Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions

Christos Katrakazas [a], Mohammed Quddus [a,*], Wen-Hua Chen [b,1], Lipika Deka [a]

[a] School of Civil and Building Engineering, Loughborough University, Loughborough LE11 3TU, United Kingdom
[b] School of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, United Kingdom

A R T I C L E   I N F O

A B S T R A C T

Currently autonomous or self-driving vehicles are at the heart of academia and industry research because of its multi-faceted advantages that includes improved safety, reduced congestion, lower emissions and greater mobility. Software is the key driving factor underpinning autonomy within which planning algorithms that are responsible for mission-critical decision making hold a significant position. While transporting passengers or goods from a given origin to a given destination, motion planning methods incorporate searching for a path to follow, avoiding obstacles and generating the best trajectory that ensures safety, comfort and efficiency. A range of different planning approaches have been proposed in the literature. The purpose of this paper is to review existing approaches and then compare and contrast different methods employed for the motion planning of autonomous on-road driving that consists of (1) finding a path, (2) searching for the safest manoeuvre and (3) determining the most feasible trajectory. Methods developed by researchers in each of these three levels exhibit varying levels of complexity and performance accuracy. This paper presents a critical evaluation of each of these methods, in terms of their advantages/disadvantages, inherent limitations, feasibility, optimality, handling of obstacles and testing operational environments.

Based on a critical review of existing methods, research challenges to address current limitations are identified and future research directions are suggested so as to enhance the performance of planning algorithms at all three levels. Some promising areas of future focus have been identified as the use of vehicular communications (V2V and V2I) and the incorporation of transport engineering aspects in order to improve the look-ahead horizon of current sensing technologies that are essential for planning with the aim of reducing the total cost of driverless vehicles. This critical review on planning techniques presented in this paper, along with the associated discussions on their constraints and limitations, seek to assist researchers in accelerating development in the emerging field of autonomous vehicle research.

1. Introduction

Autonomous vehicles are a promising evolution of current vehicle technology and advanced driver assistant systems, and are envisaged to be the sustainable future for enhanced road safety, efficient traffic flow and decreased fuel consumption, while improving mobility and hence general well-being (e.g. Thrun, 2010; Burns, 2013; Le Vine et al., 2015). Research on

---

autonomous vehicles has been growing rapidly in recent years and encompasses different domains, including robotics, computer science, and engineering. Moreover, it should be noted that scientific advances have been made by car manufacturers who do not always publicly disclose the details on their approaches or algorithms, owing to commercial sensitivity.

Critical decision making is the key to autonomy and is realised through planning algorithms, incorporated within the middleware of an autonomous vehicle's navigation, situation understanding and decision making module. The main purpose of planning is to provide the vehicle with a safe and collision-free path towards its destination, while taking into account the vehicle dynamics, its manoeuvre capabilities in the presence of obstacles, along with traffic rules and road boundaries (Zhang et al., 2013). Planning is a memory consuming as well as a computationally intensive routine, which is run in parallel with other routine operations of the vehicle (e.g. obstacle tracking, data fusion and control modules). The inputs and outputs of a motion planning normally depend on these other modules. Reliable, robust and adaptable planning is essential, especially in an urban mixed traffic scenario. These algorithms receive inputs from the sensor framework and supplement these inputs with data from digital road maps in order to provide a full workspace in which the planning takes place.

Existing planning algorithms originate primarily from the field of mobile robotics, and have subsequently been applied to different on-road and off-road vehicles and operational environments (e.g. desert vehicles) (Thrun et al., 2006), planetary rovers (Pivtoraiko and Kelly, 2009) and buses (Fernandez et al., 2013). Furthermore, a large number of algorithms have been developed for non-holonomic and car-like robots planning in abstract, simulation-based environments (e.g. Scheuer and Fraichard, 1997). In the review presented in this paper, only approaches concerned with planning for on-road autonomous vehicles are analysed. In general, planning for autonomous or intelligent driving is divided into four hierarchical classes, as suggested by Varaiya (1993): (1) route planning, (2) path planning, (3) manoeuvre choice and (4) trajectory planning (termed as control planning in the work of Varaiya). Route planning is concerned with finding the best global route from a given origin to a destination, supplemented occasionally with real-time traffic information. Route planning is not within the scope of this paper and readers are referred to Thorpe and Durrant-Whyte (2009) for details on a route planner. Path, manoeuvre and trajectory planning components of autonomous on-road driving (often combined as one) take vehicular dynamics, obstacles, road geometry and traffic interactions into account, and are the main focus of this paper. It is important to emphasise that this paper presents a state-of-the-art review of motion planning techniques based on the works after the DARPA Urban Challenge (DUC) in 2007 (Thorpe and Durrant-Whyte, 2009) and is intended to serve as a key reference for researchers who are conducting research on the domain of autonomous vehicles. The focus on studies after the DUC is given because the challenge was a milestone in autonomous driving and resembles the state-of-the-art work until 2007, thus enabling research in autonomous driving to profoundly advance. The novelty of this work lies in the fact that, even though autonomous vehicles are at the core of technological research now-a-days and many attempts have focused on motion planning techniques for mobile robots, to our knowledge, no other work compares and contrasts the approaches concerning planning in all three levels (i.e. path, manoeuvre, trajectory) simultaneously for autonomous on-road driving.

The remainder of the paper is structured as follows: foundational definitions form the body of Section 2; while Section 3 presents an extensive literature review of motion planning approaches applied to autonomous vehicles, followed by their specific characteristics. Key limitations of the approaches are then described in Section 4. Finally, in Section 5 the paper discusses future research directions in order to overcome identified challenges.

2. Definitions

2.1. Definition of planning in the context of autonomous driving

This section describes the key conceptual terms commonly used in the literature within the field of planning for robots and, hence, autonomous vehicles. As mentioned previously, this paper focuses on planning at a local on-road level and not globally (e.g. routeing).

The set of independent attributes which uniquely define the position and orientation of the vehicle according to a fixed coordinate system is termed the *configuration* vector (Eskandarian, 2012). Consequently, the set of all the configurations of the vehicle constitute the *configuration space*.

The set of attribute values describing the condition of an autonomous vehicle at an instance in time and at a particular place during its motion is termed the '*state*' of the vehicle at that moment (Eskandarian, 2012). The most common set of attributes, defined as a vector, which are used to express the state of a vehicle are the position (x, y, z), the orientation $(\theta_x, \theta_y, \theta_z)$, linear velocities (v_x, v_y, v_z) and angular velocities $(\omega_x, \omega_y, \omega_z)$. Subsequently, *state space* represents the set of all possible states that a vehicle can be in. As will be seen in the next sections, the mathematical representation of a *state space* differs from the approach taken by vehicle planning. A trade-off between explicit representation and efficiency of the algorithms should be considered for every planning problem. Representations that can be used for constructing a configuration or a state space will be discussed in Section 2.2.

The *bicycle model* is a dynamic/kinematic model of vehicles, in which the two front and rear wheels are replaced by one front and one rear wheel respectively. The vehicle moves on the plane and its coordinates are described by the vector (x, y, θ) where x, y is the position of the centre of gravity and θ is the orientation of the vehicle. Steering angle of the front wheels is denoted by ϕ. A basic assumption of the bicycle model is that the inner slip, outer slip and steer angles are equal.

A robot is *holonomic* if the controllable degrees of freedom are equal to the total degrees of freedom. Cars or car-like robots are thus *non-holonomic* because they are described by 4 degrees of freedom (2 Cartesian coordinates, orientation

and heading) but have 2 kinematic constraints: (i) they can only move backwards and forwards, tangentially to the direction of their main body and (ii) the steering radius is bounded. Another definition of *holonomy* is described in LaValle (2006), Siegwart et al. (2011) and Laumond (1998), where it is stated that car-like vehicles are non-holonomic because their motion is constrained by non-integrable differential constraints due to the assumption that the wheels roll without slipping.

Actions are system inputs (such as acceleration, steering angle) that result in a vehicle's state transition. Actions are either defined as a function of time or as a function of state and time. *Action space* represents the set of all possible actions that can be applied to the *state space*.

Given a *configuration space* or a *state space*, planning is a computationally intensive task, demanding high memory utilisation. Within the field of robotic motion (both in the case of on-road and off-road vehicles and objects), planning is performed at different levels. The highest level of planning is concerned with origin to destination route planning and the workspace is essentially limited to digital maps representing the underlying road network. The lowest level of planning is concerned with planning a smooth trajectory adhering to vehicular dynamics and such a plan is chalked out on a small (local) search space of high dimensional states. To facilitate the description and discussion, the following terms are defined as used in the rest of the paper.

Path is expressed as a continuous sequence of configurations beginning and ending with the boundary configurations, i.e. the initial configuration and the terminating configuration respectively (Eskandarian, 2012). In other words, a path is a geometric trace that the vehicle should follow in order to reach its destination without colliding with obstacles. **Path-planning** is therefore the problem of finding a geometric path from an initial configuration to a given terminating configuration such that each configuration and state (if time is taken into account) on the path is a feasible one. A feasible configuration/state does not result in a collision and adheres to a set of motion constraints such as road and lane boundaries, as well as traffic rules. It should be noted that, throughout the paper where path planning is discussed, importance is given in finding the best and safest geometric trace, under the constraints described above which also have a logical argument regarding the rules of traffic.

Manoeuvre is a high-level characterisation of the motion of the vehicle, regarding the position and speed of the vehicle on the road. Examples of manoeuvres include 'going straight', 'turning', 'overtaking' etc. A manoeuvre is nominal if it is performed safely according to traffic or other rules. As a result, **manoeuvre planning** addresses the problem of taking the best high-level decision for the vehicle, while taking into account the path that is specified from path planning.

On the other hand, **trajectory** is represented as a sequence of states visited by the vehicle, parameterised by time and, possibly, velocity. **Trajectory planning** (also known as trajectory generation) is concerned with the real-time planning of the actual vehicle's transition from one feasible state to the next, satisfying the vehicle's kinematic limits based on vehicle dynamics and constrained by the navigation comfort,[2] lane boundaries and traffic rules, while avoiding, at the same time, obstacles including other road users as well as ground roughness and ditches. Trajectory planning is parameterised by time as well as acceleration or velocity and is frequently referred to as motion planning. During each planning cycle, the path planner module generates a number of trajectories from the vehicle's current location, with a look-ahead distance, depending on the speed and line-of-sight of the vehicle's on-board sensors, and evaluating each trajectory with respect to some cost function to determine the optimal trajectory. Trajectory planning is scheduled at regular intervals; the length of which largely depends on the frequency of receiving fresh sensor data. For example, trajectory planning was scheduled every 100 ms (ms) in the controller that was tested during the VisLab Intercontinental Autonomous Challenge (Broggi et al., 2012). Error between the current vehicle location and the determined trajectory is monitored; triggering a trajectory revised plan upon detecting an error beyond a pre-defined threshold. It should be noted that there is a rich body of literature on trajectory planning of aircraft movements in the context of air traffic control. Aircraft trajectory planning is, however, quite different from trajectory planning of on-road vehicles. The operational environment and the allocated space for aircraft to manoeuvre is different from the overpopulated, multimodal, congested road network, which is also constrained by road geometry, road lanes and the existence of a large number of obstacles which do not appear in the air. Furthermore, the degrees of freedom, the dynamics and the size of aircraft are different from on-road vehicles in a way that trajectory planning for vehicles and aircraft (which must take all these parameters into account) requires different approaches. For example, as described in Schuster (2015) an aircraft's trajectory is 4D (comprising of the spatial coordinates (x,y,z) and time while a road vehicle primarily acts on a 2D space or 3D space if time is added. Furthermore, the state vector of an aircraft motion planning includes 3D position coordinates $(x, y,$ and height), air speed and aircraft mass. However, the state vector of an on-road vehicle does not consider air speed and vehicle mass. It can therefore be understood that an aircraft trajectory is treated with 6 degrees of freedom, while the trajectory of an on-road vehicle is normally treated with only 3.

The implementation of most existing trajectory planning algorithms follows two steps: (i) trajectory generated on a low resolution/lower dimensional search space in the first step and (ii) the resulting optimal trajectory smoothed out on a higher resolution/higher dimensional search space during the second step. The planning module is integral to rendering complete autonomy to the vehicle with the outputs of the trajectory planner feeding into the low-level steering/manoeuvre control unit.

2.2. Search space for planning

Planning a journey for an autonomous vehicle on the road requires that the environment should be represented in a way that enables the query for a path. This means that the physical space must be transformed into a configuration or a state space. The state space, as defined in the preceding section, consists of every representation of the vehicle position, orienta-

---

[2] In terms of the acceleration (lateral & longitudinal) that the vehicle develops, the curvature of the trajectory and others parameters which are indicated by standards such as ISO 2631-1 1997.

tion, linear or angular velocities in addition to any other measures of interest (Howard, 2009). As the vehicle travels on the road, readings from the sensors and information obtained from a digital map are used to transform the continuum of the environment into a digital representation of the road network which is the essential space for planning. This discretisation must efficiently be dealt with in terms of efficiency, density and expressiveness (Howard, 2009), as high density network may result in high computational costs and power. Similarly, inadequate representation, though it would improve computational speed, may introduce sub-optimality and inexpressiveness, not to mention collision risks.

Some of the existing algorithms initiate a search in continuous coordinates using only the road boundaries and positions of the obstacles, for example, driving corridors (Hardy and Campbell, 2013; Jeon et al., 2013; Wille and Form, 2008; Wille et al., 2010a). Decomposition (or tessellation) techniques analyse the space with higher resolution and include Voronoi Diagrams (Dolgov et al., 2010a; Lee and Vasseur, 2014), occupancy grids (Kolski et al., 2006; Bohren et al., 2008; Hundelshausen et al., 2008; Kammel et al., 2008; Leonard et al., 2008; Zhao et al., 2011; Xu et al., 2014), cost maps (Bacha et al., 2008; Rauskolb et al., 2008; Schröder et al., 2008; Himmelsbach et al., 2009; Murphy and Newman, 2011; Broggi et al., 2012) and lattices (Pivtoraiko and Kelly, 2005; Pivtoraiko et al., 2009; Ziegler and Stiller, 2009; McNaughton et al., 2011). Diagrammatic representations of these search spaces are depicted in Fig. 1.

