MSA UNIVERSITY ENGINEERING JOURNAL

Behavior Planning for Autonomous Driving: Methodologies, Applications, and Future Orientation

Nada S. Kassem [1,a], **Sameh F. Saad** [2,b] **,and Yasser I. Elshaaer**[3,c,*]

[1] Arab Academy for Science, Technology & Maritime Transport, Smart Village, Giza, Egypt
[2] The knowledge Hub Universities, Coventry University Branch, New Administrative Capital, Cairo, Egypt
[3] Arab Academy for Science, Technology & Maritime Transport, Smart Village, Giza, Egypt
E-mail: [a]N.Kassem33756@student.aast.edu , [b]sameh.eid@tkh.edu.eg, [c]y_elshaer@aast.edu

Abstract

Decision-making is a crucial task for autonomous driving. Taking the wrong decision might cause a cartographical accident. Decision-making implies planning the appropriate behavior based on the current state of the vehicle and its environment. The paper aims to review the current state of the art of behavior Planning for autonomous driving. A basic categorization for behavior planning algorithms is introduced. Besides, a review of different algorithms is discussed with a comparison between different algorithms from the point of view of the suitable scenario. Finally, promising research topics are discussed.

Keywords: Autonomous driving, Decision making, Behavior Planning.

ENGINEERING JOURNAL Volume # Issue #
Received Date Month Year
Accepted Date Month Year
Published Date Month Year
Online at https://engj.org/
DOI:10.4186/ej.20xx.xx.x.xx

1. Introduction

Autonomous vehicles are highly technological products with embedded electronics and intelligent algorithms [1]. It aims to avoid the problem of collisions that are made due to human errors, save energy, and allow passengers to have comfortable traveling experiences [2]–[5]. In 2016 the Society of Automotive Engineers (SAE) defined different levels of autonomy [6]. It considers the conventional cars that have no autonomous features as zero-level. While cars that can handle one task at a time such as braking in an emergency could be considered level-one cars. Besides, cars with two automated features or two automated control tasks are considered level two cars. Moreover, level three cars can handle different circumstances, but still needs human intervention. Level four cars can drive autonomously in certain environments and scenarios without human intervention. Finally, cars that can drive autonomously in any environment and scenario belong to level five of autonomy. DARPA Grand Challenge started in 2004 and highlighted the importance of investigating more research on autonomous driving and encourages researchers to investigate this field.[7]–[8].

Many architectures for autonomous cars are developed by researchers in the last few years. Some of them consider autonomous cars to have three main modules which are: perception, decision-making, and action execution [9]. Other researchers consider autonomous driving systems to consist of global localization and planning modules, environment, self-perception, planning, and control [10]. Another architecture for autonomous driving considers the car to have four main modules: perception, planning, low-level collision avoidance, and control [11]. The most detailed architecture by combining all their point of view is the architecture shown in fig. 1. Autonomous cars can be divided into five main modules which are sensors, perception, mapping, planning and control, and actuators. In the first module, sensors perceive the surrounding environment in addition to perceiving the ego vehicle status. The sensor's output is the input to the next module, the perception module. The perception module carries out the task of detecting and classifying a different object on the road and localizes the ego vehicle to know its exact status. In the third module, mapping helps in making a map of the car relative to its surroundings. The fourth module is the planning and control module which is responsible for executing different planning tasks besides vehicle control. Finally, the actuators are responsible for executing the different vehicle commands.

The autonomous driving planning goal is to find suitable control commands to drive a vehicle from a certain start point to a goal point [12]. It also aims to establish its task safely and efficiently [1]. The planning and control module consists of four subparts which are mission planning, Behavior Planning (BP), local planning, and vehicle control as shown in fig. 1 [7], [13]. Mission planning is responsible for generating a sequence of waypoints based on the user-specified destination. While BP is responsible for decision making which yields the generation of maneuver specifications. Besides motion planning is responsible for trajectory generation. Finally, the control part is responsible for the execution of the generated trajectory.

In real driving scenarios, taking a driving decision is a crucial process. Thus, autonomous cars need to adopt some sort of decision-making algorithm to deal with the dynamic surrounding environment [14]. Although significant achievements have been established in the field of autonomous driving, especially after the tangible progress in perception technology, BP is still one of the biggest confrontations in the field of autonomous driving [15], [16].

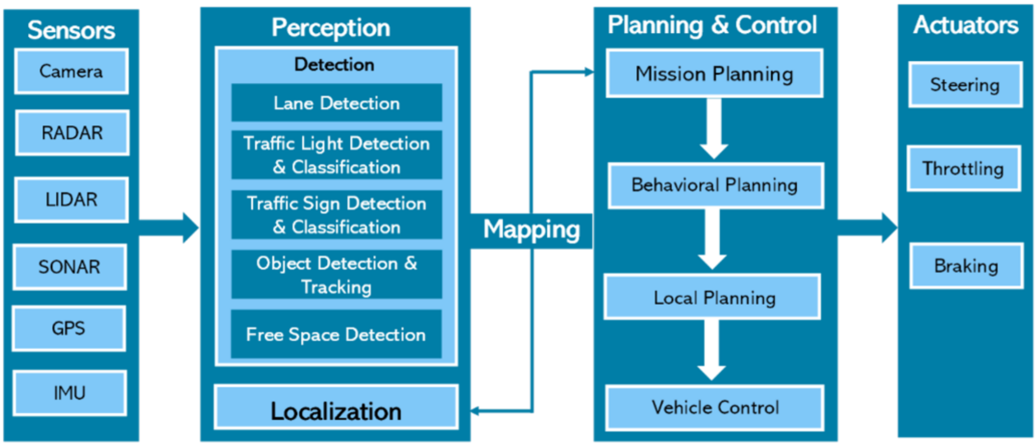

Fig. 1 Autonomous vehicle architecture

Many published papers reviewed BP for autonomous vehicles. A general review of BP is tackled by [13] who discussed and compared only two popular decision-making methodologies. These methodologies are Finite State Machines (FSM) and Partially Observable Markov Decision Processes (POMDP). Although they highlighted one of the main disadvantages of FSM which is its inability to deal with complex environments and how it is solved in the previous research, their study was vague, and they did not compare to any other used methodologies to validate its effectiveness. On the contrary [13], discussed the advantages and disadvantages of the different existing methodologies. The drawback of their study was that they did not adopt any categorization approach to classify the existing methods. Another survey on BP was presented by [5]. Their survey in behavioral planning was summarized and did not account for each methodology application. In contrast to [5], a detailed review of BP is introduced by [17]. The topic was discussed comprehensively as they tackled the categorization of the different existing methodologies. Moreover, they have also explained the advantages and disadvantages of each algorithm and tackled the suggested future directions in a detailed manner. Besides, they mentioned the applications for the different methods. Nevertheless, they did not tackle the appropriateness of the methodologies with the different scenarios and applications. Lastly, BP methodologies are studied by [18]. Although they compared the algorithms from the point of view of the performance, they did not mention or refer to any mathematical equations for the method used to measure the performance's parameters. Moreover, they categorize the methodologies by different points of view nevertheless they did not mention all the existing algorithms under each category and settle for a limited number of examples for each category. Although BP is a critical module in autonomous vehicles, it is apparent that the existing surveys for this point are still limited. Considering the importance of the BP module for autonomous vehicles, a framework for BP is introduced, a detailed comparison between the BP methodologies is presented, and further details about the suitable algorithm for each scenario are tackled.

