

Review Article

Intelligence of Autonomous Vehicles: A Concise Revisit

Neelma Naz,¹ Muhammad Khurram Ehsan ,² Muhammad Rizwan Amirzada,³
Md Yeakub Ali ,⁴ and Muhammad Aasim Qureshi⁵

¹National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan

²Faculty of Engineering Sciences, Bahria University, Lahore Campus, Lahore 54000, Pakistan

³Faculty of Engineering and Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan

⁴Department of Electronics and Telecommunication Engineering, Rajshahi University of Engineering & Technology (RUET),
Rajshahi 6204, Bangladesh

⁵Department of Computer Sciences, Bahria University, Lahore Campus, Lahore 54000, Pakistan

Correspondence should be addressed to Md Yeakub Ali; yeakub@ete.ruet.ac.bd

Received 25 February 2022; Revised 16 March 2022; Accepted 21 March 2022; Published 23 April 2022

Academic Editor: Waliullah Khan

Copyright © 2022 Neelma Naz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial intelligence- (AI-) empowered machines are devised to mimic human actions. In the automotive industry, AI plays a significant role in the development of vehicular technology. AI joins hands with the field of mechatronics to assist in the accurate execution of the vehicle functionalities. Autonomous vehicles get the scene information by using onboard sensors such as laser, radar, lidar, Global Positioning System (GPS), and vehicular communication networks. The data obtained is then used for various path planning and control techniques to make the vehicles capable of autonomously driving in complex environments. Autonomous vehicles use very up-to-date AI algorithms to localize themselves in known and unknown environments. AI algorithms are also exploited for perception, path planning, and motion control. A concise review of the state-of-the-art techniques to improve the performance of autonomous vehicles is presented.

1. Introduction

The world is progressing in technology and automation impressively with every passing day. It results in the establishment of smart cities by interconnecting the intelligent Home Area Networks (IHAN), Intelligent Industrial Area Networks (IIAN), Intelligent Vehicular Communication Networks (IVCN), and Smart Grids (SG). The key enabler of IVCN is included in an autonomous vehicle as an intelligent node of the Internet of Vehicles (IoV), Vehicle to Everything (V2X), Vehicle to Vehicle (V2V), and Vehicle to Infrastructure (V2I). People started working on autonomous driving in 1920, and since then, many advancements have been introduced in that domain. But technology still needs human support even with a certain level of intelligence. Current research is focused on introducing vehicles as completely driverless which means no human intervention is required anymore. Intelligent vehicles can move around independently with their decision-making capabilities [1–3].

According to the Society of Automobile Engineers (SAE) [4], automated vehicles are categorized into six different levels. The initial level is level 0; in this level, the driver is responsible for all decisions which means no autonomy. The highest level is level 5, where the vehicle alone is responsible for all driving tasks and decisions (fully autonomous). These levels are presented in Figure 1.

Although many companies such as Uber, Google, and Tesla have invested a lot in the advancement of this technology, the autonomous system is still an active research area due to its very large challenges. A good autonomous system is one that is able to make correct decisions intelligently in real-time scenarios [5–8]. Active researchers are still focusing on devising better algorithms for localization, perception, and detection.

The most important questions the autonomous vehicle technology is built upon are as follows:

- (1) Where am I at the time?

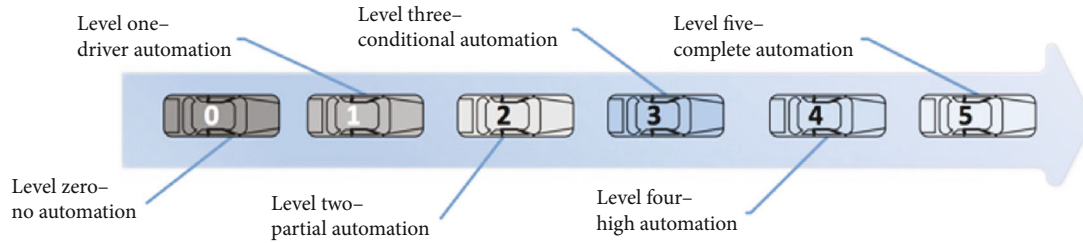


FIGURE 1: Six levels of autonomous vehicles as described by the Society of Automobile Engineers (SAE) [4].

- (2) What is around me?
- (3) What is going to happen next?
- (4) What should be done?

The first question, “Where am I at the time,” is the localization problem. The vehicle must be able to locate/localize itself in the current environment. The next question is getting information about surroundings, and it deals with perception.. Based upon the information perceived/detected, the prediction about the environment falls under the territory of the third question, that is, “What is going to happen next?” Finally, the course of action to be taken by the vehicle is discussed by, “What should be done?” All these fundamental questions are addressed by the use of different sensors and algorithms that make these cars reliable and safe to drive.

Autonomous vehicles sense the world by using various sensors mounted on the vehicle’s assembly as shown in Figure 2. Information received from these sensors is then used to make decision like the safest path to reach the destination considering the optimality with respect to time and distance required to reach the place. To complete the task, more cutting-edge solutions, like localization, object detection and identification path planning, and data fusion received from different sensors, are needed.