Voronoi Diagrams or Dirichlet tessellation techniques, generate paths which maximise the distance between the vehicle and surrounding obstacles (Takahashi and Schilling, 1989). Algorithms which are used for searching on Voronoi Diagrams
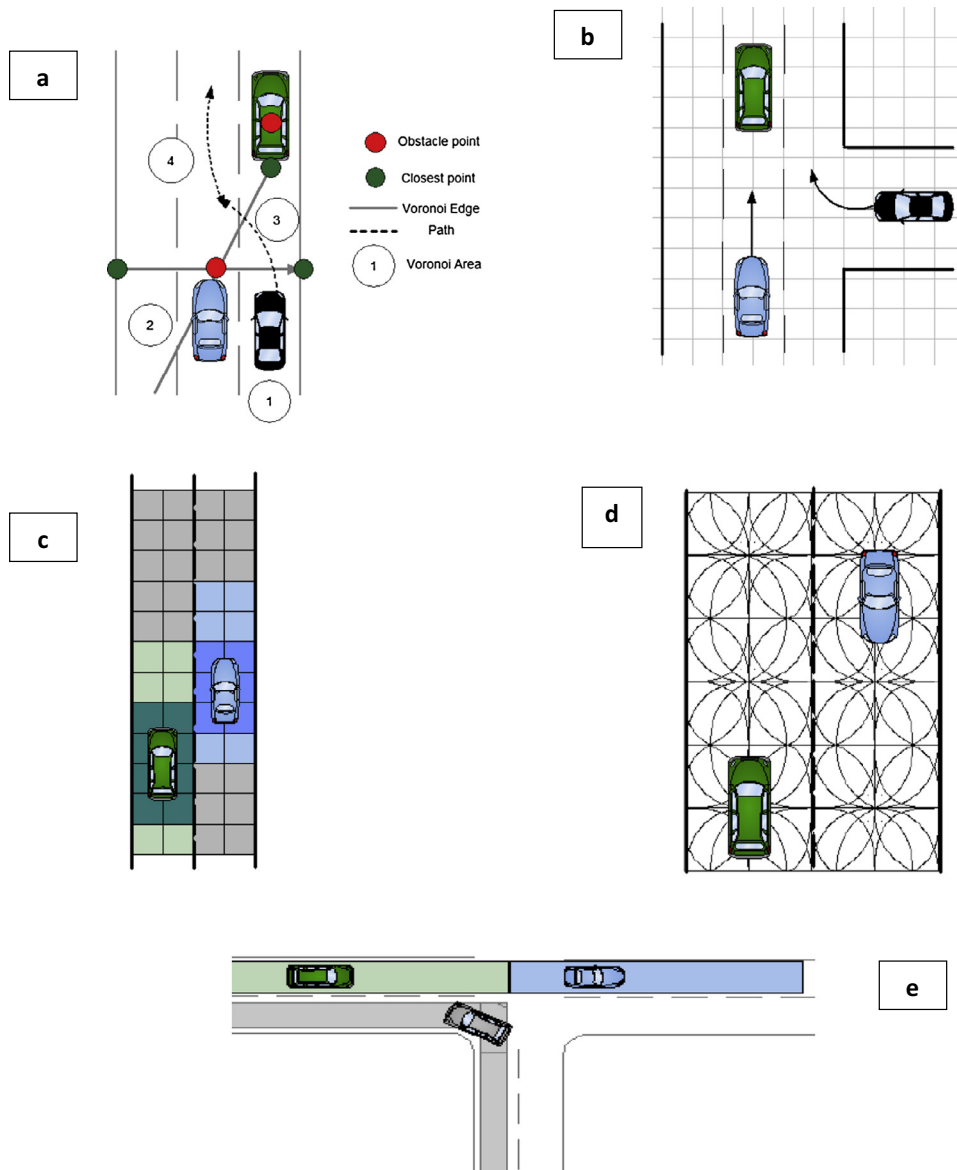
Fig. 1. Graphs used in planning: (a) Voronoi Diagram (Lee and Vasseur, 2014); (b) Occupancy Grid (Schröder et al., 2008); (c) Costmap (Ferguson and Likhachev, 2008); (d) State Lattice (Ziegler and Stiller, 2009); and (e) Driving Corridor (Wille et al., 2010b).

Table 1
Comparison of search space for planning.

Representation	Advantages	Disadvantages
Voronoi Diagrams	• Completeness • Maximum distance from obstacles	• Limited to static environments • Discontinuous edges
Occupancy Grids Cost Maps	• Fast discretisation • Small computational power[a]	• Problems with vehicle dynamics • Errors in the presence of obstacles
State lattices	• Efficiency without increasing computational time[b] • Pre-computation of edges is possible	• Problems with curvature • Restrict motion • Difficulties in dealing with evasive manoeuvres
Driving corridors	• Continuous collision free space for the vehicle to move	• Computational cost[c] • Constraints on motion

[a] Computational power refers to computations needed to construct the cells and estimate their costs. The space, in which the planning problem is solved, is discretised. Furthermore, the number of attributes needed to define each of the cells is small (the attributes just need to show if the cell is occupied or not, plus the cost of traversing the cell). As a result, the dimensions of the state matrix of each of the cells are manageable in real-time.

[b] Similar to (a), computational time refers to computations needed to construct the lattice: Because of the predefined shape of the curve with which the lattice is constructed and the pre-computation of edges, the space for planning is discretised and thus less time is needed to find the correct solution.

[c] Computational cost for driving corridors is analysed in footnote 3.

are complete in the sense that, if a path exists in the free space, it would also appear on the Voronoi Diagram (Siegwart et al., 2011). As depicted in Fig. 1a, grey lines represent Voronoi edges (i.e. edges with maximum distance from detected obstacles), and produce a space where the vehicle can perform its trip. Dolgov et al. (2010a) used Voronoi Diagrams for path-planning of autonomous vehicles in parking lots by combining Voronoi Diagrams with potential fields; an obstacle avoidance algorithm derived from mobile robotics. This combined approach, referred to as Voronoi fields, was developed to overcome the issue of conventional potential field approaches in narrow passages (that generate high potential), which rendered such passages virtually non-traversable. Voronoi Diagrams are typically used for planning in static environments, such as parking lots. Furthermore, Voronoi diagrams on their own are not suitable for on-road path-planning, since Voronoi edges, along which a vehicle navigates, can potentially be discontinuous and unsuitable for non-holonomic vehicle.

Occupancy grids (Kolski et al., 2006; Bohren et al., 2008; Hundelshausen et al., 2008; Kammel et al., 2008; Leonard et al., 2008; Zhao et al., 2011; Li et al., 2013; Xu et al., 2014) and *costmaps* (Bacha et al., 2008; Rauskolb et al., 2008; Schröder et al., 2008; Himmelsbach et al., 2009; Murphy and Newman, 2011; Broggi et al., 2012) work in a similar way; they both discretise the *state space* into a grid and each cell of the grid is associated with a probability of the cell being occupied by an obstacle, or a cost proportional to the feasibility or risk of traversal. Risk or feasibility is primarily calculated by considering the presence of obstacles, lane and road boundaries. Grid-based approaches are fast in finding a solution with low computational power (Pivtoraiko et al., 2009) but have difficulties in accounting for nonlinear dynamics in a robust way (Kushleyev and Likhachev, 2009), and in the presence of obstacles (Pivtoraiko et al., 2009). As seen in Fig. 1b and c, occupancy grids consist of a grid with the position of the obstacles and (sometimes) an attached velocity showing their expected motion; while in cost maps, the higher the cost of a certain cell, the more intense is its presentation on the map.

State Lattices can be seen as a generalisation of grids (Pivtoraiko and Kelly, 2005). In the same way that grids are built by the repetition of rectangles or squares to discretise a continuous space, lattices are constructed by regularly repeating primitive paths which connect possible states for the vehicle, in terms of position, curvature or time, as can be seen in Fig. 1d. The problem of planning then reduces to a boundary value problem of connecting the original state with the required final state (McNaughton et al., 2011). State Lattices overcome the limitations of grid based techniques in efficiency without increasing computational power (Pivtoraiko et al., 2009).

Driving Corridors represent a continuous collision-free space, bounded by road and lane boundaries as well as other obstacles, where a vehicle is expected to move. Driving corridors are based on lane boundary information given on the detailed digital maps, or a map built by using a Simultaneous Localisation and Mapping (SLAM) technique. Lane boundaries form the outer bound of the driving corridors, restricted in the presence of obstacles. In Fig. 1e, a driving corridor is constructed for each vehicle according to the chosen manoeuvre. The centre line of the determined corridor forms the path around which the trajectory to be followed by an autonomous vehicle is planned. The major drawback of planning in a continuous way is that, since intensive computational power[3] is needed for planning for the entire range of coordinates regarding the road network, representation of roads or lanes may constrain the motion of the vehicle (Fletcher et al., 2008).

It should be noted that the above techniques of search space representation for planning are not always employed independently. For example, Voronoi Diagrams and potential fields have been combined to produce Voronoi fields by Dolgov et al. (2010a) to generate a safe trajectory. In most of the cases, they are combined in order not only to provide better results for a single planning level but also to offer planning capabilities in all three levels (i.e. path, manoeuvre and trajectory planning). Their advantages and disadvantages are summarised in Table 1.

---

[3] The continuous nature of driving corridors, leads to an exponential increase in the dimensions of state vector for each one of the coordinates included in the driving corridor. Thus, at each time moment a large number of attributes need to be calculated for each of the coordinates, necessitating more computational resources.

Once a search space is constructed, then the planning algorithms are initiated in order to select the best path, behaviour and trajectory respectively.

3. Planning techniques

This section presents a review of planning techniques used in existing studies in the areas of autonomous on-road driving. Given a route provided by the route planner, motion planning for on-road driving (*hereinafter* planning) concentrates on finding the best path for the vehicle to follow while taking into account the constraints of the vehicle's motion model, way-points that the vehicle should follow and the traffic environment, including static and dynamic obstacles. Planning can be divided into incremental approaches which try to find the best sequence of state transitions (which are not fully specified from the beginning) by re-using information from previous searches and local ones which attempt to find the best single state transition for the vehicle to follow. A global or local path also has a strong correlation with the decisions or manoeuvres that the vehicle performs, so manoeuvre planning will also be addressed. As shown in Fig. 2, path search is initiated after a route has been chosen from the route planner and acts as input to the search for the best manoeuvre (i.e. the manoeuvre which places the vehicle with the most correct and safe behaviour). The final path may however change, based on the best manoeuvre, as shown with a feedback loop between these two modules. Once the path is finalised, the final trajectory planning is generated.

As such, planning is divided into three levels of planning, namely:

(1) Finding the best geometric path for the vehicle to follow
   (a) Finding the best sequence of actions through incremental sampling or discrete geometric structures (i.e. Incremental search).
   (b) Finding the best action from multiple final states (i.e. Local search).
(2) Finding the best manoeuvre to perform.
(3) Finding the best trajectory to follow through the optimisation of a geometric curve, according to given constraints.
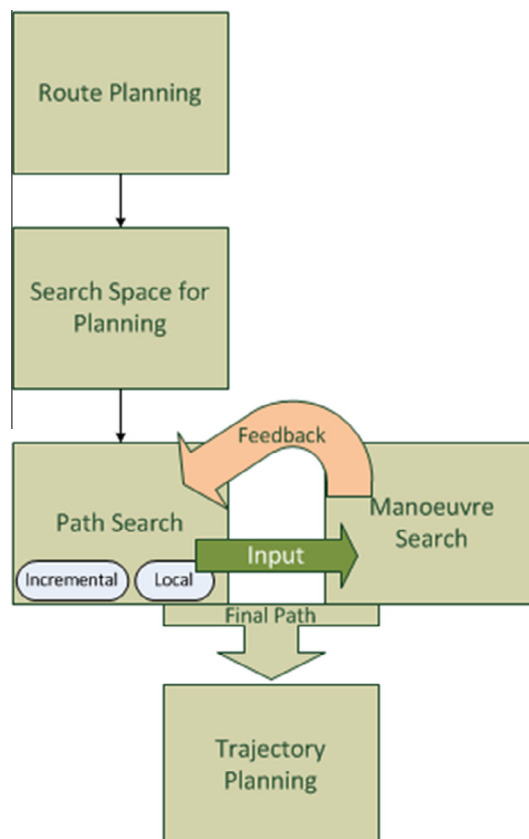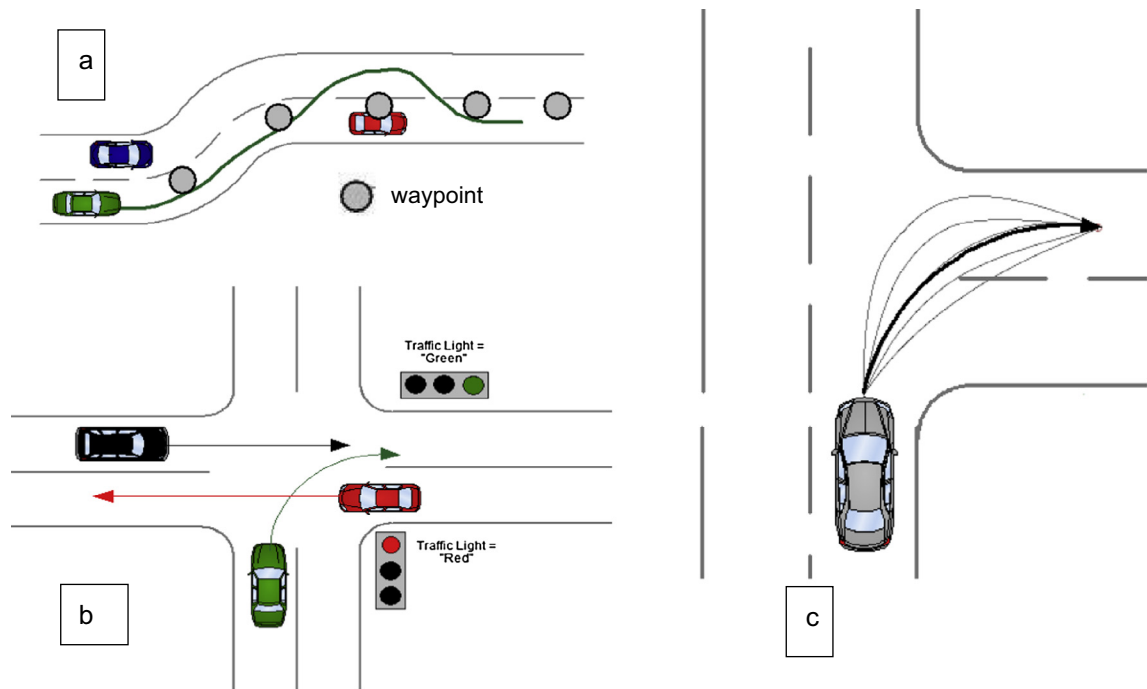

Fig. 2. A flow chart of planning modules.

Fig. 3. (a) Path planning, (b) manoeuvre planning and (c) trajectory planning (adapted from Lee and Vasseur (2014)).

For example, when a vehicle is on the road it follows the sequence of waypoints taken from the route planner and then constructs the geometric path of the vehicle (Fig. 3a). These waypoints must be obstacle-free since the vehicle needs to interact with the other vehicles so as to cooperatively move along the road. According to the geometric path that has been derived and the interactions with other vehicles, the automated vehicle must decide its next 'high level' action (Fig. 3b); i.e. should it overtake the leading vehicle to reach the next waypoint in time? As is implied, these high-level decisions depend on the path, because the vehicle needs reference waypoints in order to decide its best action. If the waypoints and the proper manoeuvre are finalised, then trajectory planning describes the procedure of searching the best way to connect the determined waypoints (Fig. 3c).

3.1. Incremental search

When searching for the best path, incremental search refers to the techniques where search configurations or states are not fully specified in advance. Such algorithms also re-use information from previous search to increase search speed. Two techniques are primarily used: (1) Rapidly-exploring Random Trees (RRTs) and (2) Lattice Planners (LP). They are discussed below:

3.1.1. Rapidly exploring random trees (RRTs)

RRTs construct a tree data structure which is expanded stochastically by adding new configurations (vertices) in each iteration that are randomly sampled from the *configuration space* until the goal configuration is reached. RRTs can also be generalised to be used in the *state space* where a tree is grown from sampled states rather than configurations (LaValle, 1998).

The basic algorithm for constructing an RRT is given in Table 2 (LaValle, 1998).

In Table 2, G is the tree topological graph; C is the configuration space; x_{random} is a configuration sampled randomly from the configuration space; x_{NEAR} is the vertex which is closest to x_{random} in terms of a distance metric; u is a selected input minimising the distance between x_{random} and x_{NEAR} making sure that the new configuration is included in X_{free}; x_{new} is the new configuration, which is acquired if u is applied within a fixed time interval Δt and x_{new} is calculated given a state or configuration transition formula. A *collision detection algorithm* is needed to ensure that x_{new} is obstacle-free. In step 7 the new configuration is added to the tree as a vertex and, finally, in step 8 an edge is created between the randomly sampled configuration and x_{NEAR}. These steps are diagrammatically shown in Fig. 4.

Table 2
A rapidly exploring random tree algorithm.

