking systems greatly affects parking accuracy.

ANN approaches have been widely considered in the planning problems for parking. Heinen et al. [39] published an article on autonomous vehicle parking and pull out using ANNs; He designed a vehicle which could recognise environment by a sensor and used the neural networks to generate steering angle and velocity.

Farooq et al. [40] implemented a mobile robot navigation with ANNs using BP algorithm to solve the navigation problem. They mentioned that neural network has been a superior method which can learn complex nonlinear relationship between inputs and outputs. But applying the tangent sigmoid activation function would have made errors in the system and caused less efficiency in the neural controller. Considering the steering angle and speed as outputs, Zhou [41] studied on automatic car parallel parking and reviewed various methods to convert human expertise to machine learning in numerous iterations by ANNs.

Ji et al. [21] investigated path planning and tracking for vehicle parallel parking based on BP Neural Network PID Controller, experimentally. Liu et al. [42] presented a method to enumerate all the possible parking trajectories and corresponding steering actions, and then have the parking controller learn the relationship between the given initial-and-final state pairs and the corresponding sequence of steering actions using an ANN. Li et al. [43] proposed an end-to-end neural-network-based automatic parking controller. Moon et al. developed an automatic parking controller with a twin ANN architecture [44]. Ma and Wang [45] Mediate perception utilised AI techniques like convolutional neural networks (CNNs) to identify one or more objects. A well-established perception task effectively handled by AI is traffic sign recognition. The accuracy of AI such as deep neural networks (DNNs) has achieved 99.46. Genetic algorithm is used for detecting lanes for the navigation of AVs.

Kockelman proposed CNNs for detect the barrier with the accuracy of 80%. The SVM is proposed for more classification of CNN learnt feature. LSTM is used to learn camera images in order to determine for the wheel's angels. The result shows the LSTM outperform the CNN in this scenario. In total, the perception and situation awareness has a better contribution in the autonomous driving. The neural Network has suffered from the false detection. Although CNN, LSTM, and DBN for autonomous vehicle has proposed, it has still some issues. The DL use more intricate model structures, making parameter calibration computationally costly. Lack of the model for the hidden layers for training is an issue, and user has to do it within some trial and error. Previous optimisation-based approaches can generate accurate parking trajectories, but these methods cannot compute feasible solutions with extremely complex constraints in a limited time. Besides, the control methods, cannot address system limitations such as steering actuation limits. Recent research uses ANN-based approaches that can generate time-optimised parking trajectories in faster time without limitation. Recently, methods based on deep neural networks have been expected to solve the drawbacks of other offline approaches by manoeuvring vehicles without prior offline trajectory planning. ANNs consist of interconnected neurons that process information collaboratively to solve problems. By training an ANN using a dataset generated by simulation or experiment, the ANN learns hyper-dimensional relationships between the current vehicle states and the appropriate vehicle manoeuvring signals. Instead of calculating the parking trajectory offline, the ANN-based parking controller can yield a direct manoeuvring signal of the steering angle and velocity online, while the vehicle is moving into a parking space.

As mentioned before, a common training algorithm previously used for ANNs for collision avoidance in robot motion was BP. It is a supervised method, but this algorithm had some drawbacks; It was very complicated and slow in convergence. This algorithm has problem in estimation, there was no defined amount of hidden layer for the network, and it was arduous to estimate the number of hidden layers in this network. These are some significant issues to make this algorithm inefficient for using in car parking project. Generally, parking a car-like robot with this algorithm was not only imprecise but also car might be collided with obstacles in some circumstances. Hence, the present study aims to improve automatic car parallel parking systems using ANN with SOM algorithm. It explores an AI-based method to control a car-like robot, focusing on automatic parking in various circumstances. Discussing about the disadvantages of BP algorithm, this paper aims to achieve a more efficient collision-free trajectory planning scheme for automated parking in narrow parking spaces.

2 RESEARCH METHOD

2.1 Self-organising map definition

To date, many methods have been proposed for parallel parking cars. However, this article presents an unsupervised algorithm based on artificial neural networks (ANN) called the self-organising map (SOM) [46]. This algorithm clusters similar datasets together by comparing input data with neuron links. When a similarity is detected between the input data and the neurons, the neurons adjust to become even closer to the inputs. Consequently, neighbouring neurons collaborate with this primary neuron and also adjust to get closer to the input data. In this way, all data are effectively clustered.

