Reduced-Scale Mobile Robots for Autonomous Driving Research

Zhuopeng Xie, Mohsen Ramezani, and David Levinson

***Abstract*— Reduced-scale mobile robots (RSMRs) are extensively used for studying autonomous driving due to their ability to test models and algorithms in physical environments, their lack of constraints related to regulations and laws, and their advantages of low cost and space-saving. Nevertheless, there is currently a lack of systematic analysis and review of these autonomous driving studies involving RSMRs. Hence, this paper comprehensively reviews 134 studies on the application of RSMRs in autonomous driving research. Through the analysis of these studies, we summarize the commonly used methods for three modules (perception, decision-making, and actuation) of the autonomous driving process of RSMRs, and thoroughly examine the main applications (navigation and obstacle avoidance, vehicle fleet coordination, intersection management, parking control, drift control, passenger unease, and hands-free control) covered in these studies. Furthermore, we identify the limitations and gaps in the existing studies related to RSMRs, and provide recommendations for future research initiatives: 1) focusing on common interactive driving events in real-world traffic such as lane changing, merging, cut-in, and overtaking, 2) extending the experiment duration and distance, 3) increasing the randomness in experimental design, 4) exploring the transferability of autonomous driving algorithms from RSMRs to real vehicles, 5) researching on the mixed fleet consisting of manually controlled RSMRs and self-driving RSMRs.**

***Index Terms*— Autonomous driving, reduced-scale mobile robot, control, intelligent transportation systems, review.**

NOMENCLATURE

α	The steering angle of RSMR's inner wheel.
β	The steering angle of RSMR's outer wheel.
δ	The steering angle in the bicycle model.
γ	The discount factor for calculating Q-values in RL.
μ_{fx}	Longitudinal friction coefficient of the front wheel in the bicycle model.
μ_{fy}	Lateral friction coefficient of the front wheel in the bicycle model.
μ_{rx}	Longitudinal friction coefficient of the rear wheel in the bicycle model.
μ_{ry}	Lateral friction coefficient of the rear wheel in the bicycle model.
ω	RSMR's wheel rotation speed.
ψ	RSMR's heading angle.
τ	An episode, i.e., a sequence of state, action, and reward in PG.
θ	The parameters of approximators or models.
a	The action in RL.
a_{d}	The desired acceleration in the third order model.
$\mathbf{a}_t$	The action performed by the expert at time t.
b	The expected reward in PG.
c_{e}	A constant that reflects the engine dynamics in the third order model.
d_i^{s}	The standstill distance between vehicle i and vehicle $i-1$ in a fleet.
$d_i(t)$	The desired distance between vehicle i and vehicle $i-1$ in a fleet.
E	The function used to explain the unobservable information in IL.
e_{PID}	The input error of a PID controller.
F	The approximator used to learn the expert's bahavior in IL.
$f_{\mathrm{fx}}^{\mathrm{w}}$	Longitudinal friction force in the wheel frame.
$f_{\mathrm{fy}}^{\mathrm{w}}$	Lateral friction force in the wheel frame.
f_{fz}	The support force of the front wheel in the bicycle model.
f_{M}	The model used to predict RSMR's motion state in MPC.
$f_{\mathrm{rx}}^{\mathrm{b}}$	Longitudinal friction force in the body frame.
$f_{\mathrm{ry}}^{\mathrm{b}}$	Lateral friction force in the body frame.
f_{rz}	The support force of the rear wheel in the bicycle model.
$\mathbf{h}$	Unobservable information in IL.
h_{c}	The constant time headway in a fleet.
i	Vehicle index in a fleet.
I_{z}	RSMR's rotational inertia along the z-axis.
J	The objective function in MPC.
K	The distance between the two kingpins of RSMR's front axle.
K_{d}	The coefficient for derivative term of a PID controller.
K_{i}	The coefficient for integral term of a PID controller.
K_{p}	The coefficient for proportional term of a PID controller.
L	RSMR's wheelbase.
l_{f}	The distance between RSMR's front axle and its center of gravity.
l_{r}	The distance between RSMR's rear axle and its center of gravity.
m	RSMR's mass.

Manuscript received 8 December 2023; revised 26 May 2024 and 8 July 2024; accepted 8 July 2024. This work was supported in part by Australian Research Council (ARC) Discovery Project under Grant DP220100882. The Associate Editor for this article was M. Brackstone. *(Corresponding author: Zhuopeng Xie.)*

The authors are with the School of Civil Engineering, Faculty of Engineering, The University of Sydney, Sydney, NSW 2050, Australia (e-mail: zhuopeng.xie@sydney.edu.au; mohsen.ramezani@sydney.edu.au; david.levinson@sydney.edu.au).

Digital Object Identifier 10.1109/TITS.2024.3426508

N_{f}	The number of vehicles in a fleet.
$\mathbf{o}_t$	The information observed by RSMR at time t.
Q	The action-value function in RL.
$\hat{Q}$	The action-value function value after iteration in RL.
$\bar{R}$	The estimation of cumulative reward for an episode in PG.
R	The reward function in RL.
r	The reward value at each moment in RL.
s	The state in RL.
s_t	Laplace transform of $v(t)$.
t_{d}	The delay that reflects the throttle actuator dynamics in the third order model.
T_n	The number of time steps of episode n in PG.
T_{p}	The length of prediction in MPC.
U	RSMR's input signal.
$\mathrm{U_l}$	The lower limit of RSMR's input signal.
u_{PID}	The output of a PID controller.
$\mathrm{U_u}$	The upper limit of RSMR's input signal.
v	RSMR's velocity.
v_{d}	RSMR's desired velocity.
X	RSMR's position and motion state.
x	The horizontal coordinate of RSMR's position.
$\mathrm{X_d}(t)$	The current desired motion state of RSMR.
$\mathrm{X_l}$	The lower limit of RSMR's motion state.
$\mathrm{X_u}$	The upper limit of RSMR's motion state.
y	The vertical coordinate of RSMR's position.
$\dot{}$	The first order derivative.
$\ddot{}$	The second order derivative.
∇	The gradient operator.

I. Introduction

TODAY, autonomous driving (AD) is increasingly becoming a research hot-spot in both academia and industry. It has the potential to improve roadway safety, mitigate traffic congestion, and reduce energy consumption. In this context, numerous studies conducted AD experiments and designed models and algorithms to tackle different AD tasks. The experimental equipment used in existing studies can be categorized into four main types: full-size vehicles, driving simulators, simulation models or software, and reduced-scale mobile robots (RSMRs).

Undoubtedly, the most authentic and effective form of experimentation among these types is the real-world, full-size vehicle test, as it fully captures the characteristics of actual vehicles and the real driving environment [1], [2], [3]. However, this type of test comes with limitations. First, it requires autonomous vehicles (AVs), test tracks, and various expensive devices such as high-performance sensors and computers, which can result in high costs. Second, due to relevant laws, regulations, and ethical considerations, some experiments that might endanger people's safety are not permitted, e.g., assessing a vehicle's ability to identify pedestrians and apply brakes to avoid collisions.

Driving simulators provide an effective approach to simulating different scenarios and the movements of vehicles in a high-fidelity environment. The flexibility and controllability offered by driving simulation tests are relatively high, allowing developers to design various scenarios and repeat the tests multiple times to meet research requirements [4]. However, the simulation environment is not physical but computer-based, which lacks fidelity and introduces discrepancies compared to real-world situations.

In addition to driving simulators, simulation models and software such as AIMSUN [5], Vissim [6], SUMO [7], Silo [8], Gazebo [9], Carla [10], CarSim [11], Cellular Automata [12], and MATLAB [13], are widely used in AD research. These methods are cost-effective and convenient for implementing and evaluating AD algorithms. However, similar to driving simulators, they lack the ability to accurately replicate realistic traffic conditions, as they are computational environments.

To tackle these challenges, RSMRs with corresponding reduced-scale test tracks have been employed in AD research. For instance, Vasconcelos Filho et al. [14] developed a cooperative robotic platooning testbed, using a 1/10 scale platform to assess the reliability and safety of wireless communications in safety-critical scenarios. Lee et al. [15] built a 1/10 scale traffic environment that includes 50 RSMRs and a set of traffic facilities such as roads, traffic signs, and signals. This platform has facilitated numerous workshops and exchange events, attracting participation from over 11 research institutions in China. Also, various AD competitions of RSMRs have been held around the world to promote the development of AD. Some well-known competitions include DARPA Robotics Challenge [16], Festival Nacional de Robótica [17], and European Land Robot Trial [18]. In addition, several manufacturers produce programmable RSMRs to support the development of AD algorithms and academic research, including companies like Yahboom, ZMP, and Quanser [19], [20], [21], [22].

Due to their wide application in the field of AD, this paper aims to review the existing studies that use such RSMRs for AD research to showcase the academic progress made in AD. In order to make the review content more clear and specific, we will first define and briefly introduce the RSMR.

The RSMR in this paper refers to a physical, scaled-down, car-like, and programmable device with AD capabilities. Common scale ratios for RSMRs include 1/5 [23], 1/10 [15], 1/16 [24], or 1/20 [25], as shown in Figure 1.

Typically, the AD system of a RSMR comprises three functional modules: perception, decision-making, and actuation. These modules represent the primary research components of the existing studies.

The perception module integrates various sensors into the RSMR, including camera, LiDAR, radar, and inertial measurement units (IMUs), to perceive the surrounding environment. By using these sensors, the RSMR gains the ability to capture visual, depth, and spatial information, enabling it to detect and track objects, identify road characteristics, and estimate its own pose and motion. The perception module processes the sensor data using different techniques such as computer vision, Machine Learning (ML), and sensor fusion, to analyze and interpret the perceived information and generate a representation of the surrounding environment. This representation is the basis of RSMR decision-making and RSMR actuation.

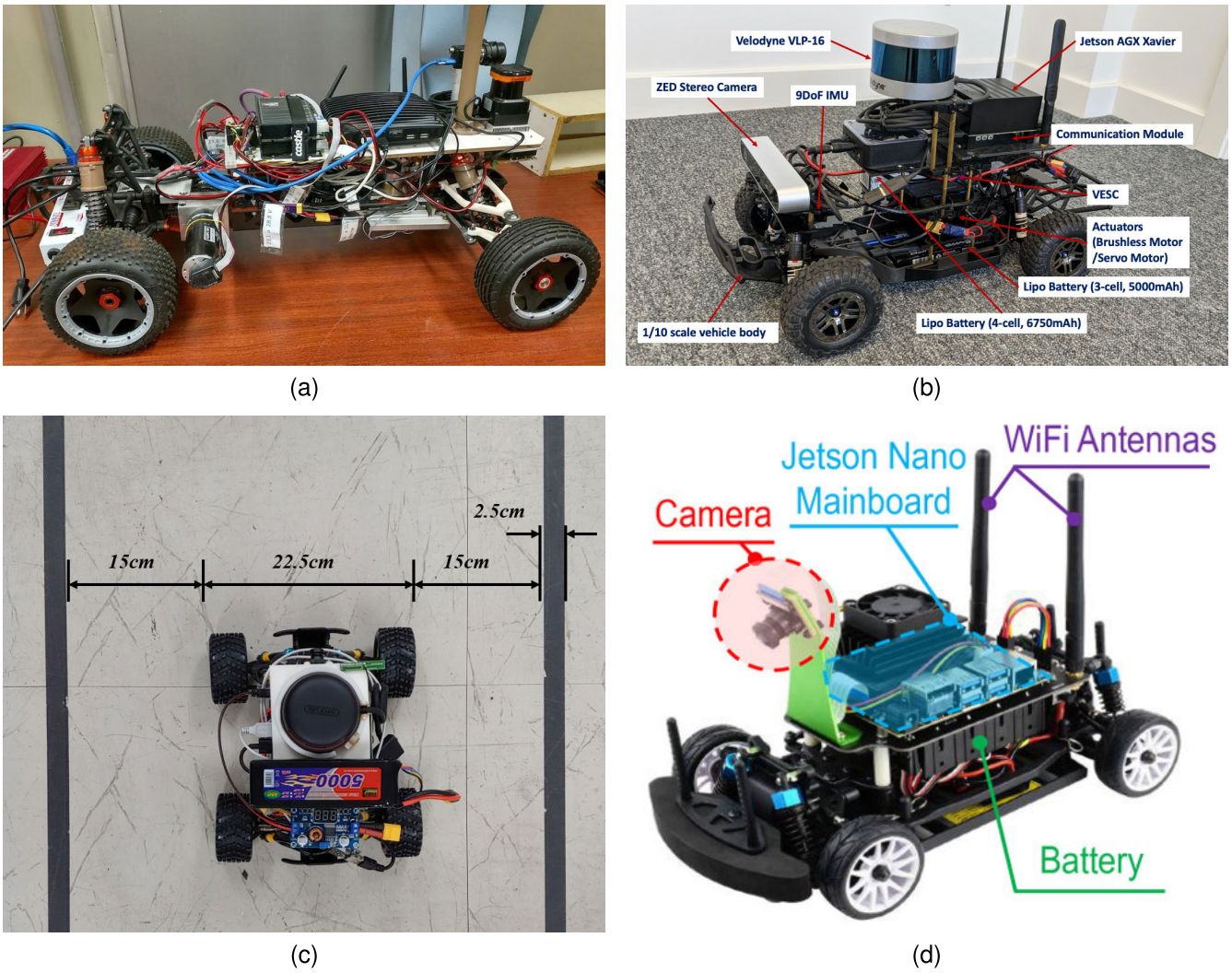

Fig. 1. Schematic diagrams of RSMRs with different scales: (a) 1/5 [23], (b) 1/10 [15], (c) 1/16 [24], (d) 1/20 [25].

The decision-making module incorporates algorithms that analyze the RSMR's current state, predict future environments, and determine appropriate actions. It facilitates informed decisions relating to actions like moving at constant speed, accelerating, or changing lanes, and provides the RSMR with motion commands in the form of vehicle kinematic parameters such as linear velocity, angular velocity, or acceleration.

The actuation system further processes the commands generated by the decision-making module to enhance the smoothness and steadiness of RSMR's motion. Subsequently, these processed commands are transmitted to the RSMR's actuator to initiate the desired movements.

Compared to other experimental devices or approaches like full-size vehicles, driving simulators, and simulation models or software, the RSMR used for AD research offers several advantages:

- First, the designed AD algorithms can be visually achieved and tested in a physical environment, ensuring the experiments' validity and reliability.
- Second, experiments conducted with RSMRs are not subject to regulatory and ethical constraints, which provides greater flexibility and controllability.
- Third, the RSMRs and reduced-scale test tracks can significantly reduce the space requirements and expenses.

The remainder of this paper is structured as follows. Section II outlines the process of gathering and selecting relevant studies. Section III presents the commonly used methods for the three modules (perception, decision-making, and actuation) of RSMRs' autonomous driving. Section IV categorizes the main applications covered in the literature. Section V concludes this paper by summarizing the findings, identifying the limitations and gaps in RSMR research, and providing suggestions for future research.