RRT(x_0)	
1	Initialise a tree (G) starting from point (x_0)
2	**Repeat**
3	Sample a random configuration x_{random} from the configuration space (C)
4	Flag x_{NEAR} the closest point of the initialised tree (G) to x_{random}
5	Select the input u which minimises the distance (x_{random}, x_{NEAR})
6	After Δt and the application of u, flag the new configuration x_{new}
7	Add x_{new} to G
8	Add the edge between x_{new} and x_{NEAR} to G
9	Return G
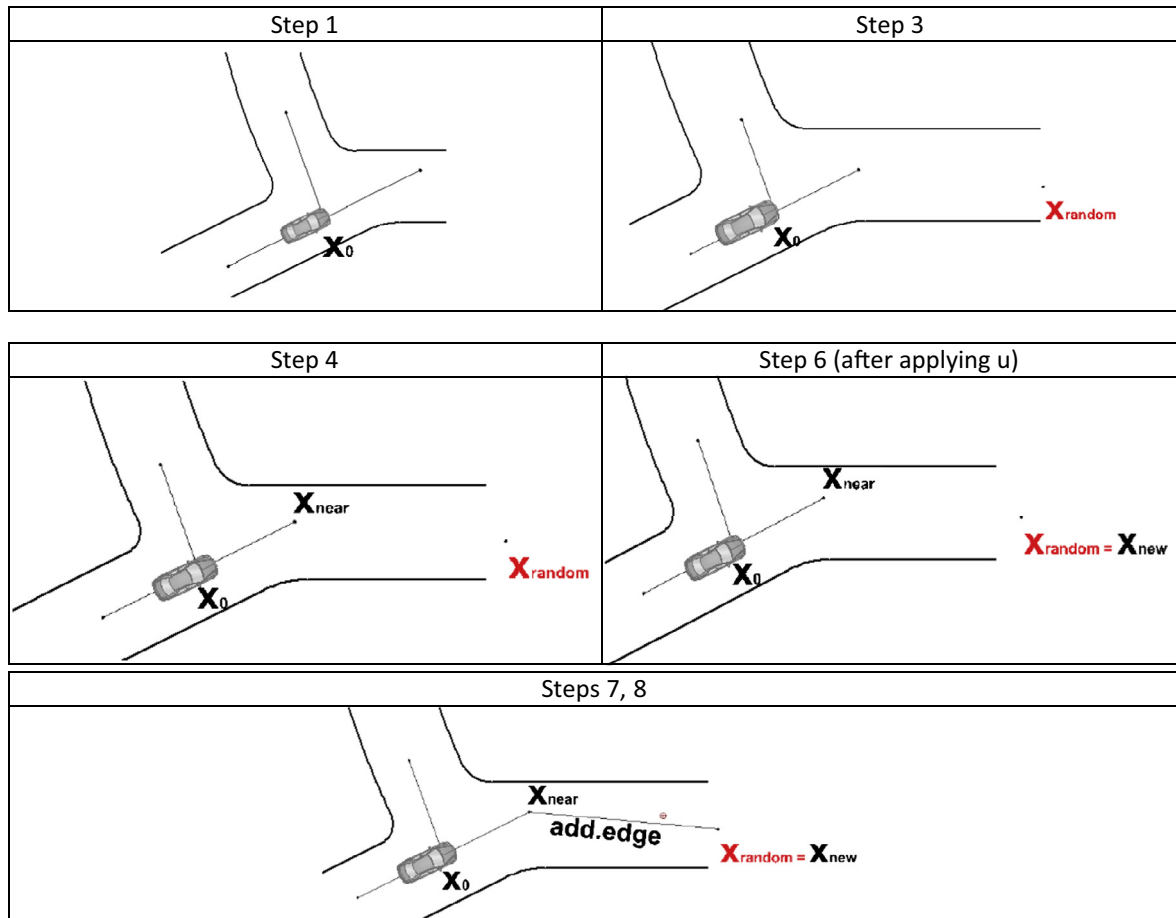
Fig. 4. RRT steps (adapted from LaValle (1998)).

RRTs are probabilistically complete[4] (LaValle, 1998), guarantee kinematic feasibility (Dolgov et al., 2010a), can easily be implemented in real-time, and handle general dynamical models (Kuwata et al., 2009). These strengths of RRT planners are the reasons that they have been used in many cases of autonomous driving. The main advantage of RRTs is the quick exploration of the free space; however, their main drawbacks lie in the jerky paths they create (Murphy and Newman, 2011), as well as in the strong dependence on the nearest neighbour metric for expansion of the tree (Ziegler and Stiller, 2009). Other limitations of RRTs include the necessity of collision checking for every expanded node which, under the presence of many obstacles or heavy

---

[4] Probabilistically complete: if a solution to the path planning problem exists, the probability that RRTs find a solution goes to 1 as running time goes to infinity.
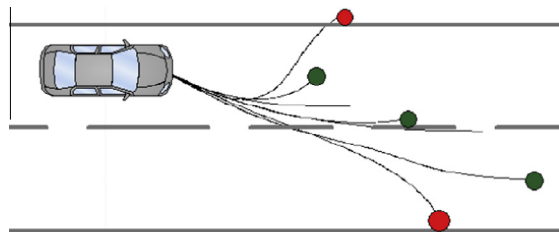
Fig. 5. RRT for on-road driving (adapted by Kuwata et al. (2008)).

traffic, may lead to computational complexity. Furthermore, optimality guarantees are often dismissed in favour of quick exploration of the free space.

Macek et al. (2006) use the simple version of the RRT algorithm along with B-splines to produce smooth paths. Randomisation parameters include acceleration, acceleration duration, sampling time and orientation angle. The collision checking is performed by assuming circular representations for both the vehicle and the obstacles. The algorithm was tested using an 800 m × 800 m simulation environment with stationary and moving obstacles that resemble a realistic traffic environment, and with an automated vehicle equipped with laser rangefinders, monocular cameras, an Inertial Measurement Unit (IMU) and a GPS sensor. Although paths are found effectively, the vehicle is limited to a low maximum speed and acceleration, along with the fact that in one experiment the vehicle had to slow down in order to thoroughly search for the best path to follow, due to a faulty tracking of an obstacle.

Kuwata et al. (2009) use a closed-loop controller in order to sway the sampling of configurations and simulate dynamic trajectories towards them, rather than sampling directly from the configuration space (see Fig. 5). While the classical approach, as shown in the RRT algorithm, samples randomly from the configuration space (Step 3 above), in the approach taken by Kuwata et al. (namely Closed Loop RRT, termed as CL-RRT) samples are selected from the input space of the closed loop controller. Furthermore, each edge added to the tree is a dynamically feasible trajectory and the traversability cost for each vertex is obtained through forward simulation which evaluates the feasibility of the possible trajectory.

The algorithm incorporates efficient techniques such as re-evaluation of states and configurations only if they are selected in the best waypoint sequence and re-propagation of vehicle states to reduce computational complexity of constructing the state space. The search is biased by a large number of heuristics which may increase execution time or convergence time. Two crash incidents took place during its implementation in the DARPA Urban Challenge (Fletcher et al., 2008). An approach to improve CL-RRT is presented in Aoude et al. (2010b) where it is combined with threat assessment for every possible trajectory using game theoretic aspects. The developed algorithm (termed as RRT-Reach) searches the whole tree for threats, assuming a perfect knowledge of each policy is adopted from the obstacles and it is tested on simulations with only two vehicles. Another improvement to CL-RRT, in terms of obstacles handling, is implemented by Aoude et al. (2013) where Gaussian processes are used to estimate the intentions of the obstacles in order to safely bias the expansion of the tree, but the trade-off between path optimality, real-time constraints and intention estimation is an area that needs further research.

Karaman and Frazzoli (2011) developed the RRT* algorithm to guarantee the optimality in terms of the path proposed. In RRT*, all feasible connections are evaluated according to their cost-to-go and only vertices on minimum cost paths are added to the tree. A solution found by an RRT* is more likely to be an optimal one. Jeon et al. (2013) employ the RRT* algorithm and construct paths which are calculated through the half-car dynamic model, Using this dynamic model results in dynamically feasible trajectories on U-turns and a circuit environment without obstacles. Although using a dynamic model results in more computational effort for the feasibility checking of the tree edges, the combination with the RRT* leads to solutions which are optimal within a given control input interval (i.e. provided by a route planner) but may not necessarily be globally optimal.

Garrote et al. (2014) also use RRT*, but modify it with two extra routines (one for collision checking and one for expansion). With regard to collision checking, for every explored node, the time for collision checking is measured in order to bound the time needed for planning. A penalisation routine is responsible for checking if it is feasible or infeasible to grow the tree from each new node. The idea behind enhancing the RRT* algorithm with these two extra features is to focus on the expansion of the tree according to the motion of dynamic obstacles. In the work of Garrote et al. (2014) the ego-vehicle is simulated as a rectangle while the obstacles take circular forms. The algorithm was tested in a simulated environment only with a static and a dynamic obstacle and no statistical tests on the performance of the simulation were presented.

A more advanced technique is given in Reyes Castro et al. (2013). The dynamical system of the vehicle and the road network is formed as Kripke structure (Kripke, 1963)[5] and a non-deterministic finite automaton is used for capturing the traffic rules. The combination of these two techniques are used to grow an RRT* (termed as the Minimum Violation RRT*) but with no obstacles on the road network.

---

[5] A Kripke structure is a graph used for the semantic representation of a system's behaviour. The nodes represent reachable states and the edges represent the transition between these states.

3.1.2. Lattice planners

As mentioned in Section 2.2., state lattices construct a discrete search space which enables relevant state continuity, acquires goal states in a deterministic way and satisfies the differential constraints of the vehicle. It also reduces computational cost as pre-computation is enabled (Howard, 2009) except in cases where the lattice is bent to follow the shape of the lane or the road (Madas et al., 2013). They are generally well-suited for non-holonomic and highly constrained environments (Likhachev and Ferguson, 2009), such as the road environment.

Lattice planners are resolution complete. This means that the control space can automatically be adjusted for every resolution change and the space is explored consistently. Lattice planners also guarantee optimality and smoothness because they do not introduce discontinuities related to back-pointers (Pivtoraiko et al., 2009). The plan generated is mostly close to the real motion of the vehicles but evasive manoeuvres cannot be performed efficiently because alternate goal states cannot be quickly considered (Werling et al., 2011). Furthermore, discretisation in heading angle can be problematic and may lead to oscillations between orientation samples (as shown in Fig. 6) and exhaustive sampling may lead to unnecessary computational complexity.

Likhachev and Ferguson (2009) use a multi-resolution state lattice with high resolution near the vehicle and the goal and low resolution elsewhere. Each state includes position (x, y), heading (θ), and speed that increase the search by a factor of 3 compared to a regular lattice. The different resolution enables the construction of long-term plans for the vehicle without increasing the computational complexity of planning through exhaustive sampling. This multi-resolution lattice is searched with a deterministic algorithm (termed as Anytime Dynamic A*) which is suitable for the limited and imperfectly perceived nature of the planning space, as well as limited computational time. Experiments were carried out with a real-world automated vehicle during the DARPA Urban Challenge, travelled at a maximum speed of 15 mph, but the main drawback was found to be curvature discontinuity.

To address curvature issues, an input- and state-lattice is constructed in Rufli and Siegwart (2010) by defining a 2D input set U (heading and steering angle are discretised) over the 4D-configuration space including 2D position, heading and steering angle. The lattice spans in 24 directions that join the nodes with cubic polynomial curves and has the capability to deform itself so as to adjust for curved and narrow road structures given a reference path.

Globally fixed state lattices require flat terrain and fixed mobility, thus restricting the vehicle's movement, since a small differentiation from the lattice set trajectories can significantly increase search time and computations.

This drawback motivated the development of spatio-temporal state lattices as shown in Fig. 7 by Ziegler and Stiller (2009) in which state-space and time are bonded in a single manifold, before being discretised analogously with a non-temporal lattice. In their work, quintic polynomials are used for connecting the vertices and a deformation paradigm for the lattice to match the shape of the road is also displayed. The addition of time increases the dimensionality (2D position, 2D velocity, 2D acceleration and time are described by each state). Efficient sampling of the state space can be performed in real-time and the edge weights can be assigned more flexibly. The drawback is the fixed values on the set of time and velocities, as well as the imbalance between quintic polynomials and the vehicle kinematic equations.

In McNaughton et al. (2011) curvature is added to each state (a state initially contains 2D position, heading and curvature) and the paths (between the vehicle and the sampled endpoints) are connected with cubic polynomial spirals. A range of
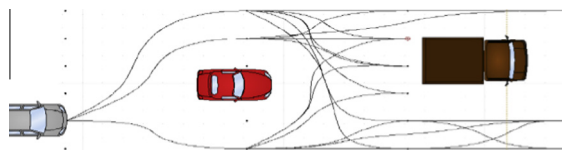

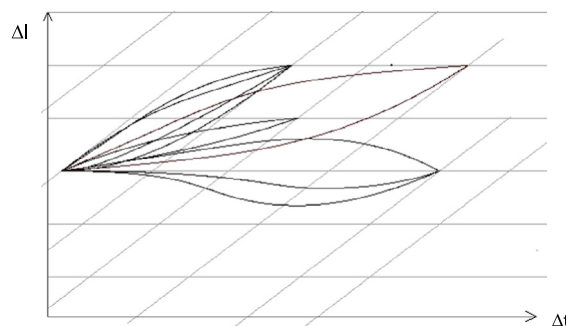**Fig. 6.** Lattice planner for on-road driving (Madas et al., 2013).

Fig. 7. Spatio-temporal State Lattice (adapted from Ziegler and Stiller (2009)).

Table 3
Comparison of RRTs and Lattice planners for incremental search planning.

RRTs	Lattice planners
Advantages	*Advantages*
• Kinematic and real-time feasibility	• Low computational power needed
• Quick search of free space	• Smoothness and optimality of the path are guaranteed (within the given lattice)
• Advanced decision techniques are applied for collision checking	• Generally appropriate for dynamic environments
• Optimality in the path is guaranteed in newer implementations such as RRT*	• Paths comply with the dynamic and kinematic abilities of the vehicle
Disadvantages	*Disadvantages*
• Jagged paths	• Time inefficiency with the calculation of a path for evasive manoeuvers
• Heavily dependent on the Nearest Neighbour heuristic to expand	• May lead to exhaustive sampling or oscillations
• Each node of the tree needs to be checked for collisions while the tree is expanding	• Transferability
• Advanced techniques for collision checking pre-suppose perfect knowledge of the environment	

times and velocities are then assigned to each vertex to enable spatio-temporal search; a constant acceleration is however required making it difficult for the vehicle to follow such an acceleration profile. The assignment of a range for time and velocity reduces the search space but introduces sub-optimality and the necessity to calculate the exact value after the evaluation of each trajectory. In order to find the best path, a dynamic programming algorithm searches exhaustively in the lattice, taking into account the presence of obstacles as well as travel time and desired behaviours.

To improve the McNaughton et al.'s approach, Xu et al. (2012) use quartic curvature polynomials to provide continuous curvature change rate among planning cycles; connections are made not only from the sampled endpoints but also from the current vehicle pose. Another differentiation is that speed profiles are inversely generated and that comfort,[6] efficiency and energy consumption are also taken into account for evaluating the alternative paths.

A spatio-temporal lattice is also used in Gu et al. (2013) in a 2-level planning approach which, firstly, generates an optimal collision-free reference path, and then samples the state space in order to find the best path according to the reference one. The reference path is constructed to deal with exhaustive sampling; leading to a more focused search and a more human-like driving style.

3.1.3. RRTs vs Lattice planners

A comparison between RRTs and Lattice Planners is provided in Table 3:

In summary, it can be concluded that both RRTs and lattice planners sample the state space using data structures (trees and lattices respectively), trying to explore it in a quick and safe fashion. Quick exploration is accomplished in both cases and a series of possible paths are provided to the planning module for the vehicle to follow. However, the planning horizon is claimed to be relatively large and, with regard to the dynamic nature of on-road driving, where obstacles or obstructions appear suddenly, re-planning routines are needed to supplement these incremental search approaches. Finally, instead of the built-in collision checking functions of the algorithms, extra collision prediction modules should be employed for higher safety.

3.2. Local search

Searching the entire graph in real-time is not always efficient; therefore, some of the approaches use a limited horizon, both in terms of time and space.

Possibly the most popular technique used for on-road autonomous driving in the local search level is the one in which a search space contains a certain geometric curve (e.g. clothoids or splines) and several lateral shifts of this curve (e.g. Montemerlo et al., 2008; Urmson et al., 2008; Broggi et al., 2012; Ma et al., 2012). Each candidate path is then evaluated through a cost function with several considerations, such as distance and time costs, acceleration and collision checking.