2.2 Dataset preparation

As shown in Fig. 1, it is crucial to choose a suitable and appropriate robot for simulation, as a robot's degree of manoeuvrability, including both mobility and steerability, differs for each type. For this simulation, an Ackerman robot is employed.

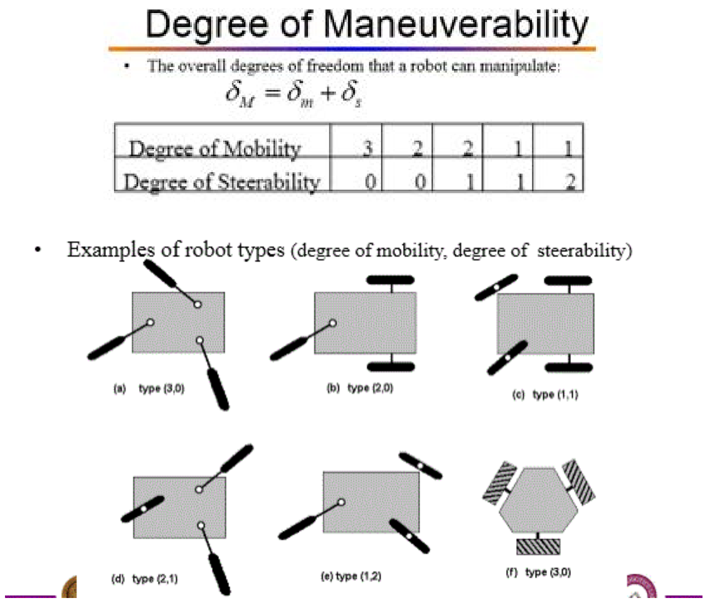


Figure 1: Different types of robots based on their degree of manoeuvrability.

In order to prepare adequate data, a displacement sensor capable of measuring distances between the four sides of the car-like robot and potential obstacles in various movements and scenarios will be utilised. Data from these sources are used to create the feature vector, which is considered as input data by the robot controller. In addition, steering angle and velocity are presumed to be outputs of the robot programs. After obtaining the distance between the four sides of the Ackerman robot and the obstacles during each movement, the data is classified based on the steering angle and speed of the robot within each class. As another feature vector, the state of movement variable is considered. As shown in Fig. 2, the steering angle and direction of the vehicle play an important role in the classification process.

Once all distances between the four sides of the car-like robot and any possible obstacles have been measured, the state of movement is acquired according to both steering angle and velocity. According to Fig. 3, both the state of movement and all distances gained from multiple scenarios are taken into consideration as inputs to the neural network. By training these input data with SOM algorithm, similar input data would be categorised by SOM. It should be noted that SOM determined multiple clusters for these similar input data, and these data will be replaced by a single cluster in datasets. The remaining elements in the dataset are clusters, steering angle, and velocity. Therefore, the clusters are designated as inputs, while the velocity and steering angle are designated as outputs. The SOM training process was conducted using the NCTOOL in MATLAB.

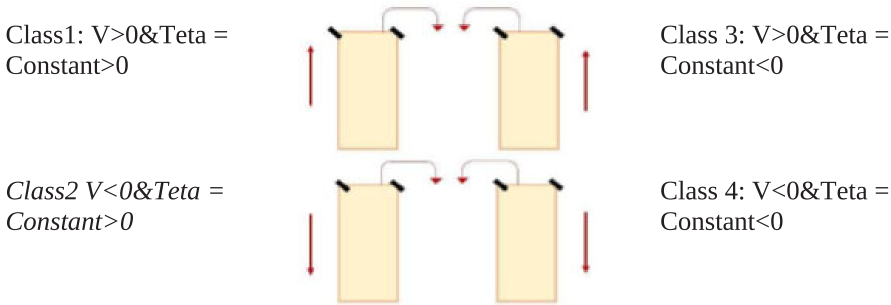


Figure 2: The state of movement according to the state of both velocity and steering angle.