With the availability of very powerful computational tools like graphics processing units (GPUs) and a very large amount of data, a subset of artificial intelligence known as deep learning(DL) has gained enormous popularity to solve these problems and to achieve the optimal performance [10]. (DL) algorithms have improved the performance of AVs by ensuring accuracy and fast processing speed. In this paper, different AI technologies being used in autonomous vehicles are reviewed. In Section 2, the generic structure of Autonomous Vehicles (AVs) is discussed. Section 3 discusses the state-of-the-art techniques used for localization. In Section 4, techniques used for path planning are discussed, and in Section 5, a brief discussion on motion controllers is made.

2. Autonomous Vehicle Decision-Making Architecture

Autonomous decision-making is required in AVs to process the observation data received from the sensors mounted on the vehicle. The car’s computer uses these observations to make optimal decisions. These decisions can be computed in two possible ways: either by using the integrated



FIGURE 2: Sensor set-up of MadeInGermany. Externally visible part of sensors [9].

perceive-plan-act method or by end-to-end learning methods. In the end-to-end method, the information obtained from sensors is mapped to control outputs directly without any intermediate steps. An AI-based AV is shown in Figure 3. As can be seen in Figure 3, each step in AVs’ perceive-plan-act method can be implemented either by classical methods with no learning or the latest AI or DL techniques. The end-to-end method of implementation always uses DL techniques. Learning and nonlearning methods can be used together in various arrangements; for example, an object detector based upon deep learning techniques provides input to the A* algorithm that is used for path planning.

An integrated perceive-plan-act method has four components of perception and localization, path planning, behavioral mediation, and motion control, and these components are discussed one by one in this paper.

3. Perception and Localization in AVs

Autonomous vehicles must be able to perceive the environment and be able to locate themselves in the environment correctly. This section reviews various techniques for perception and localization implemented in the literature.

3.1. Hardware for Sensing: Cameras or LiDAR. For better understanding of surroundings, 3D perception is usually preferred. Images taken through cameras can only capture a 2D environment. LiDAR sensors are generally used for

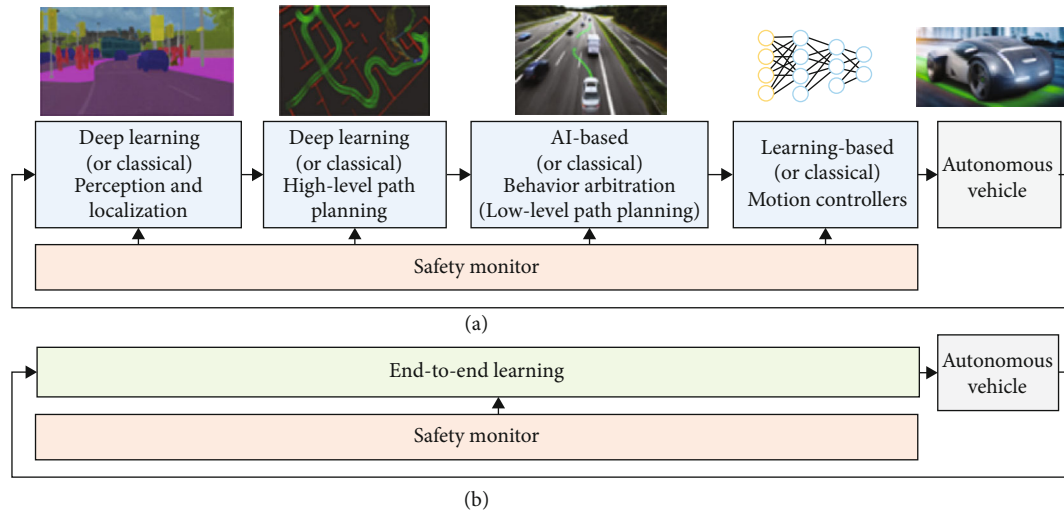


FIGURE 3: Architecture of AVs implanted in (a) the integrated perceive-plan-act method and (b) the end-to-end method [11].

3D perception. LiDAR's performance is measured by its range, rotation/frame rate, field of view, and resolution. Velodyne is also a 3D sensor that has a 360° field view. Autonomous vehicles cannot afford any delays in information communication, so to ensure processing at very high speeds, a range with a minimum of 200 m is required.

The debate of camera usage or LiDAR technology is still a hot topic. For example, Tesla is using its camera system for environment perception while Waymo's vehicle technology is based on LiDAR. Every sensing approach has its own positives and negatives. LiDARs ensure very high resolution and accurate environment perception but show poor performance in the case of bad weather. Moreover, LiDAR technology right now is very expensive. On the other hand, cameras are cheap, but they have very low depth perception and also show poor performance under bad weather conditions. In addition to LiDAR/cameras, ultrasonic sensors and RADAR are also used to enhance the system's perception capability. Waymo makes use of three LiDAR sensors.

3.2. Understanding the Driving Scene. The environments that autonomous vehicles work in are as follows:

- (1) Multiagent environment
- (2) Dynamic
- (3) Unknown
- (4) Stochastic
- (5) Sequential
- (6) Partially observable

All these features of the environment make the task of autonomous driving extremely challenging. Cars should be able to detect every possible scenario like all other agents in the environment, drivable areas, and pedestrians. The task becomes more and more challenging while driving in an

urban area where a variety of objects appear and blockings are very high.

