A Review of Motion Planning for Highway Autonomous Driving

Laurène Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser

***Abstract*— Self-driving vehicles will soon be a reality, as main automotive companies have announced that they will sell their driving automation modes in the 2020s. This technology raises relevant controversies, especially with recent deadly accidents. Nevertheless, autonomous vehicles are still popular and attractive thanks to the improvement they represent to people's way of life (safer and quicker transit, more accessible, comfortable, convenient, efficient, and environment-friendly). This paper presents a review of motion planning techniques over the last decade with a focus on highway planning. In the context of this article, motion planning denotes path generation and decision making. Highway situations limit the problem to high speed and small curvature roads, with specific driver rules, under a constrained environment framework. Lane change, obstacle avoidance, car following, and merging are the situations addressed in this paper. After a brief introduction to the context of autonomous ground vehicles, the detailed conditions for motion planning are described. The main algorithms in motion planning, their features, and their applications to highway driving are reviewed, along with current and future challenges and open issues.**

***Index Terms*— Advanced driver assistance systems, autonomous driving, decision making, intelligent vehicles, motion planning, path planning.**

I. Introduction

SINCE the last decade, the development of autonomous vehicles has spread worldwide among universities and the automotive sector as one of the most promising advancements in automotive engineering and research.

One of the first contributions to driverless cars dates back to the 1920s at the Houdina Radio Control Company, which successfully proceeded with the demonstration of a car controlled by radio signals sent by a trailing vehicle [1]. In the late 80s and 90s, research institutes and automotive manufacturers partnership numerous autonomous driving projects financed in, such as the European EUREKA Prometheus program [2] with the twin cars VaMP and VITA II [3], "No Hands Across America" from Carnegie Mellon University's Navlab [4], ARGO from the University of Parma's VisLab [5], and the automated highway platooning Consortium (NAHSC) [6], among others.

Nowadays, many technological evolutions in partial autonomous vehicles are already used in Advanced Driving Assistance Systems (ADAS), and automakers now try to personalize ADAS to the driver's style [7]. Concerning highway planning, the main assistance technologies concern both the longitudinal and lateral comfort and security with the Cruise Control (CC), the Intelligent Speed Adaptation (ISA), the Lane Keeping Assist (LKA), or the Lane Departure Warning (LDW). While interacting with obstacles, collision avoidance systems either warn the driver about an imminent collision, e.g. the Lane Change Assist (LCA), or autonomously take action, helping the driver stay safe, such as the Adaptive Cruise Control (ACC) or the Automatic Emergency Braking (AEB).

After Nevada, USA, authorized driverless vehicles on public roads in June 2011, other American states and countries adopted laws to test autonomous vehicles in traffic. In response, the United Nations Economic Commission for Europe (UNECE) made the use of ADAS in the Vienna Convention on Road Traffic [8] more flexible. It allowed automated driving systems, "provided that these technologies are in conformity with the United Nations vehicle regulations or can be overridden or switched off by the driver" [9]. With such an evolution in the automotive field, the Society of Automotive Engineers (SAE) published a standard classification for autonomous vehicles with a 6-level system, from 0 (no control but active safety systems) to 5 (no human intervention for driving) [10]. Levels 4 and 5 have not been technically fulfilled by automakers yet; however, since the Defense Advanced Research Projects Agency (DARPA) organized autonomous vehicle competitions in 2004, 2005, and 2007, and thanks to new technologies, autonomous functions are evolving quickly and treat more complex scenarios in real environments. The 11 finalist teams of the DARPA Urban Challenge 2007 [11] succeeded in navigating through a city environment. The University of Parma's VisLab ran the VisLab Intercontinental Autonomous Challenge (VIAC) in 2010, a 15 900 km trip in 100 days from Parma, Italy to Shanghai, China [12]; and ran the PROUD project [13] in 2013, a mixed traffic driving route open to public traffic through Parma, Italy. In 2011, the European projects HAVEit [14], ABV [15], and CityMobil [16] successfully demonstrated highway and

Manuscript received November 17, 2017; revised June 21, 2018, November 5, 2018, and February 18, 2019; accepted April 8, 2019. Date of publication May 15, 2019; date of current version May 1, 2020. The Associate Editor for this paper was B. Fidan. *(Corresponding author: Laurène Claussmann.)*

L. Claussmann is with the Department of Autonomous Driving, Institut VEDECOM, 78000 Versailles, France, and also with the IFSTTAR-LIVIC Laboratory, 78000 Versailles, France (e-mail: laurene.claussmann@vedecom.fr).

M. Revilloud is with the Department of Autonomous Driving, Institut VEDECOM, 78000 Versailles, France (e-mail: marc.revilloud@vedecom.fr).

D. Gruyer is with the IFSTTAR-LIVIC Laboratory, 78000 Versailles, France (e-mail: dominique.gruyer@ifsttar.fr).

S. Glaser was with the IFSTTAR-LIVIC Laboratory, 78000 Versailles, France. He is now with CARRS-Q, Queensland University of Technology, Brisbane city, QLD 4000, Australia (e-mail: sebastien.glaser@qut.edu.au).

Digital Object Identifier 10.1109/TITS.2019.2913998

city driving. In 2012, highway driving in platooning was achieved by the SARTRE Project [17]. In 2013, another demonstration by Daimler and Karlsruhe Institute of Technology completed the Bertha Benz Memorial Route [18] with both city and highway driving. In 2016, the international Grand Cooperative Driving Challenge (GCDC) [19] presented merging, intersection, and emergency traffic scenarios. In addition to the research projects, in 2016, many automotive companies unveiled their semi-autonomous vehicles corresponding to level 2 of the SAE classification. Following the demonstration of the Google Car, Apple announced its Project Titan, and Baidu started public road tests with the Cloud ride. Meanwhile, the transportation network companies Uber and Lyft collaborated with automotive companies to test their algorithms.

In spite of the current trendiness of this topic, autonomous projects are not recent in the robotics literature and many reviewing works exist on motion planning strategies [20]–[22] or, more recently, on the specific use case of autonomous driving [23]–[25]. However, the standardization of the research to fit industrially realistic prototypes of self-driving cars introduces new concerns such as functional safety, real-time computing, a systemic approach, or low-cost developments, and specific popular applications for parking, intersection management, or highway driving. The motivation for highway driving lies in its simple driving structure and the driver's limited behavior in nominal situations, making it the most reachable context for the first fully autonomous systems in traffic. Furthermore, highways seem to be the first environment where drivers would be confident driving in a fully autonomous mode [26].

The purpose of this review is to build a taxonomy of motion planning algorithms for highway driving with autonomous vehicles used like the workaday car of the future. The review is organized as follows: Section II explains the scope of our classification, related to the perception and control constraints, and driver/highway driving rules, to highlight the specific situation of highway driving. Section III describes the state of the art, and section IV shows a comparison table for highway applications. Finally, section V discusses the gaps to be filled in motion planning by the next autonomous driving car generation, and section VI concludes this work.

II. Considerations for Highway Motion Planning

A. Terminology

Before dealing with planning algorithms, one needs to define the wide terminology involved. The adjective ego relates to the vehicle that is mastered and sensors-equipped. In contrast, other vehicles are denoted as obstacles. The kinematics of the vehicle are represented by its states, i.e. its position and orientation, and their time derivatives (position, speed, and acceleration, linear and angular). The geometric state space is called the configuration space. The evolution space identifies the configuration space-time in which the ego vehicle can navigate. Both the configuration and evolution spaces are usually divided into three subspaces: the collision space, in which the ego vehicle collides with obstacles; the uncertain space, in which there exists a probability for the ego vehicle to be in

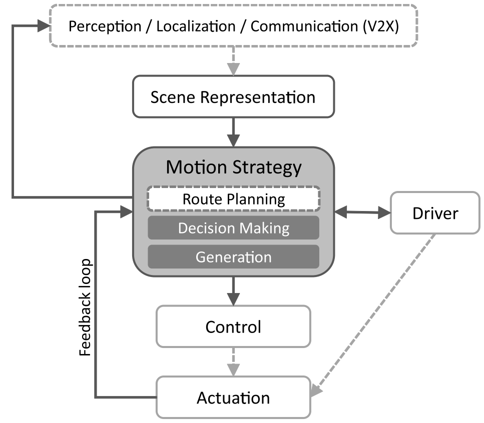


Fig. 1. A hierarchical scheme of Autonomous Ground Vehicle systems.

collision; and the free space, in which there is no collision. The generic term motion characterizes the states' evolution over time. Motion can refer in the literature to either free-space (spatial geometric zones), path (sequence of space-related states in the free space, i.e. geometric waypoints), trajectory (sequence of spatiotemporal states in the free space, i.e. time-varying waypoints), maneuver (predefined motion, considered as a subspace of paths or trajectories, i.e. motion primitives), or action/task (symbolic operations of maneuvers). We distinguish generation, which builds sequences of paths, trajectories, maneuvers, or actions, from planning, meaning the selection of one sequence among the generated motions. Finally, the prediction horizon denotes the space or/and time horizon limit for the simulation of motion.

B. Motion Planning Scheme

A general abstraction of the hierarchical scheme of autonomous vehicle can be found in [18], [24], [27]. We simplify and adapt the proposed scheme to the one shown in Fig. 1, which focuses on the motion strategy block.

The input data for the motion strategy block displays data for ego vehicle, obstacles' behaviors and infrastructure description, obtained from the perception, localization, and communication (Vehicle-to-X - V2X) blocks. It does not pay any attention to how this information has been collected, but to its quality such as the measures, their uncertainties, and their trustworthiness. Thus, a scene representation is introduced between the perception/localization/communication and motion strategy blocks, as presented in [13]. This component manages sensors data and provides a perception map with obstacles, lanes, traffic, road, and ego information. Besides, a closed-loop system from the perception/localization/communication block to the motion strategy block is used to inform the ego perception and localization of the current and future motion. It also conveys the current and future ego motion intentions to the environment via communication.

On the other hand, the control block is formally fed with the reference motion decided by motion strategy, and then acts on actuators to move the ego vehicle. The motion strategy block also reacts as a closed-loop system on the control and actuation blocks. In fact, the information from the sensors on actuators'

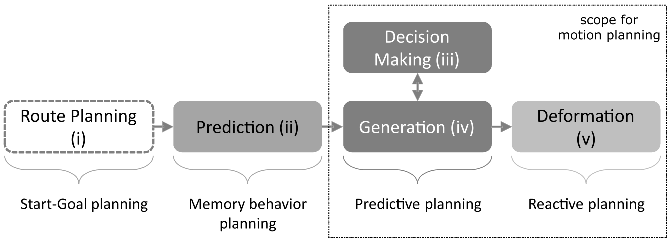


Fig. 2. Motion planning functions. Motion planning acts as a global, local, and reactive motion strategy.

outputs provides an up-to-date ego state vector, so that the motion strategy remains accurate.

Moreover, with semi-autonomous vehicles, the driver will interfere in the low levels of automation on the actuators pedals and steering, and on the high levels, so the motion strategy block acts more or less as a co-driver. In that sense, both communication from the driver to the motion strategy block and the reverse are necessary to warn the driver of the motion decision and to take into account his/her intention to drive.

The motion strategy decides the most convenient motion according to some chosen criteria (addressed in II-D). In more details, in [28], J.-A. Michon names the "three levels of skills and controls: strategical (planning), tactical (maneuvering), and operational (control) respectively." These three layers are largely exploited in the recent architecture [13], [18]. The tactical part is usually divided into subparts, with at least one for behavioral planning and another for motion generation [18], [24], [27], [29]. In this review, according to the structures previously described, we distinguish five main functions in the motion strategy hierarchy, as shown in Fig. 2: (i) route planning, (ii) prediction, (iii) decision making, (iv) generation, and (v) deformation. The space and time horizons for each of the motion planning functions are summarized in Table I.

The route planner (i) is a trip scheduler; it provides a general long-term planning through the road network from the initial position to the driver's desired destination. This function is outside the scope of this paper; see [30] for a related review.

The second function, the prediction step (ii), stores the current and historic dynamics data to predict the dynamics of all the elements surrounding the ego vehicle. This process allows to perform long-term risk estimation and dynamic replanning. The road infrastructure, the route direction, the driving rules, and the lane marking evolution are usually extrapolated from map information. The prediction of obstacles' behavior is the most critical task of the prediction function. Although it is necessary for motion planning, it is not explained in this paper; see [31] for a corresponding review. The core functions addressed in this review are decision making (iii), generation (iv), and deformation (v), which we call the motion planning scope. In recent works [32], [33], the motion planning approaches are organized as follows:

– A high-level predictive planning built around three objectives: risk evaluation, criteria minimization, and constraint submission (see II-D). Those are used for decision making (iii), i.e. to select the best solution out of the candidates' generation (iv). One either generates a set of motions and then makes a decision on the behavior motion, or, defines the behavior to adopt and then fits a set of motions. This high-level stage benefits from a longer predicted motion but is time-consuming.
– A low-level reactive planning deforming the generated motion from the high-level planning according to a reactive approach, i.e. the deformation function (v). This acts on a shorter range of actions and thus has faster computation.

TABLE I
SPACE AND TIME HORIZON FOR THE MOTION PLANNING FUNCTIONS

	Route Planning (i)	Prediction (ii)	Decision Making (iii)	Generation (iv)	Deformation (v)
Space	>100 m	>1 m <100 m	>10 m <100 m	>10 m <100 m	>0.5 m <10 m
Time	>1 min <1 hour	>1 s <1 min	>1 s <1 min	>1 s <1 min	>10 ms <1 s

C. Specificities of Highway Driving

Motion planning techniques highly depend on the use cases. Our considerations for highway driving are limited to lane-divided roads featuring unidirectional flow (opposing directions of travel being separated by a median strip) in fluid traffic, i.e. with a dynamic speed over $60 km/h$. The road shape is made of straight lines, clothoids, and circles with small curvature. In a nominal situation, there are only motorized vehicles, which adhere to the same driving rules. Obstacle's behavior prediction is also limited to one-direction, two-lane changes – right or left – and to accelerate, maintain speed, or brake. Thus, traffic behaviors are more uniform than city driving. As detailed in [34], we distinguish eight nominal non-exclusive situations during a highway trip, without focusing on exceptional situations:

– Lane keeping: The safety space in front of and behind the ego vehicle is guaranteed. The longitudinal decision is to maintain the desired speed, whereas the lateral decision keeps the ego trajectory inside the ego lane marks.
– Car following: Besides lane following, the ego vehicle must follow the front ego lane vehicle while maintaining its longitudinal safety distances.
– Lane changing: This decision is made under either directional or obstacle constraints. The motion planner must ensure that the space in the target lane is sufficient and that the speed is adequate to keep the ego vehicle in a safe state.
– Lateral-most lane changing: To ensure fluid traffic flow and safety, some driving rules require leftmost or rightmost lane driving. Hence, the ego vehicle always seeks to change lanes until the lateral-most lane is reached.
– Passing: The ego vehicle respects a lane keeping or car following decision while obstacles are in the adjacent lanes. Keeping lateral safety distances is required.
– Overtaking: This complex maneuver consists of a lane change, then passing a vehicle or an obstacle, and finally another lane change to return to the previous ego lane.

- Merging: Two actions occur on a highway: entering or exiting the highway, and merging two rows of vehicles into one. The ego vehicle must adapt its longitudinal and lateral speeds and distances to ease its way into traffic.
- Highway toll: The approach decision is to merge into a fictional lane in anticipation of a toll lane delimitation and to reduce speed until stopping, whereas the leave decision is to accelerate to the reference speed and to merge into a real traffic lane.

Autonomous vehicles are also studied in platooning. This configuration decreases the distances between the vehicles and thus increases the capacity of roads. The motion planning strategy in platoons must be more robust than for individual vehicles in the sense of smoother acceleration and turns to guarantee the platoon's stability. Platooning is not as interesting from the perspective of motion planning as from that of control. We therefore do not consider any specific studies on platooning in this review. Please refer to [35] for more details.

The main differences between highway, except for platooning, and city driving consist in a further look-ahead time, with a stronger focus towards the ahead direction of the road, whereas city driving involves a closer range but in all directions. The highway vehicle dynamics is also simpler with lower turn-angle, no reverse, and less braking/acceleration, but higher and more constant speed. Thus, even if there are less hazards, the risk due to high speed is stronger. Moreover, the higher distances imply poorer sensors capacities. Finally, less traffic insures more stable scenario. The algorithms which consider all these specificities in real-time will be favored for a practical application on highways.