Paths generated with lateral shifts can generally be split into two categories: (i) Lateral shifts in the action space of the vehicle and (ii) Lateral shifts in the state space of the vehicle. A relative comparison is shown in Fig. 8 (Howard, 2009):

Rolling out trajectories with lateral shifts may not perform well in complex dynamic environments (Gu and Dolan, 2012). Instead of searching the entire state space towards the goal, with an infinite time horizon requiring substantial computational power (for constructing the state space), Partial Motion Planning (PMP) developed by Benenson et al. (2006) can be used. PMP uses a short time horizon combined with RRTs and taking advantage of the concept of Inevitable Collision States (ICS) (i.e. definition of states of the vehicle where a collision cannot be avoided) for a safety check. Inevitable collision states guarantee that a collision will not happen but require a full knowledge of the surroundings of the vehicle (Althoff et al., 2011)

---

[6] Comfort is evaluated in terms of curvature and rate of change of curvature of the path, as well as the acceleration, the jerk and the centripetal acceleration which act on the vehicle.
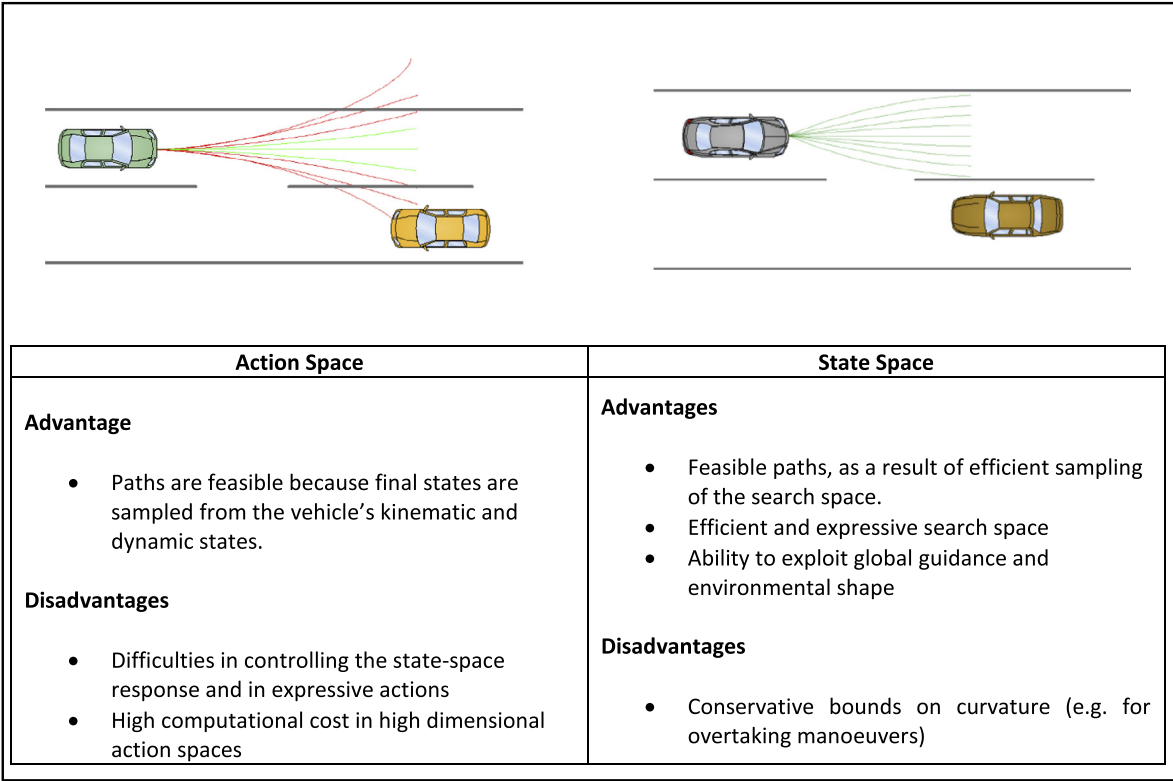
Action Space	State Space
Advantage • Paths are feasible because final states are sampled from the vehicle's kinematic and dynamic states. **Disadvantages** • Difficulties in controlling the state-space response and in expressive actions • High computational cost in high dimensional action spaces	**Advantages** • Feasible paths, as a result of efficient sampling of the search space. • Efficient and expressive search space • Ability to exploit global guidance and environmental shape **Disadvantages** • Conservative bounds on curvature (e.g. for overtaking manoeuvers)

Fig. 8. Comparison of action space search and state space search in the trajectory level.
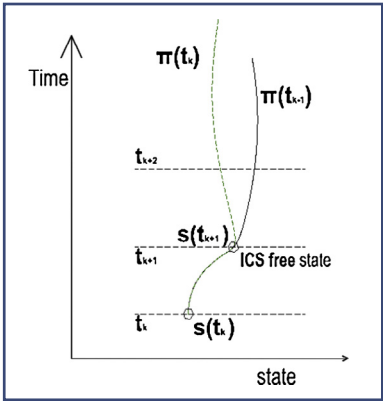

Fig. 9. Partial motion planning concept (Benenson et al., 2006).

and PMP approaches usually drop near-optimality guarantees (Kushleyev and Likhachev, 2009). As can be seen in Fig. 9, if an ICS is found, an alternative path is searched and at each time step nodes are expanded in an RRT fashion.

In summary, both rolling-out trajectories and PMP approaches aim at finding the best next action (in terms of path as an action) suitable for the journey of the vehicle. Swerving trajectories sample final conditions either in the action or in the state space, whereas PMP approaches use RRTs to plan with a limited time horizon. Techniques which sample the state space may lead to smooth and safe paths, with respect to lane and road boundaries as well as the motion of other traffic participants. This comes in contradiction with PMP and techniques which sample the action space, which may lead to infeasibility of paths accommodated with a lot of obstacle handling.

3.3. Manoeuvre planning and decision making

During the DARPA Urban Challenge in 2007, analysis showed that there was a lack of interactions between vehicle and driving in a human-like manner, with many incidents of a behavioural nature being faced during the challenge (Fletcher et al., 2008).

While driving autonomously on public roads, the vehicle at each moment should be capable of deciding the best and safest manoeuvre to undertake after finding the best geometric sequence of waypoints to follow. This decision must be made without overlooking the ego-vehicle's interactions with the surrounding traffic environment. Manoeuvre planning therefore, incorporates techniques which anticipate the behaviour of both the motorised and non-motorised traffic participants and assesses the surrounding traffic situation, thus arming the driverless vehicle to decide on its best manoeuvre. Techniques which are described in this section work on a more high-level basis. Manoeuvre planning moves away from searching a path or generating a trajectory; instead acting as a 'brain' which filters the results of path search, interacts with other traffic participants and gives the approval for the geometric path before it is transformed to a feasible trajectory.

Techniques for manoeuvre planning can be divided into two categories:

(1) Those that emphasise motion modelling and obstacle prediction.
(2) Those that are concerned with the decision making module of autonomous vehicles, based on the modelling of the traffic environment.

3.3.1. Motion modelling – obstacles prediction

Lefèvre et al. (2014) present a detailed survey to classify recent research on traffic environment modelling and prediction and introduce several risk estimators for intelligent vehicles. According to their work, motion models are classified into physics-based, manoeuvre-based and interaction-aware models. The first category describes motion models according only to the laws of physics, while the second relies on estimating the intentions of other traffic participants, based either on clustered trajectories or on manoeuvre estimation and execution. These two categories of motion models do not take into account the environment, but rather, view vehicles as independent entities. Interaction-aware models were developed in order for the inter-vehicle relationships to be exploited, so that dangerous situations can easily be modelled and identified in real-time.

As far as risk estimators are concerned, Lefèvre et al. indicate that a crash can be predicted through collision prediction (binary or probabilistic) through estimated trajectories, but also through unexpected behaviour or conflicting manoeuvres between vehicles. The readers are referred to the survey of Lefèvre et al. for further details on the description of techniques and risk indicators.

Most of the approaches for obstacle prediction (also included in the survey of Lefèvre et al.) refer to straight roads and do not apply well to the context of each manoeuvre.

A grid-based Bayesian filter is used by Alin et al. (2012) to model behaviours as spline functions to anticipate curvy roads and infer the trajectories of other vehicles. The technique shows better results than Bayesian filters that do not take into account context, but considers only cut-in and lane change manoeuvres.

A hybrid-state system using hierarchical hidden Markov models and Finite State Machines is used by Gadepally (2013) to predict future state of traffic participants at intersections. This model is motivated by the fact that vehicle behaviours (such as turning in different directions) can easily be estimated by human drivers but are not efficiently anticipated by automated vehicles. Nevertheless, the approach needs extensive training and extensive data acquisition to train the models.

Ontology, a formal description of entities, hierarchies and interrelationships used in computer and information science, is used by Armand et al. (2014) to reason about the behaviour of traffic participants. Only a limited number of situations (going straight, following and reaching a vehicle or pedestrian) are evaluated using few rules and time efficiency issues are also noticed.

Gindele et al. (2015) use Partially Observable Markov Decision Processes (POMDPs) to cope with simplified models of the traffic environment and make accurate predictions for other traffic participants. They can predict motions for about 6 s into the future at an intersection environment and their technique can respond well with noisy sensors, allowing long-term predictions. However, online learning capabilities are excluded and, in that way, adaptation to new traffic conditions may be difficult.

3.3.2. Planning with emphasis on obstacle prediction and decision making

Furda and Vlacic (2011) use Multiple Criteria Decision Making (MCDM) and Deterministic Finite Automata (DFA) for driving manoeuvre execution. The inputs come from *a priori* known data, sensor measurements and vehicular communications. Traffic rules and a hierarchy of objectives during driving are considered for decision making (namely, motion within road boundaries, safety distances, collision avoidance and minimisation of waiting time). The approach needs accurate information and manually specified weights for each objective in the decision making routine.

In Hardy and Campbell (2013), driving corridors are constructed according to the predicted motion of dynamic obstacles and the presence of static ones. Vehicles are modelled as rectangles; their trajectories are clustered for easier identification and conflicting trajectories are used to estimate the risk at each moment. In this work, planning is seen as a non-linear constrained optimisation problem. The function which is to be optimised includes terms for static and dynamic obstacles, pos-

sible collisions and distance to goal. Single and multiple obstacles are considered but building the driving corridor increases the computational effort linearly (as described in Section 2.2), according to the number of obstacles and interactions between vehicles that are ignored.

A similar approach is used by Ziegler et al. (2014b) where hierarchical concurrent state machines are used with respect to static and dynamic obstacles, as well as yield and merge rules. Driving corridors are also indicated in order for the vehicle to have optimal free space for each part of the journey, while avoiding collisions. The main drawback of this technique, however, is that other vehicles are presumed not to accelerate and to keep safe distances from the road boundaries.

Kala and Warwick (2013) consider a relatively unstructured road environment. They assume that no road lanes exist and that the majority of the traffic participants are non-autonomous and there exists no communication between vehicles. At each moment, the vehicle is supposed to display certain behaviour according to the motion of vehicles nearby. Obstacle avoidance, centring (driving in the centre of the road/lane), lane changes, overtaking and being overtaken, slowing down, detecting conflicting behaviours and travelling straight are the pre-designed behaviours. Distance and velocity constraints are used to classify different behaviours online. This work studies only straight roads with infinite length and shows that there is a delay in the decision making of the vehicle in cases such as centring on curvy roads or overtaking. The fast and correct identification of conflicting behaviours between road users is another drawback of this approach.

A Prediction and-Cost-function Based (PCB) approach is adopted by Wei et al. (2014). Using a reference trajectory, as well as static and dynamic obstacles as inputs, multiple candidate trajectories are generated and, after predicting the evolution of the traffic environment, the best strategy is chosen according to comfort,[7] safety, fuel consumption and the progress towards the goal. The motion of the vehicle in the vicinity of other vehicles is considered, and controller reactions and time delays are also simulated for better performance. The approach was validated with simulation and on-road testing and leads to smoother results, as compared to the spatio-temporal lattice planner and with a reduction of 90% on computational cost. However, only single lane behaviours are considered.

White and White (1989) employ Markov Decision Processes (MDPs) in estimating the best manoeuvre for the vehicle to undertake MDPs incorporate a presumed set of actions which are performed under uncertainty and try to maximise the total rewards or weights for every action. MDPs work on the state space to try to determine a rule which describes the decision to act from one state to another.

Unlike MDPs, which assume that the states are fully observable, partially observable Markov Decision Processes (POMDPs) assume that the state of a robot or a vehicle is not known (Ong et al., 2010). Thus, POMDPs transform the state space into a belief space, which contains all the possible probability distributions for every possible state of the system that is being modelled. If, however, some features of the state of a vehicle are known (for example, the orientation is known but the position is not), we are referring to mixed observability MDPs or MOMDPs (Ong et al., 2010).

In the work of Bandyopadhyay et al. (2012) intention prediction about human traffic participants is embedded into planning. A discrete Mixed-Observability Markov Decision Process (MOMDP) models the interaction between the autonomous vehicle and pedestrians, while making a prediction about the pedestrians' intentions. The behaviours of the ego-vehicle towards the pedestrians that are considered include: 'Reasonable but Distracted', 'Oblivious', 'Impatient' and 'Opportunistic Driving'. Experiments are carried out with simulations and a real-world golf-cart; wherein it is assumed that information about pedestrians' positions and velocity is perfectly known. Furthermore, instead of simultaneously treating the set of pedestrians, a MOMDP is separately calculated for each pedestrian and it is assumed that intentions do not change over time. Lastly, experiment results are presented for only half of the behaviours considered.

In contradiction with the previous work, Brechtel et al. (2014) implement a continuous partially- observable MDP, assuming that the belief state is infinitely large because driving is a continuous-space problem. The inputs are the position and velocities of the traffic participants, which are presumed known. Merging scenarios are simulated where the ego-vehicle has occluded vision due to a hypothetically illegally parked car. Having a continuous belief space may lead to a large number of samples to be needed to make the autonomous vehicle decide. This large number of samples may consequently lead to large computational effort and also increases the number of close calls for decision making.

Game Theory has also been used by researchers to take into account the interactions between vehicles. For example, Aoude et al. (2010a) examine an intersection environment and try to formulate a perfect information game between traffic participants. Each game terminates if a crash happens, and each vehicle tries to maximise the time to collision, while all other vehicles take on the role of 'enemies' which try to minimise this time. This threat assessment model is then embedded into an RRT-like global planner which generates the path to follow. Real-time capabilities of the approach are provided by evaluation which takes place with two model cars with maximum speeds of 0.5 m/s.

The same concept of Game Theory is followed in the work of Martin (2013) where, again, a perfect information game is used to predict the motion of other vehicles for planning on highways. For the payoff function to be maximised by the ego-vehicle, position, speed and accelerations are taken as input; producing as output the best possible manoeuvre using a manoeuvre set which includes driving straight, as well as left or right lane changes. The road is assumed to be infinitely straight and simulations are carried out with up to 4 vehicles in the traffic scene.

---

[7] In the study of Wei et al. (2014), comfort is evaluated according to the acceleration of the vehicle.

Table 4
Planning approaches with emphasis on obstacle prediction and decision making.

Study	Method	Criteria	Environment description	Drawbacks
Furda and Vlacic (2011)	Multiple Criteria Decision Making	Traffic rules Road boundaries Safety distance Collisions Waiting time	Non-intersection segments	Need precise information and manually specified weight for each criterion
Hardy and Campbell (2013)	Driving Corridors and Non-Linear Constrained Optimisation	Behaviour towards static and dynamic obstacles Vehicle dynamics Distance to obstacles Distance to goal	Intersections	Computational effort rises with number of obstacles\n\nIgnorance of social interactions between traffic participants
Ziegler et al. (2014b)	Driving Corridors and Hierarchical State Machines	Static and dynamic obstacles behaviour Yield and Merge rules	Intersections and non-intersection segments	Other vehicles are presumed not to accelerate and to keep safe distances from road boundaries
Kala and Warwick (2013)	Behaviour Choice according to Obstacle Motion	Distance and velocity constraints	No road lanes	Infinite straight roads Problems on curvy roads, overtaking and conflicting behaviours
Wei et al. (2014)	Prediction and Cost-function	Comfort, safety, fuel consumption, distance to goal	Straight roads	Only single-lane behaviours tested
Bandyopadhyay et al. (2012)	Mixed-Observability MDP	Pedestrian position and velocity	Pedestrian crossings	Different modelling required for each pedestrian Intentions are assumed unchangeable
Brechtel et al. (2014)	Partially Observable MDP	Vehicle position and velocity	Merging scenarios with occluded vision	Continuous belief space may lead to large number of samples and large computational effort
Aoude et al. (2010a)	Game Theory	Time to collision	Intersections	Model-car evaluation at low speeds Perfect information assumed
Martin (2013)	Game Theory	Position, speed, acceleration and manoeuvre choice	Straight roads	Perfect information required

Planning approaches which emphasise obstacle prediction and decision making are summarised in Table 4.

