CHAPTER 8

Automated Driving

Jorge Villagra[1], Leopoldo Acosta[2], Antonio Artuñedo[1], Rosa Blanco[3], Miguel Clavijo[4], Carlos Fernández[5], Jorge Godoy[1], Rodolfo Haber[1], Felipe Jiménez[4], Carlos Martínez[4], José E. Naranjo[4], Pedro J. Navarro[5], Ana Paúl[3] and Francisco Sánchez[3]

[1]CSIC, Madrid, Spain
[2]Universidad de La Laguna, Santa Cruz de Tenerife, Spain
[3]CTAG - Centro Tecnológico de Automoción de Galicia, Porriño, Spain
[4]Universidad Politécnica de Madrid, Madrid, Spain
[5]Universidad Politécnica de Cartagena, Cartagena, Spain

Contents

8.1 FUNDAMENTALS

Following the Webster definition, an autonomous vehicle "has the right or power of self-government," "exists or act separately from other things

Intelligent Vehicles
DOI: http://dx.doi.org/10.1016/B978-0-12-812800-8.00008-4

or people," "is undertaken or carried on without outside control," and "responds, reacts, or develops independently of the whole."

If we take these words at face value, it is therefore clear that there is a common mistake confusing this definition with the kind of systems currently appearing on mass media, where the driver has very little intervention. As a matter of fact, it is doubtful that autonomous driving, as it is defined above, would bring the commonly accepted benefits (capacity, efficiency, cleanness). New mobility paradigms, where autonomous on-demand vehicles are at the heart, would rather need to be connected and automated vehicles. Only in this context would cars be able to "drive at close range and increase the infrastructure's capacity, prevent a big deal of accidents by communicating with other vehicles and with the infrastructure, save or eliminate parking space by driving one passenger after other" (Holguín, 2016). But what are the differences then between autonomous and automated and connected vehicles?.

Automated vehicles can be defined as those in which at least some safety-critical aspects occur without direct driver input. Or in other words, an automated vehicle is one that can, at least partly, perform a driving task independently of a human driver. When these vehicles, with different levels of automation can communicate among them and with the infrastructure/cloud, a very relevant socioeconomic impact can be obtained, namely safety, congestion and pollution reduction, capacity increase, etc. By contrast, autonomous cars have theoretically the ability to operate independently and without the intervention of a driver in a dynamic traffic environment, relying on the vehicle's own systems and without communicating with other vehicles or the infrastructure.

Original equipment manufacturers (OEMs), Tier ones, and new big players are not developing the same product in this area, but it is not always easy to differentiate their unique selling points because all of them are using erroneously the term "autonomous" cars/driving.

To cope with this problem, the Society of Automotive Engineers issued in 2014 the international norm J3016 (SAE, 2016), bringing order to different prior proposals of standardization from NHTSA and the SAE. It serves as general guidelines for how technologically advanced an automated vehicle is, providing a common taxonomy for automated driving in order to simplify communication and facilitate collaboration within technical and policy domains.

There exist six levels of driving automation spanning from no automation to full automation (see Fig. 8.1). These levels are descriptive and

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability (*Driving Modes*)
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 8.1 SAE J3016 automation levels for driving (SAE, 2016).

technically-oriented, indicating minimum rather than maximum system capabilities for each level.

A key distinction appears between level 2, where the human driver performs part of the dynamic driving task, and level 3, where the automated driving system performs the entire dynamic driving task. It is worth noting that in no way does it propose a particular order of market introduction.

Fig. 8.1 classes the six automation levels following different classification aspects, relevant to understand the implications of each level. The execution, monitoring, and fall-back can be performed either by the human driver or the system, being the differentiator between levels 1–4. Driving modes are an additional aspect, which allows to talk about full automation, when all of them (e.g., expressway merging, high speed cruising, low speed traffic jam, closed-campus operations...) can be handled by a system.

The classical ADAS belong to Levels 0–1 and many examples are already on the market. Some additional solutions of Level 2 have begun to appear in the last couple of years. However, the market emergence of products in Levels 3–5 is quite controversial.