II. Literature Search and Selection

This paper uses three databases for the literature search, including Web of Science, Scopus, and IEEE. The search expression ("wheeled robot" OR "robocar" OR "robot car" OR "robotic vehicle" OR "mobile robot" OR "car-like robot") AND ("driverless" OR "unmanned driving" OR "self-driving" OR "autonomous driving" OR "autonomous vehicle" OR "automated vehicle") was employed to retrieve relevant papers in the abstract from Web of Science and IEEE. However, due to the limitation of maximum eight logical operators in

TABLE I
THE REFERENCE NUMBER IN EACH SECTION

Sections			References	Number of references
I			[14, 15, 19, 20, 21, 22, 23, 24, 25]	9
III	III-A	III-A1	[26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]	23
		III-A2	[49, 50, 51, 52, 26, 53, 54]	7
		III-A3	[55, 56, 49, 57, 58]	5
		III-A4	[48, 59, 60, 61]	4
		III-A5	[62, 35, 49]	3
		III-A6	[63, 35, 60, 64, 65, 62, 66, 67, 26, 68, 38]	11
	III-B	III-B1	[69, 70, 71, 44, 72, 73, 74, 75, 76]	9
		III-B2	[77, 78, 79, 72, 71, 80, 81, 82, 83, 84, 85, 86]	12
		III-B3	[77, 87, 72, 88, 89, 90, 91, 81, 51, 92, 93, 94, 95, 96, 97, 98, 35, 71, 99, 100, 101, 102, 103, 86, 104, 105]	26
	III-C	III-C1	[106, 107, 108, 109, 110, 111, 112, 68, 46, 90, 113, 35, 114, 65, 24, 115]	16
		III-C2	[116, 117, 118, 119, 39, 120, 50, 121, 122, 123, 124]	11
	III-D		[95, 96, 125, 126]	4
		III-D1	[32, 127, 75, 76]	4
		III-D2	[35, 73, 128, 86, 71]	5
IV			[51, 129, 76, 86, 72, 130, 131]	7
	IV-A		[51, 132, 87, 133, 134, 135, 136, 137, 138, 55, 52, 139]	12
	IV-B		[140, 141, 142, 143, 144, 129, 145, 146, 147, 148]	10
	IV-C		[149, 76, 75, 74, 150, 151]	6
	IV-D		[86, 73, 128]	3
	IV-E		[72, 96]	2
	IV-F		[130]	1
	IV-G		[131]	1

Scopus, the aforementioned search expression was divided into two parts for separate retrieval in the title, abstract, and keywords. After removing duplicate papers, a total of 862 papers were obtained. The literature was then selected and supplemented according to the following steps in order to adhere to the requirements of the review.

- First, 533 papers that focused on RSMRs but did not use physical RSMRs were excluded. These papers typically relied solely on mathematical models, numerical simulations, or software to simulate the motion of RSMRs, which lies outside the scope of this paper.
- Second, as explained in Section I, the RSMRs employed in the studies should be reduced in size, car-like, and possess AD functions. Otherwise, the studies would provide limited reference value for AD research and fail to adequately demonstrate the advantages of RSMRs. Consequently, 162 papers that employed full-size vehicles for AD experiments, concentrated on unmanned aerial vehicles or humanoid robots for automated transportation, or developed physical models of RSMRs lacking self-driving capabilities were excluded.
- Third, the central research topic must revolve around the motion and control of RSMRs for roadway AD research. According to this requirement, 36 papers that primarily focused on aspects such as construction, assembly, hardware development, sensor calibration of RSMRs, or explored robot development and operation in hazardous environments like nuclear radiation, or merely treated RSMRs as research samples (e.g., using computer vision technology to detect RSMRs) were excluded.
- Fourth, the selected studies must tackle and solve at least one specific problem, such as speed planning, navigation, or parking control. Consequently, 22 papers that only provided an overview of the structure or components of RSMRs' AD systems were excluded from this paper.
- The first four steps resulted in the selection of 109 papers. From these papers, forward and backward snowballing techniques were employed to recursively search for applicable references and citations. Note that the papers obtained in each round of snowballing must undergo the first four steps before proceeding to the next round. Consequently, 25 additional papers were obtained through this process.

Finally, 134 relevant papers were identified for review. The process of paper search and selection is illustrated in Figure 2. Table I shows the reference number in each section. Note that the same references may be cited in multiple sections. In addition to these 134 references that used RSMRs for AD research, this paper includes an additional 24 references used to introduce the research context or support arguments. Furthermore, to help readers understand the notations used in this paper, we have provided a notation table at the beginning of this paper.

III. REVIEW OF METHODS

As described in Section I, the AD task of RSMRs is to perceive information from the environment using sensors, process and analyze this information through models and algorithms to make decisions and give commands, and subsequently execute the commands through actuators. Perception, decision-making, and actuation are the three modules of AD and also the main focus of research in this field. However, due to the fundamentally different principles of various decision-making methods, we divide them into two sections for discussion: model-based decision-making and learning-based decision-making.

A. Perception

RSMRs rely on sensors to observe the surrounding environment and monitor their own states. This observation is

TABLE II
OVERVIEW OF IMPORTANT SENSORS OF RSMR

	Sensors	Functions	Data	Advantages	Disadvantages
Exteroceptive sensors	Camera	Capturing color images and depth images of the surrounding environment	Unstructured	Camera is the core sensor that provides abundant surrounding information.	Camera collects unstructured data, which require a large amount of computation. Its effectiveness can be easily undermined by extreme weather and bumpy road.
	LiDAR	Measuring the distances between RSMR and surrounding objects	Structured	LiDAR can accurately obtain the distance to objects and generate high-resolution point cloud maps.	LiDAR is costly and has a small perception range, and it can be easily affected by adverse weather conditions.
Proprioceptive sensors	GPS	Providing positional information of RSMR, including longitude, latitude, and elevation	Structured	GPS returns the absolute position of RSMR that is necessary for localization and navigation.	GPS experiences signal degradation indoors or in areas with significant shading.
	IMU	Measuring tri-axial acceleration and tri-axial angular velocity of RSMR	Structured	IMU returns RSMR's motion attitude for kinematic modeling. It has a compact size and high accuracy.	IMU can introduce errors that accumulate over time, and is susceptible to external disturbances such as vibration, temperature changes, and electromagnetic interference.
	Odometry	Estimating the displacement of RSMR	Structured	Odometry is characterized by a simple measurement principle, low cost, and suitability for a wide range of environments.	Measurements based on wheel rotation are susceptible to errors caused by wheel slippage, variations in ground friction, and inconsistent wheel diameters.

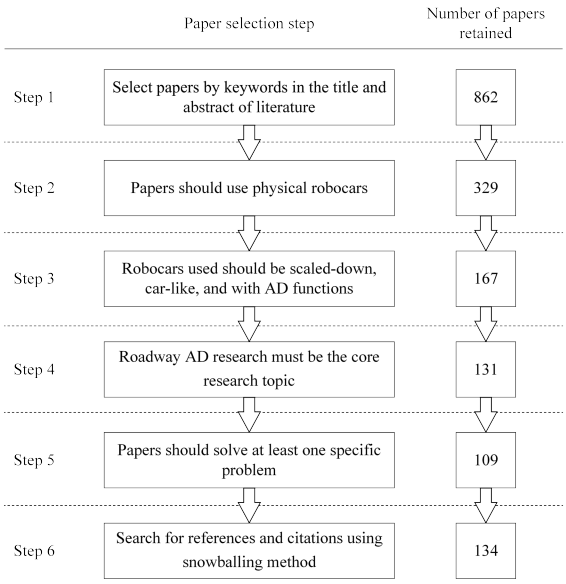

Fig. 2. Literature search and selection.

transformed into processable and measurable quantities necessary for analysis and decision-making. The sensors embedded in RSMRs can be categorized into two types: exteroceptive sensors and proprioceptive sensors [152]. Exteroceptive sensors, such as Radar, LiDAR, cameras, and hearing sensors, are employed to perceive and gather information regarding the external environment. This information includes distances to obstacles, images of lanes and traffic signs, and human's voice commands. In comparison, proprioceptive sensors are used to monitor and retrieve information of the RSMRs themselves, including their position, motion states, and battery status. Common examples of proprioceptive sensors include the Global Positioning System (GPS), odometry, and Inertial Measurement Units (IMUs).

Among the sensors embedded in RSMRs, cameras provide images and videos containing a large amount of information necessary for most AD tasks. Images and videos are unstructured data, so the processing methods are relatively complex, requiring the extraction and processing of information from the data. In contrast, the data from other sensors are structured and thus can generally be used directly. The following subsections will introduce some of the most commonly used sensors used in RSMRs, and the functions, advantages, and disadvantages of these sensors are summarized in Table II.

1) Camera: For AD tasks of RSMRs, two types of cameras are typically used: RGB cameras and RGB-D cameras. RGB cameras are able to capture color images and videos, allowing for the recording of visual appearances and color information of objects. These images and videos can be used for various purposes such as object detection and classification, color and object tracking, as well as vehicle's and pedestrian's movement prediction [26], [36], [39]. For example, Rosas-Cervantes et al. [26] used a monocular camera for extracting two-dimensional (2D) features in trajectory estimation on multilevel surfaces. The camera captures color images, enabling the detection of main corners and features in the environment, which enhance object detection, feature recognition, and robot localization. Everett et al. [39] used a camera in conjunction with a 2D LiDAR sensor to estimate pedestrian positions and velocities through clustering and image classification techniques, enabling the RSMR to navigate safely among pedestrians in real-time scenarios.

In comparison, RGB-D cameras can capture depth images of the surroundings and record the distance to objects in pixels, which enables the generation of a three-dimensional environment model, reconstruction of object appearance, and estimation of object posture [27], [30], [31]. For example,

Molska and Belter [27] used a RGB-D camera to identify and measure obstacles, including walls and narrow obstacles, in the surrounding environment. By capturing depth images, robots perceive the environment more accurately and move towards the target location while avoiding obstacles. However, the higher cost of RGB-D cameras has led some studies to conduct AD experiments using only RGB cameras [28], [36].

The method for processing images is mainly categorized into two methods: traditional methods and Deep Learning (DL) algorithms.

Traditional methods rely on manual feature extraction and pattern matching, typically involving multiple steps of extraction, transformation, detection, and classification. Commonly used methods include color transformation, Sobel Operator, Hough Transform, Scale Invariant Feature Transform, or Histogram of Oriented Gradient. Traditional methods are widely used for the problems of lane and traffic sign detection [29], [37], [40], [42], [44], [46], vehicle tracking [30], and collision warning [31].

In contrast to the manual extraction of image features in traditional methods, DL algorithms can automatically extract image features. The most common DL algorithm is Convolutional Neural Network (CNN), which can be divided into two-dimensional and three-dimensional CNN. Two-dimensional CNN is used for processing individual color, grayscale, or binary images [27], [32], [36], [43], [45], [47], [48], while three-dimensional CNN introduces a temporal dimension and is used for processing videos to capture the dynamic changes of images and the correlations between different frames [33], [38], [41]. In addition, there are other DL Algorithms used to replace or combine with CNN. For example, Generative Adversarial Networks are used for image generation, image restoration, and data augmentation [33]. Furthermore, You Only Look Once (YOLO) is used to identify the positions and categories of different objects such as vehicles, pedestrians, and traffic signs in an image or a video [34].

Despite being the core sensor of RSMRs, cameras have certain limitations. Extreme weather such as snowy, foggy, or rainy conditions, can significantly impact the accuracy of the camera. Additionally, when driving on bumpy roads, the camera may experience vibrations and movements that negatively affect the quality of captured images and videos. In such cases, cameras can be used in tandem with IMUs to compensate for errors in image acquisition and processing caused by road bumps [35].

2) LiDAR: LiDAR measures the distances between the RSMR and surrounding objects in a 360-degree range by emitting and receiving laser pulses. Compared to radar, LiDAR is costlier and has a smaller perception range, as it can be easily affected by adverse weather conditions such as fog or snow. However, LiDAR offers significantly higher accuracy than radar and is capable of generating high-resolution point cloud maps, which can be used for various applications such as environment perception [49], [50], distance estimation [51], [52], obstacle detection and identification [26], localization and mapping [53], [54]. In [49], LiDAR functions as a critical component within the perception block of a control architecture designed for multi-vehicle navigation in formation. It was used to capture environmental data to identify obstacles in real-time, enabling dynamic adjustments to the fleet formation. Since RSMRs are reduced-scale, the requirements for perception range is relatively low, while accuracy remains crucial. Consequently, LiDAR is the preferred choice for equipping RSMRs.

3) Global Positioning System: GPS provides precise positional information, such as longitude, latitude, and elevation, which plays a crucial role in navigation and localization for RSMRs [49], [55], [56]. Farooq et al. [55] used GPS to acquire the RSMR's latitude and longitude coordinates. These data were fed to the goal reaching controller, one of the two main neural network controllers proposed. Then the positional information was processed to generate steering angle commands, enabling the vehicle to navigate towards the goal effectively.

GPS also allows for accurate timestamp acquisition from satellite signals. Leveraging this time information, multiple RSMRs can achieve time synchronization, facilitating data fusion and enabling cooperation and interaction among the vehicles. Additionally, continuous collection of GPS position data enables real-time map creation and updates, aiding in environment modeling and path planning [58], [153].

Note that GPS can experience signal degradation indoors or in areas with significant shading, which may lead to imprecise localization or delays in information [57]. To mitigate these issues, additional sensors such as LiDAR and IMU are usually used alongside GPS. This sensor fusion approach enhances the accuracy and reliability of positioning.

4) Inertial Measurement Units: IMUs of a RSMR consist of several integrated inertial sensors, primarily accelerometers and gyroscopes. Accelerometers apply the law of conservation of inertial forces to measure the tri-axial acceleration of the RSMR, while gyroscopes use the law of conservation of angular momentum to measure the tri-axial angular velocity. These data on tri-axial acceleration and angular velocity are crucial for monitoring the RSMR's motion states and estimating its posture [48], [59]. Liu et al. [48] used IMUs to collect essential data regarding the RSMR's current motion states, including acceleration and orientation, which enables precise real-time pose estimation, and thus ensures the RSMR can navigate complex indoor environments reliably and efficiently.

While the IMU offers advantages like a compact size and high accuracy, it has some limitations. The calculation of the RSMR's tri-axial angles through the integration of angular velocity can introduce errors that accumulate over time [60]. Additionally, the IMU is susceptible to external disturbances such as vibration, temperature changes, and electromagnetic interference [61]. Furthermore, the IMU has a restricted measurement range, meaning accelerations or angular velocities exceeding this range may result in measurement distortions or saturation.