To summarise, manoeuvre planning relies heavily on the relative positions of other traffic participants at the moment of making a decision and estimating the risk of a certain situation. Risk estimation can be performed using risk indicators, such as the Time-to-Collision (TTC), as suggested by Ward et al. (2014, 2015), probabilistic gap acceptance models, as proposed by Lefèvre (2012), or by forming situation assessment and choosing the best manoeuvre as a decision theoretic problem (using Markov decision processes or Game Theoretic principles). In the first category of planning (obstacle prediction and risk assessment), more accurate results are provided but context is often omitted from planning and heavy computational burden may arise while predicting the motion of the obstacles in the vicinity of the autonomous vehicle. Decision-theoretic approaches cope well with context and may provide solutions to problems like negotiating intersections (such as in urban or suburban environments) or complying with manoeuvres on a highway.

3.4. Trajectory level

After finding the best path to follow and the best manoeuvre to undertake, a trajectory must be generated that satisfies the motion model or state constraints and guarantees comfort[8] for the passenger and smoothness for the trip. The problem of generating a trajectory, according to the path and the manoeuvre that is chosen, is primarily solved by selecting a geometric curve to assure smooth motion through the road network. This trajectory is then optimised by using a cost function according to the dynamic model and/or the presence of obstacles along that trajectory.

The geometric representations of the trajectories include arcs (e.g. Sun et al., 2014) clothoids (e.g. Broggi et al., 2012), Nelson polynomials (e.g. Xiu and Chen, 2010) polynomial spirals (e.g. McNaughton et al., 2011; Gu et al., 2013), spline curves (e.g. Wang et al., 2011; Hardy and Campbell, 2013; Kala and Warwick, 2013) and Bezier curves (e.g. Zhao et al., 2011; Ma et al., 2012).

---

[8] In terms of the acceleration (lateral & longitudinal) that the vehicle develops, the curvature of the trajectory and other parameters indicated by standards such as ISO 2631-1 1997.

In these type of planning problems, the geometric curve used is restricted to a certain type (e.g. splines, Bezier curves, polynomials or clothoids) and a piecewise path is constructed through solving a two-point boundary value problem at each time moment.

Delsart et al. (2009) proposed a trajectory generation algorithm (named as TiJi) which optimises a second order polynomial trajectory, according to the error between the current state and the final state. A vehicle model which includes 2D position, orientation, steering angle and velocity is used, and velocity and acceleration profiles are computed for each point. No obstacles are taken into account and the computational complexity increases when the required convergence points are far from the final state.

Sawodny et al. (2010) use fourth order polynomials and a dynamic bicycle model with the flatness property[9] to describe the motion of the vehicle. Constraints are applied to maximum acceleration and the maximum driving force for the vehicle. The approach also considers overtaking and car-following scenarios with different cost functions for each case, based on the probability of a vehicle presence in the opposing lane and a safety distance constraint. In the simulation experiments, it is assumed that vehicles employ a constant velocity; something which may not be applicable to real-world conditions.

A smoother and more controllable trajectory is developed by Ma et al. (2012), where the authors use cubic Bezier curves for trajectory planning. Their cost function includes weights for the length of the trajectory, its smoothness and the offset from the central line of the lane used as a reference path. Obstacles are considered as circles and collision checking is performed via colliding trajectories which can lead to over-computations.[10] Cubic Bezier curves are also used in Perez et al. (2013) to generate trajectories for autonomous vehicles in roundabout scenarios; however, collision checking, lateral accelerations and curvature constraints are omitted.

A comparison between 3rd and 5th order Bezier curves for lane-change manoeuvres presented by Bae et al. (2013) indicates that 5th order curves construct a much smoother and more feasible path for dynamic on-road environments. A miniature-size vehicle is used to test the approach, but real-world vehicle dynamics are not taken into consideration in this kind of testing.

Fourth degree Bezier curves are used for better results by Gonzalez et al. (2014). The road constraints are taken into consideration, in addition to lane constraints that reflect the path indicated by path planners. This work also incorporates a deformation of the curve if obstacles are present. Although constraints are put on velocity and acceleration for comfortability issues, the authors assume low speeds and consider only static obstacles.

A more complete approach, in terms of manoeuvres and traffic rules considered, is proposed by Wang et al. (2011), where Akima splines interpolation is used to generate the trajectories which comply to checkpoints given by the path planning module, stop signs, traffic lights, turns, lane changes, intersections, give-way signs, reverse driving and dead-ends. The trajectories are evaluated by their distance and time to get to next checkpoint as well as collisions with obstacles. Demonstrations are presented in a simulated environment, as well as a real-world vehicle, and were claimed to be sufficient for real-world autonomous driving, although extensive results are not provided.

Gu and Dolan (2014) propose to learn trajectory patterns from human drivers and then apply these patterns for online trajectory generation. Splines are used for trajectory representation in order to interpolate between the waypoints given from higher planning modules. Trajectories are constrained to curvature, speed and acceleration. The environment, however, is supposed to be traffic-free, and regulations or external factors like weather conditions are not taken into consideration; not to mention that a substantial quantity of naturalistic driving data must be used for learning human driving patterns.

Trajectory planning for BMW's autonomous vehicle Bertha takes place inside the driving corridor which limits the action space of the vehicle (Ziegler et al., 2014a). Weights for the cost function concern the offset from the original path, velocity and acceleration, as well as the desired behaviour of the vehicle. Furthermore, the kinematics and dynamics of the vehicle constrain the curvature given to the trajectory and the allowed friction of the tyres. Obstacles are considered through circle approximations and a re-planning scheme is followed if new information arrives to the sensors. Computational complexity (for the construction of the driving corridor) and universal applicability of the approach are considered to be the main drawbacks.

A different approach for trajectory planning, which combines aspects of control engineering within the planning module, is Model Predictive Control (MPC). Within MPC, a dynamic model for the vehicle is used and, through it, inputs from the controller are sampled about the future evolution of the vehicle's motion. From the dynamic model and the controller inputs, the optimisation problem of finding the best trajectory for the vehicle is solved. MPC was used within a driving corridor from Madas et al. (2013) using a linear bicycle model with linear tyre characteristics which also considers lateral and yaw dynamics. Constraints were derived for the prediction and control horizon, as well as the side slip and steering angle, the steering angle rate and the lateral position of the vehicle. Results showed small lateral acceleration; thus, more comfort for potential passengers but, compared to spline and lattice methods it requires more jerk. A weakness of the technique lies in the fact that the more variables used to model the vehicle, the harder it gets to optimise its trajectories. Nevertheless, because it was performed inside the driving corridor, it is more appropriate for structured domains (e.g. highways) and its performance is not dependent on the increase or decrease of the number of obstacles.

A MPC trajectory approach was also presented by Li et al. (2014) using a reference path and a vehicle model, describing its position coordinates, its heading and the curvature of the path it is on. Cubic splines were used for representing trajectories

---

[9] A nonlinear system $\dot{x} = f(x, u)$ is differentially flat if an output y exists such that x and u can be expressed in terms of y and a (finite) number of its derivatives (Peters et al., 2011).

[10] Over-computations may happen due to the continuous need to predict each obstacle's future position and trajectory.

Table 5
Comparison of trajectory planning approaches.

Study	Geometric curve/ approach	Parameters	Vehicle model	Limitations
Delsart et al. (2009)	2nd order polynomials/ TiJi Algorithm	Initial to final state error	2D position Orientation Velocity Acceleration Steering angle	No obstacles considered Computational complexity increases with convergence points far from final state
Sawodny et al. (2010)	4th order polynomials	Maximum acceleration Maximum driving force Safety distance	Bicycle dynamic model with constant tyre model	Vehicles presumed with constant velocity
Ma et al. (2012)	Cubic Bezier curves	Trajectory length Smoothness Offset for centre line	Rectangle	Obstacles treated as circles Over-computations due to colliding trajectories calculation
Perez et al. (2013)	Cubic Bezier curves	Fuzzy rules using: Angular error Lateral error Offset from the curve	Not provided	No collision checking Lateral accelerations and curvature constraints omitted
Gonzalez et al. (2014)	4th degree Bezier curves	Road boundaries Static obstacles, Vehicle dynamics Lateral acceleration Curvature Maximum speed	Rectangle	Dynamic obstacles are not taken into consideration Low speeds assumed
Bae et al. (2013)	3rd and 5th degree Bezier Curves	Curvature Slip angle Steering angle Steering rate Mean square error Maximum error	Bicycle model	Tests with a miniature-sized vehicle Only lane-change manoeuvres considered
Wang et al. (2011)	Akima Splines	Distance/time Obstacles Traffic rules	Not provided	Extensive results of the performance are not provided
Gu and Dolan (2014)	Splines/learning trajectories from human driving	Curvature Speed Acceleration Deceleration	Not provided	Traffic-free environment without regulations Large amount of data needed for learning human driving patterns
Ziegler et al. (2014a)	Optimisation inside a driving corridor	Offset from reference path Speed Acceleration Curvature Tyre friction Desired behaviour	Not provided	Computational complexity Universal applicability
Madas et al. (2013)	MPC	Prediction and control horizon Side slip and steering angle Steering angle rate Lateral position	Bicycle model with linear tyre characteristics and yaw Lateral dynamics	Optimisation sensitivity to number of variables/constraints
Li et al. (2014)	MPC	Distance to obstacles Distance from reference path Smoothness and prediction distance Max speed and max lateral Longitudinal acceleration	2D position Heading Curvature	Optimisation sensitivity to number of variables/constraints Only static obstacles considered

and the best one was selected by optimising a function of distance to obstacles, distance from the reference path, trajectory smoothness and look-ahead distance. Apart from the sensitivity of MPC to the variables considered, the aforementioned work was tested only for static obstacle avoidance.

Trajectory planning approaches are compared in Table 5.

In summary, it can be concluded that in the lowest level of planning, given the path and the behaviour from the higher levels, the final trajectory is formulated at each time epoch. A geometric curve is then optimised with respect to real-time kinematic and motion model constraints as well as the position of obstacles on the road. The choice of curve is typically related to the smoothness and curvature of the path, the comfort it provides to the passengers, as well as the dynamic approach used to model the vehicle and the computational time.
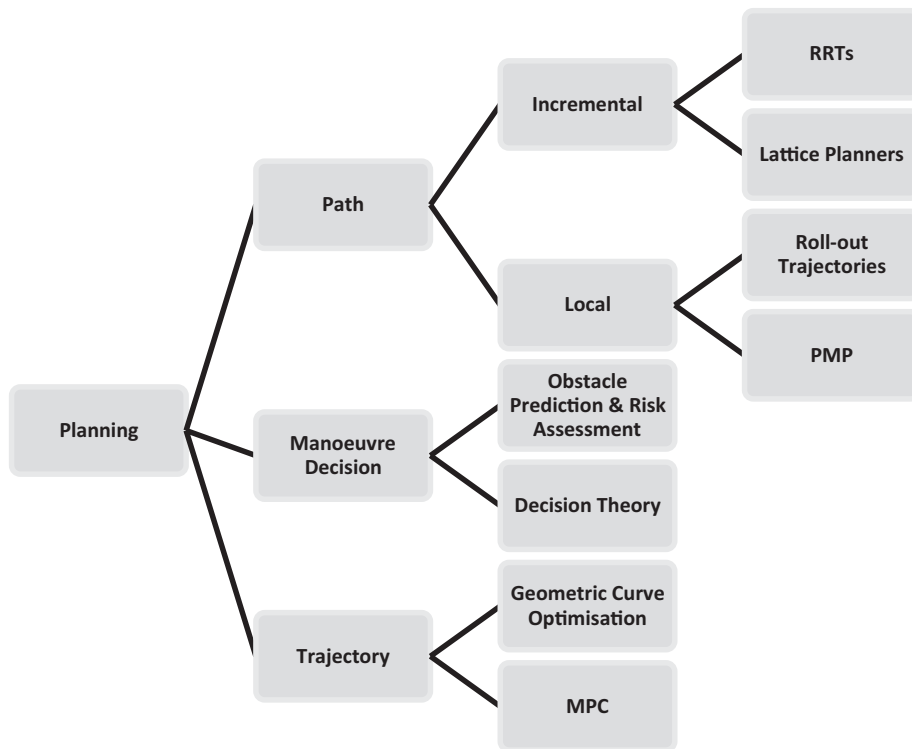
Fig. 10. Classification tree for planning approaches.

3.5. Final remarks on planning approaches

As discussed, planning for an autonomous vehicle can be divided into three main levels: search for the best path, search for the best manoeuvre and search for the best trajectory. Searching for the best path can be further divided into searching for the best series of paths towards the goal and searching within a limited 'local' time and space horizon. As far as manoeuvre planning is concerned, obstacle prediction and risk assessment are employed, while decision-theoretic approaches (such as Markov decision processes and game theory) have recently emerged to account for interactions within the traffic environment. Lastly, in trajectory planning, the chosen geometric path is bounded with kinematic and motion model constraints and further optimised to assure a smooth and feasible journey along it. This optimisation is based either on the choice of geometric curve to represent the path or on model predictive control. The classification tree (shown in Fig. 10) summarises the approaches and their sub-classes. It should be noted that these approaches are rarely treated independently in current research; instead they are typically combined in order to provide a complete plan for the vehicle.

4. Constraints and limitations

The approaches discussed in the previous sections have potential to work well in finding a path, choosing the best manoeuvre and constructing a feasible trajectory. However, limitations still exist and autonomous driving is yet to achieve the levels of human driving competence. We describe some of the identified limitations in this section.

4.1. Obstacle handling

In terms of handling obstacles, existing approaches primarily rely on predicting the trajectories of other traffic participants, either by taking their trajectories into account, or by making assumptions of constant velocities or constant accelerations (Kushleyev and Likhachev, 2009). This leads to a huge computational power requirement, since the obstacles' trajectories need to be calculated and checked at each moment. Such trajectory predictions are performed while disregarding the context within the traffic environment; thereby, leading to interactions between vehicles or other traffic participants being ignored. Some of the approaches (e.g. Aoude et al., 2010a,b; Bandyopadhyay et al., 2012; Martin, 2013) also assume that there is no uncertainty in the obstacles' motions; this assumption is not valid in real-world situations, especially in a mixed traffic scenario with the presence of human drivers. The lack of understanding between autonomous vehicles and

human drivers can be demonstrated by recent experiments from Google (2015). In these experiments, the Google autonomous car could not ascertain the intention of human drivers in its vicinity and this confusion resulted in minor crashes.

Another important limitation of the existing approaches is the simple representation of obstacles as rectangles or circles. In the latter case, the problem is that close proximity motions cannot be performed, due to lack of accuracy in the approximation (as shown in Ziegler et al., 2014a). Interaction-aware models, as presented in Lefèvre et al. (2014) can take interaction between traffic participants into account, but pre-suppose perfect knowledge or communication between the vehicles. Furthermore, motorcycles and non-motorised traffic participants are usually ignored in most approaches. Another major limitation in terms of obstacle handling is the inability to see around corners and detect obstacles such as pedestrians and bicycles approaching from blind corners. Such a disadvantage leads the planning algorithm to take a 'cautious' and hence inefficient approach, such as slowing down even in the absence of any obstacle.