The paper summarizes as follows. Section 2 discusses the behavior planning framework. Section 3 categorizes the different methodologies and compares them. Section 4 discusses the applications of the existing methodologies. Section 5 discusses and analyzes the existing methodologies. In section 6 the paper is concluded, and future orientation is discussed.

2. Behavior Planning Framwork

Previous papers describe different frameworks for BP and decision-making modules. One of these frameworks is presented by [17]. The framework consists of three layers. The first layer is the input layer. It consists of three levels. The first level is the driving environment with all static and dynamic objects in it. The second level is the status of the ego vehicle. While the third level is the HD Maps level. The second layer is the decision-making layer, which is represented by the road scenario. The road scenario can be merging, roundabout, urban intersection, etc. The last layer is the output layer. The authors classify these layers into two levels. The first level is high-level behaviors, including merging, overtaking, emergency braking, etc. while the second level is low level behaviors including angular and longitude velocities and accelerations. Another framework for decision-making is described by [18]. They stated that the input to the decision-making module is perception and vehicle prediction. The perception stage informs the decision-making module about the surrounding vehicle states. While the vehicle prediction forecast the future trajectory of other vehicles on the road.

By observing previous works in BP, a framework is adopted to describe the Behavior Planning Algorithm (BPA) as shown in fig.2. The framework has two main layers which are the input layer and the output layers. The input to BPA comes from the perception, mapping, and mission planning modules. The perception module fed the BPA with information about the dynamic objects' status including their relative position, speed, acceleration, etc. It can also feed it with a predicted path or intention of these dynamic objects. Besides, the perception module is also responsible to share some information about the static objects' relative positions. The mapping module can provide the BPA with some information about the static object as well. Moreover, the localization module in the perception module provides the BPA information about the current status of the vehicle including its location, velocity, acceleration, etc. Mission planning is responsible to provide the BPA knowledge about the car waypoints. The BPA takes all these inputs and translates them to a suitable output depending on the road scenario. For example, the output can be lane-keeping or lane-changing if the ego vehicle is in a lane-changing scenario. It can also be a car's acceleration or deceleration command for car following scenario.

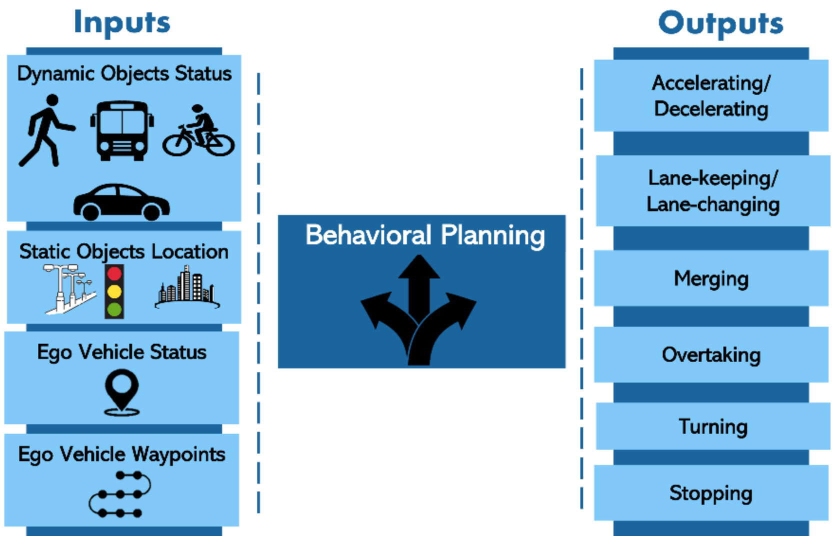

Fig. 2. BP Framework

3. Existing Methodologies

The current BP methodologies have been subjected to many classifications based on different points of views.

3.1. Classification of Methodologies

BP methodologies can be classified into two main categories classical-based and learning-based methods [14], [17], [18]. They have also classified the BP methods based on four main categories: knowledge-based, heuristic-based, approximate reasoning-based, and human-like methods. Another method of classification of BP is a classification approach according to human capabilities. These capabilities are rationality, obeying the rules, cognitive abilities, and learning [18]. The existing methodologies is classified into rule-based, cooperative-based, probabilistic-based, game theoretic-based, and learning-based approaches [19]. One last method of classification and the most recent one is classifying the existing methods into rule-based methods, utility-based methods, probabilistic-based methods, game theory-based methods, learning-based methods, learning-based methods, and cooperative-based methods [20]. After reviewing all these classification approaches, it is decided to adapt to the most general, simple, and comprehensive one that classifies the methodologies into classical approaches and learning approaches as shown in fig.3.

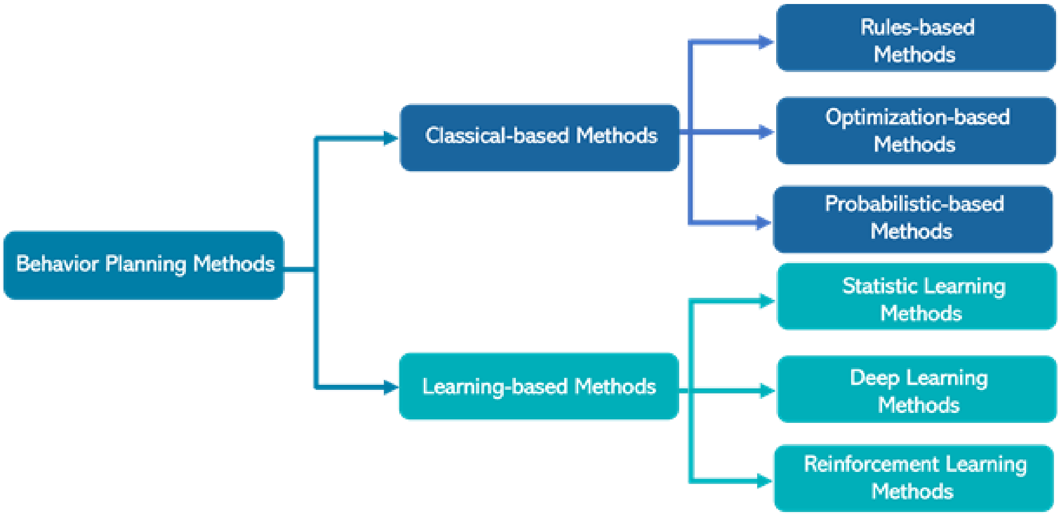

Fig. 3. BP Methods

3.1.1. Classical-based Methods

These methods are based on mathematical modelling thus, they do not need datasets and do not depend on learning. These methods have three subcategories which are rules-based methods, optimization-based methods, and probabilistic-based methods.