D. Constraints on Highway Driving

Despite indicators such as high reliability, safety, and low computation time needed for algorithmic specifications, we also consider more specific highway planning constraints. On the one hand, the environment's safety constraints respect the driving rules and avoid collision. These are called hard constraints as they are absolutely essential for autonomous driving acceptance. On the other hand, the driver makes it necessary to respect ride optimization constraints for the minimization of time, distance, or energy consumption, and maximization of comfort. These are called soft constraints and can be relaxed. Other feasibility constraints rely on kinematic restrictions of the vehicle, which are the nonholonomic dynamics, i.e. the vehicle evolves in a three dimension space with only two degrees of freedom, a smooth path, i.e. the trajectory should be differentiable and its curvature continuous, and the dynamics limitations of a vehicle. The choice of the vehicle's model to handle these constraints induces algorithmic complexity, the more degrees of freedom are used, the more complex is the model solver. For most highway planning developments, there is no or very few (particle kinematics with longitudinal and lateral position and velocity states) consideration of the vehicle model, except for the explicit resolution of the potential field methods (see III-B.3) and the numerical optimization (see III-B.5). This question of degrees of freedom is a fundamental design parameter in motion planning and control architecture, and should be addressed to guarantee a safe and drivable motion [36] as well as consistency [37]. The last dynamic constraint is the evolution of the ego vehicle in time. To summarize, the authors in [27] identify the quality requirements for the generated motion: "feasible, safe, optimal, usable, adaptive, efficient, progressive, and interactive."

E. What Is at Stake in this Paper

Our interest in this classification is to highlight the many types of algorithms used in motion strategy for highway autonomous driving without limiting ourselves to their mathematical complexity, but with a specific focus on their real-time effectiveness. We do not pretend to rank the methods; we prefer to classify them from a practical perspective, i.e. we analyze how the algorithms work well, their advantages and drawbacks, considering simulations or experiments. The methods' optimality is of secondary interest then, compared to the feasibility of the motion. In the particular case of highway driving, we must guarantee at least one motion to ensure continuity. Following the ego lane and adapting its longitudinal states to the front vehicle is, by nature, always feasible.

III. State of the Art

The choice of the motion planning methods depends on the formulation of the motion planning problem.

Firstly, the problem formulation strongly varies with the inherited data (discrete/continuous, algebraic/analytic or static/dynamic) for the scene representation. Thus, the sensory technologies are of great importance to implement a motion planning algorithm. Even if the uncertainties are more and more considered in the motion planning scope (see V.A), too few papers addressing motion planning tackle the issue of the sensor's architecture and technology. We invite the readers to refer to [38] for a corresponding review.

Secondly, motion planning combines five unavoidable aspects, as seen in the previous section: (i) state estimation, (ii) time evolution, (iii) actions planning, (iv) criteria optimization, and (v) compliance with constraints. How these are handled changes the outlook of the problem.

Furthermore, two approaches for reviewing motion planning algorithms coexist: distinguishing and not distinguishing the driving modes. For the first, the driving modes and use cases are separated. For example, in [39], the authors focus on the specific actions of lane changing and merging maneuvers on highways. In this review, we consider a situation only as a specific set of criteria and constraints, not as a differentiating element for the algorithms. In this way, all methods could be applied to the different situations, with limitations to their proper functioning. Yet, we noticed that the main functions of motion strategy described in II-B involve discriminating specificities among motion planning methods, e.g. the deformation function requires a reactive online real-time method.

In this section, we first explain our algorithm classification, and then review the most popular approaches from their founding principles to their advancement for an application to highway planning since the DARPA Urban Challenge 2007.

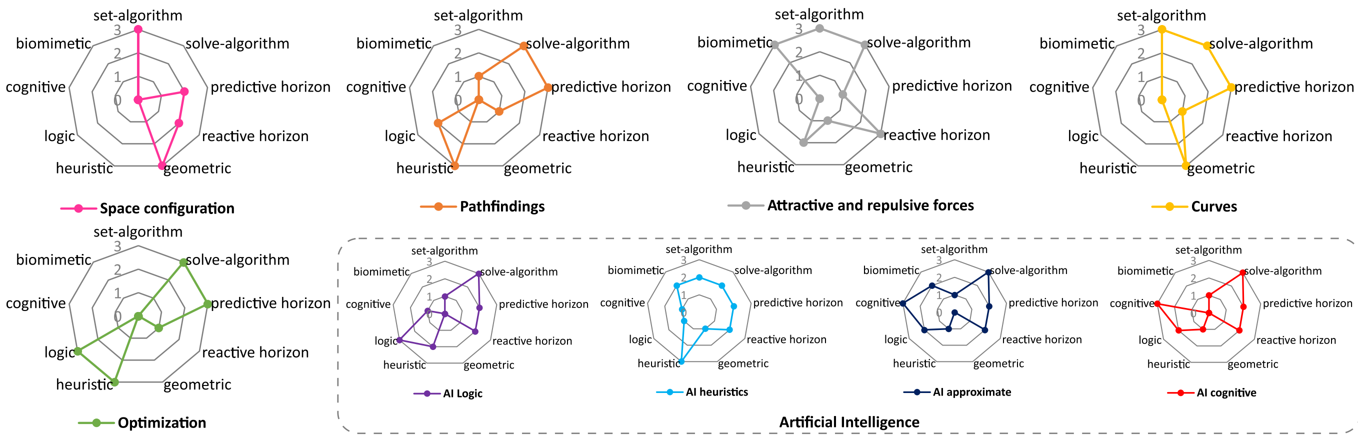


Fig. 3. Taxonomy attribute classification. The radar charts show the distribution of each of the families considering the selected attributes. At the center of the chart, the methods do not suit the corresponding attribute at all (value = 0). In contrast, an outer position of the attribute means it is well adapted to the use of the method (value = 3). Each value is assigned according first to the definitions of the families, and to their occurrence in the literature.

We do not pretend to make an exhaustive taxonomy as an infinite number of ways of treating motion planning exist.

A. Taxonomy Description

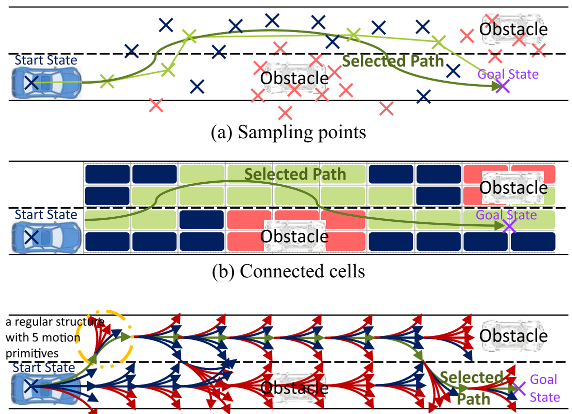
According to what we consider in section II, we propose in this review to classify the algorithms in the families summarized in Fig. 3. The term family refers to a set of algorithms that rely on the same basic principle. They usually return the same type of output, qualify with the same attributes, and relate to the same mathematical domain. We propose a classification based on the following characteristics:

– The type of output: a space, a path, a trajectory, a maneuver, or a symbolic representation. This information should be adapted to the control and driver blocks' interface shown in Fig. 1. In this respect, it is also essential to know whether the considered motion planning algorithm returns only a decomposition or a reference motion. In the first case, called a **set-algorithm**, a complementary algorithm should be added to find the feasible motion. The second case is identified as a **solve-algorithm**. These characterizations are extended from [40], where the author defined the *Findspace* and *Findpath* algorithms to find a safe position in the evolution space, and to identify a sequence of safe positions that link the start and goal positions.
– The space-time property of the algorithm: the **predictive** or **reactive** horizon, as detailed in II-B.
– The mathematical domains: **geometric**, **heuristic**, **logic**, **cognitive**, **biomimetic**. This defines the philosophy of the approach and the theoretical framework of the solver. The geometric domain is based on the properties of space, and it directly works with the space constraints of the environment and ego vehicle (i.e. kinematic constraints). The subsequent problem is dealing with large space exploration and optimization. The heuristic domain depends on special knowledge, such as constraints or data correlation, about the problem. Usually, it is useful to solve more quickly, to find approximate solutions, to avoid algorithm complexity or an ego vehicle blockage situation. Yet, it is generally not sufficient for handling complex problems and does not guarantee that the optimal solution will be found. The logic domain refers to deductive approaches built on assertions. Such assertions are usually made on elementary rules driving the evolution of the environment. Their main advantage is that they easily link the effects to the causes, but they are subject to combinatorial explosion. Cognitive approaches rely on the evaluation of a situation based on prior knowledge on this specific situation and common sense, which is close to the logical way of thinking, for example adhering to driving rules. The main advantage of cognitive processes is their ability to use existing knowledge to gather new information. For the application to autonomous vehicles, the interest is in modeling the decision process formed on human behavior characteristics. This helps to justify the acceptability of autonomous vehicle behavior mimicking human behavior. However, we do not currently have enough experience to validate the effectiveness of such theories. Therefore, we do not consider algorithms that hinge on cognition as evidence. Many driver-based theories agree that driver behavior is too complex and too difficult, or even impossible, to model, as it is not rational. The last domain we would like to explore is the biomimetic domain, which describes physics-inspired approaches. They obey the physical laws, which are convenient to implement but can be stuck in infinite loop motion behaviors. The convergence of the system has to be obtained with feasible solutions.

For each family, we review the basic idea, the specificities, the advantages/drawbacks, the evolution, and the interest for highway planning. One notices that the solutions were first dealing with a specific method in a specific scientific field, and then evolved to multi-field mixed algorithms.

B. Algorithm Classification

1) Space Configuration: The space configuration analysis is the choice of a decomposition of the evolution space.



(a) Sampling points

(b) Connected cells

(c) Lattice

Fig. 4. Space configuration illustrations. The points/cells/maneuvers are candidate solutions in blue, dismissed in red, and the solution in green.

It is a set-algorithm used mostly for motion generation or deformation when specified. The methods are based on geometric aspects; they refer to either a predictive approach with a coarse decomposition to limit computational time, or a reactive approach with a finer distribution to be more accurate. The main difficulty is finding the right space configuration parameters to obtain a good representation of motion and environment [41]. If the discretization is too coarse, the collision risk will be badly interpreted and it will not be possible to respect kinematic constraints between two successive decompositions; however, if the discretization is too fine, the algorithm will have poor real-time performance. We distinguish three main subfamilies for space decomposition, illustrated in Fig. 4: sampling points, connected cells, and lattice. The basic idea is:

1) to sample or discretize the evolution space;
2) to exclude the points, cells, or lattices in collision with obstacles or not feasible; and
3) to either send those space decompositions as free-space constraints, or to solve the resulting space configuration with a pathfinding algorithm (see III-B.2) or a curves planner (see III-B.4) to directly send waypoints, connected cells sets, or lattice sets to the control block.

a) Sampling-Based Decomposition (Fig. 4(a)): Two main routines are used in the literature to return a set of sampling points. The first one chooses the points in the evolution space of the ego vehicle with respect to the kinematic constraints, but with higher calculation time due to the optimization choice of the samples under these constraints. The second one picks random points in the evolution space, so that the algorithm is computed faster but the method is incomplete, not replicable and sensitive to the random points' distribution. Moreover, the links between two points are not necessarily kinematically feasible. Both routines are suboptimal and do not guarantee that a solution will be found if one exists, or that a solution will be returned in a finite computational time. However, sampling configurations are flexible to dynamic replanning and do not require any explicit modeling of the collision space. They can thus be used for a reactive trajectory deformation.

The most popular random method is the **Probabilistic RoadMap** (PRM) [41]. It uses random samplings picked in the evolution space during the construction phase. These samplings are connected to their neighbors to create an obstacle-free roadmap, which is then solved during a second query phase by a pathfinding algorithm, e.g. Dijkstra (see III-B.2) in [42]. In [33], the authors first sample the configuration space based on a reference path, e.g. the ego lane centerline, then select the best set of sampling points according to an objective function, and finally assign a speed profile to the path to respect safety and comfort criteria.

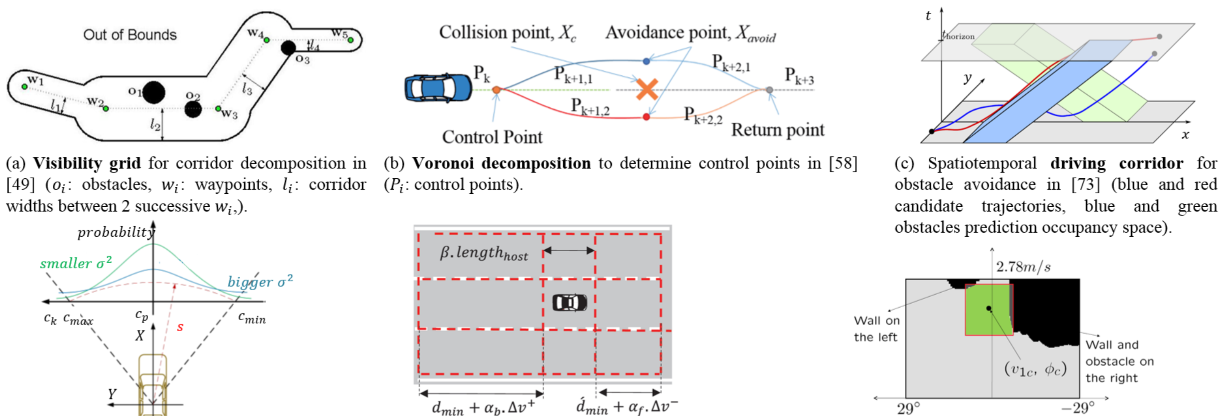
A better strategy is to consider both space and time dimensions in the decomposition. Dealing with **spatiotemporal sampling** points makes it possible to obtain a predictive algorithm, as was done in [43], where points are constrained in a 5-dimensional evolution space with vehicle position, orientation, velocity, and arrival time. Considering the drawback of tiny space and space close to obstacles, adaptive samplings can be adapted to autonomous vehicles, as presented in [44] for robots from 2 to 8 degrees of freedom.

These methods are usually preferred for mobile robotics or autonomous vehicles in unstructured environments. Their use for highly structured highway planning is thus diminished.

b) Connected Cells Decomposition (Fig. 4(b)): The methods first decompose the space into cells using geometry, and then construct an occupancy grid and/or a cells connectivity graph, see Fig. 5 for application examples. In the occupancy grid approach, a grid is generated around the ego vehicle. The information on the obstacles' detection is overlaid on the grid. In case uncertainties are considered, stochastic weights are added to the cells to obtain a costmap representation. The main drawbacks of an occupancy grid are the large memory requirements and high computation time, the false indicative occupation with moving obstacles, and a space- and time-varying resolution. In the connectivity graph approach, the nodes represent the cells, and the edges model the adjacency relationship between cells. The graph can be interpreted as a path along the edges of the cells or a path to find inside the connected cells.

Two strategies are distinguished: those that are based on the obstacles, and those that are not. The representation of obstacles plays a key role for cells decomposition algorithms. Obstacles are usually represented as convex polygons [45], rectangular [46], triangular [47], circular [48]–[50], ellipsoidal shapes [32], [51], based on the obstacle dimensions and speed; or the entire road lane [52]. For non-obstacle-based representation, the cells' organization can be determined offline, and then filled online. The grid is rapidly obtained, but does not take advantage of the environment properties. On the other hand, the obstacle-based representation builds an online grid, which is more computational as replanning is necessary to consider the dynamic of the environment. Whatever the decomposition, probabilistic occupancy can be applied to define a more realistic decomposition proportional to the probability of occupancy, as proposed in [53].

The most intuitive non-obstacle-based method is the **exact decomposition** in Fig. 5(e), which separates the space with vertical and/or horizontal segments. An egocentric exact



(a) **Visibility grid** for corridor decomposition in [49] (o_i: obstacles, w_i: waypoints, l_i: corridor widths between 2 successive w_i,).

(b) **Voronoi decomposition** to determine control points in [58] (P_i: control points).

(c) Spatiotemporal **driving corridor** for obstacle avoidance in [73] (blue and red candidate trajectories, blue and green obstacles prediction occupancy space).