4.2. Vehicle dynamics

Most of the existing planning approaches rely on the bicycle or car-like kinematic model for modelling the vehicle, but such a model cannot exploit the basic manoeuvre capabilities of the car, as it does not take into account, for example, tyre forces (Jeon et al., 2013). A few approaches (e.g. Jeon et al., 2013; Sawodny et al., 2010) have utilised the dynamic approach based on the bicycle model, and therefore take into consideration friction and the mass of the vehicle. However, a dynamic model which efficiently describes the motion and the distribution of forces of the vehicle in a real-world environment is yet to be implemented.

One of the big challenges is to capture and abstract a vehicle's capability and constraints and then incorporate this information in manoeuvre and trajectory planning. Examples include acceleration, braking or steering constraints and the influence of weather or road surface on these capabilities or constraints.

4.3. Risk indicators

In order to assess a collision risk in driving situations, the commonly employed metrics include: Time to Collision (TTC), Distance to Collision (DTC) or Time to React (TTR) as well as unexpected/conflicting behaviours (Lefèvre et al., 2014). The problem with using TTC or DTC is that TTC typically refers to car following situations only and, in most cases, a constant velocity or a constant acceleration is measured (Ward et al., 2015). Furthermore, in curved road segments, straight lines are mostly considered for the calculation of TTC/DTC/TTR, which may lead to the confusing of safe, with hazardous, situations (see Fig. 11). Gap acceptance models for each manoeuvre and generic context-specific risk estimation are not included in existing studies.

4.4. Sensing and perception

Sensing and perception within existing approaches treat the vehicle as an individual and isolated entity; limiting the perception horizon of autonomous vehicles to the perception horizon of its individual sensors. Furthermore, most approaches either assume perfect knowledge of the environment (e.g. Aoude et al., 2010b; Bandyopadhyay et al., 2012; Brechtel et al., 2014) or depend on expensive sensing (e.g. Ziegler et al., 2014b) to perceive near-perfect knowledge of the environment and the obstacles. Approaches fail to take into account the limited field of view that driverless vehicles have and possible blind-spots that may occur, for example, in curved road segments or blind and closed intersections (i.e. intersections with restricted views).
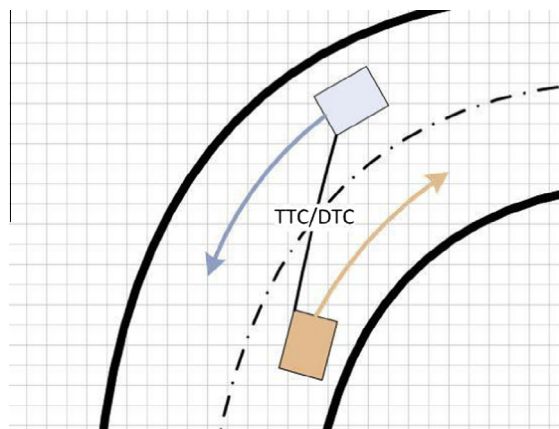
Fig. 11. TTC/DTC problem on curvy road segments.

4.5. Testing environment

The environment in which the testing of algorithms takes place can be divided into three domains: simulations; experiments with model vehicles; and experiments in real-world. The first two domains are used in most studies, since they are relatively easier to implement. On the contrary, the results from those tests, even with a high degree of accuracy, cannot ensure the same level of performance on real world vehicles, in real world environments. Additionally, testing with only two vehicles cannot resemble a real world traffic environment (Althoff et al., 2009). However, there have been cases of successful testing (which highlighted limitations of existing techniques) within real world environments (Bertozzi et al., 2010; Fischer, 2013; Ziegler et al., 2014b) and such cases need to be more widespread in order to advance the field to its final market-ready status. For more robust future algorithms, data collected by on-road experiments must be shared and mined to learn and move the entire area forward. Finally, modelling and simulation techniques, that can better model the physical and cyber components of not only the ego-vehicle but also the entire transportation system and the interactions within it, have been recognised as needed.

5. Future research directions

Based on the constraints and limitations discussed in the previous section, future research directions are discussed in this section. They are concerned with risk assessment, state space sampling, sensing, vehicle modelling, testing and dealing with dynamic environments.

5.1. Risk assessment

Many of the accidents taking place today are as a result of misplaced perception and decision making on the part of the human driver. Autonomous driving is envisaged to drastically reduce such mistakes since accurate risk assessment is vital for preventing collisions. Although current systems have been successfully applied to finding paths and detecting obstacles in real environments, collisions still occur. Hence, greater emphasis must be given to accurate risk assessment in real-time. The majority of existing risk assessment approaches predict trajectories and then detect possible collision. However, such an approach incurs huge computational cost generation and checking of all possible trajectories. Instead of exhaustively calculating and predicting the trajectories of other traffic participants at each epoch (sensing cycle), a useful proposition would be to perform the trajectory calculation and collision checking only if an unusual or dangerous manoeuvre is detected (Lefèvre et al., 2012a), or if traffic conditions are flagged as dangerous from a transport engineering perspective.

Interactions between agents should also be exploited, using appropriate mathematical foundations, such as Bayesian Game Theory (Harsanyi, 1967); such concepts have been successfully used in many agent-based problems within robotics (Antoniades et al., 2003; Emery-Montemerlo, 2005). Such approaches have the potential to assist real-time handling of interactions such as at intersections; merging into traffic, or overtaking with limited perception.

Furthermore, given that the ego vehicle will not only interact with other autonomous vehicles but also with human-driven vehicles, motorcyclists, cyclists and pedestrians, a planning algorithm shall explore the reasoning ability of human (as a driver/rider or a pedestrian at different traffic scenarios and operational environments) behavioural models and shall use the resulting intelligence in the forms of either trajectory predictions or inferring actions at a road-crossing of fellow road participants.

Existing approaches to the problem of planning originate from earlier developments within robotics which treat the vehicle as an individual isolated entity. In reality, an autonomous vehicle will be a participant of a wider (mixed) traffic system. Hence, future studies must explore the incorporation of traffic engineering concepts such as identification of crash precursors, gap acceptance and network-level crash prediction, thus bringing in the context of the surrounding traffic system complexity within risk assessment and crash prediction. Behavioural models can be provided for every manoeuvre that the vehicle is about to undertake (e.g. Hidas, 2002; Llorca et al., 2014; Saifuzzaman and Zheng, 2014) to flag the manoeuvre as nominal or not. Real-time highly disaggregated traffic flow data and historical crash data which can indicate hazardous situations can easily be implemented within the planning module of autonomous vehicles' software architecture, leading to potential improvement in decision making as an early indication of dangerous road segments. Real-time crash prediction is a problem that has been researched by Transport Engineers for many years; resulting in tried and tested methods which accurately indicate network-level crash risks (Fig. 12). Such models can be incorporated into autonomous vehicle planning, easing the computation and evaluation of hazardous situations for vehicles in real-time, and, at the same time, increasing the perception horizon of the ego-vehicle.

Trajectory planning will also benefit from the same spirit of decentralising approaches from vehicle-level only to network level. Instead of attempting to optimise and assess risks on a single, independent vehicle trajectory, the optimisation process could expand spatially. Within that concept, a bundle of trajectories from vehicles which share a common transport system (e.g. a road segment, a specific highway, a town or city) could be optimised.

As far as risk indicators are concerned, to overcome the limitation of TTC (as described in Section 4.3), the application of differential geometry could be explored. By applying differential geometry the kinematic properties of a particle moving on a curve can be studied. With that concept in mind, instead of calculating distances (for the TTC) using straight
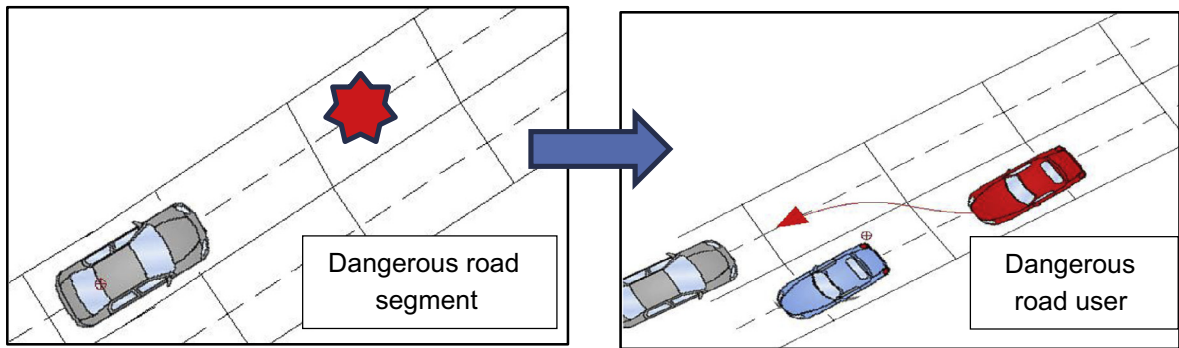
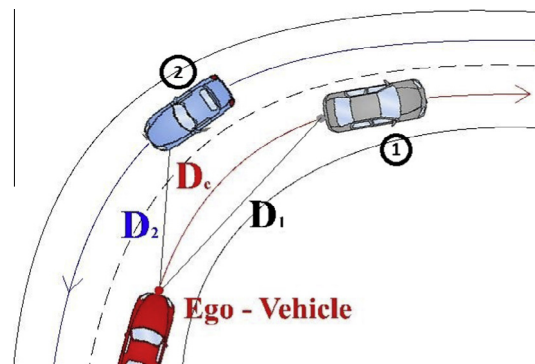**Fig. 12.** From network level risk to vehicle level risk.

Fig. 13. Using distances along a curve to calculate TTC.

lines in a global coordinate system, a Frenét frame[11] could be adopted (Zucker, 2006). With a Frenét frame distances can be calculated on the curve that describes the road geometry and thus the motion of the vehicle. The approach of calculating TTC using distances on a curve is shown in Fig. 13. Instead of using the straight-line distance (D_1) to vehicle 1, the distance D_c could be used to calculate TTC as it is more representative for the vehicle path. Moreover, car 2 may be considered to be a 'threat' to the ego-vehicle if the straight-line distance (D_2) is used. This is not the case if the curved distance along the roadway is employed.

Furthermore, approaches which combine risk assessment with space exploration can be efficiently used. Examples of such approaches are the Risk-RRT (Rios-Martinez et al., 2011) which has successfully been tested for mobile robots among pedestrians, and informed state lattices (Howard, 2009) which incorporate changes in vehicle mobility.

5.2. Search space reduction

Most of the existing techniques used in the local path search and the trajectory level apply the action space search at the lowest level of planning. Thus, a state space which guarantees safety can also be investigated further along with model-predictive-control related trajectory generations such as the one used in Howard (2009) for planetary rovers. Model predictive control can ensure that the necessary inputs for a terminal state are determined from the robotic car itself and can also provide feedback during control to cope with noise that may arise during planning. Furthermore, more complex manoeuvres can be embedded into planning and, thus, sampling the state space could become both quicker and more effective.

Furthermore, *a priori* search space constructed within the road or lane boundaries could enhance the efficiency of path searching techniques, by addressing exhaustive sampling and greedy searches of the state space. Exhaustive sampling uses too many computational resources and a greedy search may lead to paths which are not feasible (i.e. those that lie outside the road boundaries or intersect with obstacles), so a road- or lane-based search space could potentially solve these problems without a loss in computational power and having to discard paths online.

---

[11] In differential geometry a Frenét frame is a local coordinate system which describes the kinematic properties of a particle moving along a continuous differentiable curve.

5.3. Sensing and V2X communication

Sensors are an autonomous vehicle's *eyes* and *ears,* and hence every care must be taken to ensure that they 'see' correctly. This is either done through the use of highly accurate, albeit expensive, sensors or through employing enhanced sensor fusion techniques by integrating relatively inaccurate data gathered by off-the-shelf low-cost, though multiple, sensors. Whatever the approach is, on-board sensors are limited by their line-of-sight and range.

The necessity of having high-performance sensing around the day and night in all weather, while keeping the cost of the sensing system low, could lead to the adoption of cameras with thermal or acoustic sensors. A thermal camera, also known as an infrared camera because it operates in the infrared spectrum, can outperform as an usual camera in object detection and is less influenced from illumination conditions (Besbes et al., 2015). Thermal cameras have successfully been applied to object recognition (e.g. Besbes et al., 2015) and road detection (Peláez et al., 2015). By using thermal cameras, motorised (e.g. vehicles) and non-motorised objects (e.g. pedestrians, cyclists and animals) can easily be identified through their emitting thermal energy, independently of the lighting conditions.

Acoustic cameras on the other hand visualise the sources and intensity of sound in real-time (GFaI tech, 2014). These cameras can therefore prove to be a promising alternative to proximity sensing of all road users given that they emit sound. As a result, motorised vehicles can be detected from the sound of their engines in all lighting conditions even if they are not "visible" to the autonomous vehicle.

Therefore, the inclusion of thermal and acoustic cameras as sensing would provide extra level of integrity in the pursuit of collision-free path planning. It should however be noted that these cameras are sensitive to weather conditions as precipitation may alter the obstacles' thermal energy and sound detection leading to unreliable data (Eskandarian, 2012). It is therefore essential that their real-time application with respect to data reliability and integrity for planning should be further investigated and validated.

Vehicular Communication Systems established primarily through *ad hoc* means offer a promising assistance to the limitations of current sensing techniques through the increase in perception horizon and added intelligence. Vehicles will be able to communicate with the surrounding vehicles by means of Vehicle-to-Vehicle (V2V) communication, with surrounding infrastructure (e.g. traffic lights) by means of Vehicle-to-Infrastructure (V2I) communications and with devices carried by non-motorised traffic participants (i.e. pedestrians or cyclists) by means of Vehicle-to-Device Communications (V2D). V2V, V2I and V2D communications are jointly termed as V2X communications. Such *ad hoc* networks, termed Vehicular *ad hoc* Networks (VANETs) employ a number of networking technologies, such as Wi-Fi, WiMax, Bluetooth and, most specifically, Dedicated Short Range Communications (DSRC) (a type of Wi-Fi technology). DSRC technology has been primarily developed and promoted for the purpose of delay sensitive vehicular communications (such as safety messages) by the US Department of Transport (DoT). DSRC has an approximate range of 300 m and envisaged to improve the sensing range and look-ahead capability of current sensing technologies. Recent studies have shown that combining perceived intelligence and capabilities from surrounding including infrastructure could prove effective for accurate localisation (particularly in the absence of GPS such as in urban canyons), blind spot detection and looking around corners. It is envisaged that V2X communication will improve the detection of non-motorised traffic participants (e.g. pedestrians, cyclists, animals), especially when approaching from blind corners through the extension of the communication network that includes non-motorised traffic participants.

Apart from enhancing sensing capabilities of autonomous vehicles, vehicular communications could improve the performance of planning approaches through accurate longer term risk assessment (Lefèvre et al., 2012b). A significant part of risk assessment and planning is detecting obstacles followed by predicting the obstacles trajectory. Current techniques rely on on-board sensor perceptions and lack in longer term prediction ability due to inaccuracy of the perceived data. Future developments must look at acquiring the actual state of obstacles, actual manoeuvres and more accurate and longer term trajectory prediction in real-time from the surrounding obstacles or vehicles through vehicular communications.

In manoeuvre planning and decision making, vehicular communications could assist, for example, through speed adjustments as the vehicle approaches a signalised intersection, so that the vehicle arrives at the intersection during the duration of a green light. Doing so could save effort for decision making (to stop or accelerate) and also save fuel by reducing stop-and-go actions (Rakha and Kamalanathsharma, 2011; Ilgin Guler et al., 2014). In non-signalised intersections, manoeuvres of all participants should be known *a priori*, thus leading to less congestion and fewer collisions (Wu et al., 2013).

Communication among vehicles could also be used for information sharing about road conditions such as oil spillages, parked vehicles and temporary road blockages (Drobot et al., 2012; Marshall, 2014). As shown in Fig. 14, the first vehicle detecting a road anomaly or a slippery pavement can transmit this information to vehicles which follow, so as to provide in advance a better situational awareness for planning a safe path or trajectory along this road segment. An example of the application of V2X communications for perception and control of automated vehicles is presented in Fig. 14 (Kim et al., 2013).

In the case of using transport engineering techniques to estimate risk, vehicular communications could prove to be an effective medium to collect traffic flow data online and transmit them to either a traffic management agency or to other vehicles in the area.