Rules-based methods are methods that consist of predefined scenarios and statements so that the vehicle can select the appropriate behavior based on these scenarios and statements which are often represent the status of the vehicle and the status of its surroundings including the behavior of the dynamic objects around it. The most popular example of this method is the finite state machine (FSM) method. FSM is the most popular method for BP starting from DARPA competition till now. While the optimization-based method has a utility function that improves the vehicle's performance based on a reward or a punishment. One example of these methods is Model Predictive Control (MPC) methodology. The most popular approach that depends on these methods is a game-theory-based approach. Lastly, probabilistic-based methods that depend on mathematical probability and statistics concepts. The most famous approach under this methodology is Probabilistic Graphical Model (PGM).

3.1.2. Learning-based Methods

Methods that depend on artificial intelligence technology to grant the vehicle select the appropriate behavior. These methods rely on human-made datasets. The autonomous vehicle in this case depends on learning from human behaviors. They are classified into three main subcategories which are statistical learning, deep learning, and reinforcement learning method.

Statistical learning methods are methods that rely on machine learning technology to make suitable decisions. Support Vector Machines (SVM), AdaBoost, and random forest are examples of algorithms for BP in autonomous cars. Mostly in the case of BP, this decision is taken on basis of certain parameters such as efficiency, and safety. Deep learning methods utilize neural networks in selecting the appropriate behavior. This approach depends totally on datasets in which its input is mostly images and based on these cognitive data, it takes the appropriate decision. It is similar to statistical learning methods except in their structure and how the features are extracted from the input dataset. The most common approach for BP based on deep learning is an end-to-end approach. In which the neural network is fed with sensors output and steering, braking, and throttling commands are obtained. This method opens the door for imitating human-driver behavior. The end-to-end approach has an architecture that is different from the architecture of the autonomous car in the introduction section as there are no intermediate blocks between the sensor inputs and the output of the actuator. Finally, reinforcement learning methods stand on the ability of an agent or autonomous vehicle to learn by doing. In other words, to learn based on a punishment or reward function. In the last few years, this method is one most popular method in the field of BP due to its ability to handle the interaction between the ego vehicle and its surroundings. Most importantly, it can account for environmental uncertainties arise from sensors' noise and inaccurate predictions of the surrounding car's intentions.

3.2. Advantages & Disadvantages

The existing methodologies possess different strengths and weaknesses points from different perspectives which are their simplicity in implementation, the computational power needed, the computational time needed, and their ability to generalize to different scenarios as stated in Table 1.

3.2.1. Rules-based Methods

These methods can be easily implemented depending on the if-else programming logic and the algorithm is rational as human thinking that it is suitable for real-time operation. On the other side, the algorithm is hard to generalize to different scenarios and cannot handle unplanned situations.

A Hierarchal Finite State Machine (HFSM) algorithm is implemented by [1]. In which they were able to handle the collisions using a cost assessment model to select the appropriate behavior. They built an HFSM with four states which are cruising, overtaking, back-lane, and braking. The HFSM is proven to respond fast to any environmental changes and reduce the computational power needed by 39%.

For the lane change scenario, a three-layer HFSM is built [14]. The first layer is responsible for identifying the scene of the ego vehicle based on its position relative to the surroundings. The second layer is responsible for calculating the cost for all the possible behaviors. The calculated cost is a combination of three parameters which are safety, lane idleness, and efficiency. The last layer selects the appropriate behavior based on the calculated cost. The method is proven to take the most appropriate behavior with the lowest possible cost.

A four states FSM is implemented for lane-keeping and lane-change scenarios [21]. The four states for the algorithm are lane keeping, leading vehicle following, aborting, and overtaking. The algorithm is executed in three steps. The first step is identifying the safe regions using Artificial Potential Field (APF). The second step is selecting the suited maneuver by using FSM. The third step is generating an intermediate reference position and velocity for the trajectory planning to consider. The algorithm is tested within different vehicle velocities and was efficient, but the method is not suitable for complex environments due to its overoptimism.

A whole planning and control algorithm is established for lane-keeping and lane-change scenarios in urban environments [3]. A five-mode FSM is built for BP handling. The five modes are ready mode, stop mode, lane keeping and changing mode, avoiding obstacles mode, and emergency handling mode. The algorithm has proven its efficiency in different weather and ground conditions in urban environments.

An FSM is experimented with for a T-intersection scenario with three states: passing, yield, and acceleration [22]. For the transition from one state to another, information like position and speed that comes from connected cars is used. To avoid collisions, the speed profile is continuously modified according to the position of the dynamic objects that surrounds the ego vehicle. The algorithm has proven to make the vehicle navigate safely through the intersection.

3.2.2. Optimization-based Methods

Although, these methods account for the interactions between other dynamic objects in the road, its implementation is computationally costly specially for dense environments.

A Bayesian game theory model is implemented as a trial to achieve human-like behavior generation [23]. The methods were compared to state of art methods and it is proven to generate more complex and human-like performance. It was also tested by using the Turing test to assure its similarity to human behavior. Most of the participants of the test cannot distinguish between human driving and model driving.

A game theory-based methodology has been experimented with in a congested, urban scenario [24]. The experiment is conducted by performing a lane change in an intersection. A comparison between human behavior and autonomous vehicle behavior is taken under consideration and the results show that 83.3% of the behavior of the autonomous car is similar to human behavior.

A Nash equilibrium and Stackelberg game theory methodology are used for decision-making [25]. Ride comfort, drive safety, and travel efficiency was taken under consideration in decision-making It is tested for lane change scenario. It is proven to be able to achieve human-like behavior.

A cost-based hierarchal behavior-based methodology is implemented [15]. The algorithm is implemented for lane change in both highway and urban environments. It can perform any change and has parking capabilities. The algorithm is tested in a simulation environment and has proven its efficiency.