(d) **VFH** to define the probability of selected motion in [57] (c_i: candidate direction, σ^2: variance value for the cost function, s: searching step length (m)).

(e) **Exact grid** used for an occupancy grid in [54] ($\beta >$ 1: adjustment constant, $length_{host}$: ego vehicle length, d_{min} : minimum safe distance between 2 vehicles, $\alpha_{f,b}$: front/back time dependent constants, $\Delta v^{-,+}$: front/back relative speed of ego vehicle and obstacles).

(f) **Dynamic window** to navigate in [61] (the external box is the velocities set that satisfies the ego vehicle constraints, the small box is the velocities set of reachable points in the next time step, the darker regions are in collisions, v_{1c}: ego linear velocity, ϕ_c: ego steering angle).

Fig. 5. Illustrations of connected cells decompositions.

decomposition adapted to the vehicle's size and the obstacle's relative speed is used to exploit an occupancy grid for a Markov Decision Process in [50] and a lane change flow chart in [54] (see III-B.6). Reference [52] fills the occupancy grid with a belief state from the semantic lane information. In [55], 8 overlapping static and dynamic cells are mapped based on the lane configuration and performance of integrated sensors. Considering the nonholonomic behavior of the vehicle, the curvilinear or **polar grids** [53] provide a more realistic decomposition around the ego vehicle. However, as non-obstacle-based occupancy grids cannot evolve according to the obstacles' dynamics, they quickly become obsolete due to the computational cost of a refined decomposition. Another drawback is the lack of accuracy on obstacles' positions.

To consider environment uncertainties, the **Vector Field Histograms** (VFH) from [56] statistically models the evolution space with a velocity histogram occupancy grid as a polar histogram; see Fig. 5(d). Reference [57] proposes a constrained VFH to treat kinematic and dynamic constraints to extend to highway applications. VFH methods are mostly used for deformation function as a reactive planner, as they are robust to sensors' uncertainties [58].

To reduce the search space, decrease the computation time and deal with dynamic obstacles that are either hard to model or not modeled, the **Dynamic Window** (DW) approach in Fig. 5(f), introduced in [59], reduces the search space into a reachable velocity space within a short time interval. It is also used as a reactive planner for trajectory deformation. In [60], DW is directly applied on an image from the camera. Reference [61] combines a DW approach for avoiding unmodeled obstacles with a Velocity Vector Fields (VVF) (see III-B.3). The authors in [62] demonstrate safety recommendations for both static and moving obstacles using DW.

The routines of the second type use the obstacles to set the cells decomposition. The **Voronoi decomposition** [63] builds cells between particular points, which represent each obstacle, mainly using the Euclidean bisection (L2 Euclidean distance norm) or absolute value (L1 Manhattan distance norm); see Fig. 5(b). The method is extended to polygonal obstacles with the generalized Voronoi diagram [64]. The decomposition generate a non-regular grid and is classically interpreted as the way that is equidistant from each obstacle (way along the edges), i.e. a skeleton. Reference [42] uses Voronoi diagrams to reduce the obstacle-free point's space, whereas the authors in [58] propose a real-time algorithm based on one Voronoi cell built on the next collision point in the ego vehicle trajectory. To increase the distance between an obstacle and the ego vehicle, weighted [65] and uncertain [66] Voronoi diagrams have been developed in mobile robotics, but they are not that common for autonomous vehicles. The major drawbacks of Voronoi decompositions are the heterogeneous cells sizes, the kinematic feasibility of linking adjacent cells, and the dynamic evolution, which involves time-consuming replanning. Moreover, the equidistance does not necessarily guarantee safety. In response to these issues, the **approximate methods** split up into thinner cells when the obstacles are closer to obtain a more accurate occupation grid. A classic approximate approach is the quad tree decomposition, as in [67]. Building a dynamic cells decomposition remains the major drawback of the Voronoi and approximate approaches.

A major improvement is then to add a time dimension to the spatial decomposition. The **visibility decomposition** in Fig. 5(a) picks points of interest in the scene representation, and links them with segments if these do not intersect obstacles [68]. They are extended to a path velocity representation in [69]. The points of interest can be based on the obstacles' vertices [70] or along the road border [49]. This implies that the path along the edges can be in contact with obstacles, which is then adapted to the highway planning problem by space shifting the trajectories, as in [70]. In addition, the **driving corridor** representation in Fig. 5(c) uses road boundaries and the spatiotemporal positions of obstacles to produce a set of free spatiotemporal evolutions as corridors [18]. From a static decomposition based on visibility decomposition as graph connectivity, [49] builds a dynamic corridor from **Velocity Obstacles** (VO) analysis. This spatiotemporal algorithm returns the set of all the velocities of the ego vehicle that lead to a collision [71]. A velocity outside the spatiotemporal representation guarantees that there will be no collision under the hypothesis that the obstacle velocity prediction is correct. The passage and region channels are then solved with a connectivity graph [45] or with a space constraint-based optimization method [72]. Reference [73] builds the corridors with the homotopic method to enumerate the possible maneuver variants from a path velocity decomposition and [74] proposes a four collision-free cells partitioning to design a spatiotemporal transition graph. A major drawback of driving corridor algorithms is still their calculation time.

Most static cells decompositions are of little interest in the context of highway planning. Methods that are only based on obstacle decomposition exclude the road geometry, which is at the core of highway driving. In contrast, algorithms that benefit from both road constraints and dynamic obstacles are of major use for highway planning.

c) Lattice Representation (Fig. 4(c)): In motion planning, a lattice is a regular spatial structure, which is a generalization of a grid [22]. It is possible to define motion primitives, that connect one state of the lattice exactly to another. All the feasible state evolutions resulting from the lattice are represented as a reachability graph of maneuvers. The main application of lattice methods is predictive planning. The advantage of a lattice representation is the consideration of the kinematic constraints implicitly handled by the motion on the lattice [75], as well as a spatiotemporal consideration. Moreover, the lattice can be calculated offline for a quick replanning [76]. Unfortunately, their application to reactive planning is mostly limited due to the fixed structure.

The classic lattice representation is based on the **maximum turn strategy** [13], [76], where only the turning radius of the ego vehicle is discretized to propose different curved paths. As an improvement, the velocity (speed and acceleration) is considered with a **curvature velocity** method as presented in [77], and extended to autonomous vehicles in [72], [78]. The main drawbacks of these methods are the rigidity of the predefined motion set and the high density of the motion graph required to reach the goal position. Nevertheless, it is possible to define an **environment-adapted lattice**, in contrast to the previously discussed lattices based on predefined motion. The authors in [29], [79]–[81] operate regular sampling points over the spatiotemporal evolution space based on highway lane marks and centerlines. The use of curves to connect the sampling points provides a curved lattice graph set-algorithm. Other approaches adapt the lattice to the driver's behavior for a priori maneuver, as done in [82] for lane changing.

Lattice representations compile both road boundaries and kinematic constraints, and can be quickly replanned, which is useful for highway planning. Nonetheless, the structure's iteration memory requirement and long-term advantage represent a burden for fast computing path planning on a highway. As will

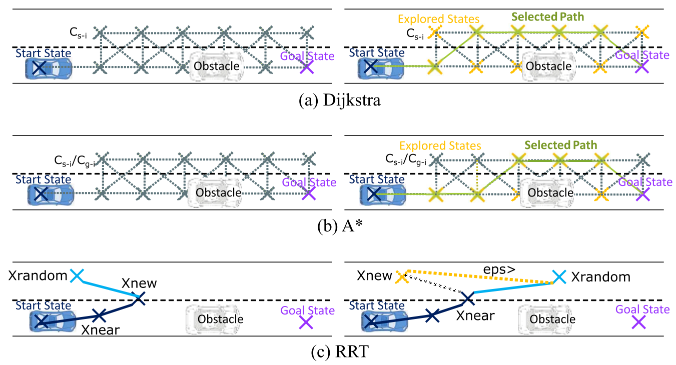


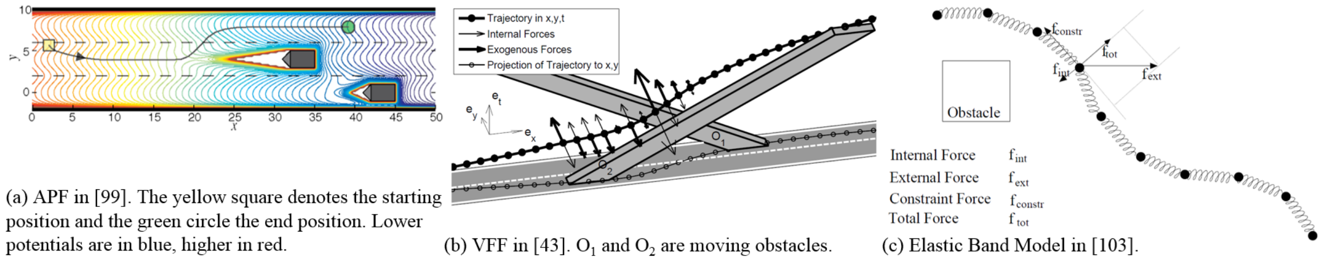
Fig. 6. Illustrations of the processes of (a) Dijkstra, (b) **A***, and (c) RRT.

be presented in III-B.4, curve iterations as tentacles are favored over lattice representation for highway motion planning.

2) Pathfinding Algorithms: The pathfinding algorithm family is a subpart of graph theory in operational research used to solve combinatorial problems under a graph representation. The graph can be weighted or oriented with sampling points, cells, or maneuvers nodes. The basic principle is to find a path in a graph to optimize a cost function. Traveled distance, fuel consumption, and comfort are the main cost functions for highway planning problems [42], [72], [83]. The graph resolution is based on logic and heuristic methods, which are mainly solve-algorithms and refer to the decision function even if they do not apply any decision but a selection. The subfamily of Rapidly-exploring Random Trees (RRT – see below for details) comprises both set- and solve-algorithms, with motion generation and selection. The main use of pathfinding algorithms is for route planning, but they adapt well for local planning and applications to highways as predictive algorithms. As benefits, these algorithms are universal and widely used, and solve either known or unknown environments. The main drawbacks are their dependency on the graph size and complexity, which affects the choice of the solver, and their need for detailed information on the space configuration, which makes them slow in vast areas. We will restrict our review to the most frequently observed algorithms for highway autonomous driving.

For known environments, the graph is previously generated by a space configuration algorithm to model the connectivity of the evolution space (see III-B.1). In the **Dijkstra** algorithm from [84], the author details a method to "Find the path of minimum total length between two given nodes P and Q", which becomes a very popular graph solver for motion planning application to autonomous vehicles [42], [72], [83]; see Fig. 6(a). As the algorithm uniformly explores all the directions, it finds the optimal path with respect to the cost function, but its computational time is high.

This drawback was first reduced with the **A*** algorithm introduced in [85], and recently tested on autonomous vehicles' replanning in [86]; see Fig. 6(b). It consists of applying Dijkstra's algorithm with a heuristic search procedure on the goal-node to expand the fewest possible nodes while searching for the optimal path. The heuristic should always be optimistic, i.e. the real cost should be higher than the heuristic cost,



(a) APF in [99]. The yellow square denotes the starting position and the green circle the end position. Lower potentials are in blue, higher in red.

(b) VFF in [43]. O_1 and O_2 are moving obstacles.

(c) Elastic Band Model in [103].

Fig. 7. Illustrations of attractive and repulsive forces approaches.

as otherwise the minimal path will be distorted. As an example of a heuristic evaluation function, [76] chooses the distance to both ego lanes' borders and the traveled distance for Dijkstra's cost function, whereas [87] adds to the travel time function the distance to the goal and hazardous motions penalizations as heuristics. It is also possible to weight the heuristic function to reduce the calculation time, as described in the **Anytime Weighted A***(AWA*), which guarantees the finding of a solution with a non-admissible heuristic [43]. Besides, when considering kinematic constraints, the approach of **hybrid-state A*** search in [88] applies a first heuristic to consider nonholonomic constraints, and then a second dual heuristic that uses an obstacle map.

A further disadvantage of Dijkstra and **A*** stems from dynamic environments. In fact, at each time step, the graph has to be reconstructed. To avoid a high time calculation and dealing with partially known environments with dynamically changing weights, a heuristic improvement consists of a dynamic cost graph search, as does the **D*** algorithm [89] in [90].

For unknown environments, the **RRT** subfamily [91] constructs its own nodes in the evolution space, as illustrated in Fig. 6(c). The reasoning is close to the PRM, except that the nodes are built from one to another and the output is a path (to solve the nodes connection if a noncollision and kinematically path exists). Thus, it guarantees kinematic feasibility and can be used for a reactive generation. Authors in [92] demonstrate a fast RRT for replanning trajectory. As for Dijkstra's algorithm, there are a large number of evolutions for RRT algorithms in mobile robotics [93], but currently few applications for highway driving, such as the example in [94], which looks at more efficient nearest-neighbor techniques with probabilistic optimality in **RRT***. For a randomized graph, the main drawback is the randomly collected sampling nodes, which may result in a poor connectivity graph and no replicability. A simple way to increase the connectivity is to add a probability function of generating intermediate points in a specific area, as done by the authors in [42].

Similarly to sampling-based decomposition, probabilistic graph search is not well suited to a highway structured environment. Besides, the highway is usually a known environment, easily represented with space configuration algorithms in III-B.1. In that sense, deterministic pathfinding is favored in highway motion planning for autonomous vehicles.

3) Attractive and Repulsive Forces: The attractive and repulsive forces approach is a biomimetic-inspired method. The evolution space is symbolized as attractive forces for desired motions (e.g. legal speed), and repulsive forces for obstacles (e.g. road borders, lane markings, obstacles). The main advantage is thus to be reactive to the dynamic evolution of the scene representation. The motion of the ego vehicle is then guided by the resultant forces vector, so no explicit space decomposition is needed. Reference [86] shows how parabolic and conical functions are well suited as potential functions. The resolution of the resultant vector is achieved by either a gradient descent method [86], [95] – a simple resolution without a vehicle model, or by the application of Newton's second law [96] – based on a vehicle model, which provides a feasible motion under kinematic constraints. The attractive and repulsive forces approach both sets and solves the motion planning problems in a continuous space representation. As the modeling of all the evolution space is time-consuming, these algorithms are mostly used as a reactive motion deformation.

The **Artificial Potential Field** (APF) concept [97] was first introduced to real-time mobile robotics in [98]. Reference [99] adapts a set of four artificial potentials over lanes, road, obstacles, and desired speed, to model the highway functions described in II-C; see Fig. 7(a). In [100], the authors use a framework of electric fields as a riskmap with weighted partial potential to distinguish between emergency reactions and preventive actions. The benefit of time consideration with a velocity potential leads the ego vehicle to progress forward smoothly, as emphasized in [61] with the use of a **Velocity Vector Field** (VVF). Furthermore, APF returns direct control inputs [96] or constraints for the optimization solver [46], [101]. The first drawback for application to vehicle planning is the oscillatory behavior when close to obstacles, but smoothing algorithms overcomes this [95]. The second difficulty is the presence of local minima. Not only is the ego vehicle stuck in a local minimum, but overcoming this issue impacts the smoothness of the path and calculation time. A trivial solution is to add a heuristic to exit the minima with a randomized path, as is done in [86], where local minima are also repulsive artificial potentials. This tends towards the elimination of the local minima afterwards. However, local minima might be necessary in highway planning to keep the ego vehicle safe from inopportune lane changes, as stated in [99].

Extensions to uncertain environments allow to transpose the algorithms from a deterministic robotics environment to highway planning. The **Virtual Force Field** (VFF) introduced by [56] uses the VFH decomposition (see III-B.1) and interprets the probability distribution as potential forces to guide the vehicle along the weakest grids. This method is adapted to highway driving in [43] to deform a trajectory previously obtained with sampling points and **A*** methods; see Fig. 7(b).