For environment perception, deep neural networks (DNNs) are playing a very important role. Various deep neural network (DNN) algorithms have been proposed for the detection of objects where objects are taken as 2D regions of interest [12–14]. In some other studies, DNNs are used for environment perception based upon pixel-wise segmentation in images [15], 3D bounding boxes in LiDAR [16], and, in some cases, 3D representation of objects in LiDAR + camera-combined data [17]. On a lighter note, for object identification, image data can be useful. However, while estimating 3D positions of the objects as 2D images, depth information of the scene is lost. The two most popular methods of driving scene detection are as follows:

- (1) Semantic and instance segmentation
- (2) Bounding boxes like object detectors

For safe navigation and to understand surrounding environments, semantic and instance segmentation are of utmost importance. For this purpose, several studies using efficient deep learning-based frameworks have been reported recently in the literature. FSNeT, a failure detection framework, is proposed for pixel-level misclassifications in the images [18]. In [19], the transformer-based knowledge distillation framework is proposed for efficient semantic segmentation of road driving scenes. A convolutional neural network method using multiscale attentions is proposed for instance segmentation [20].

3.3. Localization. Localization is the task of finding the vehicle's pose (orientation + position) when it moves in the environment. Localization is an elemental requirement for navigation. It is important to mention here that some of the latest research trends in AVs [21, 22] propose DL-based algorithms that do not need localization and mapping and instead produce end-to-end driving decisions based

upon the sensor information. This is termed as the behavior reflex approach [22].

GPS is most commonly used for localization in autonomous vehicles. GPS data is integrated with other sensor data to compensate for the signal loss in case of any possible outage. Various techniques for sensor fusions exist in the literature. The most commonly used traditional methods for sensor fusion are the Kalman filter, extended Kalman filter, unscented Kalman filter, particle filters, and multimodal Kalman filters [23–26]. A robust cooperative positioning (RCP) [27] scheme to acquire accurate position has been proposed that augments GPS with ultra wide band (UWB). However, the latest trends deal with visual-based localization that uses DL techniques. This method of localization is also called visual odometry (VO). Visual localization is achieved by key point landmarks matching in adjacent video frames. Based upon the vehicle's current frame information, key points are fed as input to the n-point mapping algorithm for the vehicle's pose detection with respect to the previous frame. Accuracy of visual odometry can be enhanced by using deep learning algorithms. These algorithms can affect the key point detector's precision. A DNN is trained for key point distractors learning in monocular VO [28]. The incremental mapping of the environment's structure can also be done by computing the camera pose. This method belongs to SLAM (simultaneous localization and mapping) [29].

SLAM is the act of online map making and localizing the vehicle in it at the same time. A priori information about the environment is not required in SLAM. Because of the enormous improvements of deep learning approaches in image classification and detection, these algorithms are being recommended to enhance traditional SLAM algorithms. Although the deep learning applications in this field are still not mature enough, some studies have proposed to replace classical SLAM blocks with deep learning modules to attain better accuracy and robustness.

To ensure safe navigation, AVs should be able to predict the surrounding environment's motions as well. This is known as scene flow. LiDAR-based estimation of the scene flow is a common approach in literature. Current research proposes to replace the method with DL techniques for automatic learning of the scene flow.

Despite that the research reports much progress in DL-based localization, classical key point matching techniques still dominate VO (visual odometry) mainly because of computational efficiency and easy deployment on embedded devices.

3.4. Perception. For the task of perception, occupancy maps are used frequently. These can also be termed as the Occupancy Grid (OG). It is environment representation in cells. In this method, driving space is divided into a set of cells and the probability of occupancy is calculated for each cell. The technique is very famous in robotics and is now a viable solution in AVs as well.

DL techniques are being used to detect and track the dynamic objects, to probabilistically estimate the occupancy map around vehicle, and to derive the driving scene context.

In the case of driving scene derivation, deep learning is used to label the environment into highway drive, intercity drive, or parking area. Deep learning plays a vital role in OG estimation. It helps in extracting the information from LiDAR data and image processing that is required to populate grid cells. A multitask recurrent neural network is proposed to predict grid maps [30]. Grid maps provide semantic information, occupancies, velocity estimates, and drivable area.

4. Path Planning

Once an AV is able to localize itself in the environment, next comes path planning. Path planning is defined as the ability of autonomous vehicles to find the optimal path between the start position and its destination (desired location) considering the kinematics and dynamic model of vehicles. The path planning process should make the autonomous vehicle capable of calculating the optimal trajectory to ensure the collision-free route while considering all possible obstacles it might come across in the surrounding environment. As mentioned earlier in the paper, autonomous driving is a multiagent problem, so according to the author in [31], the host vehicle must be capable of and apply good negotiation skills with all other users of the road while performing any action like taking a turn or changing lanes. Mission planning is defined as the full pursuit of the generated path by path planning.

Path planning also includes mission planning, motion planning, and behavior planning. Every time the vehicle undergoes a driving experience, a huge amount of data also termed as big data is stored on the server. AVs can use the information contained in the previously stored data to make correct decisions in the future. Route finding algorithms are very complicated because of all the obstacles that cross the vehicle's path. The AV should be capable of identification as well as avoiding these obstacles that make the planning algorithm's task more complicated. The AV must know exactly what to do in a specific driving environment and/or driving situation. For example, for a vehicle driving on the road, it should obey the sequence of waypoints designed by the planning algorithm as shown in Figure 4.

The problem of path planning has been the subject of study for many years and is often divided into two categories, global and local path planning. The techniques used for path planning were divided into four groups: graph search methods, interpolation, numerical optimization, and sampling. Most common motion planning techniques in autonomous vehicles are described below. Figures 5, 6, and 7 show the various techniques as presented in the literature.