A game theory-based algorithm is used in merge scenarios in a congestion environment [26]. To achieve a semi-optimal policy the algorithm is combined with Monto Carla reinforcement learning methodology. The algorithm is combined with a rule-based methodology. The difference between the methods is in the response time. The game theoretic method is proven to have a faster response than a rule-based method. For merging scenarios, a game tree search method is tested [27]. The methodology was experimented with in a simulation environment and has proven an ability to run in real-time.

3.2.3. Probabilistic-based Methods

Similar to optimization-based methods, these methods need high computational power although it has the ability to interact and predict the behavior of other dynamic objects in the road.

A Factor Graph (FG) combined with Gaussian Mixture Model (GMM) is used for interaction with surrounding vehicles in a merge scenario [28]. FG is proven to have better performance than PGM from the point of view of safety, and the average distance between vehicles. A Sequential Level Bayesian Decision Network (SLBDN) is used for decision-making in highway scenarios. To decrease the measurement uncertainties, the Extended Kalman Filter is used. The method is proven to have satisfactory results even in risky situations [29].

3.2.4. Statistic Learning-based Methods

It is powerful from the point of view of learning from small size datasets and does not consume high computational power. However, it cannot generalize its behavior for all environments, and it acts only based on what it learns.

Decision-making for lane change is implemented using three different statistic learning algorithms which are decision tree, random forest, and Artificial Neural Network (ANN). Their performance is compared with fuzzy logic algorithms and all these algorithms are proven to be better than a fuzzy logic algorithm [30] . A Support Vector Machine is used to solve the problem of the lane change. The inputs of the algorithm are a benefit, safety, and tolerance cost, and based on these parameters the decision is taken. The algorithm has proven to surpass the FSM performance [2].

3.2.5. Deep Learning-based Methods

These methods is known for its ability to imitate human behaviors, and its performance can be easily judged. It can work in more than one scenario but needs huge datasets and high computational power to be trained. Sensor’s uncertainties affect the performance of these algorithms dramatically.

An end-to-end approach is tackled for lane following, avoiding collisions, and traffic signals handling. The network takes input images and velocity measurements as inputs and generates steering, braking, and throttling commands. The network consists of seven elements which are an encoder, two decoders, a traffic light state classifier, flatten module, a velocity module, and a driving module. The approach is tested on the CARLA simulator in benchmarked datasets. It implies a high success rate and ability to handle traffic lights [31].

A deep cascaded neural network is constructed to handle multiple tasks simultaneously (Hu et al., 2021). The input to the network is images from front cameras and the output is steering, braking, and throttling commands. The network consists of one CNN and three LSTM modules. The methodology is tested on five roads which are: country road, freeway, mountain road, tunnel, and congested road. It is compared to two state-of-the-art methods such as NVIDIA (NVD) and the Weighted Loss Function (WLF) method. The proposed method has proven to have the least Root Mean Square Error (RMSE) in acceleration and braking values.

A deep learning approach is established for the lane change scenario [32]. The approach aims to imitate the human driving style. It considers not only the traffic information but also the driving styles of the surrounding vehicles. Two Convolutional Neural Networks (CNNs) are built for lane change decision-making. The first network aims to capture information about the surrounding vehicle behavior. While the second one is concerned about ego-vehicle behavior. A cost model is established to help in making the best decision from the point of view of speed, safety, and tolerance. The output of the two CNNs and the cost model is the input of a Fully Connected Network (FCN). Finally, the FCN is responsible for taking the best decision. The algorithm is proven to have an accuracy of 98.66% in comparison to human performance and surpasses human performance in safety and speed.

An end-to-end approach is presented to take throttling and steering decisions [33]. The model depends on images fed by a camera and velocity information to take its decision. To make its decisions interpretable, an attention branch network is used. The network identifies the parts of the image that make the vehicle take these decisions. The approach is proven to have an autonomy of 97.2% when experimented with in both urban and road environments.

A light end-to-end neural network is implemented to be compatible with the automotive embedded system environment [34]. The implemented CNN is fed with images from the camera and the output of this network is steering decisions. The implemented approach is proven to be capable to work in real time. Its performance was compared to Alex-net, and it is proven to be 250 times faster than it. The implemented network can work in real-time with 44 Frame Per Second (FPS).

3.2.6. Reinforcement Learning-based Methods

These methods are known of its ability to account for sensors uncertainties. Nevertheless, it is unsafe to be experimented in real street environment as the agent needs to experiment a bunch of behaviors to learn which may lead to high risk for the ego vehicle itself and for all the dynamics objects around it. Besides, it needs high computational power, especially in dense environments.

Decision-making in a dense urban environment is established by using POMDP based on occupancy grid maps. The collision check is done through the occupancy grid map to decrease the computational time needed to check for any collisions throughout the driving time. By testing the algorithm it is proven that using occupancy grid maps improves the POMDP algorithm efficiency by a factor of 50 [35].

A POMDP based on Interacting Multiple Model (IMM) is proposed for decision-making in lane change scenarios [36]. The algorithm considers the risk of collision by using a function based on time to collision (TTC) and Intravehicular Time (IT). A Monte Carlo tree search is used for decision-making as well. It was stated that the algorithm can generate safe and reasonable decisions.

The algorithm is tested on a single-lane car following the scenario and merge scenario. It is proven to be able to balance safety and usability. It can also adapt to surrounding vehicles' behavior. An online tree-based POMDP is used for dense crowd environments. The algorithm is built for real-time operation. It is proven to be safe and efficient, especially in highly dynamic environments [37].

A risk-aware decision-making algorithm is built on basis of POMDP. The responsibility-Sensitive-Safety (RSS) distance model is used to measure the distance between the ego vehicle and the surrounding vehicle to assure safety [38]. This model output is considered by the POMDP algorithm's reward function.

Markov Decision Process (MDP) is used to deal with the scenario in which ethical dilemmas might arise [39]. The implemented framework uses a measure of harm when a collision arises. Three different theories of ethics were used to take the appropriate decisions. It was proven to have the ability to generate decisions based on ethical theories.

A Partially Observable Markov Decision Process (POMDP) is used in intersection scenarios and urban environments [40]. The algorithm can consider the uncertainties in the environment. It can also interact with hidden vehicles and predict their presence. The algorithm is proven to deal with uncertainties.

A POMDP combined with the Monto Carlo tree is also used for lane change and car-following scenarios [41]. It is also used to account for environmental uncertainties. The algorithm accounts for the risk that might occurs due to the behavior of other cars on the road. The algorithm is proven to have the ability to take decisions based on the expected behavior of the surrounding cars in the future.