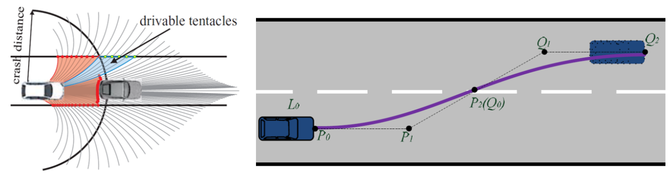
Another drawback of the APF highlighted by [98] is the lack of dynamic reasoning: namely, only spatial dimensions are used for dynamic obstacle avoidance, so that the ego vehicle always tried to avoid the obstacle by bypassing it. To overcome this inconvenience, the **elastic band** algorithm [102] models the environment as a spring-mass system, considering N discrete nodes on which potential forces are applied [48]; see Fig. 7(c). Reference [103] applies elastic band to car following as a deformation of the leader vehicle path and [104] develops a time-related elastic band framework with temporal waypoints for lane changing.

Attractive and repulsive forces are widely implemented for reactive planning in all kinds of robotics. Except the high time calculation, they prove their efficiency for highway planning, thanks to a straightforward integration of the scene representation, provided that the potential functions are well chosen within the environment.

4) Parametric and Semi-Parametric Curves: Parametric and semi-parametric curves are major geometric methods in path planning algorithms on highway for at least two reasons: (i) the highway roads are built as a succession of simple and predefined curves (line, circle, and clothoid [105]); and (ii) a predefined set of curves is easy to implement as candidate solution sets to test. Moreover, as some of the curve-based algorithms directly take into account the kinematic constraints of the vehicle, they are widely used to complement other methods. Geometric considerations are usually separated from the dynamic constraints. For instance, the authors in [42], [106] first solve a static problem considering the geometry of the road, and then solve the dynamic dual problem by searching for a speed profile built to adapt to the curvature profile of the road and to respect the constraints of the dynamic obstacles. In contrast, [107] first fixes the speed profile and then deforms the curve path. Those decoupled approaches could lead to a long time reaction for the vehicle in case of a blocking situation or replanning. Otherwise, curve planners are well suited to a predictive approach too, as they define a motion from a start to a goal point or cell or maneuvers. This property is also suitable for replanning stages.

We distinguish between two exploitations of curve algorithms. First, the point-free curves subfamily is used to build kinematically feasible trajectories as a set of candidate solutions (maneuvers). Second, the point-based subfamily uses curves to fit a set of chosen waypoints (sampling points or cells). The first routine is a set-algorithm and needs a decision-maker to return the most convenient maneuver, whereas the second one requires a space decomposition before fitting a path or a trajectory, and is thus a solve-algorithm. Both are considered for the generation function.

The point-free curves subfamily refers to the principle of lattice, called tentacles algorithm, as introduced in [108]; see Fig. 8(a). Instead of a space decomposition, the tentacles are based on primitives parametric curves, such as lines and circles, clothoids, and sigmoids. Each tentacle is obtained with different lateral, i.e. steering wheel angle, and/or longitudinal, i.e. speed, parameters. Moreover, as the search space for solutions is reduced, the computational time is limited compared to the space decomposition methods. The tentacles can also be calculated offline, as a trajectory data base [109].



(a) Tentacles scheme in [108]. (b) Bézier curves. P_i, Q_i are control points [116].

Fig. 8. Illustrations of the point-free and point-based curves methods.

Reference [110] shows that **line and circle** paths are the solution for curves of minimal length with constraints on curvature and start/end positions. In [107], the authors work with a lane-based zone model built on the linear minimum and maximum trajectories. Those curve configurations are tested on tracks in [42], [109]. Despite the simplicity and good behavior with high curvature, the second order of line and circle curves is not continuous and hence not realistic for the curvature continuity of the vehicle model. Regarding the road design, lines and circles are linked with **clothoid** functions to obtain a continuous curvature function. Indeed, the clothoid has its curvature proportional to the curvilinear abscissa [50]. This condition is important for the limitation of the lateral acceleration and thus the vehicle's comfort [106]. For example, [111] selects clothoid tentacles for overtaking trajectories based on clearance, change of curvature, and trajectory orientation criteria. The authors in [112] propose a fast method to generate a piecewise clothoid curve in agreement with a reference line (e.g. ego lane centerline for highway planning), and kinematic constraints. Conversely, [113] focuses on clothoid path sparsification to perform better optimization. However, the clothoid presents an iterative construction process, which increases its calculation time.

The use of straight lines, curves, and arcs based on clothoids paths are also favored to generate reference trajectories in a road-aligned coordinate system [114]. Nevertheless, the highway curvature is usually small enough for acceptable approximation of a straight road. Under this assumption, only two path's geometries exist for the vehicle: going straight with a straight line or changing lane. For the second case, the **sigmoid** or S-function appears to be an easy solution [115]. The authors in [32] use sigmoids to generate different candidate trajectories with acceleration profiles based on experiments from different driver behaviors.

The point-based curves subfamily is well suited to geometrically constrained environments and to ensure that the dynamic constraints of the ego vehicle are respected. The principle is to determine control points in the environment, and to fit them with a curve. They can also be used for a smoothing step in other motion planning algorithms. In [29], the authors use a cubic spiral for path generation, as do [80], and a cubic function of time for velocity generation, whereas in [54] quartic and quintic polynomials are respectively used to generate longitudinal and lateral motions. In [81], the authors compare cubic **polynomial functions** and smoother quartic curvature polynomials to generate the path with cubic polynomial speed profiles, based on a sampling point lattice.

The use of semi-parametric **spline curves** is an improvement on polynomial interpolations, which are difficult to find and increase in complexity. This entails defining curves as a set of piecewise polynomials. In this way, the obtained polynomial equations are of a lower degree but there are a large number of polynomials to deal with. For example, [79] generates quintic spline trajectories adapted to the road shape. The authors in [76] develop adaptive polar splines with non-zero curvature at the beginning and end segments to suit highway maneuvers. Among spline curves, we distinguish **Bézier curves**; see Fig. 8(b). They use control points instead of interpolation points as inflection points. The inconvenient is thus not to pass through the defined control points, except for the start and end points. The advantages are their simple implementation and thus a low computation cost compared to the previously discussed methods. Reference [49] smooths the primitive path from three waypoints with a quadratic Bézier curve to guarantee kinematic feasibility and avoid static obstacles. In [116], the authors also use piecewise quadratic Bézier curves based on safe lane change distances for autonomous vehicles and ride comfort. It is also possible to provide the control points of the Bézier curve from an SVM (see III-B.6), as in [117], which calculates three 4th-degree Bézier curves.

As previously mentioned, the curve algorithms are largely used to interface with all others motion planning algorithms. The choice of the curve type mostly depends on the type of problem and the knowledge of the environment.

5) Numerical Optimization: The optimization problem for motion planning is defined as a solve-algorithm based on logic and heuristic approaches. They are part of decision and generation functions. The optimization is usually expressed as the minimization of a cost function in a sequence of states variables under a set of constraints, and is part of competitive combinatorial operational research to avoid a combinatorial explosion. In motion planning applications, we distinguish two domains of interest, as described in [85]. The first one focuses on finding efficient algorithms to solve complex problems and to improve search time with a heuristic approach (see III-B.2 for some heuristics details). The second one is the mathematical study of the problem to deduce particular properties to find a predictive solution in a restrictive space. As the first domain is commonly used to decrease the computation time of algorithms, only the second domain is further discussed in the following.

The considered approaches model the ego vehicle and environment constraints in a well-defined mathematic form. The main advantages are that they easily handle the constraints of the problem, they deal with multicriteria optimization, and they consider the state dimensionality and kinematics of the vehicle model [118]. The basic resolution is the **Linear Programming** (LP). In LP formulation, the algorithm solves a linear cost function under linear equalities or inequalities. The Simplex algorithm is one of the most popular ones; see [81]. In [119], a spatially based trajectory planning with Sequential LP (SLP) is proposed. For nonlinear problems (NLP), nonlinear optimization is used either in the special case of nonlinear regression problems, such as the Levenberg-Marquardt algorithm on path optimization in [29], [104], or in nonlinear integration, as explained in [33] with a Boundary Value Problem (BVP) solver. For multi-objective problems, the use of **Quadratic Programming** optimization (QP) involves an iterative search of a convex approximation solution to the original problem, as in [18] with sequential quadratic programming (SQP) optimization on distance offset, velocity quadratic error, acceleration, jerk, and yaw rate functions. In [120], [121], the authors formalize the problem of lane changing and overtaking with a Mixed Integer Quadratic Programming (MIQP).

For specific predictive applications, resolution under **Model Predictive Control** (MPC) is highly popular [46], [47], [51], [113], [118], [122], [123]. MPC algorithms solve the problem at each sampling time to find a predictive motion solution over a longer horizon time, but only apply the first sequence of actions. In that respect, MPC models a receding horizon control and shifts the solution set to remain accurate to upcoming information. The main advantage of MPC algorithms is their replanning ability, but they are still too poor for non-convex and high complexity problems.

Dynamic Programming (DP) draws its efficacy for complex computational problems by breaking them into simpler subproblems, even with interdependency. The resolution of each subproblem is combined to find the global problem solution. This implies that the description of the problems has good characteristics. In [49], the author searches for the shortest path in a cells decomposition using a pre-calculated DP solution. The authors in [124] use DP to calculate the optimal cost-to-go value of a set of maneuvers for different speed profiles.

Numerical optimization is widely used in motion planning, either to decrease the solving time of a graph's exploration, or to exploit the mathematical properties of the problem. These algorithms can be solved by generic numerical resolution tools. The main encounter frameworks in the literature are CVX and CVXGEN softwares, Gurobi and YALMIP solvers, Matlab Optimization Toolbox, NPSOL package, and ACADO Toolkit.

6) Artificial Intelligence: The main contribution of Artificial Intelligence (AI) for autonomous driving is its ability to reproduce and simulate drivers' reasoning and learning. These techniques rely on thinking and acting consistently with the environment, a memory structure, and drawing inferences. In this sense, AI algorithms are particularly interesting for the decision making function, as discussed in [34]. They are also well suited to mobile robotics, as they are flexible, adaptive, and reactive to their environment. Moreover, AI techniques are well organized to deal with huge, incomplete, or inaccurate data. The advantages of AI-based algorithms are their capacity to answer generic questions and to absorb new modifications without affecting the structure of the algorithm. They are mostly employed as solve-algorithms for predictive planning, but also as symbolic set-algorithms, and less frequently for reactive deformation. AI gathers a wide diversity of methods from logic to cognitive representation. We propose to

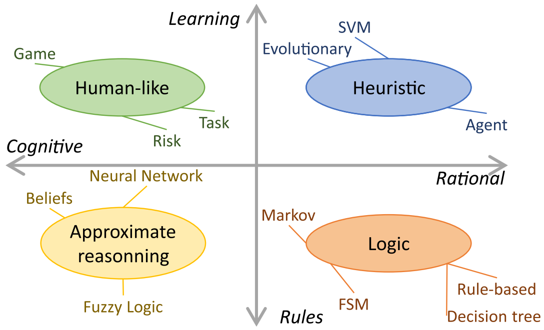


Fig. 9. A map of AI-based algorithms.

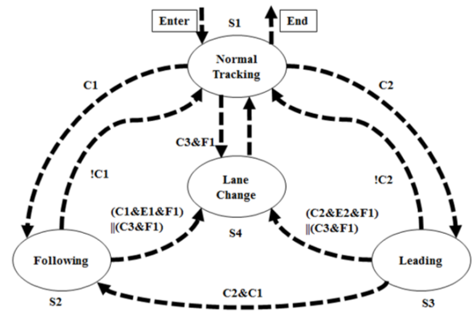


Fig. 10. Illustration of an FSM for highway [129]. C_i, E_i, F_i are specific conditions to transit between the vehicle behaviors S_i.

organize this section in two main axes – cognitive/rational and rules/learning distinctions – based on [125]'s distinction between thinking and acting humanly or rationally. We thus distinguish the four subfamilies depicted in Fig. 9: logic approaches, heuristic algorithms, approximate reasoning, and human-like methods.

a) AI Logic-Based Approach: AI logic-based approaches are symbolic planning used in decision making. They define expert systems to solve specific complex tasks depending on a knowledge base with an inference engine to automate the reasoning system. In case of modification, addition, or removal in the knowledge base, the inference engine should also be updated, and a recursive mechanism must be applied to guarantee the convergence of the new expert system. AI logic methods serve as set- and solve algorithms, while generating or selecting a set of time/space states or actions. Furthermore, their fast architecture allows their use for predictive or reactive planning. The main advantage of these systems is their intuitive setup to emulate human logic and rational reasoning. On the other hand, the knowledge base requires the discretization of numerous environment variables with a high number of cause-and-effect rules and tuning parameters.

The best-known inference engine is **rule-based** reasoning. The statements are if observations, then actions. The authors in [126] report satisfying results of rule-based intention prediction on highways, and use it to perform lane change maneuvers in [127]. The major advantages are to clearly identify the cause and effects, the notational convenience and the straightforward implementation. The main drawbacks are the cyclic reasoning and the exhaustive enumeration of rules, which lead to infinite loop and impact the computation time. Furthermore, if the current situation does not match the observations in the rule base, an unsuitable default decision may be made. In such a case, the knowledge data should be enriched and modified offline. Moreover, as a declarative reasoning, the rules' order matters and resolution conflicts can happen. A solution would be to add a heuristic to prioritize the rules.

To display rule mechanisms, **decision trees** are promoted as compressed graphical representations and decision support tools. In [78], a decision tree is depicted by enumerating all the possible navigation lanes. To facilitate the organization of the tree, binary decision diagrams or flowcharts are developed to represent Boolean functions, as applied in [32], [128]. While decision trees are simple to interpret, however, calculations become highly complex with uncertain or approximate values. On the other hand, decision rules must be interpreted to ensure safe behavior and to detect and anticipate non-legal and dangerous behavior of other vehicles, as developed in [107] with the concept of legal safety.

To avoid the exhaustive rules declaration, the **Finite State Machine** (FSM) gives an abstract model of the system behavior, representing the system states linked by actions/conditions. In [55], the FSM separately describes two longitudinal and four lateral state transitions with a specific contribution to an emergency stop assistant on highways. Compared to the rule-based approach, FSMs directly perform a predetermined sequence of actions and states, which are then mapped with path generation and control, as done in [129]; see Fig. 10. They can also be considered as state classifier algorithms [54], and thus represent an easy communication tool for collective and driver-shared driving. FSMs are not imperatively deterministic, which allows for more complex states' relationships. Reference [18] exploits multiple state charts in parallel to deal with concurrent states, which are well suited to performing simultaneous actions in a decision process (e.g. yielding and merging). The main disadvantage of the previously discussed FSM representations is that they are only based on certainty in knowledge and cannot be generalized to unknown situations.

In case of uncertainty, **Bayesian networks** using Markov models are employed. They are statistical representations of causal links, based on probabilistic transitions. They present a knowledge-based identification step of their parameters to determine the most likely sequence of states to the sequence of outputs. Reference [27] matches the most expected obstacles' behavior intention to return the best ego behavior maneuver (stop | cruise | accelerate | decelerate and turn-right | turn-left | straight). The authors in [50] develop a **Markov Decision Process** (MDP) on the choice of tentacle trajectories, and the one in [130] for a lane-staying or -changing decision. The advantage of MDP is their ability to evaluate several predictions at the same time. Behavior improvements are shown using a Partially Observable Markov Decision Process (POMDP) with a probability distribution over the set of possible states in [131]–[133].

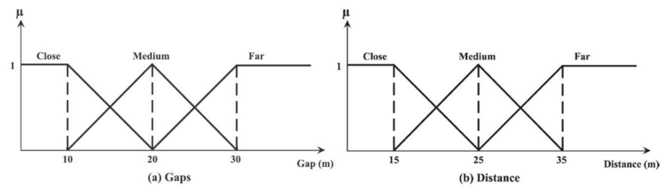
The AI logic-based methods are mostly suitable in constrained and predictable environments, such as nominal highway driving. Their intuitive and fast architectures are widely promoted for use in critical safety environments, where cause-and-effect reasoning is necessary. On the other hand, their lack of autonomy and rigid program structure for an adaptable and reconfigurable algorithm are their main disadvantages for use in open environments such as highways. As the logic-based approaches are straightforward to implement, no specific framework stands out in the literature.

b) AI Heuristic Algorithms: Heuristic algorithms are experience-based and conducive to natural environment process exploration. They aim to find an approximate solution, and are therefore used as faster and more efficient algorithms when the traditional exhaustive methods fail. In motion planning, they usually return a set of actions, but are also able to return paths or trajectories and to act as set- and solve- algorithm in predictive or reactive planning. Their main advantage is their low computational time and complexity, and their ability to handle complex problems. On the other hand, they provide, by definition, a local solution, which does not guarantee global optimality and accuracy.