5) Odometry: Odometry estimates the displacement of the RSMR by measuring the rotation of its wheels. It is characterized by a relatively simple measurement principle, low cost, and suitability for a wide range of environments. However, measurements based on wheel rotation are susceptible to errors caused by wheel slippage, variations in ground friction, and

inconsistent wheel diameters. As a result, these factors can cause the accumulation of errors over time [154]. RSMRs are typically small in size, so roads that may appear smooth for full-size cars can be rough and bumpy for RSMRs. Consequently, in situations where roads are in poor conditions, odometry is frequently combined with other sensors such as IMU, LiDAR, and GPS to enhance the accuracy and robustness of measurement [35], [49], [62], [155]. Tran et al. [62] used odometry in conjunction with a 2D Simultaneous Localization and Mapping framework to estimate the robot's position for both two and three-dimensional mapping tasks. Specifically, the odometry information contributes to the localization process that was underpinned by a Rao-Blackwellized particle filter with improved proposal distribution. This enables the system to accurately predict the RSMR's pose by selectively sampling within significant areas indicated by the scan-matcher.

6) Sensor Fusion: The capability of individual sensors is limited, so sensor fusion is necessary. Sensor fusion integrates measurements from multiple sensors in order to enhance the performance and efficiency of monitoring the state of a RSMR and perceiving its environment. The data collected from each sensor can either be redundant or independent of each other. Sensor fusion patterns can be divided into three types: data level fusion, feature level fusion, and decision level fusion.

Among these sensor fusion types, feature level fusion is most commonly used [35], [60], [64]. This approach is typically based on various deep neural networks used to combine the data from cameras and other sensors. For example, to generate feasible trajectories, Ayalew et al. [38] used an Attentive Fully Connected Neural Networks to encode the historical trajectories, an attentive three-dimensional CNN to extract spatial-temporal information from videos, and an LSTM to fuse both attentive intermediate features. To estimate trajectory on a multilevel surface, Rosas-Cervantes et al. [26] used a faster region CNN and normal distribution to extract the features of images collected by a RGB camera and three-dimensional point clouds collected by a LiDAR, and then used a clustering algorithm to fuse these features.

In addition to deep neural networks, other sensor fusion algorithms employed in AD research involving RSMRs include extended Kalman filter [63], particle filter [67], Gaussian Mixture Model [66], and Backpropagation Neural Network (BPNN) [68]. These algorithms are mainly used to fuse structured data collected by other sensors except for the camera.

Tran et al. [62] compared the three sensor fusion approaches, namely, data level, feature level, and decision level fusion. They created a map of the RSMR's surrounding environment by combining the data from a two-dimensional LiDAR and a three-dimensional ultrasonic sensor.

Sensor fusion allows different sensors to calibrate and complement each other, thereby acquiring more accurate and reliable information and facilitating collaboration for AD tasks [156]. In addition, sensor fusion can leverage the redundancy among different sensors. If one sensor fails or generates inaccurate data, the information from other sensors can compensate for this deficiency, thereby improving the robustness of the system. Although some studies emphasise that they have used only one sensor to achieve the task (usually an RGB camera) and do not rely on other sensors to indicate that their approach is advanced or low-cost [65], the gains in driving performance from sensor fusion cannot be ignored.

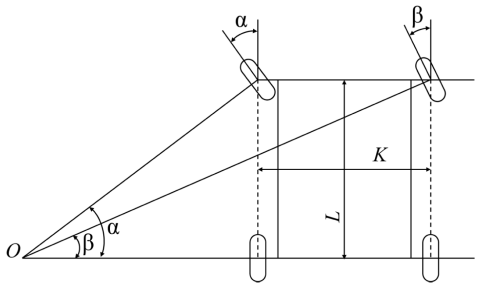

Fig. 3. Ackermann steering angle [44].

B. Model-Based Decision-Making

Model-based decision making relies on predefined physical motion models, optimization methods, and path planning techniques to predict and decide RSMR's maneuvers. These methods usually use mathematical representations of the environment and vehicle dynamics that possess explicit physical interpretations to compute optimal paths and decisions based on specific criteria such as safety and efficiency. Their principle, advantages, and disadvantages are shown in Table III.

1) Physical Motion Model: Some studies have used physical models and motion equations based on Newton's laws of motion to plan the movement of vehicles. RSMRs with different chassis correspond to different motion patterns and motion models. Common chassis types include the differential drive, four-wheeled omni-directional, four-wheeled Ackerman, and four-wheeled Mecanum wheel robots. In AD studies, the four-wheeled Ackermann robot is the most frequently used chassis, as it is the most commonly used chassis in real-world vehicles [69], [70], [71]. When an Ackermann RSMR steers in a circular motion, the steering angles of the left and right front wheels differ. The steering angle of the inner wheel is greater than that of the outer wheel, and this difference is referred to as the Ackermann steering angle, as illustrated in (1) and Figure 3.

$$\cot\beta - \cot\alpha = \frac{K}{L} \tag{1}$$

where α and β are the steering angles of the inner and outer wheels, respectively. K is the distance between the two kingpins of the front axle, and L is the wheelbase.

In order to minimize the computational load, most studies have simplified the motion of Ackermann RSMRs by employing a bicycle model. Suppose that the RSMR's state is denoted by $\mathrm{X} = \left[x, y, \psi, \dot{x}, \dot{y}, \dot{\psi}\right]^T$, and the input signal is represented by $\mathrm{U} = [\delta, \omega]^T$. Here, x and y denote the positional coordinates of the RSMR in a two-dimensional plane, ψ represents the heading angle, and $\dot{x}$, $\dot{y}$, and $\dot{\psi}$ refer to the first-order derivatives of x, y, and ψ with respect to time. Furthermore, δ corresponds to the steering angle, while ω represents the wheel rotation speed. The force condition

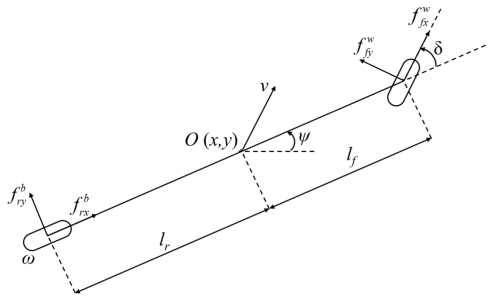

Fig. 4. Bicycle model [72].

and motion of the RSMR are depicted in Figure 4 and are described by eq. (2)-(4).

$$m\ddot{x} = f_{\mathrm{fx}}^{\mathrm{w}}\cos(\psi+\delta) - f_{\mathrm{fy}}^{\mathrm{w}}\sin(\psi+\delta) + f_{\mathrm{rx}}^{\mathrm{b}}\cos\psi - f_{\mathrm{ry}}^{\mathrm{b}}\sin\psi \tag{2}$$

$$m\ddot{y} = f_{\mathrm{fx}}^{\mathrm{w}}\sin(\psi+\delta) - f_{\mathrm{fy}}^{\mathrm{w}}\cos(\psi+\delta) + f_{\mathrm{rx}}^{\mathrm{b}}\sin\psi - f_{\mathrm{ry}}^{\mathrm{b}}\cos\psi \tag{3}$$

$$I_{\mathrm{z}}\ddot{\psi} = (f_{\mathrm{fy}}^{\mathrm{w}}\cos\delta + f_{\mathrm{fx}}^{\mathrm{w}}\sin\delta)l_{\mathrm{f}} - f_{\mathrm{ry}}^{\mathrm{b}}l_{\mathrm{r}} \tag{4}$$

where m is the RSMR's mass, I_{z} is the RSMR's rotational inertia along the z-axis, $\ddot{x}$ and $\ddot{y}$ are the second order derivatives of x and y with respect to time, respectively, l_{f} is the distance between the RSMR's front axle and the center of mass, and l_{r} is the distance between the RSMR's rear axle and the center of mass. $f_{\mathrm{fx}}^{\mathrm{w}} = \mu_{\mathrm{fx}} f_{\mathrm{fz}}$, $f_{\mathrm{fy}}^{\mathrm{w}} = \mu_{\mathrm{fy}} f_{\mathrm{fz}}$, $f_{\mathrm{rx}}^{\mathrm{b}} = \mu_{\mathrm{rx}} f_{\mathrm{rz}}$, $f_{\mathrm{ry}}^{\mathrm{b}} = \mu_{\mathrm{ry}} f_{\mathrm{rz}}$ are the frictional forces. Subscripts f and r denote the front and rear, respectively, and w and b denote the wheel frame and the body frame, respectively. μ_{fx}, μ_{fy}, μ_{rx}, μ_{ry} are the friction coefficients, and f_{fz} and f_{rz} are support forces.

Ackermann RSMRs typically do not exhibit lateral movement, unless during drift states. Therefore, the bicycle model is further simplified as shown in eq. (5)-(7) [73].

$$\dot{x} = v\cos\psi \tag{5}$$

$$\dot{y} = v\sin\psi \tag{6}$$

$$\dot{\psi} = \frac{v}{l_{\mathrm{f}}} + l_{\mathrm{r}}\tan\delta \tag{7}$$

where v is the RSMR's velocity.

Using vehicle motion models offers a straightforward approach with a deterministic analytic solution. However, accurately modelling the decision-making mechanisms of RSMRs in complex environments can be challenging when relying solely on these vehicle motion models. Nonetheless, these models hold significance in AD research as they serve as the foundation for relatively complex or hybrid algorithms like the Dynamic Window Approach (DWA), Model Predictive Control (MPC), as well as some research under simple assumptions [74], [75], [76].

2) Optimization Methods: Optimization methods are a type of mathematical techniques aimed at adjusting the values of variables to achieve the optimal value of a specific objective function while satisfying a set of constraints. The core idea of optimization methods is to search for the optimal solution within the feasible solution space, which can be conducted through mathematical analysis, computer simulation, or heuristic approaches.

Optimization methods can perform global search within the feasible solution space, which makes them highly effective in dealing with complex multidimensional problems and reducing the risk of obtaining local optima. Also, they can incorporate multiple objectives in the objective function, finding solutions that balance the interests of various parties. Furthermore, optimization methods typically have strong robustness, enabling them to find desirable solutions even when the input data contains noise or uncertainty. In addition, the objective function and constraints of optimization methods can have direct links to measurable parameters and variables.

However, optimization methods come with certain drawbacks. Non-convex objective functions can lead to significant computational burdens and no assurance of achieving the optimal solution. In real-world AVs, traffic situations are extremely complicated, and factors such as perception errors, sensor measurement noise, and dynamic obstacles in real environments can impact the performance of optimization methods. Therefore, in such cases, optimization methods are typically combined with other techniques such as ML to enhance the decision-making capabilities of the self-driving system.

In contrast, experiments conducted with RSMRs primarily focus on model and algorithm validation in reduced-scale and simplified environments. These experiments serve as a foundation and provide guidance for real-world AD research. Typically, the research involving RSMRs tackles one specific problem at a time, which simplifies the construction of optimization models. Additionally, the scaled-down environments offer high controllability at a low cost, enabling adjustments to the environmental setup and repeated evaluations of optimization methods. As a result, optimization methods are extensively employed and are used independently in AD research with RSMRs.

For AD research with RSMR, the objective function often aims to minimize path lengths, path tracking error, speed tracking error, absolute value of acceleration, change in heading angle, travel time, energy consumption, or maximize the distance to an obstacle, among other factors. Constraints encompass avoiding obstacles, satisfying vehicle motion characteristics, and adhering to energy consumption limitations.

The most common optimization methods are linear programming [77], [78] and quadratic programming [71], [72], [79]. By altering various objective functions and constraints, these programming models can address numerous problems related to RSMR path planning and motion strategies. Walambe et al. [85] used optimization method with spline interpolation for trajectory generation. The objective function focuses on minimizing the trajectory path length and ensuring smoothness and continuity, while avoiding the typical singularities encountered in cubic polynomial-based methods. Constraints include the non-integrable conditions posed by the RSMR's differential flatness properties and physical design limitations. Shen et al. [86] realized optimization setup through an MPC framework. The objective is to navigate safely and efficiently through tightly-constrained dynamic

environments typically populated with multiple AVs and human-driven vehicles. Constraints are generated in real-time and are strategy-dependent, focusing on feasible regions within the RSMR's navigation space. Yang et al. [72] used optimization method for RSMR drift maneuver control. The framework proposed integrates feedforward and feedback controllers that adjust the curvature of RSMR's trajectory in response to changes in tire-ground interaction. The objective function aims to stabilize the drift around predefined paths (like circular trajectories) while maintaining the desired drift angle. Constraints revolve around the RSMR dynamics and the interaction between tire and ground that vary significantly under different environmental conditions.

For solving complex nonlinear optimization problems within a limited time, heuristic algorithms are frequently employed to find a feasible solution instead of the optimal solution. Common heuristic algorithms include Tabu Search [80], Simulated Annealing [81], Genetic Algorithm [82], Particle Swarm Optimization [83], and Ant Colony Optimization (ACO) [84]. Chen et al. [84] employed the ACO algorithm to address the optimization problem of identifying the most efficient path for real-time obstacle avoidance. The ACO algorithm optimizes the path by iteratively recalculating the route that the RSMR should follow to reach its designated sub-goal, while navigating around obstacles identified through LiDAR scans.

3) Path Planning Methods: Path planning is the foundation of RSMRs' autonomous driving, and there are many algorithms specifically designed to solve path planning problems.

Some studies chose to use analytical curves generating the paths for RSMRs. The most common is the polynomial curve, because they are continuous and differentiable, and can flexibly fit different shapes of paths by adjusting the order and coefficients [77], [87]. Other curves include circular arc [72], Bezier curves [88], B-splines [89], and Dubins curves [90]. For example, Elbanhawi et al. [89] considered path continuity while incorporating constraints related to the RSMR's maximum steering and path curvature. They used Non-Uniform Rational B-Splines as the tracking path for the RSMR, comparing and analyzing different curve parameters for practical applications. In addition to curve fitting methods, some other classic path generation algorithms include A* [51], [81], [91], [92], Dijkstra [93], and Artificial Potential Field [87].

Upon generating the paths, the motion commands that adhere to the constraints of the RSMR's kinematic characteristics should be determined in order to track the path effectively. Common algorithms for this purpose include Pure Pursuit Control [94], Linear Quadratic Regulator [95], Feedback Linearization [96], [97], and Fuzzy Inference System [98]. Several studies have combined MPC with data-driven algorithms to enhance the performance of path tracking. Drews et al. [35] conducted aggressive driving tests using a 1:5 scale RSMR. They employed model predictive path integral control to sample and predict the RSMR's trajectories in the upcoming time steps, generating a cost map of the RSMR's surrounding environment based on GPS localization. Next, they used CNN to calculate the associated cost values, and selected the optimal trajectory based on these costs. Zhou et al. [71] employed a neural network to fit the errors of the nominal vehicle dynamic model to enhance its accuracy. Subsequently, they employed MPC to maintain the desired equilibrium states of drift and enable the RSMR to follow the intended path.

The above methods can be referred to as offline path planning, where the desired path is predetermined and generated based on a map or predefined waypoints prior to path tracking. Offline path planning allows for less real-time computations, but it tends to compromise algorithms' flexibility and RSMRs' adaptability to the environment. Online path planning can effectively solve this problem. It generates paths for RSMRs in real-time with various constraints such as kinematic characteristics and obstacle avoidance.