4.1. Graph Search-Based Planning Techniques. The autonomous driving path planning techniques work on the basic idea of traversing a complete state space from source point A to goal point B. The state space tells where the objects in the dynamic environment are and is usually represented as a lattice or as an occupancy grid. The graph search algorithms visit the state space in the occupancy grid and return an optimal/nonoptimal solution if it exists or return no solution at all in case it does not exist. The most common search

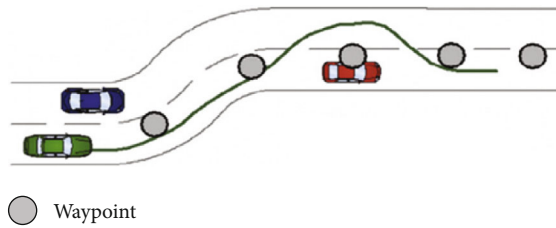


FIGURE 4: Path planning [32].

algorithms implemented for autonomous vehicle path planning are described below.

4.1.1. Dijkstra Algorithm. It is a graph search algorithm that finds the shortest path in a grid or series of nodes. It works well for global path planning in both structured and unstructured environments. In [33], the authors detailed the basic description of the algorithm and how to implement it. However, the algorithm has been implemented in [34] in multivehicle simulations. Despite its advantages, a large number of nodes are needed to be traversed in the vast areas making the algorithm slow. Moreover, the algorithm does not use any heuristics function to optimize the search cost. The path obtained is not continuous, so it is not suitable for real-time scenarios. Figure 6 shows different planning algorithms as they are presented in the literature.

4.1.2. A-Star Algorithm. It is an extended version of the Dijkstra algorithm as it implements heuristics to ensure optimality and a faster node search, reducing the computation time [35–37]. The advantage of the A-star algorithm comes from the fact that to define the node weights, it calculates the cost. It is costly in terms of speed and memory for searching large areas but is very suitable for searching spaces that are mostly known by the vehicle theoretically beforehand. Various modified versions of A-star are being utilized in mobile applications such as the dynamic A* (D*) and anytime repairing A* (ARA*) [38]. For path planning in unstructured spaces and parking spaces, A* using Voronoi cost functions has been implemented in [39]. The winner of the DARPA Urban Challenge, the Boss used the AD* algorithm [40]. Despite its advantages, the path found by the A-star algorithm is not continuous. Moreover, sometimes finding the heuristic rule becomes very complex.

4.1.3. State Lattice Algorithm. The algorithm uses spatiotemporal lattices (including velocity and time dimensions) [41, 42]. Depending upon the maneuver's complexity, the environment is decomposed in a local grid, making it suitable for dynamic environments and local planning. Despite its advantages, the algorithm has to evaluate every feasible solution in the database that makes it computationally expensive.

4.2. Sampling-Based Planning Techniques. This approach works by sampling the state space or configuration space randomly and tries to look for the connectivity inside the space [46]. These techniques try to solve timing restrictions by planning in higher dimensional spaces. However, the

techniques result in suboptimal solutions. Most commonly used sampling-based techniques are the Rapidly-Exploring Random Tree (RRT) and Probabilistic Roadmap Method (PRM). Both are probabilistically complete while RRT is much faster than PRM. RRT is used for online path planning. It executes a random search in the navigation space allowing itself to plan quickly in semistructured spaces. In autonomous vehicles, the algorithm has been used by the MIT team in the DARPA Urban challenge [47]. However, the path resulted is jerky, noncontinuous, and suboptimal. A modified version of this algorithm named RRT* is discussed in [48]. The solution generated is optimal, although at the cost of computational efficiency.

4.3. Interpolating Curve Planning Techniques. Interpolation is defined as the generation of a new set of data points that are in the range of known data points (reference points). These algorithms take previously known waypoints that describe a global roadmap and generate new data points. The points generated ensure a smooth and continuous trajectory and are also beneficial for the dynamic environment in which the AV moves as well as for AV constraints [51]. During path execution, if an obstacle occurs, it generates a set of new data points to avoid it and then continues on the previously planned path. Different techniques are used for curve generation and path smoothing, some of which are reviewed below.

4.3.1. Lines and Circles. Through the interpolation of known waypoints with circular and linear/straight shapes, segments of different road networks can be represented. It is computationally inexpensive and is easy to implement. It guarantees the shortest path for car-type vehicles [52]. However, on the downside, the path generated is jerky, thus making uncomfortable changes between path segments. It also needs global waypoints.

4.3.2. Clothoid. In this technique, the linear change in curvature is used to make the transitions from and to the curves [53]. These types of curves are implemented in road designs and highways. It is suitable for local path planning. On the downside, although the path generated is continuous, it is not smooth because of the linear behavior. It also has time complexity because of the integrals defining the curve. It also needs global waypoints for path planning.

4.3.3. Polynomial. To meet the limitations in the points being interpolated, polynomial curves are commonly implemented [54]. The limitations in the points include angle, curvature, and position. The coefficients of the curve are determined by limitations in the beginning and ending segments or desired values. This method of interpolation is computationally less expensive and is suitable for comfort. However, on the downside, a 4th or higher degree implementation of curves makes the coefficient computation very difficult and challenging.