Although the new players sell the vision that these systems will be on the market in a short period of time (before 2020), the European OEMs, affiliated and represented under the European Road Transport Research Advisory Council (ERTRAC) have a more conservative roadmap (see Fig. 8.2). They envision different pathways for urban environments (high automation in areas with low speed and/or dedicated infrastructure) and elsewhere (building on Level 0 use of ADAS to full automation of Level 5 for trucks and cars).

This chapter aims to shed some light into this fascinating debate, providing an overview of the current state of the technology in automated driving, focusing on the potential of the current technology and the

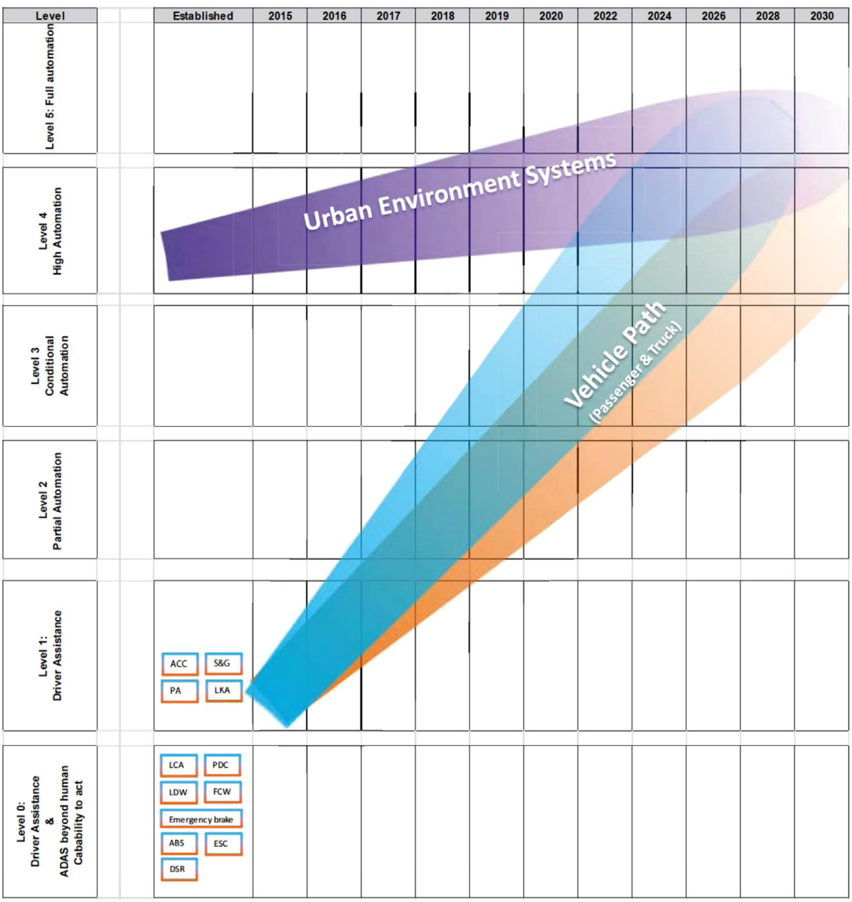

Figure 8.2 Pathways for automated driving (ERTRAC, 2015).

different socioeconomic aspects that will condition the deployment of these complex systems.

To that end, the most relevant technology bricks needed for the automation of a car will be introduced in Section 8.2. In Section 8.3 some relevant aspects of cooperative automated vehicles will be presented. It will be followed by a chapter describing one of the most challenging barriers for automated driving, Verification and Validation (Section 8.4). A brief introduction to the most relevant projects and prototypes through the world is presented in Section 8.5. Finally, a brief description of the socioregulatory aspects will be introduced in Section 8.6.

8.2 TECHNOLOGY BRICKS

This section is devoted to briefly summarizing the key technologies needed for a vehicle to incorporate some degree of automation. The enabling technologies, presented in Part A, are complemented with additional subsystems to conform a closed-loop control architecture, as detailed in Section 8.2.1. From the perception and localization outputs, the vehicle needs to assess the driving situation and infer the subsequent risk (Section 8.2.2). Then, different decision-making strategies are used (Section 8.2.3) to plan, considering a safe driver-vehicle interaction (Section 8.2.4), the most adapted vehicle motion (Section 8.2.5), which is processed and executed, at the end of the decision pipeline, by longitudinal and lateral control algorithms (Section 8.2.6).