Common algorithms used for online path planning include DWA [99], [100], Rapidly-exploring Random Trees (RRT) [101], [102], [103], and Fuzzy Inference System [86]. For example, Lee et al. [104] improved the DWA algorithm for obstacle avoidance in RSMRs, introducing the finite distribution estimation-based DWA, which estimates the distribution of obstacles in dynamic environments using a covariance matrix adaptation evolution strategy and a finite memory filter. Hess et al. [105] employed the RRT algorithm for generating and planning the RSMR's path. To prevent circular motion, the maximum heading change in one expansion step was limited. Ghaffari and Homaeinezhad [90] proposed Fuzzy Adaptive Curvature-based Point Selection (FACPS) algorithm for autonomous path planning. The inputs of FACPS consist of the angle between the robot's heading and the line connecting the tracking point to the RSMR's center of gravity, as well as the angle between the tangent line at the tracking point and the RSMR's heading.

Path planning method has a clear physical interpretation and can ensure safety by avoiding collisions. However, it cannot guarantee the acquisition of the optimal path, and sometimes depends on precise environmental modeling.

C. Learning-Based Decision-Making

Learning-based decision making uses data-driven techniques to derive the optimal strategies of RSMRs for each time step. This approach typically includes methods such as Imitation Learning (IL) and Reinforcement Learning (RL). Their principle, advantages, and disadvantages are shown in Table III.

1) Imitation Learning (IL): IL enables RSMRs to perform AD tasks by imitating human or expert behavior. This method is based on supervised learning and typically used to achieve RSMRs' end-to-end decision-making that eliminates the need for step-by-step design, tuning, and assessment.

In the IL, expert behavior can be collected through various approaches. Many researchers choose to manually control the RSMR while collecting sensor data and car motion data as expert behavior. However, human-controlled car movements tend to be uniform, stable, and rarely result in collisions. This can cause the model to struggle with handling unseen events such as lane deviation or encountering dead ends. To address this issue, Seiya et al. [106] have used a Pure Pursuit Algorithm to train the RSMR to return to

the desired path from the current image viewpoint. Hoshino and Sumiyoshi [107] adopted the method of learning action sequences to prevent the RSMR from stopping or getting stuck in dead end. In addition, in order to save the cost of manual control, some studies have employed rule-based algorithms to control RSMRs [108] or conducted AD simulation experiment [109] to collect expert behavior.

In the process of training, IL algorithms learn the mapping function from environments to expected actions by observing expert driving behavior. In the process of testing, the sensor data are directly inputted into the trained algorithms, and the algorithms then output the control commands and send them to the actuators.

Assume that the training data are represented as $\mathcal{D} = \{\langle \mathbf{o}_t, \mathbf{a}_t \rangle\}_{t=1}^{N}$, where $\mathbf{o}_t$ is the observed environmental information at time t, and $\mathbf{a}_t$ is the actions performed by the expert at time t. The task of IL is to use an approximator F with parameter θ to fit $\mathbf{a}_t$ based on $\mathbf{o}_t$, as shown in (8) [110].

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \sum_{t} \ell(F(\mathbf{o}_t; \boldsymbol{\theta}), \mathbf{a}_t). \tag{8}$$

where ℓ is the loss function, F is typically a supervised ML algorithm that has strong learning capability required for end-to-end AD systems.

Traditional ML methods can only process structured data such as the RSMRs' motion states and environmental information at each time step gathered from sensors like GPS, IMU, odometry, and LiDAR. Common traditional ML algorithms include Support Vector Machine [157], Random Forest [111], Bayesian Network [112], BPNN [68], and Fuzzy Inference System [46], [90], [113].

In comparison, DL is a powerful and flexible ML method that can handle unstructured data such as images and videos captured by cameras or time-series data segments captured by other sensors.

CNN is the most commonly used DL algorithm for directly processing images and videos by automatically extracting features [35], [114]. For example, Kim et al. [65] employed a CNN called AlexNet to achieve a RSMR's indoor navigation and obstacle avoidance in an end-to-end manner. In order to complete the lane following task, Kang et al. [24] integrated two-dimensional CNN and one-dimensional CNN in a single network, respectively used for processing RGB images and LiDAR data. Consequently, the data were fused through fully connected layers and the throttle and steering were output to control the RSMR to follow the designated lane.

Some studies have realized that using the data of each moment independently as input makes it difficult to capture the temporal characteristics of expert behavior. As a result, RNN is used to address this issue. Lai and Bräunl [108] first used CNN to extract image features for each frame, and then used Long Short Term Memory (LSTM) network, a variant of RNN, to further extract features from the sequence of image features. To enable the RSMR to restart after stopping and to escape from the dead end, Hoshino and Sumiyoshi [107] employed deep RNN with the input of the distance data from the LiDAR for the last twenty time steps.

The above studies assume that the expected actions of RSMRs are entirely determined by the observed information, as shown in (8). However, Codevilla et al. [110] indicated that this assumption is inconsistent with reality. For example, the decision to turn or go straight at an intersection cannot be made solely based on sensor information, but requires direction instructions based on the desired route. Therefore, it is necessary to include unobservable information as inputs, as shown in (9).

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \sum_{t} \ell\big(F(\mathbf{o}_t; \boldsymbol{\theta}), E(\mathbf{o}_t, \mathbf{h}_t)\big) \tag{9}$$

where $\mathbf{h}$ is the unobservable information, and E is the function used to explain the unobservable information.

In addition to [110], some studies have also noticed this point and added the directional guidance information to the input of the approximator F [106], [107].

IL requires a substantial amount of labeled data, which in turn requires a significant amount of manual labor. To expand datasets while reducing costs, some studies have used data augmentation to artificially generate data [106], [115] or data diversification to enhance the diversity of the data [110]. These techniques are particularly common in the processing of image data. However, Tampuu et al. [158] pointed out that data augmentation may lead to model overfitting, and adding noise to the data can be excessively risky when the algorithms are eventually deployed in real-world scenarios. As is well known, for ML, any data augmentation technique is not as effective as expanding the data scale. Therefore, it is necessary to leverage the advantages of low cost and small size of RSMRs to conduct more experiments in various scenarios to provide abundant training data for IL algorithms.

While IL, especially DL, provides a powerful framework for AD with RSMRs, its training requires a significant amount of time and computational resources. Moreover, its generalizability is relatively weak, as models trained well may be only effective for the current dataset, which limit its potential. This is a problem that needs to be addressed in future research.

2) Reinforcement Learning (RL): RL represents a third view of ML in addition to supervised and unsupervised learning methods. It is a method for an agent to learn the optimal strategy by interacting with environment. For AD research, RSMRs select the predefined alternative actions based on the perceived environmental states, receive rewards as feedback at each moment, and accordingly adjust their motion strategy to maximize the cumulative rewards. Through continuous exploration and trial and error, the RSMR gradually learns the strategy of taking optimal actions in different states, and thus achieves autonomous intelligent decision-making. The period from the start of the RSMR executing an AD task to the completion of a specific objective or a failure is defined as an episode. The RSMR's actions can be discrete such as moving forward or turning left, or continuous such as changing linear and angular velocity. The reward is usually related to the distance to the target or obstacles, the magnitude and variation of velocity, cumulative traveled distance, and tracking error, among other factors.

Q-learning is an effective RL algorithm. It stores the quality values (Q-values) for different states and actions in a table and iteratively calculates the Q-values to convergence using

the Bellman equation, as shown in (10).

$$\hat{Q}(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma \max_{a'} Q'(s',a') - Q(s,a)] \tag{10}$$

where $Q(s,a)$ and $\hat{Q}(s,a)$ are the Q-values before and after iteration, respectively. α is the learning rate. $R(s,a)$ is the reward for taking action a in state s, γ is the discount factor, and $\max_{a'} Q'(s',a')$ is the maximum Q-value that can be obtained in new state s'.

Nevertheless, the observed states are often in the form of images or videos, making it unrealistic to represent such high-dimensional state spaces with tables. Therefore, few studies utilize Q-learning to accomplish AD tasks [116], while Deep Reinforcement Learning (DRL) is more commonly used.

Deep Q-learning is the most common DRL algorithm that uses a deep neural network with states as input to estimate the Q-values for each action. Zhang et al. [117] used a Double Deep Q-learning algorithm to accomplish the path tracking task for the RSMR. They used a deep fully connected network, with lateral position error and angle error as inputs. They selected eight hundred discrete steering curvatures as actions and used the negative value of the accumulated error over time as the reward. Lu [118] employed a CNN to extract the features of the street and car images. The actions are the RSMR's steering angle, and the reward is based on the RSMR's current velocity and the distance to the nearest obstacle or road edge. Kendall et al. [119] used a variational autoencoder to process the images collected from a monocular camera, selected the distance travelled as the reward, and used the steering angle and speed as the RSMR's actions.

Q-learning and Deep Q-learning belong to value-based algorithms that focus on efficiently estimating Q-values. Policy-based algorithms provide another approach by directly using actions as the output of neural networks. The most common policy-based algorithm is Policy Gradient (PG) algorithm. Assuming that $\pi_\theta(a|s,\theta)$ with the parameter θ is RSMR's motion policy, and each episode generates a sequence $\tau = \{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$, where s, a, r are the state, action, and reward, respectively. The goal of PG is to maximize the cumulative reward and update the parameters of the policy network. Equation (11) demonstrates the gradient of the cumulative reward of a batch of episodes with respect to the parameters θ. Policy-based algorithms can directly output continuous action values, thus are widely applied in AD tasks of RSMRs [39], [118], [119], [120].

$$\nabla \bar{R}(\tau) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} (\sum_{t'=t}^{T_n} r_{t'}^n \gamma^{t'-t} - b) \nabla \log \pi_\theta(a_t^n | s_t^n) \tag{11}$$

where ∇ is the gradient operator, $\nabla \bar{R}(\tau)$ is the gradient of the cumulative rewards with respect to the parameters θ, N is the batch size, T_n is the number of time steps of episode n, and b is the expected reward.

Value-based algorithms can introduce high bias when estimating Q-values, while policy-based algorithms require a large amount of sampling, resulting in high variance. Actor-Critic is a combined approach that aims to address the limitations of both. In this approach, the actor network is responsible for generating policies, while the critic network approximates the Q-values. For example, in the case of sensor occlusion, Ryu et al. [50] used the actor-critic algorithm to complete the avoidance of static and dynamic obstacles. The input to the actor network includes point cloud, RSMR state, goal state, and confidence vector that represents the reliability of sensor data, and the input to the critic network is the RSMR's actions.

In DRL algorithms, some studies used RNN and its variants such as Gate Recurrent Unit and LSTM to capture the temporal characteristics of state and action sequences [50], [121]. This is encouraged because in similar environments, RSMR may not necessarily take the same actions. The current desired actions may depend on previous states and actions. Additionally, RNN can be used to predict the motion trajectory of dynamic obstacles that greatly influences the subsequent actions and potential rewards for RSMR.

RL is suitable for AD task because it does not require a large amount of manually labeled data. It only needs to define the reward function in advance, and then the RSMR will continuously explore the environment, collect data, and train the models on its own. However, its training efficiency is proved to be lower than that of IL. Therefore, some studies pre-trained the models of RL in a simulated environment and then transfer the models to RSMRs and fine-tune the parameters of models [120], [122], [123], [124].

D. Actuation

RSMRs are controlled through actuators, mainly comprising electric drive motors and steering servos. The electric drive motor is used to convert electrical energy into mechanical energy to propel the vehicle forward or backward. The speed and torque of the electric drive motor are controlled and adjusted to achieve the desired acceleration and deceleration of the RSMR. Meanwhile, the steering servo is responsible for controlling the RSMR's steering angle, thereby altering its direction of travel. Notably, in contrast to real AVs, most AD research with RSMRs employs linear and angular velocity as control commands, rather than relying on the accelerator, brake pedal, or the steering wheel. This command approach simplifies control logic, eliminating the need to understand specific control mechanisms and sensor principles. Moreover, it facilitates the application of AD algorithms across various vehicle types and brands. Additionally, the use of velocity as a command is more direct and precise than relying on pedal forces and the steering wheel angle. Specifically, RSMRs can dynamically adjust sensor settings in real-time to achieve the desired velocity, enhancing adaptability to the surrounding environment.

Nevertheless, when designing AD algorithms, it is sometimes necessary to incorporate control algorithms that ensure smoother and more stable vehicle movements. The common actuation methods include Proportional-Integral-Derivative (PID) controller, MPC, Linear Quadratic Regulator [95], Feedback Linearization [96], Internal Model Control [125], and Sliding Mode Control [126]. Due to space limitations, only the two most commonly used methods, i.e., PID controller and MPC, will be introduced.

TABLE III
OVERVIEW OF THE DECISION MAKING MODELS OF RSMR

Decision-making methods	Principle	Advantage	Disadvantage	Categories of methods	Models and algorithms
Physical motion model	Newton's laws of motion	It offers a straightforward approach with a deterministic analytic solution and serves as the foundation for some complex and hybrid algorithms.	It is excessively simple and can hardly model the decision-making mechanisms of RSMR in complex environment.		differential drive model, four-wheeled omni-directional model, four-wheeled Ackerman model, four-wheeled Mecanum model, bicycle model [74, 75, 76, 69, 70, 71]
Optimization method	Adjusting the values of variables to maximize or minimize the objective function while satisfying a set of constraints	It has clear meanings and strong robustness, and can perform global search within the feasible solution space.	Non-convex objective functions and complex constraints can lead to significant computational burdens and no assurance of the optimal solution.		linear programming [77, 78], quadratic programming [79, 72, 71]
Path planning method	Based on geometry and optimization methods	It has a clear physical interpretation and can ensure safety by avoiding collisions.	It cannot guarantee the optimal path, and sometimes depends on precise environmental modeling.	Offline path planning	Pure Pursuit Control [94], Linear Quadratic Regulator [95], Feedback Linearization [96], [97], Fuzzy Inference System [98]
				Online path planning	Dynamic Window Approach [99, 100], Rapidly-exploring Random Trees (RRT) [101, 102, 103], and Fuzzy Inference System [86].
Imitation learning	Learning the mapping function from environments to expected actions by observing expert driving behavior	It has strong learning ability that enables RSMR to navigate in the desired manner.	Its training requires a large amount of time and computational resources, and its generalization capability is relatively weak.	Traditional Machine Learning	Support Vector Machine [157], Random Forest [111], Bayesian Network [112], Back Propagation Neural Network [68], Fuzzy Inference System [46, 90, 113]
				Deep Learning	Convolutional Neural Network [35, 114, 65, 24], Recurrent Neural Network [108, 107], Generative Adversarial Networks [33], You Only Look Once [34]
Reinforcement Learning	Learning optimal actions through environment interaction, feedback rewards, and trial and error	It does not require a large amount of manually labeled data or expert guidance.	Its training efficiency leaves room for improvement.		Q-learning [116], Deep Q-learning [117, 118, 119, 121, 50], Policy Gradient Algorithm [39, 118, 120, 119], Actor-Critic Algorithm [50]

1) PID Controller: A PID controller is a feedback loop component widely employed in industrial control. Its function involves comparing acquired data with a predefined reference value and computing an appropriate control value. It uses historical data and the difference from the target value to dynamically adjust the control value, to stabilize the system [32], [75], [76], [127]. The general (time-continious) form of a PID controller is represented as (12).