4.3.4. Bézier. Bézier curves are the parametric curves that are defined by the set of control points. The Bézier curves are related to the Bernstein polynomial. The advantages of using

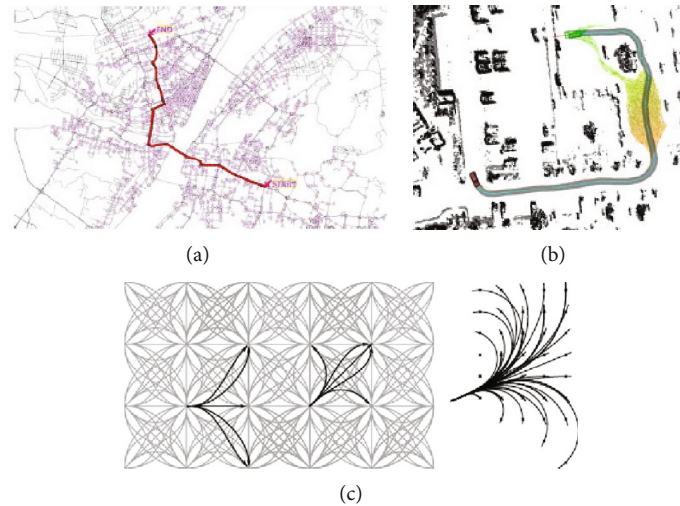


FIGURE 5: Graph search-based planning algorithms as presented in the literature. (a) A global path representation by the Dijkstra algorithm in [43]. (b) Hybrid implementation in the DARPA Challenge by Junior [44]. (c) Motion primitives and lattices as presented in [45].

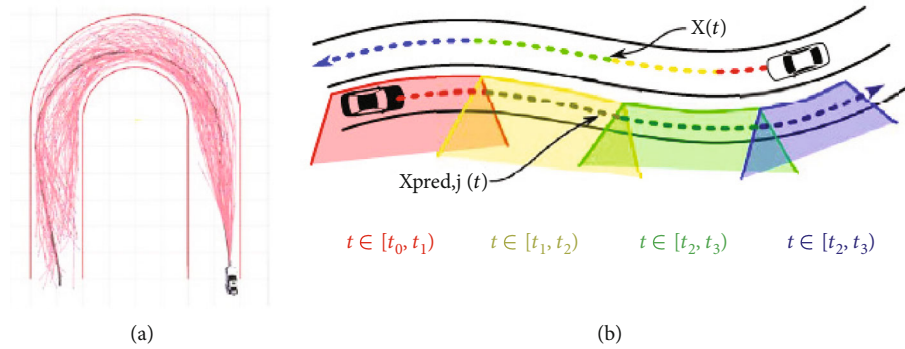


FIGURE 6: Sampling-based and numerical optimization-based planning algorithms as presented in the literature. (a) RRT as presented in [49]. (b) Trajectory optimization by considering the vehicle in the other lane [50].

these curves are their reduced computational cost and intuitive manipulation of the curve because of the control points defining it [55]. It is also possible to continuously concatenate the curves which makes it suitable for comfort. However, with the increase in the curve's degree and computational time, more and more control points need to be evaluated and placed. It also depends upon global waypoints.

4.3.5. Spline. A spline is a piecewise curve that is defined by the polynomials, clothoid or B-splines. A knot is a junction between each subsegment of the curve, and it possesses a higher degree of smoothness constraint between the spline pieces at the joint [56].

4.4. Numerical Optimization Techniques. In path planning, numerical methods are most often used to smooth already computed paths/trajectories as in [57]. The most commonly used technique is the function optimization. To minimize the outcome of variables, this technique finds real valued roots of a function. Using this technique, a plan can be generated by taking ego-vehicle limitations, road constraints,

and other users on the road into account. On the downside, at each motion state, the optimization of the function needs to take place, because of which, the optimization needs to be stopped at a given time. This planning technique also depends on global waypoints.

4.5. Deep Learning-Based Techniques. Latest research shows increased interest in the application of DL techniques in path planning. The two most discussed DL techniques in the path planning scenario are imitation learning and planning based upon reinforcement learning. The fundamental task of imitation learning (IL) [58] is to imitate the human driver's behavior. The human driver's behavior is recorded in the form of big data, and then, a convolutional neural network (CNN) is used to make the vehicle learn, how to plan from imitation. Imitation learning is also termed as the inverse of reinforcement learning [59, 60]. This method uses the human driver's behavior to learn how to maximize reward functions and then to generate driving trajectories just like humans. The DRL method is also used to plan the path. In this method, the agent learns driving trajectories in a simulator environment [61]. On the basis of a transfer