8.2.1 Control Architectures

Every control system is articulated based on a functional block architecture, which describes the relationships and dependencies between each of the elements necessary for the control action to be performed correctly. The complex tasks of control of a vehicle must be structured in the form of logical steps that are built on each other and whose complexity can be simplified by a decomposition into functional blocks. In the case of autonomous vehicles, it is also possible and necessary to define all the elements that must be taken into account when carrying out the task of autonomous driving, as well as their relationships and the exchange of information to be shared among them. In this way, we can define the control architecture of an autonomous vehicle as the organization of the different systems of an autonomous vehicle, perception, computation, and

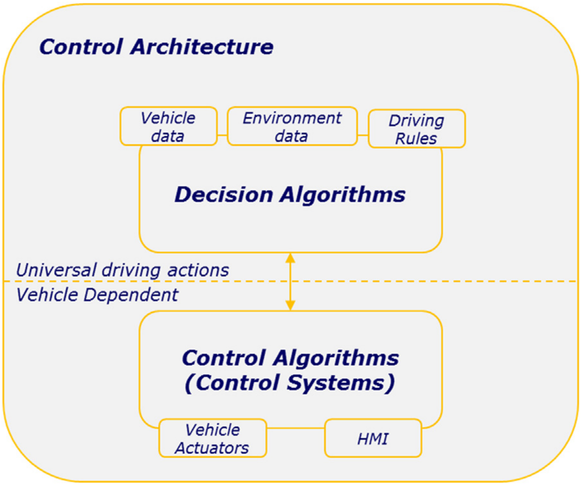

Figure 8.3 General schema of the control architecture.

actuation, to achieve the objectives for which that system has been designed.

Fig. 8.3 shows the general structure of the control architecture of an autonomous vehicle, where the two fundamental elements from the point of view of the management of the driving system are presented: the High-Level control or decision algorithms and the Low-Level control or control algorithms. The first one aims at guiding the autonomous vehicle based on the information supplied by the sensors, regardless of the type of vehicle being piloted. This guiding system generates high level commands such as "turn the steering wheel x degrees," "stop the vehicle" or "select a speed of x kmh." These orders are received by the low-level control, which is the one that is directly related to the actuators of the vehicle, and is able to handle them at its convenience in order to follow the instructions that come from the higher layer. This functional separation is present in most autonomous vehicle control systems, being an inheritance of conventional robotic control systems. The fundamental advantage is that with this structure, the high-level control system is independent of the vehicle in which it is installed, with its actuators being transposable and therefore can be moved from one vehicle to another without making any modifications, easing the interoperability.

The high level control can be subdivided into three elements as shown in Fig. 8.4. In the guidance system, the appropriate algorithms are executed so that the autonomous vehicle tracks the route that has been preset

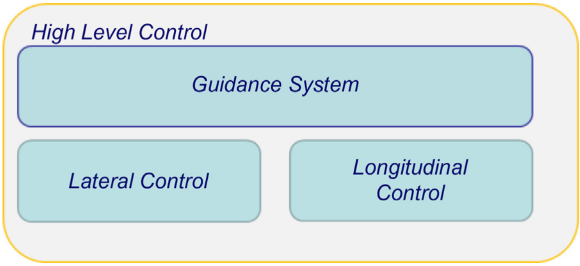

Figure 8.4 Detail of the High-Level Control.