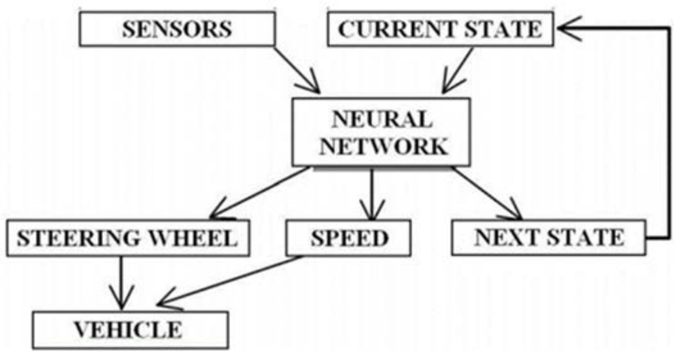


Figure 3: Artificial neural networks model scheme.

The experiments have resulted in various steering angles, but each cluster maintains a consistent speed. Therefore, it is optimal to consider the mode of the steering angles within each cluster. For each cluster, the input data consist of a consistent speed and the mode of the steering angles as outputs. Following the training phase, it is crucial to evaluate the car's performance by testing the trained neural network. Similar to the training phase, input data are processed by the SOM in the test phase, but now the input data are obtained from a single parking scenario. This generates a list of clusters specific to that scenario. Finally, the steering angle and speed associated with each cluster, obtained from the training phase across multiple scenarios, are applied to the corresponding cluster in the current parking scenario. These parameters – steering angle and speed – are essential for guiding the car-like robot simulator through navigation tasks facilitated by the ANN.

3 IMPLEMENTATION

Once a car moves forward or backward, it undergoes a rotation, resulting in a change in its orientation angle. If a car moves from an initial point to a second point, the new orientation angle of the car is the sum of its initial angle and the rotation angle it undergoes. Following this, a kinematic model is essential to convert the local coordinates to the global coordinates, considering the updated orientation angle of the car. This model helps in accurately determining the position and orientation of the car relative to its movement and rotation.

3.1 Ackerman kinematic model needed for simulation

According to the kinematic model, it is necessary to convert local coordinates to global coordinates in order to use any simulation. In this model, the front wheels can be turned, while the back wheels cannot. This is similar to the Ackerman robot used for providing datasets. Fig. 4 demonstrates the kinematic model. In this figure, the car-like robot is represented by a rectangle with four wheels. In eqns (1) and (2), X and Y indicate the coordinates of the rear wheels, θ represents the rotation angle of the vehicle, φ is the steering angle. Additionally, in eqn (3), L is the wheelbase. Furthermore, the car-like robot can be moved by two commands (φ, V). Eqns (1)–(3) were derived according to the nonholonomic motion estimation [47].

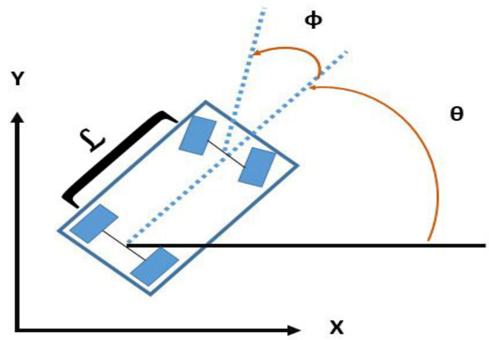


Figure 4: The kinematic model.

$$\dot{x} = v \cos \varphi \cos \theta \tag{1}$$

$$\dot{y} = v \cos \varphi \sin \theta \tag{2}$$

$$\dot{\theta} = \frac{v}{l} \sin \varphi \tag{3}$$

4 RESULTS AND DISCUSSION

4.1 Results of the back propagation algorithm employment

Although we presume all distances and states of movement as inputs to the neural networks, we cannot directly use them in the backpropagation network. Feature scaling or normalisation is essential for preparing data, including measurements of distances between the robot and possible obstacles, for efficient training. Without this step, we would have redundant neurons and low training accuracy. Eqn (4) is used for feature scaling. Similar to the SOM, input data in backpropagation must be classified. After training the input data with the backpropagation algorithm, we obtained a network with 5 inputs, 25 hidden neurons, and 2 outputs, as shown in Fig. 5. This process was carried out using MATLAB’s NFTOOLS.

$$x' = \frac{a - \bar{x}}{\sigma} \tag{4}$$

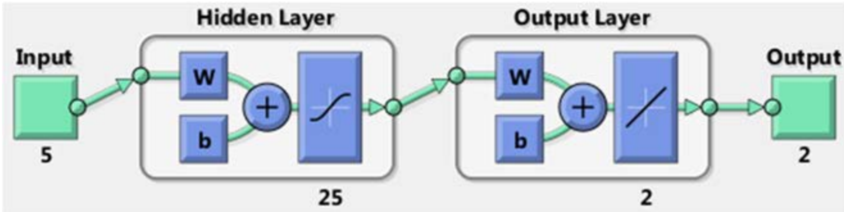


Figure 5: Topology of network after training data with the backpropagation algorithm.