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau) d\tau + K_{\mathrm{d}} \dot{e}(t) \tag{12}$$

where $e(t)$ is the error and $u(t)$ is the control value at time t. K_{p}, K_{i}, and K_{d} are the coefficients for the proportional, integral, and derivative terms, respectively. In the context of AD research with RSMRs, u is typically RSMR's velocity or acceleration, and e represents the difference between the desired values of position, velocity, heading angle and their actual counterparts.

Assuming that u is RSMR's acceleration, and e is the velocity difference, it is not difficult to find that the magnitude of the gap between the current velocity and the desired velocity governs the level of acceleration. A larger gap leads to more aggressive acceleration, while a smaller gap corresponds to a more conservative acceleration strategy. The parameter K_{i} is specifically designed to eliminate steady-state errors and ensure the current speed gradually approaches the target value. Additionally, K_{d} is used to prevent significant fluctuations in the RSMR's speed during high-frequency control. Specifically, if the proportional term's value is excessively high, the derivative term will moderate the acceleration or deceleration.

2) Model Predictive Control: MPC is a comprehensive framework that enables the prediction of RSMR's future trajectory and provides optimized control commands at the system level [35], [71], [73], [86], [128]. MPC used for AD with RSMR usually encompasses four main stages of modeling, prediction, optimization, and actuation, specifically as follows:

- Modeling: A mathematical model is first established to describe the dynamic behavior and evolution state of the RSMR over time. The model can either be the vehicle motion model as expressed in (2), (3), and (4), or a data-driven model like a neural network.
- Prediction: Based on the current motion state and alternative mode of motion, MPC employs the system model to predict the position and moving posture of the RSMR over a series of future time steps.
- Optimization: MPC solves an optimization problem to select the optimal sequence of control inputs, aiming to meeting the pre-defined objective, such as maximizing the

linear speed and the distance to obstacles, and minimizing the angular velocity variation and energy consumption.
- Execution: The first value from the optimized control input sequence is applied to the RSMR, and the whole system transitions to the next time step. The entire process repeats periodically.

The general form of MPC is expressed as (13) and (14). As shown, MPC is essentially an optimization model.

$$\min J(\mathrm{X}, \mathrm{U}) = \sum_{k=1}^{T_\mathrm{p}} \|\mathrm{X}(t+k) - \mathrm{X_d}(t+k)\|^2 \tag{13}$$

$$\text{s.t.} \begin{cases} \mathrm{X}(t+1) = f_\mathrm{M}(\mathrm{X}(t), \mathrm{U}(t)) \\ \mathrm{U}(t) \in [\mathrm{U_l}, \mathrm{U_u}] \\ \mathrm{X}(t) \in [\mathrm{X_l}, \mathrm{X_u}] \end{cases} \tag{14}$$

where $\mathrm{X}(t)$ and $\mathrm{X_d}(t)$ typically represent RSMR's current and desired motion state at time t. Meanwhile, $\mathrm{U}(t)$ corresponds to the control signal at time t, which usually denotes the linear and angular velocity of RSMRs. T_p denotes the prediction length, and $J(\mathrm{X}, \mathrm{U})$ represents the objective function. The motion state at time $t+1$, denoted as $\mathrm{X}(t+1)$, is predicted based on the motion state at time t and the model f_M. $\mathrm{U}(t) \in [\mathrm{U_l}, \mathrm{U_u}]$ and $\mathrm{X}(t) \in [\mathrm{X_l}, \mathrm{X_u}]$ indicate that the inputs and motion state of RSMRs should fall within a specific range. To enhance computational efficiency, MPC adopts a greedy approach, optimizing RSMRs' motion only for a short-term future time horizon.

Drews et al. [35] used MPC for high-speed AD by optimizing a control sequence based on real-time visual inputs from a CNN-generated cost map. The objective is to minimize the driving path cost, while considering the RSMR dynamics and track boundaries as constraints. Oyama and Nonaka [73] used an MPC framewrok for parking control of a RSMR modeled as a nonholonomic vehicle. The objective aims to minimize the trajectory's deviation from an optimal parking path, while considering the steering limitations and parking slot alignment as constraints.

IV. Review of Applications

By combining the above methods, various applications in AD can be achieved. The applications of existing studies include navigation and obstacle avoidance, vehicle fleet coordination, intersection management, parking control, drift control, passenger unease, and hands-free control. Figure 5 shows the schematic diagram of these applications.

A. Navigation and Obstacle Avoidance

In RSMR AD research, navigation and obstacle avoidance are typically combined. The navigation task is accomplished through global path planning, while obstacle avoidance is accomplished through local path planning. Obstacles are divided into static obstacles and dynamic obstacles. This configuration makes the experiment more realistic. Specifically, in order to reach their destination, drivers have a long-term route planning, namely, global path planning. Meanwhile, they also continuously engage in local path planning to avoid various obstacles during the whole driving process. Static obstacles include buildings, barriers, roadblocks, traffic signs, streetlights, and parked vehicles, while dynamic obstacles include moving vehicles, pedestrians, and animals. To ensure driving safety, the priority of local path planning is usually higher than global path planning. Before the task of navigation and obstacle avoidance begins, detailed map information is usually required, including the RSMR's initial position and target point, as well as the position, shape, and size of static obstacles. Dynamic obstacles are typically recognized and detected by the RSMR in real-time during the task.

Global path planning is to find a feasible shortest path from the starting point to the destination, while avoiding collisions with static obstacles. Most of the global path planning is done offline, namely, completed before the RSMR starts its motion. In this case, A* and Dijkstra algorithms regard the static obstacles as impassable nodes or edges on the map [51], [132]. For the Artificial Potential Field Model, the repulsion and attraction are respectively related to the distances to obstacles and the destination [87].

Local path planning generates continuous smooth trajectories or next waypoints that comply with RSMRs' motion characteristics, while avoiding collisions with dynamic obstacles.

Some studies used predefined rules and logic to achieve local path planning of RSMRs. Muhammad et al. [51] uniformly sampled eight candidate waypoints in real-time from the LiDAR's scanning circle. The probability of selecting each waypoint is inversely proportional to the distance between that point and the goal point and directly proportional to the distance to static obstacles. In addition, dynamic obstacles' motion velocity and direction were estimated in real-time, and meanwhile the RSMR was guided to move towards the direction closer to the goal point. Cherubini et al. [133] used semicircles with different radii as candidate trajectories for the RSMR. Based on the earliest collision time with dynamic obstacles as the risk function, they provided the explicit analytical equations for the RSMR's velocity at each moment.

Some studies chose to employ the optimization method framework, and their research process is similar. Specifically, they set the distance to the goal point and obstacles as the objective function and considered the motion characteristics of RSMRs as constraints [87], [134]. In addition, some studies used RRT [135], DWA [132], Neural Networks [55], [136], [137], [138], and RL [52] for local path planning.

The studies of navigation and obstacle avoidance mentioned above only involve a single RSMR. In order to enhance navigation efficiency, Wen et al. [139] presented a hierarchical search-based algorithm called car-like conflict-based search for multi-RSMR navigation. This algorithm employs a body conflict tree, considering the shapes of RSMRs, to effectively avoid mutual collisions. In addition, the spatiotemporal hybrid-state A* algorithm is used to generate paths that adhere to both kinematic and spatiotemporal constraints.

Despite the extensive research on navigation and obstacle avoidance for RSMRs, there is still a significant gap between theoretical advances and practical applications. This disparity can be attributed to the fact that existing studies have set

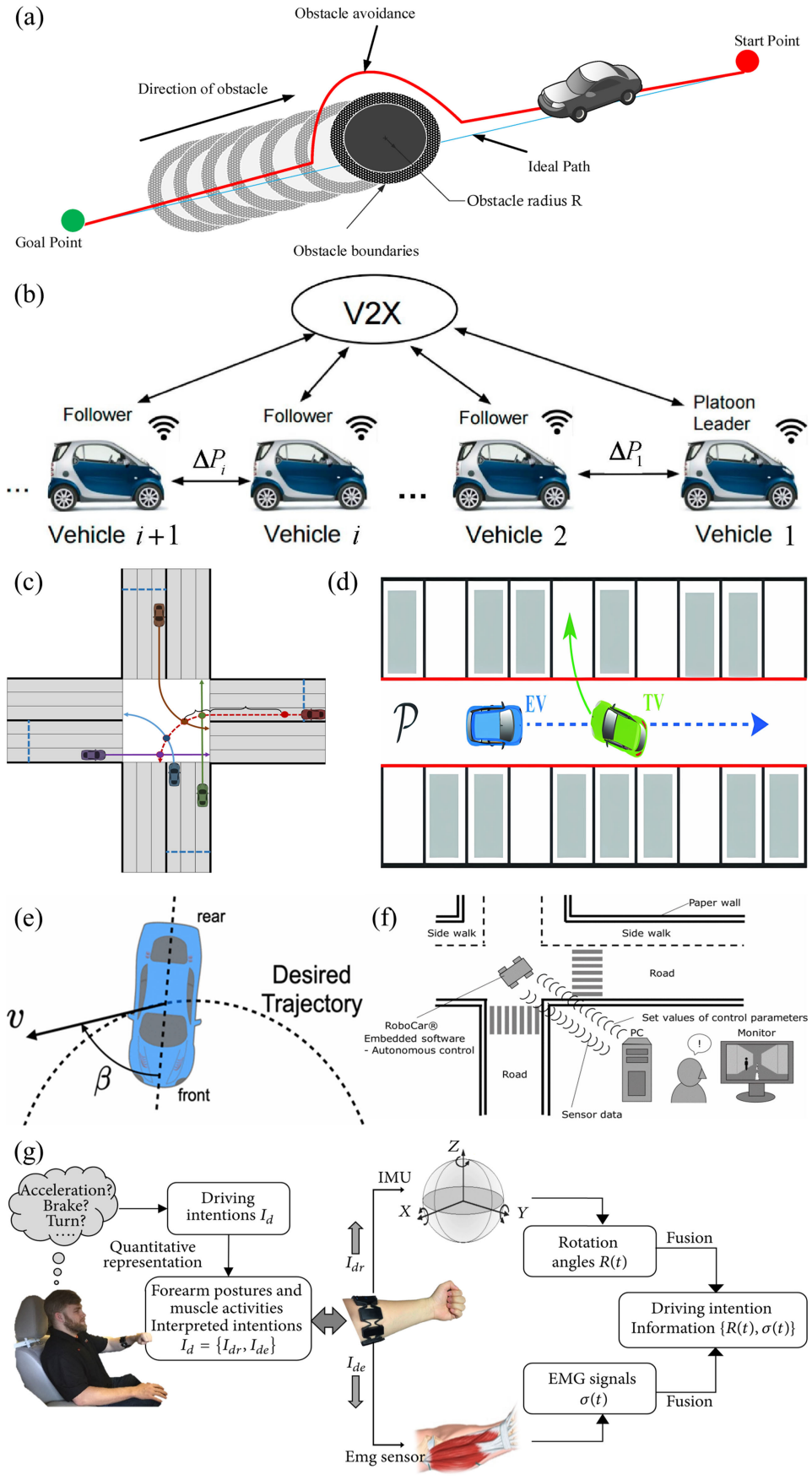

Fig. 5. Overview of the applications of RSMR in AD research: (a) navigation and obstacle avoidance [51], (b) vehicle fleet coordination [129], (c) intersection management [76], (d) parking control [86], (e) drift control [72], (f) passenger unease [130], and (g) hands-free control [131].

dynamic obstacles with regular shapes that move at a constant speed and lacks the capability to adapt their motion patterns according to the surroundings. However, real-world scenarios involve dynamic interactions between vehicles, pedestrians, and roads, that is, the motion patterns of the dynamic obstacles are much more complex. Consequently, local vehicle motion planning in reality poses higher demands compared to the existing research.

B. Vehicle Fleet Coordination

Vehicle fleet control is to arrange multiple RSMRs in a fleet to follow the same or similar trajectory, ensuring small inter-vehicle spacing while avoiding collisions, in order to increase road throughput and reduce traffic congestion. Fundamentally, this is an adaptive cruise control problem, and extensive research has been conducted on it. The control pattern can be classified into leader-follower approach, virtual structure approach, or behavior-based approach, for example [140].

However, the methods for fleet control research using RSMRs are relatively limited. They primarily rely on deriving the functions of RSMRs' displacements, velocities, and accelerations with respect to time based on physical motion models. Some studies directly chose the safe braking distance as the following distance between RSMRs [141], [142], [143]. This constant distance approach is intuitive and can roughly reflect the proportional relationship between the distance and the RSMR size. However, it is proved to be susceptible to error accumulation with an increasing number of vehicles, leading to potential issues with string stability [129], [144], [145]. To address this issue, many studies used time-based approach. For example, Klančar et al. [144] derived the RSMR's motion equations based on the Ackermann motion model, and used quadratic polynomial to fit the trajectory. They controlled the following RSMR to consistently follow the state of the leading RSMR T time units ago. Wu et al. [146] and Tuchner and Haddad [129] adopted a constant time-headway spacing policy, as shown in (15). Based on it, the RSMR's motion formula was derived as (16).

$$d_i(t) = d_i^{\mathrm{s}} + h_{\mathrm{c}} v_i(t), \quad 2 \le i \le N_{\mathrm{f}} \tag{15}$$

where $d_i(t)$ and d_i^{s} are the desired distance at time t and the standstill distance between RSMR i and RSMR $i-1$, h_{c} is the constant time headway, v_i is the speed of RSMR i, and N_{f} is the number of RSMR in the fleet.

$$G(s_t) = \frac{p(s_t)}{a_{\mathrm{d}}(s_t)} = \frac{1}{s_t^2(c_{\mathrm{e}} s_t + 1)} e^{-t_{\mathrm{d}} s_t} \tag{16}$$

where s_t is the Laplace transform of $v(t)$, $p(s_t)$ is the position of the vehicle, a_{d} is the desired acceleration, and c_{e} and t_{d} are a constant and a delay that reflect the engine and throttle actuator dynamics, respectively.

A more flexible and efficient approach is to use a time-varying time gap that changes with the variation in vehicle spacing and relative motion states. For examples, Cruz-Morales et al. [147] and Velasco-Villa et al. [148] used a time-varying gap that is a bounded non-negative differentiable function and increases as the following RSMR approaches to the leading RSMR.

In the above studies, some studies were conducted in a connected environment [129], [143], while other studies assumed no communication between vehicles, with all vehicle information sourced from their sensors [129], [141], [142], [144], [145], [147], [148].