The most convenient interpretation of heuristic methods involves an **agent** representation. An intelligent agent is an autonomous entity modeled with a rational and social behavior, which adapts to the observations of the environment. Its behavior can be based on condition-action rules (only depending on the current perception), a world model (how it affects the environment), goals to achieve, or utility to goals, e.g on a game theoretic formalism in [133]. The faculties of heuristic agents allow the inclusion of multipolicy decision making, such as in [132], including distance to goal, lane choice bias, max yaw rate, and policy cost. They are especially well suited to uncertain environments. The main drawback is the difficulty of ensuring convergence towards the solution.

Learning methods are also introduced to the decision in heuristic approaches. **Support Vector Machines** (SVM) are statistical learning classifiers for agent intentions which depend on information search algorithms. In analogy with an FSM, they are based on states' classes and in-between margins. The separation of the classes has to be trained beforehand, and full labeling of input data is needed to return a convenient classification. In [117], the authors develop an SVM to provide the control points to a Bézier curve-fitting method. Authors in [134] define an SVM for personalized lane change decision based on the relative velocities and positions.

Evolutionary methods are a more widely used class of learning algorithms. They are defined as meta-heuristic functions, inspired by biomimetics with a natural rational evolution process, such as reproduction, mutation, recombination, and selection. The first step is to determine a set of a priori solutions associated with a fitness function to evaluate their quality. A set of evolution processes is then applied to find a better solution to the optimization problem. In [135], the authors develop the reasoning system SAPIENT, based on a population-based incremental learning, to define the most appropriate parameters for a given task to solve tactical driving problems. More recently, [136] applies two genetic algorithms to refine a fuzzy control module, and [137] details a genetic algorithm with a 4-chromosome structure based on speed, angle, break and time to avoidance. The risk of such algorithms is linked to the mutation mechanism, whose random process can lead to local minima. They are of particular interest for agent swarm methods; see V-C for references.



Rule no.	IF				THEN
	$\tilde{G}_{FA}$	$\tilde{G}_{PA}$	$\tilde{D}$	$\tilde{G}_{PB}$	$\tilde{C}$
1	Close	Close	Close	Close	No
2	Close	Medium	Close	Close	Yes
3	Close	Close	Medium	Close	Yes
⋮					
49	Far	Medium	Medium	Close	Yes
50	Far	Close	Close	Close	Yes
51	Far	Close	Close	Medium	Yes

Fig. 11. Illustration of the fuzzy logic decision in [138].

AI heuristic algorithms are a good alternative to address the disadvantages of classical methods. Their attributes are acceptable for highway driving, where an optimal solution is not necessary and approximate solutions could be sufficient. There is no AI-heuristic framework built for the autonomous vehicle applications, however, major frameworks are available to solve this family, e.g. JADE for agents, DEAP, Jenetics or the Matlab Global Optimization Toolbox for evolutionary algorithms.

c) AI Approximate Reasoning: AI approximate reasoning mimics human reasoning. We distinguish the logic approach, as described earlier with expert systems, with the difference that the knowledge base is non-Boolean; and the learning approach to classify new knowledge to adapt to future situations. The first presents the advantage of exhibiting intelligent behavior with intuitive demonstration and explanation, whereas the second is related to a system able to understand, think, and learn.

Compared to Boolean logic, **fuzzy logic** relies on many-valued variables. It consists of fuzzy expert systems based on Boolean compromises, which mathematically model vagueness and are close to cognitive reasoning with an inductive logic programming on different attribute-level solutions. They return a decision, usually expressed as a maneuver or task, and are thus part of predictive or reactive solve-algorithms. Their major advantages are the flexibility and permissiveness of the designed rules and by extension to uncertain data. On the other hand, their main drawbacks are their lack of traceability and the absence of a systematic design methodology. The authors in [138] propose a fuzzy inference decision system to check whether the ego vehicle has enough time to change lanes based on a close, medium, or far evaluation of the gaps between the surrounding obstacles; see Fig. 11. References [139] and [140] use fuzzy rules on lateral and angular error, respectively, to propose automatic overtaking maneuvers and lateral following of a reference map.

Artificial Neural Networks (ANN) are a popular form of approximate learning solutions. ANNs imitate the low-level structure of biological neural networks, i.e. connected neurons in overlaid layers. The learning process is based on error retropropagation to adjust the neural connections' weights with different learning strategies, e.g. supervised, unsupervised, reinforcement learning. They are used for both generation and decision making functions, as a set- and solve-algorithm in predictive and reactive approaches. Their main drawback for autonomous driving is the absence of a causal explanation to a solution, while their main advantage is their ability to learn by training on multidimensional data. Another disadvantage is thus the large amount of data required, which makes them computationally expensive. Reference [122] proposes a combination of Hidden Markov Models (HMM) and Gaussian Mixture Regression to mimic the driver's acceleration and braking model, whereas [141] learns the driving speed profiles with a Levenberg-Marquardt algorithm on a 3-layer ANN with 20 hidden nodes trained on 600 driving trips. The authors in [142] imitate driver trajectories for lane changing with **Convolutional Neural Networks** (CNN), and [143] compares the learning stage with human driver control inputs. **Reinforcement learning** is used in [133] to determine the policies for each agent in the scene representation, or for multi-goal overtaking maneuvers in [144] and automated lane change maneuvers in [145]. Some new promising research is in progress using the deep learning methods [146], or by integrating the perception and planning blocks in **end-to-end learning** [147]. Neural networks are also extended in case of uncertainty with Bayesian probabilities in **belief networks** [148]. However, one of the challenges of these methods is ensuring that they learn in the correct way with the diversity of environments and behaviors of driving.

AI approximate reasoning algorithms seem to be highly promising for the near future. They rely on logic and statistical bases, which are extended to cognitive properties. They also provide adaptive reasoning to evolve appropriately to their environment. At the moment, the main impediment to these algorithms is the lack of feedback in real driving scenarios. As the real-time implementation of approximate reasoning algorithms is complex, the use of specific frameworks is favored, e.g. Matlab Fuzzy Logic Toolbox for fuzzy logic approach or TensorFlow for machine learning.

d) AI Human-Like Methods: AI human-like methods propose high-level models that mirror human processes for solve-algorithms. We do not pretend in this review to provide a complete description of driver models; for this, the readers are invited to refer to [149]. The AI human-like methods discussed in this article are decision making functions, based on human cognitive aspects. They are intelligent, potentially learning, and cognitive procedures. Their knowledge of the environment rules and ability to handle various and complex situations make them useful for predictive and reactive planning. Their main advantage is an abstract and universal representation of the decision making process. However, they are difficult to model and to use accurately. In [28], the author highlights different driver models based on taxonomic or functional, and behavioral or psychological distinctions. With respect to his decomposition, we categorize AI human-like methods into three main approaches based on risk, task, and game theories.

Risk estimators are employed to interpret rational decisions from the scene analysis with a cognitive bias. The main advantage is to be intuitive. This implies a balance between a subjective level of acceptable risk and the objective safety, which is a negative aspect for a global risk assessment. The authors in [31] identify two notions to evaluate the risk of a situation: risk of collision and risky unexpected behavior. In the scope of this review, we only details algorithms for the risk of collision. The most basic risk estimator uses a binary collision prediction. Considering a risk criterion, a threshold is applied to the obtained value to classify it as risky or safe. Indicators such as **Time-To-Collision** (TTC) [51], [55], [150] or **Time-To-React** (TTR) [151], [152], e.g. steering or braking, give measurable estimators to the driver too. This method is also exploited in [153] to return Inevitable Collision States (ICS) for grid occupation representation (see III-B.1). The **Time-To-Intercept** (TTI) indicator is extended in [154] to return a position-velocity shadow target to prevent obstacle collision. Binary risk estimators provide a coarse evaluation of the scene, but uncertainty of the obstacles' future motion is a parameter not to neglect. Probabilistic risk estimators are better suited to providing a more realistic scene representation.

To adapt the risk perception to the driver, **compensation risk** estimators introduce driver states such as stress, drowsiness, or illness. In [155], the authors use a risk-speed compensation model, i.e. the product of perceived risk and the driver's speed is constant. Reference [156] exploits a risk homeostasis theory to deal with uncertainties over subjective risk assessment. Furthermore, the authors in [157] introduce a **Time of Zone Clearance** (TZC), combined with the perceived risk on speed, distance, safety, and comfort, and based on the proposition that risk appears if trajectories overlap.

Even if compensation risk estimators seem to be more faithful to the human decision, the above risk approaches are usually based on one factor of risk, whereas risk warning should take into account all factors of the scene representation. Reference [158] proposes a two-level risk estimator affected by weather, traffic, or road conditions, and then refined with real-time information about the ego vehicle surroundings. Promising risk assessments have been developed combining usual risk estimators and belief theory, as proposed in [159] to relax thresholds applied to fuzzy sets.

The second approach comprises **taxonomic models**, which are also popular symbolic models for human decisions. They identify the sequential driving tasks relations, with behavioral and ability requirements. Those methods call for a logical organization, and represent operational research task scheduling. They must then be interpreted either as space-time-action constraints or with a local planner, e.g. the curve planner discussed in III-B.4. One of the disadvantages is their difficult explicit description, as they are either focused on a driving task, such as lane changing or merging, or hardly exhaustive. However, they present the advantages of providing an abstract and universal representation and easy replanning. For example, McKnight and Adams list 45 major driving tasks in [160]–[163], decomposed into 1700 elementary tasks; see Fig. 12. References [139], [164] define the sequence of operations to perform an overtaking maneuver; each action is then validated under a set of numerical criteria to prompt

```
                    Task 32: Passing
32-1   Decides wether to pass (two- or three-lane roads)

32-11   Looks along roadside for no passing control signs
32-111   Does no pass if « no passing » zone is indicated or has been
         indicated previously
32-112   May pass if sign indicates end of « no passing » zone

32-12   Observes lane markings
32-121   Does not pass if left side of lane is marked by the following:
32-1211   One or two solid lines
32-1212   Solid line to the right of broken line
```

Fig. 12. Illustration of the first elementary tasks for "passing" in [161].

the next action. The authors in [83] propose a functional architecture of the driving strategy as a discrete set of behavioral strategies for a specific traffic situation.

Third, some learning approaches are inspired by interaction models introduced in **game theory**. The idea for highway planning is to consider the moving vehicles as players that observe each other's actions and consequently react with an appropriate strategy. The main advantage of this method is that it quickly obtains a trained driver model, starting with a merely basic one. The main drawback is that it assumes that all the players respect the rules, and it can therefore lead to unsafe reactions in real-life applications. Moreover, one should make sure to learn using various behaviors of the players to enrich the knowledge. In [133], the models develop more complex strategies as they are trained against the other behaviors. The authors use POMDP to model the players' knowledge and Jaakola reinforcement learning in the training phase.

AI human-like methods are well suited to decision making in highway scenarios, where drivers' behaviors are more predictable due to the basic rules of this environment. They are also easy to understand and to share with the driver. Moreover, the application of such algorithms is usually not as complicated as modeling a driver, but still interesting enough to involve in complex scenarios. With their simple architecture and heterogeneous implementation, one notices that no major framework is highlighted in the literature of autonomous driving for AI human-like methods.

IV. Comparison Table for Highway Applications

The highway applicability of the previously described methods is summarized in Table II. We propose to quantify the constraints' assessment with a '–/∼/+' scale. We assess '– –' as being highly inappropriate, '–' inappropriate, '∼' intermediary, '+' appropriate, and '++' highly appropriate. The references illustrate the use cases of the families to suggest their situation adaptiveness (collision avoidance, car following, lane change, merging, overtaking) and their implementation in simulation or experimentation, specified with the square brackets' superscript and subscript respectively. To quote but a few, simulators such as CARLA, PreScan, SCANeR studio, TORCS are developed for autonomous driving.

Unlike the classification in [25], we do not focus on the usual points of comparison, such as completeness, optimality, or time complexity, but on how the algorithms offer an effective and efficiently implementable response to practical applications. Therefore, one can interpret Table II as a guide to choose the most appropriate family given the attributes of the algorithm, and its intrinsic and extrinsic limits. Furthermore, one will notice that references often apply to different families. In that sense, Table II also helps to understand the complementary methods for a systemic motion planner design.

Among the characteristics highlighted in this review, the taxonomy criteria can be found in III-A: (i) the **type of use** of set- and solve-algorithm, (ii) the predictive or reactive **horizon**, and (iii) the **mathematical domain**; and the motion planning **functions** in II-B (Fig. 2).

As intrinsic limits, we distinguish the performance, ease of use, and data analysis. The performance first gathers the **real-time** implementation. In the case of highway motion ($> 17m/s$), with an error of a meter, the motion planning algorithm should return a solution within the order of magnitude of $10ms$ to be real-time. As the intrinsic cost of each algorithm strongly relies on the hardware platform, we inform about qualitative, but not quantitative computation. The other performance requirements are derived from the analysis proposed in [97]: **robustness** of the algorithm to find a solution despite the variation of the environment (merging, jam approach); **stability** to keep the solution despite environmental changes; **adaptability** to perform a solution in various conditions, e.g. introducing new scenarios or criteria; and the **feasibility** of converging to a solution in a finite time. The ease of use is defined by the ability to **replan** in real-time without changing the structure of the algorithm. The last intrinsic limit relates to the data analysis: the **input type** for the scene representation (discrete, sampled, continuous) and the **output type** of the algorithm (space, path, trajectory, maneuver, task), along with the ability to deal with **uncertain** data.

The extrinsic limits show the algorithm's dependency on high-level sensors of the ego vehicle and/or the infrastructure. This aspect is important to reflect the real applicability in a near or further future. If the algorithm works well only with precise information over a large time horizon, which does not correspond to the current high-level sensors' properties, it will be highly inappropriate for **sensors constraints**. We invite the readers to consider [38] as a reference for sensors' properties. As stated before, the specificities of the driving environment and vehicle kinematics must be considered. We therefore consider whether the algorithm performs well in **complex environments** (dense traffic, various topologies) and takes the **constraints of the environment** and the **ego vehicle kinematics** into account.

V. Future Research Directions

This part spotlights some critical and forthcoming issues of research for highway driving motion planning.