While significant research has been conducting, much of the validations have been actually carried out within controlled simulated environments. There is now a need to increase the amount of the experiments in real environments taking into account occasional chaotic traffic that do not adhere to standard traffic models. Moreover, upgrading infrastructures (for
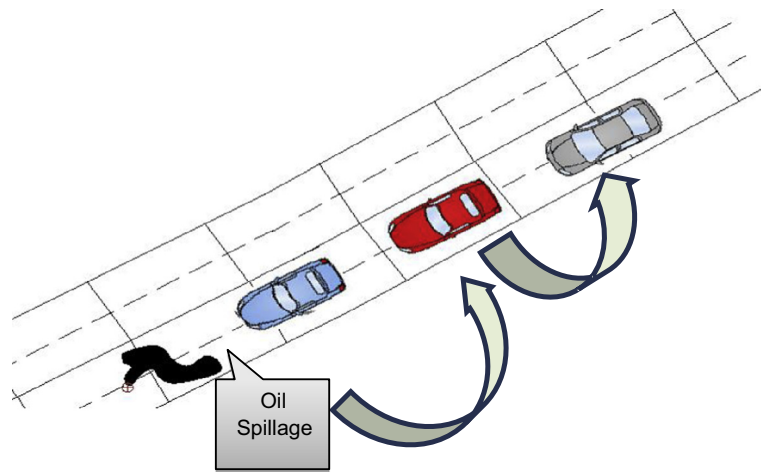
Fig. 14. Transmission of road condition information with vehicular communications.

example to adopt V2I communications) is expensive and there is a need to design planning techniques relying on infrastructure-less vehicular communication that allows seamless movement at intersections. Nevertheless, studies need to be conducted to assess the real-time implementation of vehicular communication and its effect on planning. Disturbances in communication, real-time constraints, speed of communication and detection performance are only a few aspects of vehicular communication that need to be checked prior to their incorporation into the planning module of autonomous vehicles and factored into the methods relying on them. Integration of the information from V2X and on-board sensors to provide better information would be a research challenge. With the whole spread of V2X technology however, there is significant potential in improving coordination between vehicles and their planning modules.

5.4. Vehicle model

The bicycle model has been proven to be an effective, although simplistic, model for car-like vehicles. Researchers concentrate on the kinematic nature of a vehicle's motion but dynamics should also be taken into account. Modelling the vehicle dynamically can resemble the real-world capabilities of the vehicle from the planning phase, thus making it easier for the trajectory module to track and follow the path or the trajectory that is designated for the vehicle. Models which also take into account the forces of the vehicle's engine, the interaction between the tyres and the pavement or the turning capabilities of a real-car could add a slight burden in computation (due to the dimensions added in the vehicle model) but might enhance the overall on-road performance for an autonomous vehicle. This becomes more important for manoeuvres, particularly for those which take place within a confined space or include close interactions between vehicles. It can be shown that in manoeuvres such as collision avoidance, the dynamics of the vehicle and other capabilities have to be taken into account.

5.5. Testing and validation

As discussed earlier, tests on autonomous driving technologies need to aim at real-world experiments after simulation and test in controlled experiments. As field experiments might be able to prove that autonomous vehicles could be a safe and reliable transportation mode (Dokic et al., 2015), more testing is needed. First of all, autonomous driving should be considered for all weather and visibility conditions. Driving in rain, snow or fog is investigated in crash prediction (Yu et al., 2015) but has yet to be included in planning approaches for autonomous vehicles, apart from at the routeing level (e.g. Maddern et al., 2014). Highways, intersections, roundabouts and merging scenarios have been tested so far, but flyovers, give-way scenarios and diverging sections could improve the applicability of planning approaches and autonomous driving. Lastly, the infrastructure readiness level should also be researched, as well as dealing with emergency situations, in order to cover the whole spectrum of driving situations.

5.6. Dealing with dynamic and uncertain environment

Autonomous vehicles operate in a dynamic and uncertain transport environment. This is not only due to the changes of the environment such as road signals, and the appearance and disappearance of other road users, but also because the vehicle continuously moves from one place to another, at potentially high speed. Situational awareness and assessment is essential in providing necessary information for planning algorithms of the vehicles. Human drivers maintain safe separation from other road users by not only comprehending their intention and current situation but also anticipating where they might

move, along with their possible position in the immediate future. Good progress has been achieved in understanding static environment and understanding the position and status of the ego-vehicle, with respect to the local environment such as SLAM. More research is required in tracking and anticipating moving objects such as pedestrians, bikes and other vehicles. Multiple object tracking will play a significant role in improving situation awareness (Ding and Chen, 2013; Ding et al., 2015). Furthermore, the background/context knowledge (for example road conditions, traffic regulations, and local maps/ facilities) can be used to better understand the intention of other users and, therefore, reduce uncertainty in anticipation of their future moves (Ding et al., 2015; Liu et al., 2013).

Receding horizon concept or model predictive control provides one of the most promising ways to deal with the dynamic and uncertain nature of vehicle planning. In this approach, planning is made by looking ahead for a fixed horizon based on all the available information (Liu et al., 2010a,b) but only the first small part of the planning is implemented. The planning is repeated by looking ahead again for the horizon after new information has arrived. There are several features that make this approach very attractive. First, all the on-board sensors have a limited range and information beyond the range known (Liu and Chen, 2013). Secondly, even within the horizon, due to the dynamic and uncertain nature of the transport environment, re-planning is still necessary to cope with the changes; for example, the autonomous vehicle will respond to the change of a traffic light by re-planning its actions when it is approaching. Thirdly, the use of a limited horizon can reduce the computational burden in planning, which makes it possible to implement some computationally extensive planning algorithms. Despite all the initial works in this area (e.g. Werling and Liccardo, 2012; Kim, 2013; Nilsson et al., 2013), much more research is required to fully explore the potential of the receding horizon concept.

6. Conclusions

The planning module of an autonomous or self-driving vehicle should ensure safety and comfort for the passengers. It should also put the vehicle in the right behaviour with respect to the kinematic and motion model constraints surrounding the vehicle. This paper critically reviewed existing planning approaches applied to autonomous on-road driving after the milestone of the DARPA Urban Challenge, and highlighted the most frequently applied techniques. Focus was given to incremental and local path search, as well as behaviour and trajectory planning, since global routing between an O–D pair has been discussed in the literature. The paper identified that incremental path planning relies on searching data structures such as trees or lattices, while local usually takes place in a continuous space with sampling from the final states. Manoeuvre planning, in most cases, consists of obstacle prediction followed by collision estimation; while context is absent from most of the approaches. Lastly, in the trajectory level, most approaches try to optimise a given geometric curve or deform it with regard to static or dynamic obstacles.

Since decision making and the handling of dynamic obstacles were found to be the main areas of concern, the implementation of agent-based mathematical formulations and the incorporation of transport engineering aspects have been proposed. In addition, a recommendation was made for the use of alternative ways of sensing, such as vehicular communications, which would enhance the field of view of vehicles and improve both their estimating capabilities and localisation performance. Finally, directions for the incorporation of dynamic models into planning were made to improve the real-world performance of current approaches.

It is essential to note that the proposed enhancements should gradually be incorporated into autonomous driving after excessive testing in simulations, model car environments, test tracks and, finally, real-world experiments.

Acknowledgement

References

Alin, A., Butz, M.V., Fritsch, J., 2012. Incorporating environmental knowledge into Bayesian filtering using attractor functions. In: IEEE Intelligent Vehicles Symposium, pp. 476–481.

Althoff, D., Wollherr, D., Buss, M., 2011. Safety assessment of trajectories for navigation in uncertain and dynamic environments. In: 2011 IEEE International Conference on Robotics and Automation, pp. 5407–5412.

Althoff, M., Stursberg, O., Buss, M., 2009. Model-based probabilistic collision detection in autonomous driving. IEEE Trans. Intell. Transport. Syst. 10, 299–310.

Antoniades, A., Kim, H.J., Sastry, S., 2003. Pursuit-evasion strategies for teams of multiple agents with incomplete information. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), vo. 1, pp. 756–761.

Aoude, G., Luders, B.D., Joseph, J.M., Roy, N., How, J.P., 2013. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. Auton. Robot. 35, 51–76.

Aoude, G., Luders, B.D., Levine, D.S., How, J.P., 2010. Threat-aware path planning in uncertain urban environments. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 6058–6063.

Aoude, G.S., Luders, B.D., How, J.P., Pilutti, T.E., 2010. Sampling-based threat assessment algorithms for intersection collisions involving errant drivers. In: IIFAC Symposium on Intelligent Autonomous Vehicles.

Armand, A., Filliat, D., Ibanez-Guzman, J., 2014. Ontology-based context awareness for driving assistance systems. In: 2014 IEEE Intelligent Vehicles Symposium, pp. 227–233.

Bacha, A., Bauman, C., Faruque, R., Fleming, M., Terwelp, C., Hong, D., Wicks, A., Alberi, T., Anderson, D., Cacciola, S., Currier, P., Dalton, A., Farmer, J., Hurdus, J., Kimmel, S., King, P., Taylor, A., Covern, D., Van Webster, M., 2008. Odin: Team VictorTango's Entry in the DARPA Urban. J. Field Robot. 25, 467–492.
Bae, I., Moon, J., Park, H., Kim, J.H., Kim, S., 2013. Path generation and tracking based on a Bezier curve for a steering rate controller of autonomous vehicles. In: IEEE Conference on Intelligent Transportation Systems, ITSC, pp. 436–441.
Bandyopadhyay, T., Won, K.S., Frazzoli, E., Hsu, D., 2012. Intention-aware motion planning. In: Proc. of the 10th Workshop on the Algorithmic Foundations of Robotics, pp. 475–491.
Benenson, R., Petti, S., Fraichard, T., Parent, M., 2006. Integrating perception and planning for autonomous navigation of urban vehicles. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 98–104.
Bertozzi, M., Bombini, L., Broggi, A., 2010. The VisLab intercontinental autonomous challenge: 13,000 km, 3 months, no driver. In: 17th World Congress on ITS. Busan, South Korea, pp. 1–11.
Besbes, B., Rogozan, A., Rus, A.-M., Bensrhair, A., Broggi, A., 2015. Pedestrian detection in far-infrared daytime images using a hierarchical codebook of SURF. Sensors 15, 8570–8594.
Bohren, J., Foote, T., Keller, J., Kushleyev, A., Lee, D., Stewart, A., Vernaza, P., Satterfield, B., 2008. Little ben: the ben franklin racing team's entry in the 2007 DARPA urban challenge. J. Field Robot. 25, 598–614.
Brechtel, S., Gindele, T., Dillmann, R., 2014. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 392–399.
Broggi, A., Medici, P., Zani, P., Coati, A., Panciroli, M., 2012. Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. Ann. Rev. Contr. 36, 161–171.
Burns, L.D., 2013. A vision of our transport future. Nature 497, 181–182.
Delsart, V., Fraichard, T., Martinez, L., 2009. Real-time trajectory generation for car-like vehicles navigating dynamic environments. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3401–3406.
Ding, R., Chen, W.-H., 2013. Moving object tracking in support of unmanned vehicle operation. In: 2013 19th International Conference on Automation and Computing (ICAC), pp. 1–6.
Ding, R., Yu, M., Chen, W., 2015. A multiple target tracking strategy using moving horizon estimation approach aided by road constraint. In: 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV), pp. 1–13.
Dokic, J., Muller, B., Meyer, G., 2015. European Roadmap Smart Systems for Automated Driving, European Technology Platform on Smart Systems Integration.
Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J., 2010a. Path planning for autonomous vehicles in unknown semi-structured environments. Int. J. Robot. Res. 29, 485–501.
Drobot, S., Chapman, M., Pisano, P., Guevara, G., 2012. Connected vehicle data promises advanced weather warning [WWW Document]. ITS International. <http://www.itsinternational.com/categories/networking-communication-systems/features/connected-vehicle-data-promises-advanced-weather-warning/> (accessed 26.02.15).
Emery-Montemerlo, R., 2005. Game Theoretic Control for Robot Teams. PhD Thesis, The Robotics Institute, Carnegie Mellon University.
Eskandarian, A., 2012. Handbook of Intelligent Vehicles. Springer, London, London.
Ferguson, D., Likhachev, M., 2008. Efficiently using cost maps for planning complex maneuvers. In: International Conference on Robotics and Automation Workshop on Planning with Cost Maps.
Fernandez, C., Dominguez, R., Fernandez-Llorca, D., Alonso, J., Sotelo, M.A., 2013. Autonomous navigation and obstacle avoidance of a micro-bus. Int. J. Adv. Rob. Syst. 10, 1–9.
Fischer, A., 2013. Inside Google's Quest To Popularize Self-Driving Cars | Popular Science [WWW Document]. Popular Science. <http://www.popsci.com/cars/article/2013-09/google-self-driving-car> (accessed 09.01.14).
Fletcher, L., Teller, S., Olson, E., Moore, D., Kuwata, Y., Leonard, J., Miller, I., Campbell, M., Nathan, A., Kline, F., 2008. The MIT – Cornell collision and why it happened. J. Field Robot. 25, 775–807.
Furda, A., Vlacic, L., 2011. Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making. IEEE Intell. Transp. Syst. Mag. 3, 4–17.
Gadepally, V.N., 2013. Estimation of Driver Behavior for Autonomous Vehicle Applications., PhD Thesis, Ohio State University.
Garrote, L., Premebida, C., Silva, M., Nunes, U., 2014. An RRT-based navigation approach for mobile robots and automated vehicles. In: 2014 IEEE International Conference on Industrial Informatics (INDIN), pp. 326–331.
Gindele, T., Brechtel, S., Dillmann, R., 2015. Learning driver behavior models from traffic observations for decision making and planning. IEEE Intell. Transport. Syst. Mag. 7, 69–79.
GmbH GFaI tech, 2014. Acoustic Camera. Listening with your eyes. [WWW Document]. <http://www.acoustic-camera.com/>.
Gonzalez, D., Perez, J., Lattarulo, R., Milanes, V., Nashashibi, F., 2014. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1430–1435.
Google, 2015. Google Self-Driving Car Project Monthly Report (May 2015).
Gu, T., Dolan, J.M., 2012. On-road motion planning for autonomous vehicles. In: Proceedings of the 5th International Conference on Intelligent Robotics and Applications (ICIRA 2012), pp. 588–597.
Gu, T., Dolan, J.M., 2014. Toward human-like motion planning in urban environments. In: 2014 IEEE Intelligent Vehicles Symposium (IV), pp. 350–355. d.
Gu, T., Snider, J., Dolan, J.M., Lee, J., 2013. Focused Trajectory Planning for autonomous on-road driving. In: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 547–552.
Hardy, J., Campbell, M., 2013. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. IEEE Trans. Rob. 29, 913–929.
Harsanyi, J.C., 1967. Games with incomplete information played by "Bayesian" Players, I-III Part I. The Basic Model. Manage. Sci. 14, 159–182.
Hidas, P., 2002. Modelling lane changing and merging in microscopic traffic simulation. Transport. Res. Part C: Emerg. Technol. 10, 351–371.
Himmelsbach, M., Von Hundelshausen, F., Luttel, T., Manz, M., Muller, A., Schneider, S., Wunsche, H.-J., 2009. Team MuCAR-3 at C-ELROB 2009. In: Proceedings of 1st Workshop on Field Robotics, Civilian European Land Robot Trial. University of Oulu, pp. 951–978.
Howard, T.M., 2009. Adaptive Model-Predictive Motion Planning for Navigation in Complex Environments. PhD Thesis. Carnegie Mellon University.
Hundelshausen, F. Von, Himmelsbach, M., Hecker, F., Mueller, A., Wuensche, H., 2008. Driving with Tentacles: integral structures for sensing. J. Field Robot. 25, 640–673.
Ilgin Guler, S., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transport. Res. Part C: Emerg. Technol. 46, 121–131.
Jeon, J., Cowlagi, R.V., Peters, S.C., Karaman, S., Frazzoli, E., Tsiotras, P., Iagnemma, K., 2013. Optimal motion planning with the half-car dynamical model for autonomous high-speed driving. In: American Control Conference, pp. 188–193.
Kala, R., Warwick, K., 2013. Motion planning of autonomous vehicles in a non-autonomous vehicle environment without speed lanes. Eng. Appl. Artif. Intell. 26, 1588–1601.
Kammel, S., Ziegler, J., Pitzer, B., Gindele, T., Jagzent, D., Schr, J., Hundelshausen, F. Von, 2008. Team AnnieWAY ' s Autonomous System for the 2007 DARPA Urban Challenge. J. Field Robot. 25, 615–639.
Karaman, S., Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. Int. J. Robot. Res. 30, 846–894.
Kim, K.D., 2013. Collision free autonomous ground traffic: a model predictive control approach. In: 2013 ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2013, pp. 51–60.
Kim, S.W., Chong, Z.J., Qin, B., Shen, X., Cheng, Z., Liu, W., Ang, M.H., 2013. Cooperative perception for autonomous vehicle control on the road: motivation and experimental results. In: IEEE International Conference on Intelligent Robots and Systems, pp. 5059–5066.