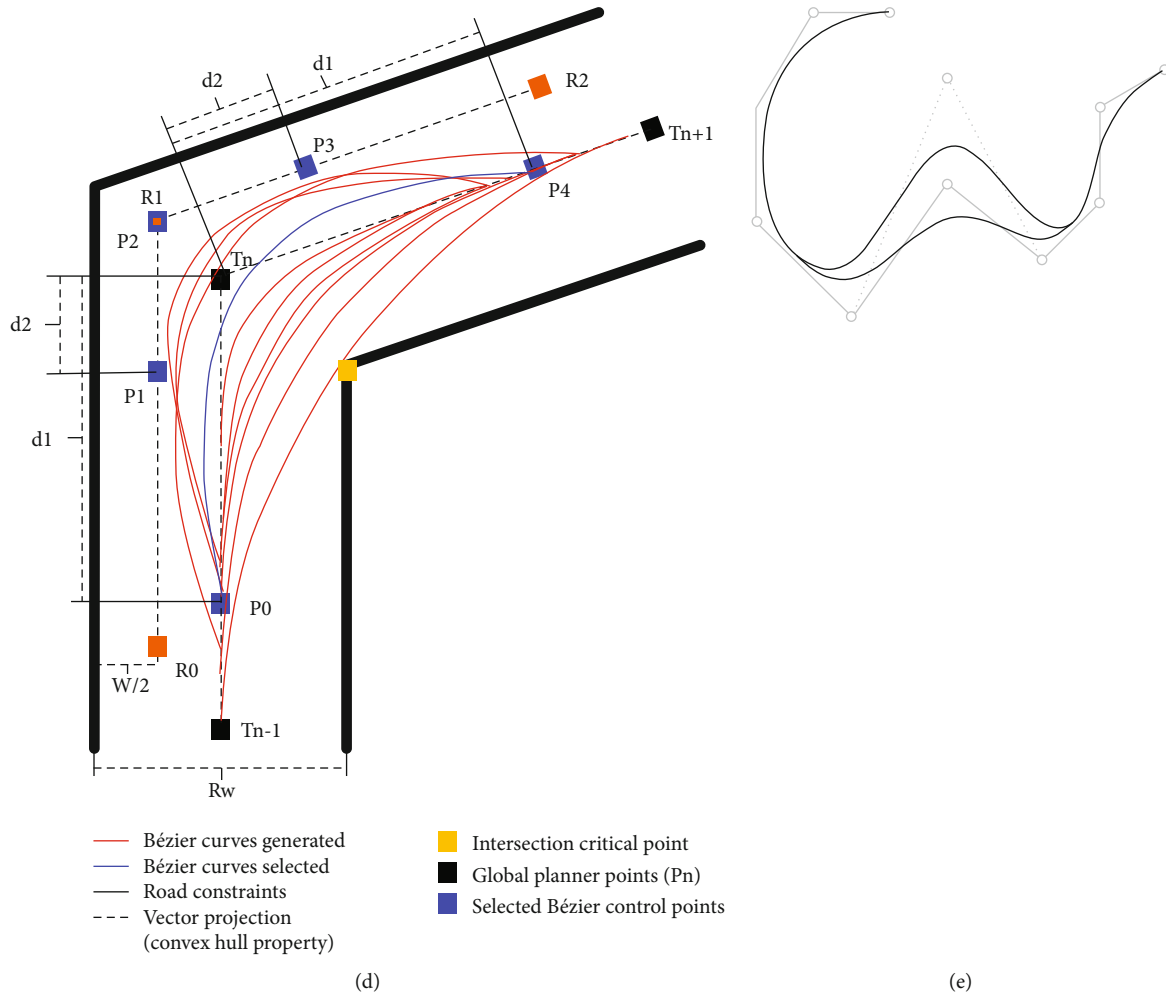


FIGURE 7: Interpolating curve-based planning algorithms as presented in the literature. (a) Optimal path to turn the vehicle around as proven in [52]. (b) Planning a turn for the Audi TTS from Stanford [53]. (c) Different motion states planned with polynomial curves as presented in [54]. (d) Evaluation of several Bézier curves in a turn, as shown in [55]. (e) Spline behavior when a knot changes place as presented in [56].

model, the real environment model is transformed into a virtual one. Both of these methods have their own advantages and disadvantages. IL has the advantage of being trained on real-world data, but as data is rare on corner cases (e.g., driving off the lanes), the trained network might give errors when it handles unseen scenarios. On the other side, DRL shows good performance in simulations, but the performance is not that good under real-world scenarios. Although the use of deep learning-based techniques to perform perception, localization, path planning, and control is getting much attention, it has also increased concerns of transparency and accountability in autonomous vehicles because of the black box nature of deep neural networks. So to build the trust on these deep frameworks, explainable AI (xAI) is the field that has gained researchers' interest in recent years. Explanations generated either in numerical form or textual forms or in the form of heat/saliency maps (visual form) provide insights into the decision-making process of autonomous vehicles. Various approaches are being used to produce these explanations. An imitation learning-

(IL-) based agent equipped with an attention model is proposed [62]. The attention model helps to understand regions of images considered important in the decision-making process.

5. Motion Controllers/Act

The task of calculating steering commands (longitudinal and lateral) comes under the territory of the motion controller. The motion controller makes use of learning algorithms as part of an incomplete entity, or they work as a complete entity as an end-to-end controller to generate the steering commands from sensory data. Traditional controllers work on a model composed of fixed parameters. Learning controllers use the training information and data to make themselves capable of learning their models over time. The more information gathered, the more accurate the system model is. Commonly used learning controllers are the iterative learning control (ILC) [63] and model predictive control (MPC) [64]. ILC works efficiently for controlling systems

that work in repetitive mode, e.g., tracking a defined trajectory in autonomous vehicles. MPC finds the appropriate control actions by solving the optimization problem. MPC also helps us in the prediction of disturbances and uncertainties in the system leading to optimal solutions. The data for training is mostly available in the form of the vehicle's past states and observations. A use of CNN can then be made by training it to find the dense occupancy grid map. This map is then passed to the cost function of MPC to find the optimal trajectory to be followed by the vehicle over a finite horizon. The maximum advantage of these learning controllers can be achieved as they make use of a model-based control as well as learning algorithms. Deep learning-based techniques have gained much importance in the motion control of autonomous vehicles [30, 65]. A visual attention model is used to train an end-to-end (from images to control commands) convolutional neural network model [66]. These attentions learned by the attention model identify the image regions influencing the network's output. To generate textual explanations, an attention-based video-to-text model is used. Finally, the controller's attention map and explanations are aligned to ground the explanations in the image regions that mattered to the controller. In most existing works on autonomous driving, three main modules of autonomous vehicles, i.e., sensing, decision making, and motion controlling, have been studied separately. However, the power of DNN can also be exploited for joint optimization of sensing, decision making, and motion control [67].