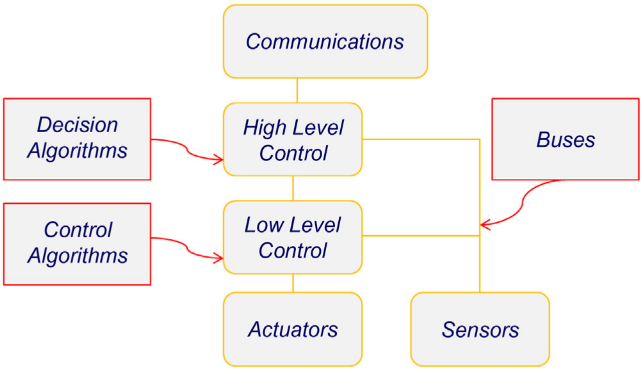

Figure 8.5 Detailed general schema of the control architecture.

to it, as a function of the digital cartography, the information of the sensors, the data of the individual vehicle, and the current driving regulations. This guiding system makes the appropriate decisions and sends them to the specific controllers that perform the lateral control and longitudinal control of the vehicle, who are in charge of managing the trajectory and the speed of the vehicle, whose outputs will be sent to the controller of the low-level to be executed by the actuators.

In this generic architecture for the control of an autonomous vehicle, four additional components must be added, as shown in Fig. 8.5. On the one hand, there is the functional block of actuation, where the actuators of the vehicle are: throttle, brake, steering, and gearbox, although many more systems can be added at convenience, such as lights, alarms, or safety systems. Within the actuation block, ad hoc automations are incorporated, if necessary, when the vehicle itself is not ready to be managed automatically. Once automated, all mechanical components of the vehicle must be able to receive the corresponding instructions generated by the low-level control system.

On the other hand, is the functional block of the sensors, which concerns the equipment that the control systems need to perform correctly the task of guidance and the perception of the environment. The task of guiding is basically through GPS receivers, inertial systems (speedometer, gyroscope, accelerometers, and compass), as well as advanced digital mapping. In recent times, this information for guidance has been enriched with information from the fusion of computer vision or Lidar in so-called visual odometry. The perception of the environment allows the autonomous vehicle to carry out the driving task, taking into account the possible obstacles that appear in its route, the traffic signals, pedestrians, and other vehicles that circulate along the way. For this, two types of sensors are mainly used: computer vision and 3D Lidar.

The third element, part of the control architecture of an autonomous vehicle, is the communications system. The communications allow surpassing the visual horizon to which both the human driver and the sensors installed in the autonomous vehicles are limited, allowing to reach the so-called electronic horizon. This enables the possibility of receiving information from the circulation environment, both infrastructure and other vehicles, so that control systems can take the appropriate decisions or take the necessary actions to prevent an accident, adapt to the circumstances of the road, or anticipate any kind of situation. In addition, communication systems enable autonomous cooperative driving, so that autonomous vehicles are able to collaborate with each other, allowing the emergence of high-level behaviors, such as platooning. In addition, communication systems allow autonomous vehicles to receive information and enable cooperative systems which, while initially designed for manually driven vehicles, must be compatible with autonomous technology.

Finally, the fourth additional element of the generic architecture of the autonomous vehicles is the data buses. These connect all the elements that make up the functional blocks of the architecture in order to allow the exchange of data, sensory information, and control orders in real time. These communication buses are basically two types: on the one hand, Controller Area Network buses (CAN), which allow the transmission of information in real time with messaging defined by priorities. Given their small bandwidth (maximum 1 Mbps), they are mainly used for the interconnection of physical components with low-level control. The other type are the Local Area Network (LAN) networks, networks of computer equipment that operate with a high bandwidth (maximum 1 Gbps), allow the interconnection of all the equipment and sensors with enough capacity.

Once this general autonomous vehicle control architecture is described, it is possible to present the different particular approaches of the different research groups that participate in the autonomous vehicle area.

8.2.2 Situation Awareness and Risk Assessment

Driving is a matter which needs the continuous evaluation of two main factors: the vehicle current state (position, velocity, acceleration, direction) and the environment conditions (near vehicles, obstacles, pedestrians, etc.). To the extent that these two factors are accurately assessed, appropriate decisions can be taken towards reliable autonomous driving. The closer we get to the fact that the vehicle itself is capable of doing this evaluation, the closer we will be to a vehicle-centric approach (Ibañez-Guzmán et al., 2012) to autonomous driving, where the main goal is the safe movement of the vehicle.

This section presents the current state of technology capable of providing onboard situation awareness (SA) and risk assessment (RA) capabilities.

To assess the driving situation, a highly automated vehicle needs the following main capacities (Urmson et al., 2009): global positioning, vehicle tracking, obstacle detection, and self-location in a road model.