After completing the training phase with multiple scenarios, the learned model will be applied to a different parking scenario during the test phase. Similar to the training process, scaling and classification of input data will be conducted. However, in this case, only data from a single scenario is available for testing. Once the input data are processed and the model is trained, the outputs will be speed and wheel angles. These outputs need to be converted from the scaled format used during training back to the original format suitable for the simulation program. This conversion is crucial because the simulation program operates based on these two parameters – speed and wheel angles – to accurately simulate the car's movements in the parking scenario. Therefore, maintaining accuracy and fidelity in this conversion process is essential for realistic simulation outcomes.

4.2 Analysing backpropagation with its plots

Fig. 6 illustrates the experimental results for backpropagation training. The figure shows the error plotted against epochs. Typically, the error decreases as training progresses through multiple epochs. Training is typically halted when the model begins to approach overfitting, which occurs when the model performs well on the training data but fails to generalise to unseen data. In this study, the best performance was achieved at 93 epochs with a mean squared error (MSE) of approximately 0.1. This point represents a balance where the model has learned effectively from the training data without overfitting, as indicated by continued improvement in validation error.

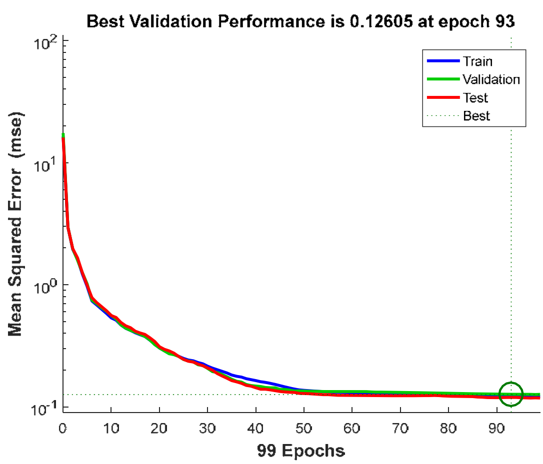


Figure 6: MSE versus epochs performance plot.

Fig. 7 depicts the outcomes of the training, validation, and test phases in a regression plot. This plot illustrates the relationship between output and target values, with the best outcome occurring when the output exactly matches the target. In regression analysis, an ideal scenario is represented by R=1, indicating perfect correlation between output and target values. In this experiment, the regression coefficient (R) is 0.93, indicating a strong correlation. A line of best fit, known as the fit line, is drawn to assess how closely it aligns with the dashed line representing perfect alignment. The closer the fit line matches the dashed line, the better the neural network performs, with lower error. Error in this context refers to the difference between the target (desired output) and the actual output. However, the scattered data points indicate variability in the data which makes it challenging for backpropagation to accurately predict outputs based on these inputs. As a result, achieving optimal performance through backpropagation can be difficult with such scattered data.

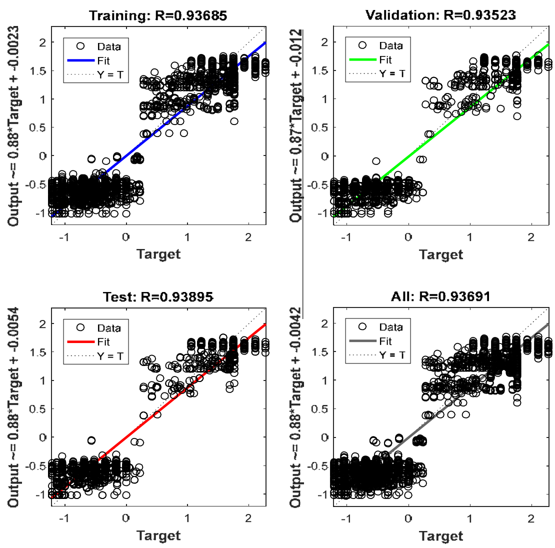


Figure 7: Regression plot includes the data position in three phases training, validation and test.