A. Data Management

Like a driver, the decision relies on a proper representation/perception/modeling of the environment and a good control of the vehicle's behavior, as discussed in [165]. Reference [166] details several perception challenges impacting the motion planners, and [167] discusses conditions to improve control algorithms for driving. In this review, we pay attention to two questions: *how much data is enough in terms*

TABLE II

COMPARISON TABLE FOR HIGHWAY APPLICATIONS OF MOTION PLANNING METHODS ('−−' VERY INAPPROPRIATE, '−' INAPPROPRIATE, '∼' INTERMEDIARY, '+' APPROPRIATE, '++' VERY APPROPRIATE)

Family	Characteristics				Intrinsic Limits									Extrinsic Limits			
					Performance					Use	Data analysis			Sensor	Environment		Vehicle
Use case	**Type of use**	**Horizon**	**Domain**	**Functions**	**Real-time**	**Robustness**	**Stability**	**Adaptability**	**Feasibility**	**Replanning**	**Input Type**	**Output type**	**Uncertainty**	**Constraints**	**Complexity**	**Highway constraints**	**Kinematics**
Configuration space sampling $[43]^{o}_{E}$	se	P/R	G	gen	∼	−	∼	∼	−	++	s	sp	−	∼	−	+	++
Probabilistic sampling $[33]^{ca}_{S}$ $[42]^{*}_{E}$	se	P/R	G	gen	+	−−	−−	+	−−	++	s	sp	+	+	∼	−−	−−
Non-obstacle-based cells																	
– Exact decomposition $[50]^{o}_{S}$ $[52]^{*}_{E}$ $[54]^{lc}_{S}$ $[55]^{e}_{S}$; Polar grid $[53]^{*}_{E}$	se	P/R	G	gen	−	∼	−	∼	∼	−	d	sp/pa	∼	++	−	−−	−−
– VFH $[57]^{*}_{S}$ $[58]^{*}_{E}$; DW $[60]^{ca}_{S}$ $[61]^{*}_{E}$ $[62]^{*}_{/}$	se	R	G	def	+	++	∼	+	+	+	c	sp/tr	+	−	+	+	∼
Obstacle-based cells																	
– Voronoi $[42]^{*}_{E}$ $[58]^{*}_{E}$; Approximate $[67]^{*}_{E}$	se	P/R	G	gen	−−	++	∼	+	∼	−−	d	sp/pa	+	−	−	+	−
– Visibility graph $[49]^{*}_{S}$ $[70]^{m}_{S}$	se	P/R	G	gen	−	∼	−−	−	∼	−−	d	sp/pa	−	∼	−−	∼	−−
– VO $[49]^{*}_{S}$; Driving corridor $[18]^{*}_{E}$ $[72]^{*}_{S}$ $[73]^{*}_{S}$ $[74]^{*}_{S}$	se	P/R	G	gen	−	++	∼	+	+	−	c	sp/tr	∼	−	∼	+	+
Lattice																	
– Maximum turn $[13]^{*}_{E}$ $[76]^{*}_{S}$; Curvature Velocity $[72]^{*}_{S}$ $[78]^{*}_{S}$	se	P	G	gen	∼	+	+	+	+	+	c	tr/ma	−−	+	−	+	++
– Road-adapted $[29]^{*}_{S}$ $[79]^{*}_{S}$ $[80]^{*}_{E}$ $[81]^{*}_{E}$; Driver-based $[82]^{*}_{E}$	se	P	G	gen	∼	+	+	+	+	+	c	tr/ma	−	∼	−	++	+
Pathfindings																	
– Dijkstra $[42]^{*}_{E}$ $[72]^{*}_{S}$ $[83]^{*}_{S}$	so	P	L	dec	−	∼	+	+	++	∼	s	pa	−	+	−−	−−	−−
– A* $[43]^{o}_{E}$ $[76]^{*}_{S}$ $[86]^{*}_{S}$ $[87]^{*}_{S}$ $[88]^{*}_{E}$ $[90]^{*}_{E}$	so	P	H/L	dec	∼	∼	+	+	+	+	s	pa	∼	+	−	∼	−−
– RRT $[92]^{*}_{E}$ $[94]^{*}_{S}$	se/so	P/R	H/L	gen/dec	+	+	∼	++	∼	++	s	pa	+	−	−	∼	++
Artificial forces																	
APF $[46]^{*}_{S}$ $[61]^{*}_{E}$ $[86]^{*}_{S}$ $[95]^{*}_{E}$ $[96]^{*}_{S}$ $[99]^{*}_{S}$ $[100]^{*}_{S}$ $[101]^{*}_{S}$; VFF $[43]^{o}_{E}$; Elastic band $[48]^{*}_{S}$ $[103]^{cf}_{E}$ $[104]^{ca}_{S}$	se/so	R	B	gen/def	−	+	+	+	+	++	c	sp/pa/tr	+	+	∼	++	+
Parametric curves																	
– Line and circle $[42]^{*}_{E}$ $[107]^{*}_{E}$ $[109]^{ca}_{E}$	se	P/R	G	gen	+	−	−−	+	+	+	s/d/c	pa/tr/ma	−−	+	−−	+	∼
– Clothoïd $[50]^{o}_{S}$ $[106]^{*}_{S}$ $[111]^{*}_{E}$ $[112]^{*}_{S}$ $[113]^{*}_{S}$; Sigmoid $[32]^{lc}_{E}$ $[115]^{*}_{E}$	se	P/R	G	gen	+	−	−−	+	+	+	s/d/c	pa/tr/ma	−	+	∼	++	++
Semi-parametric curves																	
Polynomial $[29]^{*}_{S}$ $[54]^{lc}_{S}$ $[80]^{*}_{E}$ $[81]^{*}_{E}$; Spline $[76]^{*}_{S}$ $[79]^{*}_{S}$; Bézier $[49]^{*}_{S}$ $[116]^{lc}_{E}$ $[117]^{*}_{S}$	so	P/R	G	gen	∼	−	−−	+	++	+	s/d/c	pa/tr/ma	∼	+	+	+	++
Mathematical optimization																	
LP $[81]^{*}_{E}$ $[119]^{o}_{S}$; NLP $[29]^{*}_{S}$ $[33]^{ca}_{S}$ $[104]^{ca}_{S}$; QP $[18]^{*}_{E}$ $[120]^{o}_{S}$ $[121]^{*}_{S}$; MPC $[46]^{*}_{S}$ $[47]^{o}_{S}$ $[51]^{o}_{S}$ $[113]^{*}_{S}$ $[118]^{*}_{S}$ $[122]^{*}_{E}$ $[123]^{*}_{E}$; DP $[49]^{*}_{S}$ $[124]^{*}_{S}$	so	P	H/L	gen/dec	+	+	+	++	∼	+	s/d/c	pa/tr/ma	∼	+	+	++	++
AI Logic																	
– if-then-rules $[32]^{lc}_{E}$ $[78]^{*}_{S}$ $[107]^{*}_{E}$ $[126]^{o}_{S}$ $[127]^{lc}_{E}$ $[128]^{*}_{S}$	so	P/R	L	dec	∼	+	+	−	+	∼	d	ta	−	++	−	++	∼
– FSM $[18]^{*}_{E}$ $[54]^{lc}_{S}$ $[55]^{e}_{S}$ $[129]^{*}_{S}$	se/so	P/R	L	dec	+	+	+	−	++	+	d	ta	−−	++	∼	++	∼
– Dynamic Bayesian $[27]^{*}_{S}$ $[50]^{o}_{S}$ $[130]^{*}_{S}$ $[131]^{lc}_{E}$ $[132]^{*}_{E}$ $[133]^{*}_{S}$	se/so	P/R	H/L	dec	−	++	+	−	++	+	d	ta	++	+	∼	++	∼
AI Heuristic																	
– Agent $[132]^{*}_{E}$ $[133]^{*}_{S}$	se/so	P/R	H/L/B	gen/dec	−	∼	−	++	∼	+	s/d/c	pa/tr/ma	+	+	+	++	+
– SVM $[117]^{*}_{S}$ $[134]^{lc}_{S}$	se/so	P/R	H/L	dec	∼	++	∼	−	+	−−	s/d/c	ma	∼	+	+	+	∼
– Evolutionary $[135]^{*}_{/}$ $[136]^{lc}_{S}$ $[137]^{ca}_{S}$	se/so	P/R	H/L/B	gen/dec	−	++	−	++	−−	−−	s/d/c	pa/tr/ma	−	+	+	++	+
AI Approximate																	
– Fuzzy logic $[138]^{lc}_{S}$ $[139]^{o}_{E}$ $[140]^{*}_{E}$	so	P/R	H/L/C	dec	∼	+	+	∼	+	∼	c	ma/ta	+	++	∼	++	+
– ANN $[122]^{*}_{E}$ $[133]^{*}_{S}$ $[141]^{*}_{S}$ $[142]^{*}_{E}$ $[143]^{*}_{E}$ $[144]^{o}_{S}$ $[145]^{*}_{S}$ $[146]^{*}_{E}$ $[147]^{*}_{E}$	se/so	P/R	H/L/C/B	gen/dec	−−	∼	∼	−−	+	−	c	tr/ma/ta	∼	−−	+	+	+
– Belief Network $[148]^{*}_{E}$	se/so	P/R	H/L/C/B	gen/dec	−−	∼	∼	−−	+	−	c	tr/ma/ta	++	−	+	+	+
AI Cognitive																	
– Risk $[51]^{o}_{S}$ $[55]^{e}_{S}$ $[150]^{*}_{S}$ $[151]^{*}_{S}$ $[152]^{ca}_{S}$ $[153]^{*}_{S}$ $[154]^{ca}_{E}$ $[155]^{*}_{/}$ $[156]^{*}_{/}$ $[157]^{*}_{S}$ $[158]^{*}_{S}$ $[159]^{*}_{S}$	so	P/R	L/C	dec	++	++	++	+	++	++	d	ta	∼	−−	+	++	∼
– Task $[83]^{*}_{S}$ $[139]^{o}_{E}$ $[160]^{*}_{/}$ $[161]^{*}_{/}$ $[162]^{*}_{/}$ $[163]^{*}_{/}$ $[164]^{o}_{S}$	so	P/R	L/C	dec	+	+	∼	+	+	−	d	ta	−−	++	∼	+	∼
– Game $[133]^{*}_{S}$	se/so	P/R	C	gen/dec	∼	∼	+	−	++	∼	c	ma/ta	+	+	−	+	+

Legend: *Use case* – $[\,]_{S}$ = simulation / $[\,]_{E}$ = experimentation; $[\,]^{ca}$ = collision avoidance / $[\,]^{cf}$ = car following / $[\,]^{lc}$ = lane change / $[\,]^{m}$ = merging / $[\,]^{o}$ = overtaking / $[\,]^{e}$ = emergency / $[\,]^{*}$ = no specific use case.
Characteristics – se/so = set-/solve-algorithm; P = predictive / R = reactive; G = geometric / H = heuristic / L = logic / C = cognitive / B = biomimetic; gen = generation / dec = decision making / def = deformation.
Data Analysis – d = discrete / s = sampled / c = continuous; sp = space / pa = path / tr = trajectory / ma = maneuver / ta = task

of uncertainties and trustworthiness? and *how do we make a decision with latency and unsynchronized data?*

1) Data Uncertainty and Trustworthiness: As we discussed in II-B, motion planning algorithms need a scene representation with data characterized by their uncertainties and trustworthiness. The quality of these data is crucial for motion planning to ensure safety decisions. For example, the authors in [168] propose a method to evaluate the appropriate number of naturalistic driving data to model a car following behavior. Some research has been conducted on data uncertainties and trustworthiness on applied methods, such as Kalman estimators [52], [169], Markov processes [131], [170], Monte Carlo simulation [170], evidential theory [52], or interval arithmetic [171]. Furthermore, the planning stage must also ensure motion continuity to guarantee the safety of the vehicle even if data are missing.

2) Data Synchronization: The scene's data are obtained with heterogeneous frequencies and latencies. The problem is therefore choosing a logic of data synchronization processing, as is done in [172]. A first solution is to run the algorithm with the same time step for the motion planner and the controller, and to extrapolate the last obtained data. The second possibility is to have a planner that will refresh at each new data arrival. A mixed logic would then be to operate on a fixed frequency and to refresh the data as soon as new data arrive, to improve the propagation equations by extrapolation of the data set.

B. Adaptive Mobility

Adaptive mobility tackles the question of the introduction of autonomous vehicles to our human environment. In fact, all the infrastructure and driving rules for transportation means are built from human models. According to current and future reflections on autonomous developments, it is not yet conceivable to consider, in a near future, autonomous transportation means independent of the current ones. Thus, it can be wise to ask whether it is reasonable to expect to model and to reproduce human behavior/reasoning on a robotic system whose environmental knowledge is not adapted in terms of perception and reasoning means. Five aspects of driving are especially subject to this question: safety, eco-driving, perception compensation, route context, and service-oriented vehicles.

1) Safety: The safety considerations impose the need for intelligent enough algorithms to distinguish a permanent or temporary danger, such as the front vehicle suddenly stopping, or another one cutting in.

From a motion planning perspective, the ego vehicle is safe if it is not in conflict with its environment. However, this safety space highly depends on the safety capacity of robot driving, which can be different from a human one. As stated in [173], longitudinal distance and speed controls are generally faster in autonomous driving, and so longitudinal safety space could be smaller. In contrast, lateral distance control is generally more stable with human drivers; thus, lateral safety space must be larger with robot drivers. Furthermore, this safety space has to be sufficient against the unexpected challenges of the environment (e.g. unexpected brakes from leading vehicle, unforeseen road's dead end). The formal methods are thus used to mathematically prove safety properties [174], [175].

Moreover, at the moment, automakers are developing autonomous vehicles with human takeover and human-robot mixed driving. This implies a possibly shared decision making between autonomous vehicles and drivers. The first problem is retaining a stable and predictive decision in case the driver takes the control back [176]. The second one is that the decisions of the human driver and the machine may contradict each other [177]. The third safety consideration is of human drivers' behaviors towards robot drivers in the sense of respect, acceptability, and predictability [178] or comfort [179].

2) Eco-Driving Planning: Eco-driving is expressed as a multicriteria optimization problem (see III-B.5), with a cost function based on energy efficiency and environmental care. Two main aspects are considered. The first deals with the dynamic model of the ego vehicle to propose a smooth maneuver, such as low acceleration and constant speed to decrease fuel consumption [180]. The second one is a matter of prediction and anticipation to not act suddenly, as suggested by the driver policy in [181].

3) Perception Compensation: One of the problems regarding perception is the constraint of a fixed point of view of the driving scene. However, when a human driver has to make a decision, he/she optimizes his/her perception by moving the ego vehicle to capture more information on a larger perceived environment. This perception compensation places the ego vehicle in a position to optimize the capacity of the sensors with a geometrically wide perception, as suggested in [182] for city driving.

4) Route Context: The route planner is well studied with a trip scheduler. However, there are very few examples in the literature of integrating these data for decision and generation functions (see Fig. 2). In fact, route planning constrains the evolution space, especially on highways; e.g. a right-hand highway exit involves the ego vehicle to navigate to the rightmost lane. These requirements imply respecting constraints in actions, distances, and time to follow the instructions of the navigator.

5) Service Orientation: The last prospect discussed for adaptive mobility concerns the future use of transportation. Reference [183] highlights the roles of driverless vehicles in the service for which they are designed, e.g. private car, taxi, shuttle bus, logistics, health or safety vehicles. Their behaviors then become heterogeneous according to their modality of services. Until now, the classic representation was the space-time domain decision making, i.e. *where* and *when* the vehicle had to navigate. A third dimension is now added: *how* the vehicle has to navigate. The behavior towards other vehicles then shows a logistic-driven attitude to prioritize the evolution space.

C. Cooperative Planning

Driving is a social task, where cooperation is inherent to the decision making. We identify three perspectives regarding the automation of vehicles and/or infrastructure, as detailed in [34]: individual, collective, and shared.

First, individual transportation applies an isolated decision-maker, with or without communication skills. It is analogous to a Lagrangian description in fluid mechanics (or traffic network science), separately considering each particle's dynamics. Each vehicle takes its decision independently of the others, but considers interaction [159] and courtesy [184].

Second, we identify the Eulerian description in fluid mechanics with collective decisions. One globally considers particles' interactions and behaviors within a given volume; a platoon comprises numerous entities – individual vehicles – but adopts a unified behavior: a swarm intelligence. Collective systems require a strong knowledge of other vehicles' states when acting as auto-organized systems, and/or strong communication with other vehicles and/or infrastructure if acting as self-organizing [185]. Many studies have modeled cooperative platoon maneuvers using multi-agent event-based approach [186], swarm intelligence [187], risk estimators [188], or fuzzy logic [189]. Similarly, an intelligent infrastructure approach includes a conductor of a group of vehicles, which acts as either a manager or a controller. In case of managing, the vehicles entering the area supervised by the intelligent infrastructure send their intention. The infrastructure then provides a coordination scheduler to fulfill the desired actions of every vehicle. In the second case, the intelligent infrastructure takes full control of the monitoring. In the literature, one can mainly find intelligent highway managers with knowledge-based inference engines [190], [191] and agents [183].

The last cooperative approach concerns the drivers' shared decision. Today, the first four SAE automation levels in [10] depict a shared cooperation between drivers and autonomous vehicles, from driver assistance to partial, conditional, and high automation. As long as the driver stays in the vehicle, he/she approves the decisions and can take the control back at any moment. To maintain decision continuity, human driver experiences represent a useful base of knowledge for the system. A fuzzification approach to mix this driver cognition experience with the system's decision is proposed in [192]. In [150], the authors present to the driver a maneuver grid based on a crash estimation with the Equivalent Energetic Speed (EES), whereas [193] introduces game theory for cooperative guidance. Although studies exist on integrating autonomous vehicles and human drivers, the question of adapting the autonomous vehicle behavior to the driver type is of interest for the popularization of such vehicles [194].

D. Validation and Evaluation

In addition to the indicators depicted in Table II, the evaluation of the algorithms must include a verification of behaviors, judgments, and responsibilities. Here, we discuss the validation of transition stability with different planners, the evaluation of motion planning methods on predefined use cases, the ethics dilemmas, and the topic of rules relaxation.