It can be observed that existing studies on vehicle fleet tend to make simplistic assumptions about the motion of RSMRs. However, the reality of the traffic environment is much more complex, involving games, conflicts, and yielding between vehicles and pedestrians. In this complex environment, vehicles frequently need to accelerate, decelerate, or adjust the steering wheel, which significantly deviates from the assumptions made in previous studies. Also, current studies only consider a fleet of two or three RSMRs, which does not reflect the actual fleet size in reality. For example, at an intersection, the queue of vehicles may consist of dozens of RSMRs. Consequently, research that includes a larger number of RSMRs should be conducted. Furthermore, existing studies assume that the RSMRs in the fleet are homogeneous in terms of type, size, and motion characteristics. However, in reality, vehicle fleets normally consist of different types of vehicles, including trucks, buses, and cars. Hence, the research on heterogeneous vehicle fleet is necessary to complement. In addition, when a vehicle in the fleet performs an emergency braking manoeuvre or suddenly stops, or when another vehicle cuts into the fleet, the fleet will be temporarily interrupted. In this case, an investigation into the restoration of the fleet becomes a significant research topic.

C. Intersection Management

In the domain of intersection management, studies typically assume that vehicles can communicate with each other through an intersection manager (IM), enabling the exchange of positional and movement data, and the IM takes on the role of coordinating and controlling these vehicles.

Fok et al. [149] proposed eight different management policies, including V2I-Sequential, V2I-Parallel, V2I-Reservation, V2V-Sequential, V2V-Parallel, V2V-Reservation, Stop Sign, and Traffic Signal. However, they only described the framework and logic of these strategies, without providing detailed decision-making and control methods for vehicle motion.

Some studies have used kinematic formulas based on Newton's laws of motion to determine potential conflict points among multiple vehicles and to optimize the trajectory and speed of RSMRs. Khayatian et al. [75], [76] introduced an intersection management algorithm designed to guide multiple AVs safely across an unsignalized intersection. By applying a first-come-first-service policy, they identified potential conflict points among different vehicles and determined the desired speed for each vehicle using the PID control algorithm. These two studies assume that all vehicles will strictly obey the commands of the intelligent management without any errors, but in reality, this situation is impossible. Therefore, in their subsequent research, Khayatian et al. [74] proposed a communication mechanism aimed at addressing the issue of "rogue vehicle", which are vehicles that disobey intersection managers' instructions or provide false motion information. In this study, the IM will regularly calculate the error between

the actual positions and the expected positions of each vehicle. If this error exceeds a certain threshold, the IM will activate an emergency state, and command the rogue vehicle to decelerate and stop depending on the potential occurrence of a collision. For similar purposes, Dedinsky et al. [150] installed a camera above a reduced-scale intersection to detect abnormalities in the position and speed data of a RSMR through image processing technology, and provide real-time notifications to the IM.

In addition to simple physical motion models, some studies used more complex motion models and optimization methods. For example, Li et al. [151] involved a lane-free autonomous intersection management method that relies on numerical optimal control and a parameterized social force model. This approach assumed that the motion of connected and automated vehicles adheres to a third-order model with a delay. Log-exp functions were employed to convert the non-differentiable collision-avoidance constraints into convex functions while establishing the relationship between vehicle spacing and input control. An interpolating control approach is then employed to identify the optimal input, while the parameterized social force model provides an adjustable initial estimate for the numerical optimal control. Cooperative trajectories are ultimately achieved by adjusting the urgency weights in the social force model.

Through the above summary, it is not difficult to notice that the current research on intersection management focuses on creating a connected environment for all vehicles to ensure system safety and efficiency. However, the connected vehicles and environments may never be fully deployed, making it challenging to implement this intersection management model in practical scenarios. So there remains a need for studies on decentralized and unconnected intersection management.

D. Parking Control

Parking control involves planning the path of a RSMR to enable it to stop at a specific parking location, and various studies have been conducted to address this problem [73], [86], [128]. Shen et al. [86] developed a hierarchical control approach to facilitate the parking of a self-driving RSMR (SR) in a designated parking space when confronted by another human-controlled RSMR (HR) approaching in the opposite direction. This approach incorporates a high-level data-driven strategy predictor and a lower-level model-based feedback controller. The strategy predictor employs a neural network to establish a mapping from the dynamic environment to high-level strategies. These strategies include: (1) the SR passes the HR from the left side, (2) the SR passes the HR from the right side, (3) the SR yields to the HR for safety. Based on the selected strategy, a set of time-varying planes is generated online in the SR's position space. The associated space constraints are integrated into the lower-level model-based controller to guide the SR toward feasible areas.

E. Drift Control

Drift control is typically associated with fun driving maneuvers, racing performance, and slippery surfaces. It involves controlling RSMRs to deliberately execute controlled side slip, often by maintaining a fixed side-slip angle during cornering, which enables agile and aggressive turning. Essentially, drift control is a special kind of path generation and tracking. The distinctive aspect of drift control is that it relaxes the curvature constraints of the generated paths and allows lateral movement of the vehicles.

Yang et al. [72] proposed three types of drift conditions: fixed-circle drifting, moving-center drifting, and varying-interaction drifting with different tire-ground interaction parameters. Then, they derived the required steering angle and wheel rotational speed to stabilize the heading angle and the trajectory curvature. Zhang et al. [96] divided a road with a corner into three distinct regions: free, drift, and transit. In the free region, trajectories based on a bicycle model are generated and selected using a rapidly-exploring random trees approach. The drift region consists of three defined phases, namely turn-in, counter-steering, and exit. A rule-based model is employed to determine the desired steering and torque during these phases. In the transit region, a PI controller is used to minimize the deviations between the current kinematic parameters and the desired kinematic parameters.

F. Passenger Unease

Takada and Nakagawa [130] employed a RSMR to investigate the factors that contribute to passenger unease. Participants were shown recordings from the RSMR's front-facing camera, which depicted the vehicle navigating through a scaled-down city. Participants were instructed to press a button when they felt uneasy (their safety is being threatened) to trigger the recording of relevant influential factors. These factors encompassed the degree of obstacles protruding towards the road, the steering angle during right turns, the start timing of right turns, vehicle speed, and the arrangement of mini-pedestrians and buildings on the sidewalk.

G. Hands-Free Control

When human intervention is required in automated driving, the driver needs to quickly regain the control of vehicle. In this case, relying on the steering wheel and pedal can cause delays. Therefore, Wang et al. [131] developed a hands-free control method for robotic vehicles based on fuzzy logic. This method employs electromyography sensors and IMUs to capture forearm postures and muscle activity information, which serve as indicators of the human driver's intentions. The collected information is fuzzified and 'IF-THEN' statements are formulated to facilitate the RSMR's understanding of the driver's intent. This research enables drivers to conveniently interact with vehicles without physical contact. However, further research is needed to investigate the impact of these sensors on driver comfort and whether these sensors can be widely adopted on a large scale.

V. Conclusion

This paper comprehensively reviews 134 studies that focus on AD using RSMRs. First, we provide an overview of the progress made in AD research with RSMRs, and highlight

the characteristics, advantages, and disadvantages of RSMRs compared to real-world tests, driving simulators, and simulation models or software. Next, we provide an analysis of three key modules of AD task, including perception, decision-making, and actuation. Within each module, we describe the commonly used devices or methods, and compare their respective advantages and disadvantages. Additionally, we also identify the main applications covered in existing studies, including navigation and obstacle avoidance, vehicle fleet coordination, intersection management, parking control, drift control, passenger unease, and hands-free control.

The primary objective of AD experiments and research with RSMRs is to develop and evaluate algorithms for perception, decision-making, and actuation. These studies serve as a foundation and reference for real-world AD experiments and tasks. Although physical RSMRs can partially simulate the characteristics of real-world traffic and vehicles, and the results obtained from experiments conducted with RSMRs provide more authentic and compelling evidence compared to simulations, it is important to acknowledge that the experimental environment still differs from reality. Consequently, we outline the shortcomings and limitations of RSMRs as follows:

- First, experiments with RSMRs have a limited scope. While RSMRs are well-suited for exploring the performance of decision-making and actuation algorithms, they are not applicable to certain issues such as human driving experiment, human-machine interaction, takeover, driver behavior and psychology when co-driving with machines, and the societal impacts of AVs.
- Second, real-world traffic scenarios are often unpredictable and significantly more complex than controlled experimental environments. Specifically, real-world traffic involves diverse road conditions and intricate interactions with various types of vehicles and pedestrians. Emergencies, traffic violations, and congestion can occur at any time. Although physical RSMRs can simulate and model some of these factors, accurately replicating the complete complexity and variability of real-world traffic scenarios is impossible. This suggests that RSMRs also exhibit a lack of high-fidelity characteristics to some degree.
- Third, RSMRs are miniature and simplified versions of real-world vehicles, typically characterized by smaller size, lighter mass, lower speed, and different kinetic characteristics and limitations compared to actual vehicles. Consequently, many desired experiments cannot be conducted in a robotic environment. Furthermore, due to the kinetic disparities between RSMRs and real vehicles, algorithms developed and tested on RSMRs cannot necessarily be directly applied in practical settings without further testing.

In addition to the inherent limitations of RSMRs, the existing research on AD using RSMRs presents several unresolved issues and areas requiring improvement. In this context, we outline the identified gaps and challenges of this research as follows to provide reference for future research.

- First, the existing studies have primarily focused on specific problems, and the scenario design and assumptions for these problems are simplistic, e.g., avoiding static or uniformly moving dynamic obstacles. However, real-world traffic involves complex interactions with various vehicles and pedestrians that have autonomous mobility. These traffic users have resulted in an environment much more complex than the testing environment of RSMRs.
  Therefore, a crucial task now is to closely simulate reality in scenario design to enhance the practical value of RSMR AD experiments and the experimental results. Specifically, we argue for research on interactive driving events commonly observed in real-world traffic. Examples include merging, lane changing, cut-in, overtaking, and pedestrian avoidance behaviors. In addition, the number of RSMRs should be increased for the research of vehicle interactions and intersection management to more closely resemble reality.
- Second, the existing research has short experiment duration and distance, resulting in insufficient reliability of algorithm evaluation results. Furthermore, the impact of time and distance on the AD system's performance remains to be studied.
  We think that by fully leveraging the advantages of low cost and small size of RSMRs, extending the experiment duration can serve as a reference for AD in long-distance scenarios such as motorways and rural roads.
- Third, existing research assumes that the environment is stable and the surroundings of RSMRs tend to move as expected. However, in reality, emergencies, traffic violations, and congestion can occur at any time.
  Therefore, we believe it is necessary to consider the environmental uncertainty in the research, for example, evaluating the emergency response capability of a RSMR when the preceding vehicle suddenly brakes in a vehicle fleet, introducing pedestrians that suddenly run out in intersections, adding some randomness to the movement of dynamic obstacles, or evaluating the measures taken by the RSMR in the event of sensor failure.
- Fourth, there is a scarcity of studies that compare the outcomes of RSMR experiments with those of real-world vehicles. These reduced-scale systems differ from reality in terms of size, vehicle speed, and various other factors. Conclusions drawn from experiments conducted with RSMRs may not always be applicable to real-world scenarios.
  Thus, we suggest additional comparative studies are necessary to thoroughly assess the reliability and transferability of RSMR experiments.
- Fifth, existing studies typically involve one or more fully autonomous RSMRs. However, fully autonomous vehicles are still in the experimental stage. For a considerable period in the future, the vehicles on the road will transition towards a mixed traffic flow consisting of both manually driven and autonomous vehicles.
  To better understand mixed traffic with RSMRs, we advise programming some RSMRs for full autonomous driving while manually controlling others (for example with a human in a driving simulator or controlling the RSMR using a joystick).

Through a detailed analysis of existing literature, we have formed initial predictions about the future development directions and prospects of RSMR in AD applications, as outlined below.

- First, future research will simulate more real-world scenarios and factors, aiming to enhance the realism of AV experiments with RSMR. To achieve this, realistic miniature cities will be established, consisting of lane markings, central dividers, traffic lights, and artificial pedestrians. Additionally, other road models and scenarios will be designed, potentially including freeway ramps, mountainous roads, tunnels, and elevated bridges.
- Second, as more advanced artificial intelligence algorithms are integrated into the perception and decision-making of RSMR, the level of intelligence of RSMR will increase accordingly. This enables the testing of more AV scenarios in environments like ramp merging and lane-free traffic.
- Third, more AD functions and the corresponding hardware and software will be developed and investigated, such as voice control, gesture control, and human-machine interaction interfaces.

In conclusion, the use of RSMRs in AD research shows promise. However, further research is required to tackle the existing challenges.

REFERENCES

[1] Y. Ma, Z. Xie, S. Chen, Y. Wu, and F. Qiao, "Real-time driving behavior identification based on multi-source data fusion," *Int. J. Environ. Res. Public Health*, vol. 19, no. 1, p. 348, Dec. 2021.

[2] Y. Ma, Z. Xie, S. Chen, F. Qiao, and Z. Li, "Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals," *Transp. Res. C, Emerg. Technol.*, vol. 146, Jan. 2023, Art. no. 103983.

[3] Y. Ma, Z. Xie, W. Li, and S. Chen, "Modeling driving styles of online ride-hailing drivers with model identifiability and interpretability," *Travel Behav. Soc.*, vol. 33, Oct. 2023, Art. no. 100645.

[4] Y. Ma, F. Wang, S. Chen, G. Xing, Z. Xie, and F. Wang, "A dynamic method to predict driving risk on sharp curves using multi-source data," *Accident Anal. Prevention*, vol. 191, Oct. 2023, Art. no. 107228.

[5] L. Gao, W. Bai, R. Leary, K. Varadarajan, and S. Brennan, "ROS integration of external vehicle motion simulations with an AIMSUN traffic simulator as a tool to assess CAV impacts on traffic," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 870–875, 2021.

[6] M. Tajalli, R. Niroumand, and A. Hajbabaie, "Distributed cooperative trajectory and lane changing optimization of connected automated vehicles: Freeway segments with lane drop," *Transp. Res. C, Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103761.

[7] X. Tang et al., "Uncertainty-aware decision-making for autonomous driving at uncontrolled intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9725–9735, Sep. 2023.

[8] M. K. Natvig, E. Stav, and T. M. Stene, "Concepts for future traffic management supporting automation, multimodal coordination and resilience," *Multimodal Coordination Resilience*, p. 23, Jun. 2023.

[9] T. Zeng and B. Si, "Mobile robot exploration based on rapidly-exploring random trees and dynamic window approach," in *Proc. 5th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2019, pp. 51–57.

[10] L. Tai, P. Yun, Y. Chen, C. Liu, H. Ye, and M. Liu, "Visual-based autonomous driving deployment from a stochastic and uncertainty-aware perspective," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2622–2628.

[11] A.-T. Nguyen, J. Rath, T.-M. Guerra, R. Palhares, and H. Zhang, "Robust set-invariance based fuzzy output tracking control for vehicle autonomous driving under uncertain lateral forces and steering constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5849–5860, Sep. 2021.

[12] L. Zheng, C. Zhu, Z. He, and T. He, "Safety rule-based cellular automaton modeling and simulation under V2V environment," *Transportmetrica A, Transp. Sci.*, vol. 17, no. 1, pp. 81–106, Jan. 2021.