As shown in Fig. 8, the steering angle data in backpropagation exhibits significantly more scattering compared to that in the SOM. Backpropagation data is dispersed far from the central line, whereas SOM data clusters closely around it. This analysis underscores the distinct behaviour of data within these algorithms. In all charts referenced above, the vertical axis represents output data, while the horizontal axis represents target data. Both axes are divided into positive and negative areas due to the sign of the steering angle.

In Fig. 9(b), the actual (target) steering values range from +22 to +41, whereas the predicted steering values by backpropagation range from +25 to +59. This discrepancy indicates that backpropagation has lower precision in estimating positive angles. Conversely, in the negative angles, where actual steering values range from −26 to −58, backpropagation predicts values ranging from −25 to −58. Data clustering is more pronounced in the negative region, suggesting that backpropagation estimates are more accurate there compared to the

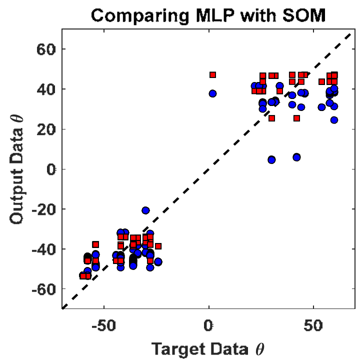


Figure 8: Comparison of data scattering in SOM with back propagation.

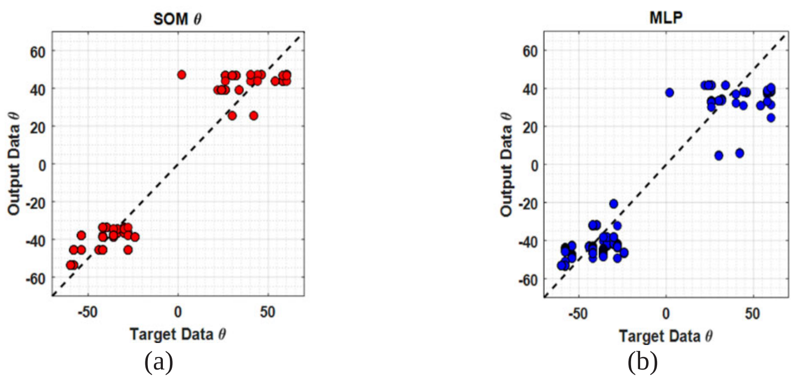
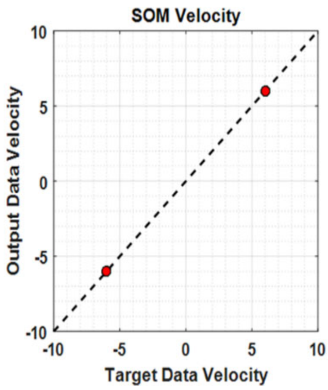


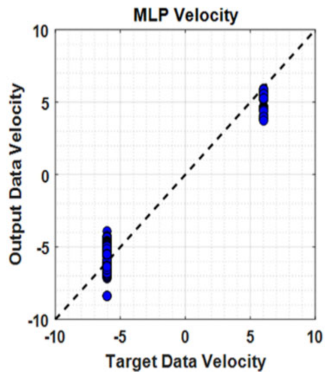
Figure 9: Target steering angle data vs predicted steering angle data. These results were obtained after training input data by SOM labelled in (a) and after training input data by backpropagation labelled in (b).

positive region. According to Fig. 9(a), SOM predicts steering data ranging from +20 to +50 in positive angles, while the actual data ranges from +24 to +58. In the negative angles, where actual steering data ranges from −32 to −60, SOM predicts data ranging from −24 to −53. Importantly, despite the differences in value ranges, SOM exhibits significantly lower data scattering; the dispersion around the central line is minimal, indicating closer proximity to the desired outputs (black line). This highlights the effectiveness of SOM in this context. The superiority of the SOM over backpropagation in training car parallel parking data stems from its ability to cluster data more effectively and reduce data scattering, thereby achieving more accurate predictions and better performance overall.