Table 1. Existing Methodologies Advantages and Disadvantages

Categories	Subcategories	Author & Date	Advantages	Disadvantages
Classical methods	Rules-based methods	[1], [3], [14], [21], [22]	- Simple implementation. - Low computational power. - Real-time operation. - Adapt the rationality of human thinking. - Its behavior can be easily traced and explained.	- Inability to handle complex environments. - Risk of rules explosion. - Inability to handle uncertainty. - Low ability to handle unplanned situations.
	Optimization-based methods	[15], [23]–[27]	- Interaction between dynamic objects can be easily handled.	- Only small traffic situations can be handled in real-time on state-of-the-art computing hardware
	Probabilistic-based methods	[28], [29]	- Ability to interact with the behaviors of other dynamic objects on the road.	- High computational power is required - Does not account for environment uncertainties
Learning methods	Statistic Learning-based methods	[2], [32]	-Requires low computational power. -Requires small human-made datasets.	- Low ability to generalize to different environments.
	Deep Learning-based methods	[31]–[34], [42]	- Adapt behaviors like human behaviors. - Simple in judging its performance in comparison to human performance.	- Requires high computational power. - Requires large human-made datasets. - Low ability to generalize its behaviors to different environments. - Most deep-learning algorithms are sensitive to sensor uncertainties. -Low explainability
	Reinforcement Learning-based methods	[35]–[41]	-Ability to handle environmental uncertainties. - High decision-making accuracy. - Ability to generalize in different environments.	-Requires high computational power. -Most of the existing algorithms are unable to work in real time. -Suffers from safety problems due to the need of the agent to test random behaviors to learn.

4. Applications for BP Methodologies

Different environments, scenarios, and simulation platforms are used for each methodology depending on the adaptability and suitability of each methodology to certain applications as shown in Table.1 and Table.2.

4.1. Testing Environments

The previous researchers have tested their methodologies in one of two different environments which are Highways [H] or Urban [U] environments. Highways are simple environments with vehicles that have high velocity, low traffic, and no pedestrians on them. The highways might also have no challenging components such as motorcycles, bicycles, and animals.

While urban and city environments are more challenging environments. It can imply many dynamic elements such as vehicles, pedestrians, animals, motorcycles, and bicycles.

4.2. Testing Platforms

The previous researchers test their work either on Simulation [S] platforms or Realistic [R]. platforms. The simulation platforms are Carla, Prescan, Matlab/ Simulink, Udacity self-driving car environment, etc. Some of the researchers depend on games environment to test their work such as GTAV. This simulation platform provides semirealistic road environments so that the obtained results from testing their work on them can be more or less reliable.

The testing process for real platforms can be conducted indoors or outdoors. Some researchers depend on small robots to test their work indoors. While others depend on a real vehicle and test their work on real streets. That can be either public or private roads owned by the research companies. Deep learning end-to-end methods are mostly tested on a dataset that is made by a real vehicle platform such as the NGISM dataset.

4.3. Testing Scenarios

There are many different scenarios in which the autonomous car can be on. Some of them are simple and some of them are challenging. The challenging scenarios can be scenarios in which sudden objects might appear, or scenarios that hazards might aries such as construction zones, or scenarios like T-intersection and roundbouts. The previous research has established their work on a basis of one of these scenarios. These scenarios are Lane Changing [LC], Lane Keeping [LK], Over-taking [O], Intersection [I], Roundabout [R], Parking Zone [PZ], and Turning [T]. Considering the lane change scenarios that is stated in Table 1 and Table 2, it can be on a regular road or in a merge. Lane-keeping is also might be following the heading car or just cruising on a lane.

Table 2. Classical Methods Applications

Method	Author & Date	Testing Environment		Testing Platform		Testing Scenario						
		[H]	[U]	[R]	[S]	[LC]	[LK]	[O]	[I]	[R]	[P]	[T]
Rules-based	[1]	✓			✓		✓	✓				
	[14]	✓			✓	✓	✓					
	[21]	✓			✓			✓				
	[3]		✓	✓		✓						
	[22]		✓	✓	✓				✓			
	[16]		✓	✓			✓	✓	✓			✓
	[43]	✓			✓	✓	✓	✓				
	[44]		✓	✓					✓			
	[45]	✓			✓	✓						
	[46]	✓		✓		✓		✓				
	[47]		✓	✓						✓		
	[48]		✓	✓			✓		✓		✓	
	[49]		✓	✓			✓	✓			✓	✓
Optimization-based	[25], [26], [50], [51]	✓			✓	✓						
	[24]		✓		✓	✓						
	[15]	✓	✓	✓	✓	✓	✓				✓	
	[52], [53],	✓			✓	✓	✓					
	[54]	✓		✓		✓	✓	✓				
Probabilistic-based	[28], [29], [55], [56]	✓			✓	✓						
	[57]	✓			✓			✓				
	[58]	✓			✓				✓			

Table 3. Learning Methods Applications

Method	Author & Date	Testing Environment		Testing Platform		Testing Scenario						
		[H]	[U]	[R]	[S]	[LC]	[LK]	[O]	[I]	[R]	[P]	[T]
Statistical Learning-based	[30], [59]	✓			✓	✓						
	[2]	✓		✓	✓	✓						
	[60]	✓		✓		✓						
Deep Learning-based	[31]		✓		✓	✓	✓	✓				✓
	[42]	✓	✓		✓				✓			
	[61]		✓		✓							
	[32]	✓		✓		✓						
	[33], [62]	✓	✓	✓								
	[34]	✓			✓							
	[63]		✓	✓	✓		✓					✓
	[64]		✓	✓								✓
	[65]		✓	✓								
	[66]	✓		✓	✓		✓					
	[67]	✓		✓	✓	✓	✓					
Reinforcement Learning-based	[36], [41], [68]	✓			✓	✓						
	[36], [40], [69]–[72]		✓		✓				✓			
	[38]	✓			✓	✓	✓					
	[73], [74]		✓	✓					✓			
	[7], [75]		✓		✓	✓						
	[76]		✓		✓				✓	✓		
	[10]	✓		✓	✓	✓		✓				

5. Discussion

In the existing BPAs, there is a tradeoff between the simplicity, computational power, and the ability of the algorithm to be generalized to different environments and scenarios. For example, FSM is the simplest of all algorithms and has low computational power but cannot be generalized for all scenarios or platforms due to the hardness of humans to predict or limit all the possible situations that can arise on the road. FSM is the oldest method for BP of all the stated methods it has been used since the DARPA competition and there are still updates that come on with it. Mostly, FSM does not account for predictions, it can only account for current situations and interactions between the surrounding objects. Optimization methods can handle interactions and predictions, but huge computational power is needed and most of the researchers test optimization methods using simulation platforms only. Probabilistic methods are not popular in the field of autonomous cars BP. Moreover, it is also always experimented with in simulation platforms only due to the high computational power needed.