6. Conclusion

The development of intelligent and efficient algorithms for the safe operation of AVs is one of the key issues in vehicle design. This work presents a complete layout of an autonomous vehicle. A survey of various state-of-the-art AI algorithms used by the AVs to achieve the best possible and optimal solutions to the problems of perception, localization, path planning, and motion control has been presented. Although the field of AVs is vast and involves a wide variety of challenges to address, this very challenging nature of the problem makes endless research opportunities in this field.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] P. J. Jin, D. Fagnant, A. Hall, and C. M. Walton, *Emerging Transportation Technologies White Papers*, University of Texas, TRB, 2015.
- [2] W. U. Khan, E. Lagunas, A. Mahmood, S. Chatzinotas, and B. Ottersten, "When RIS meets GEO satellite communications: a new optimization framework in 6G," 2022, <https://arxiv.org/abs/2202.00497>.
- [3] M. K. Ehsan, A. A. Shah, M. R. Amirzade et al., "Characterization of sparse WLAN data traffic in opportunistic indoor environments as a prior for coexistence scenarios of modern wireless technologies," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 347–355, 2021.
- [4] Mobilus, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE International, 2018.
- [5] H. Khayyam, B. Javadi, M. Jalili, and R. N. Jazar, *Artificial Intelligence and Internet of Things for Autonomous Vehicles*, Springer International Publishing, Cham, Switzerland, 2020.
- [6] M. K. Ehsan, "Performance analysis of the probabilistic models of ISM data traffic in cognitive radio enabled radio environments," *IEEE Access*, vol. 8, pp. 140–150, 2020.
- [7] A. Mahmood, Y. Hong, M. K. Ehsan, and S. Mumtaz, "Optimal resource allocation and task segmentation in IoT enabled mobile edge cloud," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13294–13303, 2021.
- [8] M. K. Ehsan and D. Dahlhaus, "Statistical modeling of ism data traffic in indoor environments for cognitive radio systems," in *in 2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pp. 88–93, Moscow, Russia, 2015.
- [9] D. Göhring, D. Latotzky, M. Wang, and R. Rojas, "Semiautonomous car control using brain computer interfaces," in *Intelligent Autonomous Systems 12*, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds., pp. 393–408, Springer, Berlin, Heidelberg, 2013.
- [10] X. Chen, Y. Chen, and H. Najjaran, "3D object classification with point convolution network," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 783–788, Vancouver, BC, Canada, 2017.
- [11] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.
- [13] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, Salt Lake City, UT, USA, 2018.
- [14] R. Girshick, "Fast R-CNN," in *in 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, 2015.
- [15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [16] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," in *in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, Salt Lake City, UT, USA, 2018.
- [17] C. Qi, W. Liu, C. Wu, H. Su, and L. Guibas, "Frustrum Point-Nets for 3D object detection from RGB-D data," in *in 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, Salt Lake City, UT, USA, 2018.
- [18] Q. Rahman, N. Sunderhauf, P. Corke, and F. Dayoub, "Fsnet: A failure detection framework for semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3030–3037, 2022.