In Fig. 10, the comparison between errors in the SOM and backpropagation is illustrated. As depicted in Fig. 10(a), the red square points are higher than the blue points, and the data around the central line is less scattered compared to the blue points. The red points align closely with the central data, indicating that SOM provides highly accurate estimates for these data points. Notably, the predicted data using SOM closely matches the target data. On the other hand, Fig. 10(b) shows that data points in backpropagation multi-layer perceptron (MLP) do not align precisely with the central line. There is a noticeable discrepancy between the target data and the predicted data in backpropagation. The target velocity data ranges from −6 to 6, whereas the predicted values range from 3 to 6 for positive velocity and from 4 to 8 for negative velocity. This variability suggests that backpropagation MLP struggles with accuracy in both velocity and steering angle predictions. Due to the classification approach utilised by SOM, it demonstrates greater precision in estimating speed. This precision stems from the fact that speed predictions are derived from well-defined classes within SOM. Consequently, SOM consistently delivers more desirable results compared to backpropagation MLP, making it the preferred method for car parallel parking tasks in terms of both velocity and steering angle estimation.



(a)



(b)

Figure 10: Comparison between target velocity data with predicted steering angle data in both SOM and backpropagation algorithm.

5 CONCLUSION

This article presents a survey on using neural networks for car parallel parking, comparing traditional methods like backpropagation with MLP structures to a proposed SOM algorithm. The study explores both the advantages and disadvantages of using backpropagation for this application. To improve car parallel parking, the SOM-based algorithm is introduced. A car-like robot was assembled and programmed based on datasets gathered from a distance sensor integrated into the robot. During the training phase, datasets from multiple scenarios were fed into the neural network to learn the relationship between input (sensor data) and output (steering angle and speed). This enabled the network to predict outputs for new inputs during the test phase. A simulation program was then developed based on the output dataset obtained from the test phase. Tools such as NetLab and RapidMiner were considered for neural network implementation, but practical experiments were conducted using MATLAB's NCTOOL. It was observed that dataset feature scaling improved vector balance, enhancing the performance of the experiments on the SOM. As part of the experiment, NCTOOL was used to classify the input dataset for a variety of scenarios. Outputs were clustered during both the training and test phases, where each cluster exhibited varied steering angles while maintaining a consistent speed established during training. In simulations involving numerous small movements, parking accuracy using the self-organising map outperformed backpropagation MLP. The density of data posed a challenge for backpropagation in establishing effective function relationships, whereas the self-organising map efficiently clustered data regardless of density, thereby controlling the vehicle parking procedure effectively. In the future, researchers may explore deep learning methodologies to improve the performance of automatic neural network-based controllers for parallel parking cars by leveraging additional sensors with higher precision and performance. With continuous advancements like this, vehicle control systems based on neural networks will continue to be refined.

REFERENCES

[1] Wang, W., Song, Y., Zhang, J. & Deng, H., Automatic parking of vehicles: A review of literatures. *Int. J. Automot. Technol.*, **15**, pp. 967–978, 2014.

[2] Song, J., Zhang, W., Wu, X., Cao, H., Gao, Q. & Luo, S., Laser-based SLAM automatic parallel parking path planning and tracking for passenger vehicle. *IET Intell. Transp. Syst.*, **13**(10), pp. 1557–1568, 2019.

[3] Katrakazas, C., Quddus, M., Chen, W.-H. & Deka, L., Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.*, **60**, pp. 416–442, 2015.

[4] Chen, Q., Gan, L., Chen, B., Liu, Q. & Zhang, X., Parallel parking path planning based on improved arctangent function optimization. *Int. J. Automot. Technol.*, **24**(1), pp. 23–33, 2023.

[5] Ren, B.T. et al., An automatic parking path optimization method based on hybrid A* and variable radius RS curve. *Chinese J. Highw.*, **35**(07), pp. 317–327, 2022.

[6] Zhang, J. et al., Autonomous valet parking path planning based on hierarchical architecture. *J. Southeast Univ. (Natural Sci. Ed.)*, **51**(5), pp. 8–887, 2021.

[7] Su, B., Yang, J., Li, L. & Wang, Y., Secondary parallel automatic parking of endpoint regionalization based on genetic algorithm. *Cluster Comput.*, **22**, pp. 7515–7523, 2019.