1) Transition Stability: The various cases encountered often require the use of different combined motion planning algorithms. In this case, we need to know whether the resulting architecture is stable, reliable, and robust, or whether it is necessary to add a high-level supervisor to validate the coherence of decisions, actions, and observations. In [195], the authors propose a behavior paradigm analogous to the FSM decision to implement an architecture able to switch from manual to autonomous driving and between maneuver modes.

2) Evaluation: As stated in section IV, the validation of motion planning algorithms mostly relies on extensive simulation and experimental testing results, where a human driver evaluates the action according to a personal reference index, such as safety, smoothness, or operation time [54]. There is a lack of formal analysis and evaluation methods to hierarchically classify algorithms' performances. This will change thanks to the open source scenario library developed by [196] to propose a benchmark for motion planning algorithms, or with Key Performance Indicators (KPIs) proposed by [197].

3) Ethics: As soon as robotic systems interact with human beings, questions of ethics in autonomous decision-makers arise. Reference [198] proposes an ethical vehicle deployment strategy in a hybrid rational and AI approach, especially for critical safety situations. The authors in [199] show how to incorporate ethics into decisions based on ethical frameworks, such as deontology – as rules constraining the actions and navigation of the system, consequentialism – as cost-based construction for the objective function, or as morality to determine the different costs of the system's behaviors.

4) Algorithm Relaxation: The strict respect and relaxation of driving rules are key points to ensure safety in any case for an autonomous vehicle. Indeed, even hard constraints might become dangerous in critical cases [13]. Reference [200] introduces an ontology for traffic rules in unusual situations with knowledge-based inference engines to avoid blockage situations, and to preserve safety with lane crossing or excess speed.

VI. Conclusion

To conclude, our literature review revealed a huge amount of algorithms for motion planning in robotics. The objective of this paper was to identify the main methods of motion planning for autonomous vehicles in the highway case. We do not claim to have exhaustively gathered all algorithms used for this application, but we hope to have shown the diversity and potential of highway planning. Through this state-of-the-art work, we also aspired to present readers with a different proposal of decomposition methods than what they may have found in previous reviews. In particular, we believe that the radar chart illustration of differentiating attributes will allow future users to identify and orient their work towards a solution that will correctly answer their problem. Finally, the last objective of this survey was to encourage readers' reflections on the practical implementation of these algorithms and, more generally, their perspectives of highway motion planning and autonomous driving.

References

[1] *The Free-Lance Star (Newspaper Distributed Throughout Fredericksburg)*, Phantom Auto Operated Here, Richmond, VA, USA, Jun. 1932.

[2] EUREKA Project E!45 PROMETHEUS. *Programme for a European Traffic System With Highest Efficiency and Unprecedented Safety*. Accessed: Mar. 11, 2018. [Online]. Available: http://www.eurekanetwork.org/project/id/45

[3] B. Ulmer, "VITA II-active collision avoidance in real traffic," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 1994, pp. 1–6.

[4] *No Hands Across America*. Accessed: Mar. 11, 2018. [Online]. Available: http://www.cs.cmu.edu/~tjochem/nhaa/nhaa_home_page.html

[5] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi, "The ARGO autonomous vehicle's vision and control systems," *Int. J. Intell. Control Syst.*, vol. 3, no. 4, pp. 409–441, 1999.

[6] R. Rajamani, S. B. Choi, B. K. Law, J. K. Hedrick, R. Prohaska, and P. Kretz, "Design and experimental implementation of longitudinal control for a platoon of automated vehicles," *J. Dyn. Syst., Meas., Control*, vol. 122, no. 3, pp. 470–476, 2000.

[7] M. Hasenjäger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–7.

[8] United Nations. *Convention on Road Traffic (Vienna)*. Accessed: Mar. 11, 2018. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf

[9] *UNECE*. Accessed: Mar. 11, 2018. [Online]. Available: https://www.unece.org/info/media/presscurrent-press-h/transport/2016/unece-paves-the-way-for-automated-driving-by-updating-un-international-convention/doc.html

[10] *SAE International J3016*. Accessed: Mar. 11, 2018. [Online]. Available: https://www.sae.org/standards/content/j3016_201401/

[11] *DARPA Urban Challenge*. Accessed: Mar. 11, 2018. [Online]. Available: http://archive.darpa.mil/grandchallenge/

[12] A. Broggi *et al.*, "Extensive tests of autonomous driving technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1403–1415, Sep. 2013.

[13] A. Broggi *et al.*, "PROUD-public road urban driverless test: Architecture and results," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 648–654.

[14] HAVE IT. *The Future of Driving. Deliverable D61.1 Final Report*. Accessed: Mar. 11, 2018. [Online]. Available: http://www.haveit-eu.org/LH2Uploads/ItemsContent/24/HAVEit_212154_D61.1_Final_Report_Published.pdf

[15] P. Resende, E. Pollard, H. Li, and F. Nashashibi, "ABV—A low speed automation project to study the technical feasibility of fully automated driving," in *Proc. Workshop PAMM*, 2013, pp. 8–9.

[16] *CityMobil*. Accessed: Mar. 11, 2018. [Online]. Available: http://www.citymobil-project.eu/

[17] *SARTRE*. Accessed: Mar. 11, 2018. [Online]. Available: https://www.sp.se/sv/index/research/dependable_systems/Documents/The%20SARTRE%20project.pdf

[18] J. Ziegler *et al.*, "Making bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Apr. 2014.

[19] C. Englund *et al.*, "The grand cooperative driving challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 146–152, Aug. 2016.

[20] J.-C. Latombe, *Robot Motion Planning*. New York, NY, USA: Springer, 1991.

[21] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot Motion Planning and Control*. Berlin, Germany: Springer, 1998, pp. 1–53.

[22] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[23] 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," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[24] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[25] 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. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[26] W. Payre, J. Cestac, and P. Delhomme, "Intention to use a fully automated car: Attitudes and *a priori* acceptability," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 27, pp. 252–263, Nov. 2014.

[27] M. Rodrigues, A. McGordon, G. Gest, and J. Marco, "Adaptive tactical behaviour planner for autonomous ground vehicle," in *Proc. IEEE Int. Conf. Control*, Aug./Sep. 2016, pp. 1–8.

[28] J. A. Michon, "A critical view of driver behavior models: What do we know, what should we do?" in *Human Behavior and Traffic Safety*. Boston, MA, USA: Springer, 1985, pp. 485–524.

[29] T. Gu, J. Snider, J. M. Dolan, and J.-W. Lee, "Focused trajectory planning for autonomous on-road driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 547–552.

[30] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of Large and Complex Networks*. Berlin, Germany: Springer, 2009, pp. 117–139.

[31] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, 2014.

[32] L. Claussmann, A. Carvalho, and G. Schildbach, "A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams," in *Proc. IEEE Eur. Control Conf. (ECC)*, Jul. 2015, pp. 2976–2982.

[33] X. Li, Z. Sun, Q. Zhu, and D. Liu, "A unified approach to local trajectory planning and control for autonomous driving along a reference path," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2014, pp. 1716–1721.

[34] L. Claussmann, M. Revilloud, S. Glaser, and D. Gruyer, "A study on al-based approaches for high-level decision making in highway autonomous driving," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 3671–3676.

[35] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1033–1045, May 2017.

[36] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[37] P. Polack, F. Altché, B. D'Andrea-Novel, and A. de La Fortelle, "Guaranteeing consistency in a motion planning and control architecture using a kinematic bicycle model," in *Proc. IEEE Amer. Control Conf. (ACC)*, Jun. 2018, pp. 3981–3987.

[38] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annu. Rev. Control*, vol. 44, pp. 323–341, Oct. 2017.

[39] D. Bevly *et al.*, "Lane change and merge maneuvers for connected and automated vehicles: A survey," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 105–120, Mar. 2016.

[40] T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.*, vol. C-32, no. 2, pp. 108–120, Feb. 1983.

[41] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[42] J. Villagra, V. Milanés, J. P. Rastelli, J. Godoy, and E. Onieva, "Path and speed planning for smooth autonomous navigation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2012.

[43] T. Hesse, D. Hess, and T. Sattel, "Motion planning for passenger vehicles-force field trajectory optimization for automated driving," in *Proc. Int. Conf. Robot. Appl.*, 2010. doi: 10.2316/P.2010.706-066.

[44] D. Hsu and Z. Sun, "Adaptively combining multiple sampling strategies for probabilistic roadmap planning," in *Proc. IEEE. Conf. Robot., Autom. Mechatronics*, Dec. 2004, pp. 774–779.

[45] D. Kuan, J. Zamiska, and R. Brooks, "Natural decomposition of free space for path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Mar. 1985, pp. 168–173.

[46] C. Liu, S. Lee, S. Varnhagen, and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 174–179.

[47] J. Nilsson, J. Fredriksson, and E. Coelingh, "Trajectory planning with miscellaneous safety critical zones," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9083–9088, 2017.

[48] X. Song, H. Cao, and J. Huang, "Vehicle path planning in various driving situations based on the elastic band theory for highway collision avoidance," in *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 227, no. 12, pp. 1706–1722, 2013.

[49] J. Choi, "Kinodynamic motion planning for autonomous vehicles," *Int. J. Adv. Robot. Syst.*, vol. 11, no. 6, p. 90, 2014.

[50] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard, "Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1114–1119.

[51] Q. Wang and B. Ayalew, "Obstacle filtering algorithm for control of an autonomous road vehicle in public highway traffic," in *Proc. ASME Dyn. Syst. Control Conf.*, 2016, p. V002T24A009.

[52] C. Yu, V. Cherfaoui, and P. Bonnifait, "Semantic evidential lane grids with prior maps for autonomous navigation," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1875–1881.

[53] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 84–89.

[54] H. Tehrani, Q. H. Do, M. Egawa, K. Muto, K. Yoneda, and S. Mita, "General behavior and motion model for automated lane change," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 1154–1159.

[55] M. Ardelt, P. Waldmann, F. Homm, and N. Kaempchen, "Strategic decision-making process in advanced driver assistance systems," *IFAC Proc. Volumes*, vol. 43, no. 7, pp. 566–571, 2010.

[56] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[57] P. Qu, J. Xue, L. Ma, and C. Ma, "A constrained VFH algorithm for motion planning of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 700–705.

[58] U. Lee, S. Yoon, H. Shim, P. Vasseur, and C. Demonceaux, "Local path planning in a complex environment for self-driving car," in *Proc. IEEE Int. Conf. Cyber Technol. Automat., Control, Intell. Syst. (CYBER)*, Jun. 2014, pp. 445–450.

[59] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[60] Y. Kang, D. A. de Lima, and A. C. Victorino, "Dynamic obstacles avoidance based on image-based dynamic window approach for human-vehicle interaction," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 77–82.

[61] D. A. de Lima and G. A. S. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *J. Control, Autom. Electr. Syst.*, vol. 24, nos. 1–2, pp. 106–116, 2013.

[62] S. Mitsch, K. Ghorbal, and A. Platzer, "On provably safe obstacle avoidance for autonomous robotic ground vehicles," in *Proc. Robot., Sci. Syst.*, Berlin, Germany, Jun. 2013, pp. 14–21.

[63] D. T. Kuan, R. A. Brooks, J. C. Zamiska, and M. Das, "Automatic path planning for a mobile robot using a mixed representation of free space," in *Proc. IEEE Comput. Society Conf. Artif. Intell. Appl.*, Dec. 1984, pp. 70–74.

[64] F. Aurenhammer, "Voronoi diagrams—A survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, 1991.

[65] K. Sugihara, "Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams," *CVGIP, Graph. Models Image Process.*, vol. 55, no. 6, pp. 522–531, 1993.

[66] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert, and M. Stilman, "Path planning with uncertainty: Voronoi uncertainty fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 4596–4601.

[67] P. G. Trepagnier *et al.*, "Navigation and control system for autonomous vehicles," U.S. Patent 8 050 863, Nov. 1, 2011.

[68] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[69] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, Sep. 1986.

[70] J. Johnson and K. Hauser, "Optimal longitudinal control planning with moving obstacles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 605–611.

[71] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

[72] T. Gu, J. M. Dolan, and J.-W. Lee, "Runtime-bounded tunable motion planning for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1301–1306.

[73] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 1386–1392.

[74] F. Altché and A. de La Fortelle, "Partitioning of the free space-time for on-road navigation of autonomous ground vehicles," in *Proc. IEEE Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2126–2133.

[75] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, Mar. 2009.

[76] R. Schubert, U. Scheunert, and G. Wanielik, "Planning feasible vehicle manoeuvres on highways," *IET Intell. Transp. Syst.*, vol. 2, no. 3, pp. 211–218, Sep. 2008.

[77] N. Y. Ko and R. G. Simmons, "The lane-curvature method for local obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Oct. 1998, pp. 1615–1621.

[78] A. Constantin, J. Park, and K. Iagnemma, "A margin-based approach to threat assessment for autonomous highway navigation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 234–239.

[79] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2009, pp. 1879–1884.

[80] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 4889–4895.

[81] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 2061–2067.

[82] W. Yao, H. Zhao, F. Davoine, and H. Zha, "Learning lane change trajectories from on-road driving data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2012, pp. 885–890.

[83] M. Bahram, A. Wolf, M. Aeberhard, and D. Wollherr, "A prediction-based reactive driving strategy for highly automated driving function on freeways," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 400–406.

[84] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[85] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[86] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, "Modified artificial potential field method for online path planning applications," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 180–185.

[87] Z. Boroujeni, D. Goehring, F. Ulbrich, D. Neumann, and R. Rojas, "Flexible unit A-star trajectory planning for autonomous vehicles on structured road maps," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jun. 2017, pp. 7–12.

[88] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[89] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 1994, pp. 3310–3317.

[90] S. Rezaei, J. Guivant, and E. M. Nebot, "Car-like robot path following in large unstructured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2003, pp. 2468–2473.

[91] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. TR 98-11, Oct. 1998.

[92] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961–1976, Aug. 2015.

[93] D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using RRT," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1429–1434.

[94] J. H. Jeon *et al.*, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proc. IEEE Amer. Control Conf. (ACC)*, Jun. 2013, pp. 188–193.

[95] E. Galceran, R. M. Eustice, and E. Olson, "Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 304–309.

[96] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *J. Dyn. Syst. Meas. Control*, vol. 123, no. 3, pp. 431–438, 1999.

[97] O. Khatib and J. F. Le Maitre, "Dynamic control of manipulators operating in a complex environment," in *Proc. CISM-IFToMM Symp. Theory Pract. Robots Manipulators*, 1978, pp. 1–16.

[98] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.

[99] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2008, pp. 3731–3736.

[100] D. Reichardt and J. Shick, "Collision avoidance in dynamic environments applied to autonomous vehicle guidance on the motorway," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 1994, pp. 74–78.

[101] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1255–1267, May 2017.

[102] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 1993, pp. 802–807.

[103] S. K. Gehrig and F. J. Stein, "Collision avoidance for vehicle-following systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 233–244, Jun. 2007.

[104] M. Keller, F. Hoffmann, C. Hass, T. Bertram, and A. Seewald, "Planning of optimal collision avoidance trajectories with timed elastic bands," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 9822–9827, 2014.

[105] K. G. Baass, "Use of clothoid templates in highway design," in *Proc. Transp. Forum*, vol. 1, 1984, pp. 47–52.

[106] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based speed profiler and control for autonomous driving," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 2194–2199.

[107] B. Vanholme, D. Gruyer, B. Lusetti, S. Glaser, and S. Mammar, "Highly automated driving on highways based on legal safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 333–347, Mar. 2013.

[108] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, 2008.

[109] A. Cherubini, F. Spindler, and F. Chaumette, "A new tentacles-based technique for avoiding obstacles during visual navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4850–4855.

[110] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.

[111] C. Alia, T. Gilles, T. Reine, and C. Ali, "Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 674–679.

[112] M. Sheckells, T. M. Caldwell, and M. Kobilarov, "Fast approximate path coordinate motion primitives for autonomous driving," in *Proc. IEEE Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 837–842.

[113] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *Proc. IEEE Eur. Control Conf. (ECC)*, Jul. 2015, pp. 2983–2990.

[114] J. Hudecek and L. Eckstein, "Improving and simplifying the generation of reference trajectories by usage of road-aligned coordinate systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 504–509.