On the other hand, learning-based methods such as statistical learning-based methods and deep learning-based methods almost need many human-made datasets. It took a lot of time and effort to make such a dataset. Besides, it is still hard to account for all the situations in a dataset. The deep-learning methods achieve high accuracy and can imitate human behaviors, but as these algorithms depend on sensor inputs and datasets, the output will be highly affected by the sensor's uncertainties. Reinforcement learning methods are the most popular in the last few years. It is known for its ability to achieve performance that is like human performance. It can also compensate for the sensors and environmental uncertainties. It can account for the interaction and predictions between dynamic objects on the road. The drawback of the reinforcement learning method is its computational power which is not suitable for autonomous embedded systems. This opens the door for researchers to try to achieve light versions of the existing reinforcement learning algorithms.

From the applications perspective, Rules-based methods are tested in both highways and urban environments. But most of the tests are made on highways except for DARPA urban competition. DARPA Urban competition environments are not as complex as real urban environments. However, it is believed that rules-based methods are more suitable for simple scenarios such as highways. It is mostly used for lane change and car following scenarios and rarely to be used in a challenging scenario such as roundabouts. It is proven a real success in scenarios such as lane changing, lane keeping, and overtaking. Optimization methods can be implemented in both highways and urban environments, but as far as we researched optimization methods are mostly used for simple scenarios such as lane changing, lane keeping, and overtaking. The implemented methods are mostly experimented in simulation platforms rather than a real-world scenario and this is mostly due to the high computational power needed for these algorithms. Similarly, the probabilistic-based methods are mostly implemented in simulated scenarios due to the same reason. Besides, they are only tested on highways, and they might be efficient only on highways. It is also almost experimented with in simple scenarios such as lane changing and overtaking. By reviewing the classical-based methods, it is found out that it is appropriate to use them in simple environments.

Statistical learning methods are dependent on datasets and sensor parameters thus it will be easier to be implemented in simple environments such as highways. While deep learning methods can be used in any environment and any scenario depending on the dataset it learned on. However, it is impossible to account for all the situations the car can be on. Finally, reinforcement learning is suitable to be experimented with in both highways and urban environments nevertheless the computational power increase dramatically within dense urban environments. Moreover, most of the conducted research use simulation platforms to test them due to the risk that might arise from testing it in real environments**.**

6. Conclusions and Future Orientation

To conclude, after reviewing the existing methodologies for behavior planning in autonomous vehicles and discussing their advantages, disadvantages, and applications, it is obvious that the current state of research and the previous work is not enough for autonomous vehicles to transfer to level-5 of automation. More investigation in real experiments is needed as many of the existing work shows feasible results in simulation environments, but this is not enough to prove its efficiency in real world implementation.

From the present point of research, these points can be covered in the future. Fusion between different algorithms is needed to be considered as each algorithm has its area of application. Combining different algorithms can open the door for autonomous cars to operate in different environments and generalize its performance. In addition, considering the interaction between vehicles using communication networks protocol such as V2V protocol will ease the decision making specially in environments where the perception is limited. Since, reinforcement learning methods are promising, finding ways to decrease their computational power is important. Situations where ethical dilemmas arise needed to be handled by the existing methodologies. More research is needed to be conducted in parking scenarios and moving in parking lots scenarios. Besides, hazardous places such as construction zones are needed to be studied. In addition, highly dynamic objects such as pedestrians, and cyclists are needed to be considered while selecting the appropriate behavior. Most of the existing research accounts only for the surrounding vehicles and their interaction with them. Complex and unstructured environments are needed to be investigated. Furthermore, dense environments are needed to be studied.

.

References

[1] J. Hu, R. Chen, W. Xu, and R. Lu, "An event-triggered real-time motion planning strategy for autonomous vehicles," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 236, no. 19, pp. 10332–10348, 2022, doi: 10.1177/09544062221098548.

[2] Y. Liu, X. Wang, L. Li, S. Cheng, Z. Chen, and S. Member, "A Novel Lane Change Decision-Making Model of Autonomous Vehicle Based on Support Vector Machine," *IEEE Access*, vol. 7, pp. 26543–26550, 2019, doi: 10.1109/ACCESS.2019.2900416.

[3] N. D. Van, M. Sualeh, D. Kim, and G. W. Kim, "A hierarchical control system for autonomous driving towards urban challenges," *Appl. Sci.*, vol. 10, no. 10, 2020, doi: 10.3390/app10103543.

[4] H. Li, S. Member, C. Wu, D. Chu, L. Lu, and K. E. N. Cheng, "Combined Trajectory Planning and Tracking for Autonomous Vehicle Considering Driving Styles," pp. 9453–

9463, 2021, doi: 10.1109/ACCESS.2021.3050005.

[5] O. Sharma, N. C. Sahoo, and N. B. Puhan, “Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey,” *Eng. Appl. Artif. Intell.*, vol. 101, no. January, p. 104211, 2021, doi: 10.1016/j.engappai.2021.104211.

[6] SAE International, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles; SAE International,” Warrendale, 2019.

[7] S. Ulbrich and M. Maurer, “Probabilistic online POMDP decision making for lane changes in fully automated driving,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. Itsc, pp. 2063–2070, 2013, doi: 10.1109/ITSC.2013.6728533.

[8] DARPA, “The Grand Challenge,” 2004. [Online]. Available: https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles.

[9] O. S. Tas, F. Kuhnt, J. M. Zollner, and C. Stiller, “Functional system architectures towards fully automated driving,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2016-Augus, no. Iv, pp. 304–309, 2016, doi: 10.1109/IVS.2016.7535402.

[10] S. Ulbrich and M. Maurer, “Towards Tactical Lane Change Behavior Planning for Automated Vehicles,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015-Octob, pp. 989–995, 2015, doi: 10.1109/ITSC.2015.165.

[11]T. Gindele, D. Jagszent, B. Pitzer, and R. Dillmann, “Design of the planner of team annieway’s autonomous vehicle used in the DARPA Urban Challenge 2007,” *IEEE Intell. Veh. Symp. Proc.*, pp. 1131–1136, 2008, doi: 10.1109/IVS.2008.4621268.

[12] S. Li, “An Adaptive Motion Planning Technique for On-Road Autonomous Driving,” vol. 9, 2021, doi: 10.1109/ACCESS.2020.3047385.

[13] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016, doi: 10.1109/TIV.2016.2578706.

[14] X. Wang, X. Qi, P. Wang, and J. Yang, “Decision making framework for autonomous vehicles driving behavior in complex scenarios via hierarchical state machine,” pp. 1–12, 2021.