[13] C. Zhang, H.-K. Lam, J. Qiu, P. Qi, and Q. Chen, "Fuzzy-model-based output feedback steering control in autonomous driving subject to actuator constraints," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 3, pp. 457–470, Mar. 2021.

[14] E. V. Filho et al., "Towards a cooperative robotic platooning testbed," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2020, pp. 332–337.

[15] H. Lee, J. Yang, and K. Moessner, "Implementation of a collision avoidance system to assist safe driving based on data fusion in vehicular networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 795–800.

[16] DARPA. (2023). *Defense Advanced Research Projects Agency*. [Online]. Available: https://www.darpa.mil/program/darpa-robotics-challenge/

[17] (2023). *Festival Nacional De Robótica*. [Online]. Available: https://www.festivalnacionalrobotica.pt/2023/

[18] ELROB. (2023). *The European Land Robot Trial*. [Online]. Available: https://www.elrob.org/

[19] J. Hu, Y. Zhang, and S. Rakheja, "Adaptive lane change trajectory planning scheme for autonomous vehicles under various road frictions and vehicle speeds," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1252–1265, Feb. 2023.

[20] H. Ke, "Cooperative adaptive cruise control using V2V communication and deep learning," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Windsor, Windsor, ON, Canada, 2022.

[21] J. Hu, Y. Zhang, and S. Rakheja, "Adaptive trajectory tracking for car-like vehicles with input constraints," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2801–2810, Mar. 2022.

[22] Y. Liu, H. Schofield, and J. Shan, "Navigation of a self-driving vehicle using one fiducial marker," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2021, pp. 1–6.

[23] L. A. D. O. Nogueira, M. F. Koyama, R. D. A. Cordeiro, A. M. Ribeiro, S. S. Bueno, and E. C. De Paiva, "A miniaturized four-wheel robotic vehicle for autonomous driving research in off-road scenarios," in *Proc. Congresso Brasileiro de Automática (CBA)*, João Pessoa, Brazil, 2019, vol. 1, no. 1.

[24] I. Kang, R. Cimurs, J. H. Lee, and I. Hong Suh, "Fusion drive: End-to-end multi modal sensor fusion for guided low-cost autonomous vehicle," in *Proc. 17th Int. Conf. Ubiquitous Robots (UR)*, Jun. 2020, pp. 421–428.

[25] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7262–7269, Oct. 2021.

[26] V. Rosas-Cervantes, Q.-D. Hoang, S. Woo, and S.-G. Lee, "Mobile robot 3D trajectory estimation on a multilevel surface with multimodal fusion of 2D camera features and a 3D light detection and ranging point cloud," *Int. J. Adv. Robotic Syst.*, vol. 19, no. 2, Mar. 2022, Art. no. 172988062210891.

[27] M. Molska and D. Belter, "Convolutional neural network-based local obstacle avoidance for a mobile robot," in *Proc. Conf. Autom.* Cham, Switzerland: Springer, 2021, pp. 262–271.

[28] S. Mitchell, I. Sajjad, A. Al-Hashimi, S. Dadras, R. M. Gerdes, and R. Sharma, "Visual distance estimation for pure pursuit based platooning with a monocular camera," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 2327–2332.

[29] C. Y. Kuo, Y. R. Lu, and S. M. Yang, "On the image sensor processing for lane detection and control in vehicle lane keeping systems," *Sensors*, vol. 19, no. 7, p. 1665, Apr. 2019.

[30] S. Boubou, H. Jabbari Asl, T. Narikiyo, and M. Kawanishi, "Real-time recognition and pursuit in robots based on 3D depth data," *J. Intell. Robotic Syst.*, vol. 93, nos. 3–4, pp. 587–600, Mar. 2019.

[31] Y. Kawai and K. Ito, "Estimation method for time to contact from visual information—A simple approach that requires no recognition of objects," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 469–474.

[32] R. Farkh, M. T. Quasim, K. Al Jaloud, S. Alhuwaimel, and S. Tabrez Siddiqui, "Computer vision-control-based CNN-PID for mobile robot," *Comput., Mater. Continua*, vol. 68, no. 1, pp. 1065–1079, 2021.

[33] K. Li, S. Eiffert, M. Shan, F. Gomez-Donoso, S. Worrall, and E. Nebot, "Attentional-GCNN: Adaptive pedestrian trajectory prediction towards generic autonomous vehicle use cases," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14241–14247.

[34] T. N. Nizar, R. Hartono, and D. Meidina, "Human detection and avoidance control systems of an autonomous vehicle," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 879, no. 1, Jul. 2020, Art. no. 012103.

[35] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Aggressive deep driving: Model predictive control with a CNN cost model," 2017, *arXiv:1707.05303*.

[36] N. Rezaei and S. Darabi, "Mobile robot monocular vision-based obstacle avoidance algorithm using a deep neural network," *Evol. Intell.*, vol. 16, no. 6, pp. 1999–2014, Dec. 2023.

[37] C. Ferencz and M. Zöldy, "Autonomous driving cycle modeling, simulation and validation on 1:10 scale vehicle model platforms," *Int. J. Traffic Transp. Eng.*, vol. 11, no. 4, pp. 1–15, 2021.

[38] M. Ayalew, S. Zhou, M. Assefa, and G. Yilma, "Spatial–temporal attentive motion planning network for autonomous vehicles," in *Proc. 18th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2021, pp. 601–605.

[39] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.

[40] Y. Sun, N. Wu, S. Tateno, and H. Ogai, "Development of driving support system for electric vehicle by using image processing technology," in *Proc. 12th Int. Conf. Control, Autom. Syst.*, Oct. 2012, pp. 1965–1968.

[41] J. M. Pierre, "Spatio-temporal deep learning for robotic visuomotor control," in *Proc. 4th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2018, pp. 94–103.

[42] T. Happek, U. Lang, T. Bockmeier, D. Neubauer, and A. Kuznietsov, "Distributed data acquisition and control system for an scaled autonomous vehicle," in *Proc. IEEE 7th Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS)*, vol. 1, Sep. 2013, pp. 437–440.

[43] S. J. Fusic, K. Hariharan, R. Sitharthan, and S. Karthikeyan, "Scene terrain classification for autonomous vehicle navigation based on semantic segmentation method," *Trans. Inst. Meas. Control*, vol. 44, no. 13, pp. 2574–2587, Sep. 2022.

[44] B.-C.-Z. Blaga, M.-A. Deac, R. W. Y. Al-doori, M. Negru, and R. Danescu, "Miniature autonomous vehicle development on raspberry pi," in *Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2018, pp. 229–236.

[45] D. Teso-Fz-Beto no, E. Zulueta, A. Sánchez-Chica, U. Fernandez-Gamiz, and A. Saenz-Aguirre, "Semantic segmentation to develop an indoor navigation system for an autonomous mobile robot," *Mathematics*, vol. 8, no. 5, p. 855, May 2020.

[46] S. Brahimi, R. Tiar, O. Azouaoui, M. Lakrouf, and M. Loudini, "Car-like mobile robot navigation in unknown urban areas," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1727–1732.

[47] M. Khan and G. Parker, "Vision based indoor obstacle avoidance using a deep convolutional neural network," in *Proc. 11th Int. Joint Conf. Comput. Intell.*, 2019, pp. 403–411.

[48] C. Liu, C. Zhou, W. Cao, F. Li, and P. Jia, "A novel design and implementation of autonomous robotic car based on ROS in indoor scenario," *Robotics*, vol. 9, no. 1, p. 19, Mar. 2020.

[49] J. Vilca, L. Adouane, and Y. Mezouar, "Stable and flexible multi-vehicle navigation based on dynamic inter-target distance matrix," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1416–1431, Apr. 2019.

[50] H. Ryu, M. Yoon, D. Park, and S.-E. Yoon, "Confidence-based robot navigation under sensor occlusion with deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8231–8237.

[51] A. Muhammad et al., "A generalized laser simulator algorithm for mobile robot path planning with obstacle avoidance," *Sensors*, vol. 22, no. 21, p. 8177, Oct. 2022.

[52] H. Song, A. Li, T. Wang, and M. Wang, "Multimodal deep reinforcement learning with auxiliary task for obstacle avoidance of indoor mobile robot," *Sensors*, vol. 21, no. 4, p. 1363, Feb. 2021.

[53] S. Masahiko, U. Shohei, O. Yutaka, and M. Takami, "Real-time localization of mobile robot using observers from LiDAR measurement," *Innov. Comput., Inf. Control*, vol. 17, no. 9, p. 1039, 2023.

[54] O. H. Graven et al., "An autonomous indoor exploration robot rover and 3D modeling with photogrammetry," in *Proc. Int. ECTI northern Sect. Conf. Electr., Electron., Comput. Telecommun. Eng. (ECTI-NCON)*, Feb. 2018, pp. 44–47.

[55] U. Farooq, M. Amar, M. U. Asad, A. Hanif, and S. O. Saleh, "Design and implementation of neural network based controller for mobile robot navigation in unknown environments," *Int. J. Comput. Electr. Eng.*, vol. 6, no. 2, pp. 83–89, 2014.

[56] C.-L. Chang, B.-H. Wu, and C.-C. Chang, "Autonomous field robotic vehicle with embedded multi-sensor system for agricultural applications," in *Proc. Int. Tech. Meeting Inst. Navigat.*, 2011, pp. 1077–1084.

[57] J.-H. Jeong and K. Park, "Numerical analysis of 2-D positioned, indoor, fuzzy-logic, autonomous navigation system based on chromaticity and frequency-component analysis of LED light," *Sensors*, vol. 21, no. 13, p. 4345, Jun. 2021.

[58] Y. Deng, Y. Shan, Z. Gong, and L. Chen, "Large-scale navigation method for autonomous mobile robot based on fusion of GPS and LiDAR SLAM," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 3145–3148.

[59] Y. Li et al., "A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach," *IEEE Access*, vol. 10, pp. 57736–57747, 2022.

[60] C. Li, S. Wang, Y. Zhuang, and F. Yan, "Deep sensor fusion between 2D laser scanner and IMU for mobile robot localization," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8501–8509, Mar. 2021.

[61] G.-S. Cai, H.-Y. Lin, and S.-F. Kao, "Mobile robot localization using GPS, IMU and visual odometry," in *Proc. Int. Autom. Control Conf. (CACS)*, Nov. 2019, pp. 1–6.

[62] T. Q. Tran, A. Becker, and D. Grzechca, "Environment mapping using sensor fusion of 2D laser scanner and 3D ultrasonic sensor for a real mobile robot," *Sensors*, vol. 21, no. 9, p. 3184, May 2021.

[63] L. Zhang, X. Wu, R. Gao, L. Pan, and Q. Zhang, "A multi-sensor fusion positioning approach for indoor mobile robot using factor graph," *Measurement*, vol. 216, Jul. 2023, Art. no. 112926.

[64] J. Li, R. Li, J. Li, J. Wang, Q. Wu, and X. Liu, "Dual-view 3D object recognition and detection via LiDAR point cloud and camera image," *Robot. Auto. Syst.*, vol. 150, Apr. 2022, Art. no. 103999.

[65] Y.-H. Kim, J.-I. Jang, and S. Yun, "End-to-end deep learning for autonomous navigation of mobile robot," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–6.

[66] A. Zia, T. Gulrez, and T. Chaudhry, "Heterogeneous sensor fusion framework for autonomous mobile robot obstacle avoidance," in *Proc. 10th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2010, pp. 1052–1058.

[67] P. Vadakkepat and L. Jing, "Improved particle filter in sensor fusion for tracking randomly moving object," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1823–1832, Oct. 2006.

[68] A. J. Barreto-Cubero, A. Gómez-Espinosa, J. A. E. Cabello, E. Cuan-Urquizo, and S. R. Cruz-Ramírez, "Sensor data fusion for a mobile robot using neural networks," *Sensors*, vol. 22, no. 1, p. 305, Dec. 2021.

[69] G. Huskić, S. Buck, and A. Zell, "GeRoNa: Generic robot navigation," *J. Intell. Robotic Syst.*, vol. 95, no. 2, pp. 419–442, Aug. 2019.

[70] F. Ma, J. Shi, Y. Yang, J. Li, and K. Dai, "ACK-MSCKF: Tightly-coupled Ackermann multi-state constraint Kalman filter for autonomous vehicle localization," *Sensors*, vol. 19, no. 21, p. 4816, Nov. 2019.

[71] X. Zhou et al., "Learning-based MPC controller for drift control of autonomous vehicles," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 322–328.

[72] B. Yang, Y. Lu, X. Yang, and Y. Mo, "A hierarchical control framework for drift maneuvering of autonomous vehicles," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 1387–1393.

[73] K. Oyama and K. Nonaka, "Model predictive parking control for nonholonomic vehicles using time-state control form," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 458–465.

[74] M. Khayatian, R. Dedinsky, S. Choudhary, M. Mehrabian, and A. Shrivastava, "R2 IM-robust and resilient intersection management of connected autonomous vehicles," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[75] M. Khayatian, M. Mehrabian, and A. Shrivastava, "RIM: Robust intersection management for connected autonomous vehicles," in *Proc. IEEE Real-Time Syst. Symp. (RTSS)*, Dec. 2018, pp. 35–44.

[76] M. Khayatian, Y. Lou, M. Mehrabian, and A. Shirvastava, "Cross-roads+: A time-aware approach for intersection management of connected autonomous vehicles," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, pp. 1–28, Nov. 2019, doi: 10.1145/3364182.

[77] F. Valero, F. Rubio, and C. Llopis-Albert, "Assessment of the effect of energy consumption on trajectory improvement for a car-like robot," *Robotica*, vol. 37, no. 11, pp. 1998–2009, Nov. 2019.

[78] P. Scheffe et al., "Networked and autonomous model-scale vehicles for experiments in research and education," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17332–17337, 2020.

[79] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Auto. Robots*, vol. 37, no. 3, pp. 279–300, Oct. 2014.

[80] M.-T. Lee, B.-Y. Chen, and Y.-C. Lai, "A hybrid Tabu search and 2-opt path programming for mission route planning of multiple robots under range limitations," *Electronics*, vol. 9, no. 3, p. 534, Mar. 2020.

[81] C. Li, X. Huang, J. Ding, K. Song, and S. Lu, "Global path planning based on a bidirectional alternating search A* algorithm for mobile robots," *Comput. Ind. Eng.*, vol. 168, Jun. 2022, Art. no. 108123.

[82] E. McCormick, H. Lang, and C. W. De. Silva, "Dynamic modeling and simulation of a four-wheel skid-steer mobile robot using linear graphs," *Electronics*, vol. 11, no. 15, p. 2453, Aug. 2022.

[83] J. L. C. Carvalho, P. C. M. A. Farias, and E. F. S. Filho, "Global localization of unmanned ground vehicles using swarm intelligence and evolutionary algorithms," *J. Intell. Robotic Syst.*, vol. 107, no. 3, p. 45, Mar. 2023.