Kolski, S., Ferguson, D., Bellino, M., Siegwart, R., 2006. Autonomous driving in structured and unstructured environments. In: 2006 IEEE Intelligent Vehicles Symposium, pp. 558–563.

Kripke, S., 1963. Semantical considerations on modal logic. Acta Philos. Fennica 16, 83–94.

Kushleyev, A., Likhachev, M., 2009. Time-bounded lattice for efficient planning in dynamic environments. In: 2009 IEEE International Conference on Robotics and Automation, pp. 1662–1668.

Kuwata, Y., Fiore, G.A., Teo, J., Frazzoli, E., How, J.P., 2008. Motion planning for urban driving using RRT. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1681–1686.

Kuwata, Y., Teo, J., Member, S., Fiore, G., Karaman, S., Frazzoli, E., Member, S., How, J.P., 2009. Real-time motion planning with applications to autonomous urban driving. IEEE Trans. Control Syst. Technol., 1105–1118

Laumond, J., 1998. Robot Motion Planning and Control, Lectures Notes in Control and Information Sciences. Springer-Verlag, Berlin Heidelberg.

LaValle, S.M., 1998. Rapidly-Exploring Random Trees: A New Tool for Path Planning, TR 98-11. Computer Science Dept., Iowa State University.

LaValle, S.M., 2006. Planning Algorithms. Cambridge University Press, Cambridge.

Le Vine, S., Zolfaghari, A., Polak, J., 2015. Autonomous cars: the tension between occupant experience and intersection capacity. Transport. Res. Part C: Emerg. Technol. 52, 1–14.

Lee, U., Vasseur, P., 2014. Local path planning in a complex environment for self-driving car. In: 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, pp. 445–450.

Lefèvre, S., 2012. Risk Estimation at Road Intersections for Connected Vehicle Safety Applications. PhD Thesis, University of Grenoble.

Lefèvre, S., Laugier, C., Ibañez-Guzmán, J., 2012a. Risk assessment at road intersections: comparing intention and expectation. In: IEEE Intelligent Vehicles Symposium, Proceedings.

Lefèvre, S., Laugier, C., Ibañez-guzmán, J., 2012b. Evaluating risk at road intersections by detecting conflicting intentions. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4841–4846.

Lefèvre, S., Vasquez, D., Laugier, C., 2014. A survey on motion prediction and risk assessment for intelligent vehicles. ROBOMECH J. 1, 1–14.

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Maheloni, K., Moyer, K., College, F.W.O., Jones, T., Buckley, R., Galejs, R., Krishnamurthy, S., 2008. A perception-driven autonomous urban vehicle. J. Field Robot. 25, 727–774.

Li, X., Sun, Z., Zhu, Q., Liu, D., 2014. A unified approach to local trajectory planning and control for autonomous driving along a reference path. In: International Conference on Mechatronics and Automations, pp. 1716–1721.

Li, Z., Chitturi, M., Noyce, D., Ran, B., Chitturi, M.V., Noyce, D., 2013. Development of Next Generation Intersection Control, Technical Report, CFIRE 04–18, National Center for Freight & Infrastructure Research & Education Department of Civil and Environmental Engineering College of Engineering University of Wisconsin-Madison.

Likhachev, M., Ferguson, D., 2009. Planning long dynamically feasible maneuvers for autonomous vehicles. Int. J. Robot. Res. 28, 933–945.

Liu, C., Chen, W.H., 2013. Hierarchical path planning and flight control of small autonomous helicopters using MPC techniques. In: IEEE Intelligent Vehicles Symposium, Proceedings. Gold Coast, Australia, 26–28th June, pp. 417–422.

Liu, C., Chen, W.H., Andrews, J., 2010a. Experimental tests of autonomous ground vehicles with preview. Int. J. Autom. Comput. 7, 342–348.

Liu, C., Chen, W.H., Andrews, J., 2010b. Optimisation based control framework for autonomous vehicles: algorithm and experiment. In: 2010 IEEE International Conference on Mechatronics and Automation, ICMA 2010, pp. 1030–1035.

Liu, C., Li, B., Chen, W.-H., 2013. Road network based vehicle navigation using an improved IMM particle filter. In: Proceedings of 2013 IFAC Intelligent Autonomous Vehicles Symposium, pp. 193–198.

Llorca, C., Moreno, A.T., Lenorzer, A., Casas, J., Garcia, A., 2014. Development of a new microscopic passing maneuver model for two-lane rural roads. Transport. Res. Part C: Emerg. Technol. 52, 157–172.

Ma, L., Yang, J., Zhang, M., 2012. A two-level path planning method for on-road autonomous driving. In: 2012 2nd International Conference on Intelligent System Design and Engineering Application, pp. 661–664.

Macek, K., Becker, M., Siegwart, R., 2006. Motion planning for car-like vehicles in dynamic urban scenarios. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4375–4380.

Madas, D., Nosratinia, M., Keshavarz, M., Sundstrom, P., Philippsen, R., Eidehall, A., Dahl, K., 2013. On path planning methods for automotive collision avoidance. In: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 931–937.

Maddern, W., Stewart, A.D., Newman, P., 2014. LAPS-II: 6-DoF day and night visual localisation with prior 3D structure for autonomous road vehicles. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 330–337.

Marshall, Andrew, 2014. Opel and DRIVE C2X Vehicle Communication Improves European Road Safety [WWW Document]. Adam Opel AG. <http://media.opel.com/media/intl/en/opel/news.detail.html/content/Pages/news/intl/en/2014/opel/07-16-opel-and-drive-c2x.html> (accessed 26.02.15).

Martin, A., 2013. Interactive Motion Prediction using Game Theory. Master Thesis, University of Padua.

McNaughton, M., Urmson, C., Dolan, J.M., Lee, J.-W., 2011. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: 2011 IEEE International Conference on Robotics and Automation, pp. 4889–4895.

Montemerlo, M., Becker, J., Dahlkamp, H., Dolgov, D., Hoffmann, G., Johnston, D., Klumpp, S., Langer, D., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., 2008. Junior: the stanford entry in the urban challenge. J. Field Robot. 25, 569–597.

Murphy, L., Newman, P., 2011. Risky planning: path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3727–3732.

Nilsson, J., Ali, M., Falcone, P., Sjoberg, J., 2013. Predictive manoeuvre generation for automated driving. In: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, pp. 418–423.

Ong, S.C.W., Png, Shao Wei, Hsu, D., Lee, Wee Sun, 2010. Planning under uncertainty for robotic tasks with mixed observability. Int. J. Robot. Res. 29, 1053–1068.

Peláez, G.A., Bacara, D., Escalera, A. De, García, F., 2015. Road detection with thermal cameras through 3D information. In: 2015 IEEE Intelligent Vehicles Symposium (IV), pp. 255–260.

Perez, J., Godoy, J., Villagra, J., Onieva, E., 2013. Trajectory generator for autonomous vehicles in urban environments. In: Proceedings – IEEE International Conference on Robotics and Automation, pp. 409–414.

Peters, S.C., Frazzoli, E., Iagnemma, K., 2011. Differential flatness of a front-steered vehicle with tire force control. In: IEEE International Conference on Intelligent Robots and Systems 02139, pp. 298–304.

Pivtoraiko, M., Kelly, A., 2005. Efficient constrained path planning via search in State Lattices. In: International Symposium on Artificial Intelligence, Robotics, and Automation in Space, pp. 1–7.

Pivtoraiko, M., Kelly, A., 2009. Fast and feasible deliberative motion planner for dynamic environments. In: Proceedings of the 2009 ICRA Workshop on Safe Navigation in Open and Dynamic Environments.

Pivtoraiko, M., Knepper, R.A., Kelly, A., 2009. Differentially constrained mobile robot motion planning in state spaces. J. Field Robot. 26, 308–333.

Rakha, H., Kamalanathsharma, R.K., 2011. Eco-driving at signalized intersections using V2I communication. In: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, pp. 341–346.

Rauskolb, F.W., Berger, K., Lipski, C., Magnor, M., Cornelsen, K., Effertz, J., Form, T., Graefe, F., Ohl, S., Doering, M., Homeier, K., Morgenroth, J., Wolf, L., Basarke, C., Berger, C., 2008. Caroline: an autonomously driving vehicle for urban environments. J. Field Robot. 25, 674–724.

Reyes Castro, L.I., Chaudhari, P., Tumova, J., Karaman, S., Frazzoli, E., Rus, D., 2013. Incremental sampling-based algorithm for minimum-violation motion planning. In: 52nd IEEE Conference on Decision and Control, pp. 3217–3224.

Rios-martinez, J., Spalanzani, A., Laugier, C., 2011. Understanding human interaction for probabilistic autonomous navigation using Risk-RRT approach. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2014–2019.

Rufli, M., Siegwart, R., 2010. On the design of deformable input- / state-lattice graphs. In: 2010 IEEE International Conference on Robotics and Automation, pp. 3071–3077.

Saifuzzaman, M., Zheng, Z., 2014. Incorporating human-factors in car-following models: a review of recent developments and research needs. Transport. Res. Part C: Emerg. Technol. 48, 379–403.

Sawodny, O., Zimmermann, J., Lutz, A., 2010. Motion planning for an autonomous vehicle driving on motorways by using flatness properties. In: 2010 IEEE International Conference on Control Applications, pp. 908–913.

Scheuer, A., Fraichard, T., 1997. Collision-free and continuous-curvature path planning for car-like robots. In: 1997 IEEE International Conference on Robotics and Automation, pp. 867–873.

Schröder, J., Gindele, T., Jagszent, D., Dillmann, R., 2008. Path planning for cognitive vehicles using risk maps. IEEE Intell. Vehicles Sympos. 2008, 1119–1124.

Schuster, W., 2015. Trajectory prediction for future air traffic management – complex manoeuvres and taxiing. Aeronaut. J. 119, 121–143.

Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D., 2011. Introduction to Autonomous Mobile Robots, second ed. MIT Press.

Sun, H., Deng, W., Zhang, S., Wang, S., 2014. Trajectory planning for vehicle autonomous driving with uncertainties. In: 2014 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), pp. 34–38.

Takahashi, O., Schilling, R.J., 1989. Motion planning in a plane using generalized Voronoi Diagrams. IEEE Trans. Robot. Autom. 5, 143–150.

Thorpe, C., Durrant-Whyte, H., 2009. The DARPA Urban Challenge, Springer Tracts in Advanced Robotics. Springer, Berlin Heidelberg.

Thrun, S., 2010. Toward Robotic Cars. Commun. ACM 53, 99–106.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffman, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P., 2006. Stanley: the robot that won the DARPA Grand Challenge. J. Field Robot. 23, 661–692.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M.N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T.M., Kolski, S., Kelly, A., Likhachev, M., Mcnaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y., Singh, S., Snider, J., Stentz, A., Whittaker, W.R., Wolkowicki, C., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., 2008. Autonomous driving in urban environments: boss and the urban challenge. J. Field Robot. 25, 425–466.

Varaiya, P., 1993. Smart cars on smart roads. Problems of control. IEEE Trans. Autom. Control 38, 195–207.

Wang, M., Ganjineh, T., Rojas, R., 2011. Action annotated trajectory generation for autonomous maneuvers on structured road networks. In: The 5th International Conference on Automation, Robotics and Applications, pp. 67–72.

Ward, J., Agamennoni, G., Worrall, S., Nebot, E., 2014. Vehicle collision probability calculation for general traffic scenarios under uncertainty. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 986–992.

Ward, J.R., Agamennoni, G., Worrall, S., Bender, A., Nebot, E., 2015. Extending Time to Collision for probabilistic reasoning in general traffic scenarios. Transport. Res. Part C: Emerg. Technol. 51, 66–82.

Wei, J., Snider, J.M., Gu, T., Dolan, J.M., Litkouhi, B., 2014. A behavioral planning framework for autonomous driving. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 458–464.

Werling, M., Kammel, S., Ziegler, J., Groll, L., 2011. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. Int. J. Robot. Res. 31, 346–359.

Werling, M., Liccardo, D., 2012. Automatic collision avoidance using model-predictive online optimization. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp. 6309–6314.

White, C.C., White, D.J., 1989. Markov decision processes. Eur. J. Oper. Res. 39, 1–16.

Wille, J.M., Form, T., 2008. Realizing complex autonomous driving maneuvers: the approach taken by team CarOLO at the DARPA urban challenge. In: 2008 IEEE International Conference on Vehicular Electronics and Safety, pp. 232–236.

Wille, J.M., Saust, F., Maurer, M., 2010a. Comprehensive treated sections in a trajectory planner for realizing autonomous driving in Braunschweig's urban traffic. In: 13th International IEEE Conference on Intelligent Transportation Systems, pp. 647–652.

Wille, J.M., Saust, F., Maurer, M., 2010b. Stadtpilot: driving autonomously on Braunschweig's inner ring road. In: 2010 IEEE Intelligent Vehicles Symposium, pp. 506–511.

Wu, Z., Yang, J., Huang, L., 2013. Study on the collision avoidance strategy at unsignalized intersection based on PreScan simulation. Procedia – Soc. Behav. Sci. 96, 1315–1321.

Xiu, C., Chen, H., 2010. A behavior-based path planning for autonomous vehicle. In: Liu, H., Ding, H., Xiong, Z., Zhu, X. (Eds.), Intelligent Robotics and Applications. Springer-Verlag, Berlin Heidelberg, pp. 1–9.

Xu, W., Pan, J., Wei, J., Dolan, J.M., 2014. Motion planning under uncertainty for on-road autonomous driving. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2061–2067.

Xu, W., Wei, J., Dolan, J.M., Zhao, H., Zha, H., 2012. A real-time motion planner with trajectory optimization for autonomous vehicles. In: 2012 IEEE International Conference on Robotics and Automation, pp. 2061–2067.

Yu, R., Xiong, Y., Abdel-Aty, M., 2015. A correlated random parameter approach to investigate the effects of weather conditions on crash risk for a mountainous freeway. Transport. Res. Part C: Emerg. Technol. 50, 68–77.

Zhang, S., Deng, W., Zhao, Q., Sun, H., Litkouhi, B., 2013. Dynamic trajectory planning for vehicle autonomous driving. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4161–4166.

Zhao, P., Chen, J., Mei, T., Liang, H., 2011. Dynamic motion planning for autonomous vehicle in unknown environments. In: 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 284–289.

Ziegler, J., Bender, P., Dang, T., Stiller, C., Preliminaries, A., 2014a. Trajectory planning for BERTHA – a local, continuous method. In: 2014 IEEE Intelligent Vehicles Symposium (IV), pp. 450–457.

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C.G., Kaus, E., Herrtwich, R.G., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., Zeeb, E., 2014b. Making bertha drive-an autonomous journey on a historic route. IEEE Intell. Transp. Syst. Mag. 6, 8–20.

Ziegler, J., Stiller, C., 2009. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1879–1884.

Zucker, S., 2006. Differential geometry from the Frenet point of view: boundary detection, stereo, texture and color. In: Paragios, N., Chen, Y., Faugeras, O. (Eds.), Handbook of Mathematical Models in Computer Vision. Springer, US, pp. 357–373.