[8] Vorobieva, H., Glaser, S., Minoiu-Enache, N. & Mammar, S., Geometric path planning for automatic parallel parking in tiny spots. *IFAC Proc.* **45**(24), pp. 36–42, 2012.

[9] Chen, J., Lan, X. & Chen, F., Autonomous parking path planning based on improved deep reinforcement learning. *J. Chongqing Univ. Technol. (Natural Sci.)*, **35**(7), pp. 17–27, 2021.

[10] Shin, H., Kim, M.-J., Baek, S., Crane, C.D. & Kim, J., Perpendicular parking path generation and optimal path tracking algorithm for auto-parking of trailers. *Int. J. Control. Autom. Syst.*, **20**(9), pp. 3006–3018, 2022.

[11] Wu, Y., Li, X., Gao, J. & Yang, X., Research on automatic vertical parking path-planning algorithms for narrow parking spaces. *Electronics*, **12**(20), p. 4203, 2023.

[12] Wang, W. & Hou, Z., Model-free adaptive control based automatic parking scheme. *Control Decis.*, **37**(8), pp. 2056–2066, 2022.

[13] Horváth, E., Hajdu, C. & KHorös, P., Novel pure-pursuit trajectory following approaches and their practical applications. *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 597–602, 2019.

[14] Nakrani, N. & Joshi, M., An intelligent fuzzy based hybrid approach for parallel parking in dynamic environment. *Procedia Comput. Sci.*, **133**, pp. 82–91, 2018.

[15] Daxwanger, W.A. & Schmidt, G.K., Skill-based visual parking control using neural and fuzzy networks. *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, pp. 1659–1664, 1995.

[16] Scicluna, J., Gatt, N., Casha, E., Grech, O. & Micallef, I., FPGA-based autonomous parking of a car-like robot using fuzzy logic control. *2012 19th IEEE Int. Conf. Electron. Circuits Syst. (ICECS 2012)*, pp. 229–232, 2012.

[17] Cheng, K., Zhang, Y. & Chen, H., Planning and control for a fully-automatic parallel parking assist system in narrow parking spaces. *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1440–1445, 2013.
[18] Vorobieva, H., Glaser, S., Minoiu-Enache, N. & Mammar, S., Automatic parallel parking with geometric continuous-curvature path planning. *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 465–471, 2014.
[19] Vorobieva, H., Glaser, S., Minoiu-Enache, N. & Mammar, S., Automatic parallel parking in tiny spots: Path planning and control. *IEEE Trans. Intell. Transp. Syst*., **16**(1), pp. 396–410, 2014.
[20] Naderi Samani, N., Danesh, M. & Ghaisari, J., Parallel parking of a car-like mobile robot based on the P-domain path tracking controllers. *IET Control Theory and Appl*., **10**(5), pp. 564–572, 2016.
[21] Ji, X., Wang, J., Zhao, Y., Liu, Y., Zang, L. & Li, B., Path planning and tracking for vehicle parallel parking based on preview BP neural network PID controller. Trans. Tianjin Univ., 21(3, pp. 199–208, 2015.
[22] Li, B., Wang, K. & Shao, Z., Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Trans. Intell. Transp. Syst*., **17**(11), pp. 3263–3274, 2016.
[23] Wu, T.-F., Tsai, P.-S., Hu, N.-T. & Chen, J.-Y., Research and implementation of auto parking system based on ultrasonic sensors. *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pp. 643–645, 2016.
[24] Nedamani, H.R., Khiabani, P.M. & Azadi, S., Intelligent parallel parking using adaptive neuro-fuzzy inference system based on fuzzy C-means clustering algorithm. *SAE Technical Papers*, 2018.
[25] Lee, B., Wei, Y. & Guo, I.Y., Automatic parking of self-driving car based on lidar. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*., **42**, pp. 241–246, 2017.
[26] Qin, Z., Chen, X., Hu, M., Chen, L. & Fan, J., A novel path planning methodology for automated valet parking based on directional graph search and geometry curve. *Rob. Auton. Syst*., **132**, 103606, 2020.
[27] Zips, P., Böck, M. & Kugi, A., Optimisation based path planning for car parking in narrow environments. *Rob. Auton. Syst*., **79**, pp. 1–11, 2016.
[28] Banzhaf, H., Sanzenbacher, P., Baumann, U. & Zöllner, J.M., Learning to predict ego-vehicle poses for sampling-based nonholonomic motion planning. *IEEE Robot. Autom. Lett*., **4**(2), pp. 1053–1060, 2019.
[29] Chen, C., Rickert, M. & Knoll, A., Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering. *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1148–1153, 2015.
[30] Lee, S., Lim, W. & Sunwoo, M., Robust parking path planning with error-adaptive sampling under perception uncertainty. *Sensors*, **20**(12), p. 3560, 2020.
[31] Fan, Z. & Chen, H., Study on path following control method for automatic parking system based on LQR. *SAE Int. J. Passeng. Cars-Electronic Electr. Syst*., **10**(2016-01–1881), pp. 41–49, 2016.
[32] Du, X. & Tan, K.K., Autonomous reverse parking system based on robust path generation and improved sliding mode control. *IEEE Trans. Intell. Transp. Syst*., **16**(3), pp. 1225–1237, 2014.
[33] Ballinas, E., Montiel, O., Castillo, O., Rubio, Y. & Aguilar, L.T., Automatic parallel parking algorithm for a carlike robot using fuzzy PD+ I control. *Eng. Lett*., **26**(4), 2018.
[34] Chen, C.-Y. & Feng, H.-M., Hybrid intelligent vision-based car-like vehicle backing systems design. *Expert Syst. Appl*., **36**(4), pp. 7500–7509, 2009.