[15] P. F. Orzechowski, C. Burger, and M. Lauer, “Decision-Making for Automated Vehicles Using a Hierarchical Behavior-Based Arbitration Scheme,” no. Iv, 2020.

[16] L. Yu, D. Kong, and X. Yan, “A driving behavior planning and trajectory generation method for autonomous electric bus,” *Futur. Internet*, vol. 10, no. 6, 2018, doi: 10.3390/fi10060051.

[17] Q. Liu, X. Li, S. Yuan, and Z. Li, “Decision-Making Technology for Autonomous Vehicles : Learning-Based Methods , Applications and Future Outlook,” 2021.

[18] G. Anjana and R. K. Megalingam, “A Study on Human-like Driving Decision-Making Mechanism in Autonomous Vehicles Under Various Road Scenarios,” *Lect. Notes Networks Syst.*, vol. 204 LNNS, pp. 391–402, 2021, doi: 10.1007/978-981-16-1395-1_29.

[19] A. Rev, C. Robot, A. Syst, W. Schwarting, J. Alonso-mora, and D. Rus, “Planning and Decision-Making for Autonomous Vehicles,” no. January, 2018.

[20] F. Garrido and P. Resende, “Review of Decision-Making and Planning Approaches in Automated Driving,” *IEEE Access*, vol. 10, no. September, pp. 100348–100366, 2022, doi: 10.1109/ACCESS.2022.3207759.

[21] J. Palatti, A. Aksjonov, G. Alcan, and V. Kyrki, “Planning for Safe Abortable Overtaking Maneuvers in Autonomous Driving,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2021-Septe, pp. 508–514, 2021, doi: 10.1109/ITSC48978.2021.9564499.

[22] Y. Chen, J. Zha, J. Wang, and S. Member, “An Autonomous T-intersection Driving Strategy Considering Oncoming Vehicles Based on Connected Vehicle Technology,” *IEEE/ASME Trans. Mechatronics*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TMECH.2019.2942769.

[23] and C. L. Yiran Zhang, Peng Hang, Chao Huang, “Human-Like Interactive Behavior Generation for Autonomous Vehicles: A Bayesian Game-Theoretic Approach with Turing Test.” 2022.

[24] C. O.-M. Nikita Smirnov, Yuzhou Liu , Aso Validi, Walter Morales-Alvarez, “A Game Theory-Based Approach for Modeling Autonomous Vehicle Behavior in Congested, Urban Lane-Changing Scenarios,” 2021.

[25] P. Hang, C. Lv, S. Member, Y. Xing, C. Huang, and Z. Hu, “Human-Like Decision Making for Autonomous Driving : A Noncooperative Game Theoretic Approach,” no. 1922500046, pp. 1–11, 2020.

[26] M. Garzón, A. Spalanzani, M. Garzón, A. Spalanzani, and M. Garz, “Game theoretic decision making for autonomous vehicles ’ merge manoeuvre in high traffic scenarios To cite this version : HAL Id : hal-02388757 Game theoretic decision making for autonomous vehicles ’ merge manoeuvre in high traffic scenarios,” 2019.

[27] David Isele, “Interactive Decision Making for Autonomous Vehicles in Dense Traffic,” 2019.

[28] Y. Lyu, C. Dong, and J. M. Dolan, “FG-GMM-based Interactive Behavior Estimation for Autonomous Driving Vehicles in Ramp Merging Control ∗,” pp. 1–6, 2020.

[29] D. Iberraken *et al.*, “Reliable Risk Management for Autonomous Vehicles based on Sequential Bayesian Decision Networks and Dynamic Inter-Vehicular Assessment To cite this version : HAL Id : hal-03100071 Reliable Risk Management for Autonomous Vehicles based on Sequential Bayesi,” 2019.

[30] A. Mechernene, V. Judalet, A. Chaibet, and M. Boukhnifer, “Detection and Risk Analysis with Lane-Changing Decision Algorithms for Autonomous Vehicles,” pp. 1–14, 2022.

[31] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautam, “Multi-task Learning with Attention for End-to-end Autonomous Driving,” 2021.

[32] Y. Zhang and J. Wang, “A LEARNING-BASED DISCRETIONARY LANE-CHANGE DECISION-MAKING MODEL WITH DRIVING STYLE AWARENESS,” no. Mlc, pp. 1–11, 2020.

[33] K. Mori, H. Fukui, and T. Murase, “Visual Explanation by Attention Branch Network for End-to-end Learning-based Self-driving,” *2019 IEEE Intell. Veh. Symp.*, no. Iv, pp. 1577–1582, 2019.

[34] N. J. and V. D. Jelena Kocic, “An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms,” *Sensors (Switzerland)*, 2019.

[35] C. Zhang, S. Ma, M. Wang, G. Hinz, and A. Knoll, “Efficient POMDP Behavior Planning for Autonomous Driving in Dense Urban Environments using Multi-Step Occupancy Grid Maps,” pp. 2722–2729, 2022.

[36] Y. Huang, W. Zhao, C. Xu, and S. Zou, “An IMM-based POMDP decision algorithm using collision-risk function in mandatory lane change,” vol. 236, no. 7, pp. 1500–1514, 2022, doi: 10.1177/09544070211039720.

[37] B. Hayes and Z. Sunberg, “Intention-Aware Navigation in Crowds with Extended-Space POMDP Planning,” no. Aamas, 2022.

[38] C. Skoglund, “Risk-aware Autonomous Driving Risk-aware Autonomous Driving Using POMDPs and Responsibility-Sensitive Safety,” 2021.

[39] N. De Moura and K. Evans, “Ethical decision making for autonomous vehicles,” no.

Iv, pp. 2006–2013, 2020.

[40] P. Schorner, T. Lars, J. Doll, and J. Z. Marius, “Predictive Trajectory Planning in Situations with Hidden Road Users Using Partially Observable Markov Decision Processes,” no. Iv, pp. 2299–2306, 2019.

[41] D. S. Gonzalez, M. Garzon, J. S. Dibangoye, and C. Laugier, “Human-Like Decision-Making for Automated Driving in Highways,” *2019 IEEE Intell. Transp. Syst. Conf.*, pp. 2087–2094, 2019.

[42] X. Hu, B. Tang, L. Chen, S. Song, and X. Tong, “Learning a Deep Cascaded Neural Network for Multiple Motion Commands Prediction in Autonomous Driving,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7585–7596, 2021, doi: 10.1109/TITS.2020.3004984.