- [19] R. Liu, K. Yang, H. Liu, J. Zhang, K. Peng, and R. Stiefelwagen, "Transformer based knowledge distillation for efficient semantic segmentation of road-driving scenes," 2022, <https://arxiv.org/abs/2202.13393>.
- [20] W. Gaihua, L. Jinheng, C. Lei, D. Yingying, and Z. Tianlun, "Instance segmentation convolutional neural network based on multi-scale attention mechanism," *PLoS One*, vol. 17, no. 1, article e0263134, 2022.
- [21] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2722–2730, Santiago, Chile, 2015.
- [22] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6526–6534, Honolulu, HI, USA, 2017.
- [23] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.
- [24] H. Qi and J. Moore, "Direct kalman filtering approach for gps/ins integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 687–693, 2002.
- [25] G. Wang, Y. Han, J. Chen et al., "A GNSS/INS integrated navigation algorithm based on Kalman filter," *IFAC PapersOnLine*, vol. 51, no. 17, pp. 232–237, 2018.
- [26] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, Lake Louise, AB, Canada, 2000.
- [27] Y. Gao, H. Jing, M. Dianati, C. M. Hancock, and X. Meng, "Performance analysis of robust cooperative positioning based on gps/uwb integration for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, 2022.
- [28] D. Barnes, W. Maddern, G. Pascoe, and I. Posner, "Driven to distraction: self-supervised distractor learning for robust monocular visual odometry in urban environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1894–1900, Brisbane, QLD, Australia, 2018.
- [29] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: a survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.
- [30] M. Schreiber, V. Belagiannis, C. Glaser, and K. Dietmayer, "A multi-task recurrent neural network for end-to-end dynamic occupancy grid mapping," 2022, <https://arxiv.org/abs/2202.04461>.
- [31] S. S. Shwartz, S. Shammah, and A. Shashua, "Safe, multiagent, reinforcement learning for autonomous driving," *CoRR*, 2016.
- [32] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.
- [33] J. Y. Hwang, J. S. Kim, S. S. Lim, and K. H. Park, "A fast path planning by path graph optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 33, no. 1, pp. 121–128, 2003.
- [34] R. Kala and K. Warwick, "Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization," *Journal of Intelligent and Robotic Systems*, vol. 72, no. 3–4, pp. 559–590, 2013.
- [35] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.
- [36] W. U. Khan, E. Lagunas, A. Mahmood, S. Chatzinotas, and B. Ottersten, "Integration of backscatter communication with multi-cell NOMA: a spectral efficiency optimization under imperfect SIC," 2021, <https://arxiv.org/abs/2109.11509>.
- [37] M. K. Ehsan and D. Dahlhaus, "A framework for statistical characterization of indoor data traffic for efficient dynamic spectrum access in the 2.4 ghz ism band," *International Journal of Digital Information and Wireless Communications (IJDIWC)*, vol. 5, no. 4, pp. 210–220, 2015.
- [38] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," *Artificial Intelligence*, vol. 172, no. 14, pp. 1613–1643, 2008.
- [39] J. Ziegler, M. Werling, and J. Schroder, "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 787–791, Eindhoven, Netherlands, 2008.
- [40] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments: part i," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1063–1069, Nice, France, 2008.
- [41] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *Proceedings of 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, 2005.
- [42] W. U. Khan, M. A. Jamshed, A. Mahmood, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Backscatter-aided noma v2x communication under channel estimation errors," 2022, <https://arxiv.org/abs/2202.01586>.
- [43] Q. Li, Z. Zeng, B. Yang, and T. Zhang, "Hierarchical route planning based on taxi GPS-trajectories," in *2009 17th International Conference on Geoinformatics*, pp. 1–5, Fairfax, VA, USA, 2009.
- [44] M. Montemerlo, J. Becker, S. Bhat et al., *Junior: the Stanford entry in the Urban Challenge*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [45] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1879–1884, St. Louis, MO, USA, 2009.
- [46] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: a review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [47] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.
- [48] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 7681–7687, Atlanta, GA, USA, 2010.
- [49] J. H. Jeon, R. V. Cowlagi, S. C. Peters et al., "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *2013 American Control Conference*, pp. 188–193, Washington, DC, USA, 2013.
- [50] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha - a local, continuous method," in *2014 IEEE*

- Intelligent Vehicles Symposium Proceedings*, pp. 450–457, Dearborn, MI, USA, 2014.
- [51] L. Labakhua, U. Nunes, R. Rodrigues, and F. S. Leite, *Smooth trajectory planning for fully automated passengers vehicles: spline and clothoid based methods and its simulation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [52] J. A. Reeds and L. A. Shepp, “Optimal paths for a car that goes both forwards and backwards,” *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [53] J. Funke, P. Theodosis, R. Hindiyeh et al., “Up to the limits: autonomous Audi TTS,” in *2012 IEEE Intelligent Vehicles Symposium*, pp. 541–547, Madrid, Spain, 2012.
- [54] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, “A real-time motion planner with trajectory optimization for autonomous vehicles,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 2061–2067, Saint Paul, MN, USA, 2012.
- [55] A. Valera, F. Valero, M. Vallés, A. Besa, V. Mata, and C. Llopis-Albert, “Navigation of autonomous light vehicles using an optimal trajectory planning algorithm,” *Sustainability*, vol. 13, no. 3, p. 1233, 2021.
- [56] R. T. Farouki, *Pythagorean-hodograph curves: algebra and geometry inseparable, geometry and computing*, Springer, 2008.
- [57] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path planning for autonomous vehicles in unknown semi-structured environments,” *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [58] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, “A fast integrated planning and control framework for autonomous driving via imitation learning,” *Dynamic Systems and Control Conference*, vol. 9, 2018.
- [59] W. U. Khan, T. N. Nguyen, F. Jameel et al., “Learning-based resource allocation for backscatter-aided vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [60] W. U. Khan, X. Li, A. Ihsan, M. A. Khan, V. G. Menon, and M. Ahmed, “Noma-enabled optimization framework for next-generation small-cell iov networks under imperfect sic decoding,” *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [61] A. I. Panov, K. S. Yakovlev, and R. Suvorov, “Grid path planning with deep reinforcement learning: preliminary results,” *Procedia Computer Science*, vol. 123, pp. 347–353, 2018.
- [62] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, “Explaining autonomous driving by learning end-to-end visual attention,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1389–1398, Seattle, WA, USA, 2020.
- [63] Y. Zhao, F. Zhou, Y. Li, and Y. Wang, “A novel iterative learning pathtracking control for nonholonomic mobile robots against initial shifts,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, 2017.
- [64] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli, “Repetitive learning model predictive control: an autonomous racing example,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2545–2550, Melbourne, VIC, Australia, 2017.
- [65] Y. Du, J. Chen, C. Zhao, C. Liu, F. Liao, and C.-Y. Chan, “Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning,” *Transportation Research Part C: Emerging Technologies*, vol. 134, article 103489, 2022.
- [66] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, “Textual explanations for self-driving vehicles,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 563–578, Munich, Germany, 2018.
- [67] L. Chen, Y. He, Q. Wang, W. Pan, and Z. Ming, “Joint optimization of sensing, decision-making and motion-controlling for autonomous vehicles: A deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, 2022.