[35] Joshi, M.M. & Zaveri, M.A., Reactive navigation of autonomous mobile robot using neuro-fuzzy system. *Int. J. Robot. Autom.*, **2**(3), p. 128, 2011.
[36] Marasigan, A.A., Saberon, J.T., San Jose, I.M.B., Sevilla, D.P.B. & Bandala, P.A.T., Autonomous parallel parking of four wheeled vehicles utilizing adoptive fuzzy-neuro control system. *2014 IEEE Reg. 10 Symp.*, pp. 640–644, 2014.
[37] Aye, Y.Y., Watanabe, K., Maeyama, S. & Nagai, I., An intelligent parking system for vehicles using an image-based fuzzy controller. *Int. J. Smart Mater. Mechatronics*, **4**(1), 2017.
[38] Nari, M.I., Mustain, Z., Kautsar, S. & Utomo, S.B., Parallel parking system design with fuzzy logic control. *J. Nas. Tek. Elektro*, **10**(2), 2021. https://doi.org/10.25077/jnte.v10n2.904.2021.
[39] Heinen, M.R., Osório, F.S., Heinen, F.J. & Kelber, C., Autonomous vehicle parking and pull out using artificial neural networks. *Proceedings of the I Workshop on Computational Intelligence (WCI)*, 2006.
[40] Farooq, U., Amar, M., Asad, M.U., Hanif, A. & Saleh, S.O., Design and implementation of neural network based controller for mobile robot navigation in unknown environments. *Int. J. Comput. Electr. Eng.*, **6**(2), pp. 83–89, 2014.
[41] Zhou, W., An improved approach for automatic parallel parking in narrow parking spaces. 2015.
[42] Liu, W., Li, Z., Li, L. & Wang, F.-Y., Parking like a human: A direct trajectory planning solution. *IEEE Trans. Intell. Transp. Syst.*, **18**(12), pp. 3388–3397, 2017.
[43] Li, R., Wang, W., Chen, Y., Srinivasan, S. & Krovi, V.N., An end-to-end fully automatic bay parking approach for autonomous vehicles. *Dynamic Systems and Control Conference*, V002T15A004, 2018.
[44] Moon, J., Bae, I. & Kim, S., Automatic parking controller with a twin artificial neural network architecture. *Math. Probl. Eng.*, **2019**(1), 4801985, 2019.
[45] Ma, Y. & Wang, Z., Artificial intelligence applications in the development of autonomous vehicles: A survey, 2020.
[46] Ghaseminezhad, M.H. & Karami, A., A novel self-organizing map (SOM) neural network for discrete groups of data clustering. *Appl. Soft Comput.*, **11**(4), pp. 3771–3778, 2011.
[47] De Luca, A., Oriolo, G. & Samson, C., Feedback control of a nonholonomic car-like robot. *Robot Motion Plan. Control*, pp. 171–253, 2005.

www.witpress.com, ISSN 1743-3533 (on-line)