[43] Q. Wang, B. Ayalew, and T. Weiskircher, “Predictive Maneuver Planning for an Autonomous Vehicle in Public Highway Traffic,” *IEEE Trans. Intell. Transp. Syst.*, vol. PP, pp. 1–13, 2018, doi: 10.1109/TITS.2018.2848472.

[44] A. Cosgun *et al.*, “Towards full automated drive in urban environments: A demonstration in GoMentum Station, California,” *IEEE Intell. Veh. Symp. Proc.*, pp. 1811–1818, 2017, doi: 10.1109/IVS.2017.7995969.

[45] K. Ransikarbum, N. Kim, S. Ha, R. A. Wysk, and L. Rothrock, “A Highway-Driving System Design Viewpoint Using an Agent-Based Modeling of an Affordance-Based Finite State Automata,” *IEEE Access*, vol. 6, pp. 2193–2205, 2017, doi: 10.1109/ACCESS.2017.2782257.

[46] S. Noh and K. An, “Decision-Making Framework for Automated Driving in Highway Environments,” pp. 1–14, 2017.

[47] B. Okumura *et al.*, “Challenges in perception and decision making for intelligent automotive vehicles: A case study,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 20–32, 2016, doi: 10.1109/TIV.2016.2551545.

[48] A. Kurt and Ü. Özgüner, *Hybrid State System Development for Autonomous Vehicle Control in Urban Scenarios*, vol. 41, no. 2. IFAC, 2008.

[49] M. Montemerlo *et al.*, “Junior: The stanford entry in the urban challenge,” *Springer Tracts Adv. Robot.*, vol. 56, no. October 2005, pp. 91–123, 2009, doi: 10.1007/978-3-642-03991-1_3.

[50] M. S. and D. M. Rohan Chandra, Mingyu Wang, “Game-Theoretic Planning for Autonomous Driving among Risk-Aware Human Drivers,” 2022.

[51] N. Li *et al.*, “Game Theoretic Modeling of Driver and Vehicle Interactions for

Verification and Validation of Autonomous Vehicle Control Systems," pp. 1–16, 2017.

[52] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A Game Theoretic Approach to Replanning-aware Interactive Scene Prediction and Planning," vol. X, no. c, pp. 1–12, 2015, doi: 10.1109/TVT.2015.2508009.

[53] M. During and P. Pascheka, "Cooperative Decentralized Decision Making for Conflict Resolution among Autonomous Agents," pp. 0–7, 2014.

[54] S. Ulbrich, S. Grossjohann, C. Appelt, K. Homeier, J. Rieken, and M. Maurer, "Structuring Cooperative Behavior Planning Implementations for Automated Driving," 2015, doi: 10.1109/ITSC.2015.349.

[55] W. Zhan, L. Sun, Y. Hu, J. Li, and M. Tomizuka, "Towards a Fatality-Aware Benchmark of Probabilistic Reaction Prediction in Highly Interactive Driving Scenarios," 2018.

[56] C. Dong, J. M. Dolan, and B. Litkouhi, "Intention Estimation For Ramp Merging Control In Autonomous Driving," 2017.

[57] D. Iberraken, L. Adouane, and D. Denis, "Safe Autonomous Overtaking Maneuver based on Inter-Vehicular Distance Prediction and Multi-Level Bayesian Decision-Making," pp. 3259–3265, 2018.

[58] N. Evestedt, E. Ward, J. Folkesson, and D. Axehill, "Interaction aware trajectory planning for merge scenarios in congested traffic situations," 2016.

[59] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," no. Iv, pp. 1590–1595, 2017.

[60] N. Motamedidehkordi, S. Amini, S. Hoffmann, and F. Busch, "Modeling Tac tical Lane-change Behavior for Automated Vehicles: A Supervised Machine Learning Approach," pp. 268–273, 2017.

[61] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable End-to-end Urban Autonomous Driving with Latent Deep Reinforcement Learning," pp. 1–11, 2020.

[62] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-based Driving Path Generation Using Fully Convolutional Neural Networks," 2017.

[63] F. Codevilla, M. Matthias, and L. Antonio, "End-to-end Driving via Conditional Imitation Learning," pp. 4693–4700, 2018.

[64] C. Hubschneider, M. Weber, and J. Z. Marius, "Adding Navigation to the Equation : Turning Decisions for End-to-End Vehicle Control," pp. 307–314, 2017.

[65] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, “Large-scale cost function learning for path planning using deep inverse reinforcement learning,” *Int. J. Rob. Res.*, vol. 36, no. 10, pp. 1073–1087, 2017, doi: 10.1177/0278364917722396.

[66] M. Bojarski *et al.*, “End to End Learning for Self-Driving Cars,” pp. 1–9, 2016.

[67] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “DeepDriving: Learning affordance for direct perception in autonomous driving,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, no. Figure 1, pp. 2722–2730, 2015, doi: 10.1109/ICCV.2015.312.

[68] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, “High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning,” 2018.

[69] Z. Qiao, K. Muelling, J. Dolan, P. Palanisamy, and P. Mudalige, “POMDP and Hierarchical Options MDP with Continuous Actions for Autonomous Driving at Intersections,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2018-Novem, pp. 2377–2382, 2018, doi: 10.1109/ITSC.2018.8569400.

[70] M. Sefati, J. Chandiramani, K. K. A. Kampker, and S. Baldi, “Towards Tactical Behaviour Planning Under Uncertainties for Automated Vehicles in Urban Scenarios,” 2017.

[71] D. L. S. Constantin Hubmann, Marvin Becker, Daniel Althoff, “Decision Making for Autonomous Driving considering Interaction and Uncertain Prediction of Surrounding Vehicles,” *IEEE Intell. Veh. Symp.*, no. Iv, pp. 1671–1678, 2017.

[72] K. H. Wray, S. J. Witwicki, and S. Zilberstein, “Online decision-making for scalable autonomous systems,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 0, no. Section 4, pp. 4768–4774, 2017, doi: 10.24963/ijcai.2017/664.

[73] K. H. Wray *et al.*, “POMDPs for Safe Visibility Reasoning in Autonomous Vehicles,” no. Figure 1, 2019.

[74] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs,” *2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014*, pp. 392–399, 2014, doi: 10.1109/ITSC.2014.6957722.

[75] H. Yi, E. Park, and S. Kim, “Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving,” *KIISE Trans. Comput. Pract.*, vol. 24, no. 12, pp. 670–674, 2018, doi: 10.5626/ktcp.2018.24.12.670.

[76] W. Liu, S. W. Kim, S. Pendleton, and M. H. Ang, “Situation-aware decision making for autonomous driving on urban road using online POMDP,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2015-Augus, no. Iv, pp. 1126–1133, 2015, doi: 10.1109/IVS.2015.7225835.