[84] W. Chen, J. Sun, W. Li, and D. Zhao, "A real-time multi-constraints obstacle avoidance method using LiDAR," *J. Intell. Fuzzy Syst.*, vol. 39, no. 1, pp. 119–131, Jul. 2020.

[85] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.

[86] X. Shen, E. L. Zhu, Y. R. Stürz, and F. Borrelli, "Collision avoidance in tightly-constrained environments without coordination: A hierarchical control approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2674–2680.

[87] X. Chen, Z. Huang, Y. Sun, Y. Zhong, R. Gu, and L. Bai, "Online on-road motion planning based on hybrid potential field model for car-like robot," *J. Intell. Robotic Syst.*, vol. 105, no. 1, p. 7, May 2022.

[88] S. Liu and D. Sun, "Optimal motion planning of a mobile robot with minimum energy consumption," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2011, pp. 43–48.

[89] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-Spline curves," *J. Intell. Robotic Syst.*, vol. 80, no. S1, pp. 23–56, Dec. 2015.

[90] S. Ghaffari and M. Homaeinezhad, "Autonomous path following by fuzzy adaptive curvature-based point selection algorithm for four-wheel-steering car-like mobile robot," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 232, no. 15, pp. 2655–2665, Aug. 2018.

[91] S. T. Dang, X. M. Dinh, T. D. Kim, H. L. Xuan, and M.-H. Ha, "Adaptive backstepping hierarchical sliding mode control for 3-wheeled mobile robots based on RBF neural networks," *Electronics*, vol. 12, no. 11, p. 2345, May 2023.

[92] Y. Ou, Y. Fan, X. Zhang, Y. Lin, and W. Yang, "Improved A* path planning method based on the grid map," *Sensors*, vol. 22, no. 16, p. 6198, Aug. 2022.

[93] J. Zhao, S. Liu, and J. Li, "Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS," *Sensors*, vol. 22, no. 11, p. 4172, May 2022.

[94] J. Morales, J. L. Martínez, M. A. Martínez, and A. Mandow, "Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, pp. 1–10, Dec. 2009.

[95] M. Park and Y. Kang, "Experimental verification of a drift controller for autonomous vehicle tracking: A circular trajectory using LQR method," *Int. J. Control, Autom. Syst.*, vol. 19, no. 1, pp. 404–416, Jan. 2021.

[96] F. Zhang, J. Gonzales, S. E. Li, F. Borrelli, and K. Li, "Drift control for cornering maneuver of autonomous vehicles," *Mechatronics*, vol. 54, pp. 167–174, Oct. 2018.

[97] A. Akhtar, C. Nielsen, and S. L. Waslander, "Path following using dynamic transverse feedback linearization for car-like robots," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 269–279, Apr. 2015.

[98] H. Xue, Z. Zhang, M. Wu, and P. Chen, "Fuzzy controller for autonomous vehicle based on rough sets," *IEEE Access*, vol. 7, pp. 147350–147361, 2019.

[99] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Autonom. Robots*, vol. 45, no. 1, pp. 51–76, Jan. 2021.

[100] P. Saranrittichai, N. Niparnan, and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on dynamic window approach," in *Proc. 10th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2013, pp. 1–4.

[101] E. Dönmez, A. F. Kocamaz, and M. Dirik, "Bi-RRT path extraction and curve fitting smooth with visual based configuration space mapping," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2017, pp. 1–5.

[102] D. Wu, L. Wei, G. Wang, L. Tian, and G. Dai, "APF-IRRT*: An improved informed rapidly-exploring random trees-star algorithm by introducing artificial potential field method for mobile robot path planning," *Appl. Sci.*, vol. 12, no. 21, p. 10905, Oct. 2022.

[103] J. Sun, J. Zhao, X. Hu, H. Gao, and J. Yu, "Autonomous navigation system of indoor mobile robots using 2D LiDAR," *Mathematics*, vol. 11, no. 6, p. 1455, Mar. 2023.

[104] D. H. Lee, S. S. Lee, C. K. Ahn, P. Shi, and C.-C. Lim, "Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9998–10006, Oct. 2021.

[105] R. Hess, F. Kempf, and K. Schilling, "Trajectory planning for car-like robots using rapidly exploring random Trees*," *IFAC Proc. Volumes*, vol. 46, no. 29, pp. 44–49, 2013.

[106] S. Seiya, A. Carballo, E. Takeuchi, C. Miyajima, and K. Takeda, "End-to-end navigation with branch turning support using convolutional neural network," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 499–506.

[107] S. Hoshino and J. Sumiyoshi, "Discrete motion planner based on deep recurrent neural network for mobile robot obstacle avoidance in dead-end environments," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2022, pp. 979–984.

[108] Z. Lai and T. Bräunl, "End-to-end learning with memory models for complex autonomous driving tasks in indoor environments," *J. Intell. Robotic Syst.*, vol. 107, no. 3, p. 37, Mar. 2023.

[109] I. A. Ribeiro, T. Ribeiro, G. Lopes, and A. F. Ribeiro, "End-to-end approach for autonomous driving: A supervised learning method using computer vision algorithms for dataset creation," *Algorithms*, vol. 16, no. 9, p. 411, Aug. 2023.

[110] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4693–4700.

[111] B. Zhou, Y. He, W. Huang, X. Yu, F. Fang, and X. Li, "Place recognition and navigation of outdoor mobile robots based on random forest learning with a 3D LiDAR," *J. Intell. Robotic Syst.*, vol. 104, no. 4, p. 72, Apr. 2022.

[112] H. Zhou and S. Sakane, "Sensor planning for mobile robot localization - a hierarchical approach using a Bayesian network and a particle filter," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 481–487, 2008.

[113] B. Öztürk and V. Sezer, "A new speed planning method based on predictive curvature calculation for autonomous driving," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 30, no. 4, pp. 1555–1570, May 2022.

[114] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Vision-based high-speed driving with a deep dynamic observer," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1564–1571, Apr. 2019.

[115] A. Carballo et al., "End-to-end autonomous mobile robot navigation with model-based system support," *J. Robot. Mechatronics*, vol. 30, no. 4, pp. 563–583, Aug. 2018.

[116] C. C. E. Chewu and V. Manoj Kumar, "Autonomous navigation of a mobile robot in dynamic indoor environments using SLAM and reinforcement learning," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 402, Sep. 2018, Art. no. 012022.

[117] W. Zhang, J. Gai, Z. Zhang, L. Tang, Q. Liao, and Y. Ding, "Double-DQN based path smoothing and tracking control method for robotic vehicle navigation," *Comput. Electron. Agricult.*, vol. 166, Nov. 2019, Art. no. 104985.

[118] H. Lu, "Integrating imitation learning with human driving data into reinforcement learning to improve training efficiency for autonomous driving," 2021, *arXiv:2111.11673*.

[119] A. Kendall et al., "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8248–8254.

[120] D. Kamran, J. Zhu, and M. Lauer, "Learning path tracking for real car-like mobile robots from simulation," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2019, pp. 1–6.

[121] G. Kahn, P. Abbeel, and S. Levine, "BADGR: An autonomous self-supervised learning-based navigation system," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1312–1319, Apr. 2021.

[122] Y. Wang, K. Zheng, D. Tian, X. Duan, and J. Zhou, "Pre-training with asynchronous supervised learning for reinforcement learning based autonomous driving," *Frontiers Inf. Technol. Electron. Eng.*, vol. 22, no. 5, pp. 673–686, May 2021.

[123] D. Li and O. Okhrin, "Vision-based DRL autonomous driving agent with Sim2Real transfer," 2023, *arXiv:2305.11589*.

[124] M.-F.-R. Lee and S. H. Yusuf, "Mobile robot navigation using deep reinforcement learning," *Processes*, vol. 10, no. 12, p. 2748, Dec. 2022.

[125] D. Pérez-Morales, O. Kermorgant, S. Domínguez-Quijada, and P. Martinet, "Multisensor-based predictive control for autonomous parking," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 835–851, Apr. 2022.

[126] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space," *Int. J. Control, Autom. Syst.*, vol. 7, no. 3, pp. 429–435, Jun. 2009.

[127] V. Robila, L. Paulino, M. Rao, I. Li, M. Zhu, and W. Wang, "Design and implementation of PID-based steering control for 1/10-scale autonomous vehicle," in *Proc. IEEE 12th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Dec. 2021, pp. 758–762.

[128] K. Sakaeta, T. Oda, K. Nonaka, and K. Sekiguchi, "Model predictive parking control with on-line path generations and multiple switching motions," in *Proc. IEEE Conf. Control Appl. (CCA)*, Sep. 2015, pp. 804–809.

[129] A. Tuchner and J. Haddad, "Vehicle platoon formation using interpolating control: A laboratory experimental analysis," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 21–47, Nov. 2017.

[130] N. Takada and S. Nakagawa, "Differences of passenger unease emotional areas, an experimental case study," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2018, pp. 595–600.

[131] W. Wang, R. Li, L. Guo, Z. M. Diekel, and Y. Jia, "Hands-free maneuvers of robotic vehicles via human intentions understanding using wearable sensing," *J. Robot.*, vol. 2018, pp. 1–10, Jan. 2018.

[132] L.-S. Liu et al., "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–12, Feb. 2021.

[133] A. Cherubini, B. Grechanichenko, F. Spindler, and F. Chaumette, "Avoiding moving obstacles during visual navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3069–3074.

[134] R. M. K. Chetty, M. Singaperumal, and T. Nagarajan, "Behavior based multi robot formations with active obstacle avoidance based on switching control strategy," *Adv. Mater. Res.*, vols. 433–440, pp. 6630–6635, Jan. 2012.

[135] L. Zhang, Y. Zhang, and Y. Li, "Path planning for indoor mobile robot based on deep learning," *Optik*, vol. 219, Oct. 2020, Art. no. 165096.

[136] K. Zhu, W. Chen, W. Zhang, R. Song, and Y. Li, "Autonomous robot navigation based on multi-camera perception," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5879–5885.

[137] O. Azouaoui, N. Ouadah, I. Mansour, and A. Semani, "Neural networks and fuzzy logic navigation approach for a bi-steerable mobile robot," in *Proc. 8th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Nov. 2011, pp. 44–49.

[138] O. Azouaoui, M. Kadri, and N. Ouadah, "Implementation of a neural-based navigation approach on indoor and outdoor mobile robots," in *Proc. 5th Int. Conf. Soft Comput. Transdisciplinary Sci. Technol. (CSTST)*, 2008, pp. 71–77.

[139] L. Wen, Y. Liu, and H. Li, "CL-MAPF: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints," *Robot. Auto. Syst.*, vol. 150, Apr. 2022, Art. no. 103997.

[140] A. Soni and H. Hu, "Formation control for a fleet of autonomous ground vehicles: A survey," *Robotics*, vol. 7, no. 4, p. 67, Nov. 2018.

[141] Y. Zhao and H. Ogai, "Development of a platooning control algorithm based on RoboCar," in *Proc. SICE Annu. Conf.*, Sep. 2011, pp. 352–355.

[142] D. L. Luu and C. Lupu, "Experimenta1Evaluation of smart cars model for a platoon of vehicles," in *Proc. 23rd Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2019, pp. 815–820.

[143] M. Goli and A. Eskandarian, "Mobile robot coordinated platooning: A small-scale experimental evaluation to emulate connected vehicles," in *Proc. Dyn., Vibrat., Control*, vol. 4A, Nov. 2015, Art. no. V04AT04A004.

[144] G. Klančar, D. Matko, and S. Blažič, "A control strategy for platoons of differential drive wheeled mobile robot," *Robot. Auto. Syst.*, vol. 59, no. 2, pp. 57–64, Feb. 2011.

[145] G. Klančar, D. Matko, and S. Blažič, "Wheeled mobile robots control in a linear platoon," *J. Intell. Robotic Syst.*, vol. 54, no. 5, pp. 709–731, May 2009.

[146] C. Wu, Y. Lin, and A. Eskandarian, "Cooperative adaptive cruise control with adaptive Kalman filter subject to temporary communication loss," *IEEE Access*, vol. 7, pp. 93558–93568, 2019.

[147] R. D. Cruz-Morales, M. Velasco-Villa, and A. Rodriguez-Angeles, "Chain formation control for a platoon of robots using time-gap separation," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 2, Mar. 2018, Art. no. 172988141877085.

[148] M. Velasco-Villa, R. D. Cruz-Morales, A. Rodriguez-Angeles, and C. A. Domínguez-Ortega, "Observer-based time-variant spacing policy for a platoon of non-holonomic mobile robots," *Sensors*, vol. 21, no. 11, p. 3824, May 2021.

[149] C.-L. Fok et al., "A platform for evaluating autonomous intersection management policies," in *Proc. IEEE/ACM 3rd Int. Conf. Cyber-Phys. Syst.*, Apr. 2012, pp. 87–96.

[150] R. Dedinsky, M. Khayatian, M. Mehrabian, and A. Shrivastava, "A dependable detection mechanism for intersection management of connected autonomous vehicles (interactive presentation)," in *Proc. Workshop Auto. Syst. Design (ASD)*, 2019, pp. 1–13.

[151] B. Li et al., "Sharing traffic priorities via cyber–physical–social intelligence: A lane-free autonomous intersection management method in metaverse," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 53, no. 4, pp. 2025–2036, Apr. 2023.

[152] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, Apr. 2020.

[153] D. Coelho and M. Oliveira, "A review of end-to-end autonomous driving in urban environments," *IEEE Access*, vol. 10, pp. 75296–75311, 2022.

[154] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[155] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[156] J.-W. Hu et al., "A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 5, pp. 675–692, May 2020.

[157] X. Chen, J. Liu, J. Wu, C. Wang, and R. Song, "LoPF: An online LiDAR-only person-following framework," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[158] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1364–1384, Apr. 2022.

Zhuopeng Xie received the master's degree in transportation engineering from Southeast University in 2023. He is currently pursuing the Ph.D. degree with the Department of Civil Engineering, The University of Sydney. His research interests include vehicle fleet, multi-vehicle interaction, and RSMRs control.

Mohsen Ramezani received the B.Sc. and M.Sc. degrees in electrical engineering (control systems) from the University of Tehran, Iran, in 2008 and 2010, respectively, and the Ph.D. degree in transportation from the École Polytechnique Fédérale de Lausanne, Switzerland, in 2014. He is currently a Senior Lecturer with the TransportLaboratory, School of Civil Engineering, The University of Sydney. His research interests include point to point transport services, traffic control, data analytics, and traffic flow theory and operations. He is an Australian Research Council (ARC) Discovery Early Career Research Award (DECRA) Fellow. He was a recipient of the 2015 IEEE Intelligent Transportation Systems Society First Prize Best Ph.D. Dissertation.

David Levinson received the Ph.D. degree from the University of California, Berkeley, USA, in 1998. From 1999 to 2016, he was a Faculty Member with the Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, where he held the RP Braun/CTS Chair. He joined the School of Civil Engineering, The University of Sydney, in 2017, where he is currently a Professor of transport. His research interests include transport accessibility and transport-land use interaction.