[115] M. Arbitmann, U. Stählin, M. Schorn, and R. Isermann, "Method and device for performing a collision avoidance maneuver," U.S. Patent 8 209 090, Jun. 26, 2012.

[116] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise bezier curve for autonomous vehicle," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jul. 2013, pp. 17–22.

[117] Q. Huy, S. Mita, H. T. N. Nejad, and L. Han, "Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles," *IEICE Trans. Inf. Syst.*, vol. E96-D, no. 2, pp. 314–328, 2013.

[118] F. Altché, P. Polack, and A. de La Fortelle, "High-speed trajectory planning for autonomous vehicles using a simple dynamic model," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–7.

[119] M. G. Plessen, P. F. Lima, J. Mårtensson, A. Bemporad, and B. Wahlberg, "Trajectory planning under vehicle dimension constraints using sequential linear programming," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[120] X. Qian, F. Altché, P. Bender, C. Stiller, and A. de La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 205–210.

[121] C. Miller, C. Pek, and M. Althoff, "Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1954–1961.

[122] S. Lefèvre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 32–42, Jan. 2016.

[123] V. Cardoso *et al.*, "A model-predictive motion planner for the IARA autonomous car," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 225–230.

[124] T. Gu, J. M. Dolan, and J.-W. Lee, "On-road trajectory planning for general autonomous driving with enhanced tunability," in *Intelligent Autonomous Systems*, vol. 13. Cham, Switzerland: Springer, 2016, pp. 247–261.

[125] S. Russell and P. Norvig, "A modern approach," in *Artificial Intelligence*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[126] J. Nilsson, J. Fredriksson, and E. Coelingh, "Rule-based highway maneuver intention recognition," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 950–955.

[127] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson, "If, when, and how to perform lane change maneuvers on highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 4, pp. 68–78, Oct. 2016.

[128] N. Li, H. Chen, I. Kolmanovsky, and A. Girard, "An explicit decision tree approach for automated driving," in *Proc. ASME Dyn. Syst. Control Conf.*, 2017, p. V003T50A006.

[129] Q. Wang, T. Weiskircher, and B. Ayalew, "Hierarchical hybrid predictive control of an autonomous road vehicle," in *Proc. ASME Dyn. Syst. Control Conf.*, 2015.

[130] S. Zhou, Y. Wang, M. Zheng, and M. Tomizuka, "A hierarchical planning and control framework for structured highway driving," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9101–9107, 2017.

[131] S. Ulbrich and M. Maurer, "Towards tactical lane change behavior planning for automated vehicles," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 989–995.

[132] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Robot., Sci. Syst.*, vol. 41, no. 6, pp. 1367–1382, 2015.

[133] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.

[134] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1590–1595.

[135] S. Baluja, R. Sukthankar, and J. Hancock, "Prototyping intelligent vehicle modules using evolutionary algorithms," in *Prototyping Intelligent Vehicle Modules Using Evolutionary Algorithms*. Berlin, Germany: Springer, 1997, pp. 241–257.

[136] E. Onieva, J. E. Naranjo, V. Milanés, J. Alonso, R. García, and J. Pérez, "Automatic lateral control for unmanned vehicles via genetic algorithms," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 1303–1309, 2011.

[137] F. Riaz, M. A. Niazi, M. Sajid, S. Amin, N. I. Ratyal, and F. Butt, "An efficient collision avoidance scheme for autonomous vehicles using genetic algorithm," *J. Appl. Environ. Biol. Sci.*, vol. 5, no. 8, pp. 70–76, 2015.

[138] E. Balal, R. L. Cheu, and T. Sarkodie-Gyan, "A binary decision model for discretionary lane changing move based on fuzzy inference system," *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 47–61, Jun. 2016.

[139] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. D. Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 438–450, Sep. 2008.

[140] V. Milanés, E. Onieva, J. Pérez, J. Godoy, and J. Villagrá, "An approach to driverless vehicles in highways," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 668–673.

[141] X. Geng, H. Liang, H. Xu, B. Yu, and M. Zhu, "Human-driver speed profile modeling for autonomous vehicle's velocity strategy on curvy paths," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 755–760.

[142] E. Rehder, J. Quehl, and C. Stiller, "Driving like a human: Imitation learning for path planning using convolutional neural networks," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS) Workshops*, Sep. 2017, pp. 1–5.

[143] M. Bojarski *et al.* (2016). "End to end learning for self-driving cars." [Online]. Available: https://arxiv.org/abs/1604.07316

[144] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Trans. Intell. Trans. Syst.*, vol. 12, no. 2, pp. 509–522, Jun. 2011.

[145] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1379–1384.

[146] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2722–2730.

[147] L. Yang, X. Liang, T. Wang, and E. Xing, "Real-to-virtual domain unification for end-to-end autonomous driving," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 530–545.

[148] T. Huang *et al.*, "Automatic symbolic traffic scene analysis using belief networks," in *Proc. AAAI*, vol. 94, 1994, pp. 966–972.

[149] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," *Human Factors*, vol. 48, no. 2, pp. 362–380, 2006.

[150] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.

[151] A. Tamke, T. Dang, and G. Breuel, "A flexible method for criticality assessment in driver assistance systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 697–702.

[152] S. Wagner, K. Groh, T. Kühbeck, M. Dörfel, and A. Knol, "Using time-to-react based on naturalistic traffic object behavior for scenario-based risk assessment of automated driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1521–1528.

[153] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: A probabilistic perspective," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2010, pp. 4022–4027.

[154] U. S. Ghumman *et al.*, "Guidance-based on-line motion planning for autonomous highway overtaking," *Int. J. Smart Sens. Intell. Syst.*, vol. 1, no. 2, pp. 549–571, 2008.

[155] D. H. Taylor, "Drivers' galvanic skin response and the risk of accident," *Ergonomics*, vol. 7, no. 4, pp. 439–451, 1964.

[156] G. J. S. Wilde, "The theory of risk homeostasis: Implications for safety and health," *Risk Anal.*, vol. 2, pp. 209–225, Dec. 1982.

[157] M. Naumann and C. Stiller, "Towards cooperative motion planning for automated vehicles in mixed traffic," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS) Workshops*, Aug. 2017, pp. 6–11.

[158] D. Prokhorov, "Risk estimator for control in intelligent transportation system," in *Proc. IEEE Control Appl. (CCA) Intell. Control (ISIC)*, Jul. 2009, pp. 1403–1408.

[159] J. Daniel, J.-P. Lauffenburger, S. Bernet, and M. Basset, "Driving risk assessment with belief functions," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 690–695.

[160] A. J. McKnight and B. B. Adams, "Driver education task analysis. Volume I: Task descriptions," Hum. Resour. Res. Org., Alexandria, VA, USA, Final Rep. DOT-HS-800-367; HumRRO-IR-D1-70-103, Nov. 1970.

[161] A. J. McKnight and B. B. Adams, "Driver education task analysis. Volume II: Task analysis methods," Hum. Resour. Res. Org., Alexandria, VA, USA, Final Rep. DOT-HS-800-368; HumRRO-IR-D1-72-13, Nov. 1970.

[162] A. J. McKnight and A. G. Hundt, "Driver education task analysis. Volume III: Instructional Objectives," Hum. Resour. Res. Org., Alexandria, VA, USA, Final Rep. DOT-HS-800-369; HumRRO-IR-D1-71-9, Mar. 1971.

[163] A. J. McKnight and A. G. Hundt, "Driver education task analysis. Volume IV: The development of instructional objectives," Hum. Resour. Res. Org., Alexandria, VA, USA, Final Rep. DOT-HS-800-370; HumRRO-IR-D1-72-14, Mar. 1971.

[164] C. Chen, A. Gaschler, M. Rickert, and A. Knoll, "Task planning for highly automated driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 940–945.

[165] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, "Autonomous vehicle control systems—A review of decision making," *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 225, no. 2, pp. 155–195, 2011.

[166] D. Gruyer, V. Magnier, M. A. Rahal, and G. Bresson, "Real-time architecture for obstacle detection, tracking and filtering: An issue for the autonomous driving," *J. Intell. Comput.*, vol. 8, no. 2, pp. 33–48, 2017.

[167] J. Pérez, D. Gonzalez, and V. Milanés, "Vehicle control in ADAS applications: State of the art," in *Intelligent Transport Systems: Technologies and Applications*. Hoboken, NJ, USA: Wiley, 2015, pp. 206–219.

[168] W. Wang, C. Liu, and D. Zhao, "How much data are enough? A statistical approach with case study on longitudinal driving behavior," *IEEE Trans. Intell. Veh.*, vol. 2, no. 2, pp. 85–98, Jun. 2017.

[169] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2507–2512.

[170] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1237–1247, Dec. 2011.

[171] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, Aug. 2014.

[172] B. Vanholme, D. Gruyer, S. Glaser, and S. Mammar, "Fast prototyping of a highly autonomous cooperative driving system for public roads," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 135–142.

[173] M. Althoff and R. Lösch, "Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance?" in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 485–491.

[174] C. Pek, M. Koschi, M. Werling, and M. Althoff, "Enhancing motion safety by identifying safety-critical passageways," in *Proc. IEEE Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 320–326.

[175] C. Pek, P. Zahn, and M. Althoff, "Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1477–1483.

[176] N. Merat, A. H. Jamson, F. C. H. Lai, M. Daly, and O. M. J. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 27, pp. 274–282, Nov. 2014.

[177] M. Da Lio *et al.*, "Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 244–263, Feb. 2015.

[178] F. M. Verberne, J. Ham, and C. J. Midden, "Trust in smart systems: Sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars," *Hum. Factors*, vol. 54, no. 5, pp. 799–810, 2012.

[179] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: Investigating ride comfort measures in autonomous cars," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 4–17, Jul. 2015.

[180] Q. Cheng, L. Nouvelière, and O. Orfila, "A new eco-driving assistance system for a light vehicle: Energy management and speed optimization," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1434–1439.

[181] C. Vagg, C. J. Brace, D. Hari, S. Akehurst, J. Poxon, and L. Ash, "Development and field trial of a driver assistance system to encourage eco-driving in light commercial vehicle fleets," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 796–805, Jun. 2013.

[182] H. Andersen *et al.*, "Trajectory optimization for autonomous overtaking with visibility maximization," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[183] K. Zhang, A. Yang, H. Su, A. de La Fortelle, K. Miao, and Y. Yao, "Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1867–1881, Jul. 2016.

[184] C. Menéndez-Romero, M. Sezer, F. Winkler, C. Dornhege, and W. Burgard, "Courtesy behavior for highly automated vehicles on highway interchanges," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 943–948.

[185] L. Li, D. Wen, and D.Y. Yao, "A survey of traffic control with vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 425–432, Feb. 2014.

[186] S. Lam and J. Katupitiya, "Cooperative autonomous platoon maneuvers on highways," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2013, pp. 1152–1157.

[187] A. Marjovi, M. Vasic, J. Lemaitre, and A. Martinoli, "Distributed graph-based convoy control for networked intelligent vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun./Jul. 2015, pp. 138–143.

[188] S. Demmel, D. Gruyer, and A. Rakotonirainy, "Comparing cooperative and non-cooperative crash risk-assessment," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1007–1013.

[189] J. Pérez, V. Milanés, and M. S. Peñas, "Control agents for autonomous vehicles in urban and highways scenarios," *IFAC Proc. Volumes*, vol. 46, no. 10, pp. 120–125, 2013.

[190] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," in *Proc. IEEE Int. Conf. Auto. Auto. Syst.*, Mar. 2008, pp. 94–99.

[191] M. Hülsen, J. M. Zöllner, and C. Weiss, "Traffic intersection situation description ontology for advanced driver assistance," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 993–999.

[192] P. Morignot, J. P. Rastelli, and F. Nashashibi, "Arbitration for balancing control between the driver and ADAS systems in an automated vehicle: Survey and approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2014, pp. 575–580.

[193] E. Altendorf and F. Flemisch, "Prediction of driving behavior in cooperative guidance and control: A first game-theoretic approach," in *Proc. CogSys*, Madgeburg, Germany, 2014, pp. 1–10.

[194] A. Rosenfeld, Z. Bareket, C. V. Goldman, S. Kraus, D. J. LeBlanc, and O. Tsimhoni, "Towards adapting cars to their drivers," *AI Mag.*, vol. 33, no. 4, p. 46, 2012.

[195] A. Broggi, S. Debattisti, M. Panciroli, and P. P. Porta, "Moving from analog to digital driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1113–1118.

[196] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 719–726.

[197] J. Quilbeuf *et al.*, "Statistical model checking applied on perception and decision-making systems for autonomous driving," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS) Workshops*, Oct. 2018, pp. 1–8.

[198] N. Goodall, "Ethical decision making during automated vehicle crashes," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2424, pp. 58–65, Sep. 2014.

[199] S. M. Thornton, S. Pan, S. M. Erlien, and J. C. Gerdes, "Incorporating ethical considerations into automated vehicle control," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1429–1439, Jun. 2017.

[200] P. Morignot and F. Nashashibi, "An ontology-based approach to relax traffic regulation for autonomous vehicle assistance," in *Proc. IASTED Int. Conf. Artif. Intell. Appl.*, 2013, pp. 10–17.

Laurène Claussmann received the Dipl.-Ing. degree in automatic control, systems and information technology from Ecole Nationale Supérieure de l'Energie, l'Eau et l'Environnement (Grenoble INP - ENSE3), Grenoble, France, in 2015. She is currently pursuing the Ph.D. degree in automatic control with the Institut VEDECOM and IFSTTAR-LIVIC Laboratory, Université Paris-Saclay, France. Her research is focused on motion planning for the application of autonomous vehicles.

Marc Revilloud received the Dipl.-Ing. degree in image, interaction, and virtual reality from the Université de Technologie Belfort-Montbéliard (UTBM), France, in 2010, and the Ph.D. degree in signal and image processing from the Université Paris-Saclay, France. He is currently a Researcher in autonomous vehicles with the Institut VEDECOM, France. His current research interests revolve around autonomous vehicles, lane detection, and visual SLAM.

Dominique Gruyer was born in France, in 1969. He received the M.S. and Ph.D. degrees from the University of Technology of Compiègne, in 1995 and 1999, respectively. Since 2001, he has been a Researcher at INRETS, into the perception team of LIVIC Department (Laboratory on interactions between vehicles, infrastructure, and drivers) and he involved in the study and the development of multisensors/sources association, combination and fusion. His works enter into the conception of on-board driving assistance systems and more precisely on the carry out of multiobstacle detection and tracking, extended perception, accurate ego localization. He involved for multisensor fusion tasks and sensors modeling and simulation in several European and French projects dealing with intelligent vehicles (HAVEit, Isi-PADAS, CVIS, CARSENSE, eMOTIVE, LOVe, ARCOS, MICADO, ABV, eFuture, CooPerCom, and SINETIC). He is responsible and the main inventor of the SiVIC platform (Simulation for Vehicle, Infrastructure, and sensors). Since 2010, he has been the Team Leader of the LIVICs Perception team. For five years, he was a Network Researcher with AUTO21, Canada. He was the Co-founder and Technical Director of the CIVITEC Company until 2015. Since 2015, CIVITEC has been a subsidiary of the ESI group company. He is currently the Head of LIVIC Laboratory (since 2015), a Research Director (since 2014) in IFSTTAR, and the Perception system and data fusion Scientific Director (since 2015) for ESI group (CIVITEC).

Sébastien Glaser received the Habilitation and Ph.D. degrees in automatic and control in 2011 and 2004, respectively. He was a Researcher in the development of Connected and Automated Vehicles (CAV). He was involved in several European Union initiatives and in the French National Research Agency (ANR) initiative on AV development, test, and evaluation. He also led the technological roadmap task force in France for the AV. Since 2009, he has worked across academic and industrial sectors and held senior researcher positions. He led the French ABV project, gathering eight academic and industrial partners, to develop a CAV solution at low speed. He was the Deputy Director and the Director of LIVIC from 2012 to 2015 and a Project Leader of VEDECOM from 2015 to 2017, developing the autonomous vehicle prototypes. He is currently a Professor in intelligent transportation system with the Center for Accident Research and Road Safety, CARRS-Q, Queensland University of Technology, Australia, where he focuses on a safe and sustainable development/deployment of Automated Driving System in interaction with other road users.