The Global Positioning System (GPS) technology is established as the most useful one for answering the Where am I? question (Skog et al., 2009). To face problems such as GPS performance degradation or GPS signal occlusion, it is common to add inertial sensors like Inertial Navigation Systems (INS) or Dead Reckoning (DR) systems (Zhang and Xu, 2012; Tzoreff and Bobrovsky, 2012), with low-cost inertial sensors based on Micro Electromechanical Systems (MEMS) being the most suitable for autonomous driving application. However, due to different drawbacks like the presence of noise in the MEMS-based sensors or failures in the integrating software, INS and DR applications are susceptible to drift, thus causing a loss of accurate vehicle location (Bhatt et al., 2014). The solution that is usually adopted to combat this problem is the use of the above technology combined with sensing technologies such as computer vision, radar, or LIDAR (Jenkins, 2007; Conner, 2011). These, combined with artificial intelligence algorithms, have been developed to

Table 8.1 Main features of different sensing technologies

	Range for optimal operation	Spatial info	Sample rate	Speed measurement	Operation under bad weather	Operation at night
VISION	0–25 m	2D/3D	High	No	Bad	Weak
Radar	1–200 m	–	High	Yes	Excellent	Excellent
LIDAR 2D	1–20 m	2D	Medium	No	Weak	Excellent
LIDAR 3D	1–100 m	3D	Medium	No	Weak	Excellent

put into the market reliable sensing and control systems employed for SA and RA assessment. Table 8.1 shows the main properties of different sensing technologies commonly used in the vehicle-centric approach to autonomous driving.

More recently, some companies have deployed high-performance autonomous cars, with Google Car (Google Inc, 2015) being the most outstanding example. It is fair to say that its astonishingly good behavior in urban environments is not owed so much to its sensing capacities as to very accurate a priori information about the route (ultraprecision 3D maps where the positions of every element—curbs, lights, signals, etc.—are registered with centimeter precision). Nevertheless, it is necessary an accurate detection of relevant information to on-line decide on the safety of the current trajectory. Among aforementioned sensing technologies, it is an accepted fact that 3D LIDAR is the more outstanding one for vehicle-centric autonomous driving, although some manufacturers already provide devices combining different technologies (vision and radar (Delphi Inc., 2015), vision and LIDAR (Continental, 2012)).

Table 8.2 shows the main features of some currently available 3D LIDAR devices for autonomous driving purposes.

Safety is a key piece of the autonomous driving paradigm. Risk (the probability of a vehicle suffering some damage in the future) must be assessed in every particular vehicle situation.

For this purpose, recent works propose mathematical models (motion models) to predict how a situation will evolve in the future. For autonomous driving, those motion models which consider interaction (among vehicles, and pedestrians) are the most useful ones. The challenges include detecting interactions and identifying interactions; the commonly used variables are communications, joint activities, or social conventions and the common tools include rule-based systems. Different motion models are considered and Lefevre et al. (2014) have summarized their main characteristics, as shown in Table 8.3.

Table 8.2 Some commercial 3D LIDAR devices

	Range	Type of data Data rate	FOV (H)-(resolution) FOV(V)-(resolution)	Weight	Accuracy
All-purpose 3D LIDAR					
VLP-16	100 m	Dist./reflect. 0.3 M points/s	360°−(0.1°) ±15°−(0.4°)	0.8 kg	±3 cm
HDL-32	80−100 m	Dist./reflect. 0.7 M points/s	360°−(0.1°) +10°/−30°−(0.4°)	1.3 kg	±2 cm
HDL-64	120 m	Dist./reflect. 2.2 M points/s	360°−(0.018°) +2.0/−25°−(0.4°)	13.5 kg	<2 cm
RobotEye	160 m	Dist./reflect. 0.5 M points/s	360°−(0.01°) on-line adjust. ±35°−(0.01°) on-line adjust.	2.8 kg	±5 cm
Minolta SingleBeam	100 m	Dist./reflect. 0.4 M points/s	180°−(0.01°) 25°−(0.01°)	N.A.	N.A.
Specific purpose 3D LIDAR					
DENSO Pedestrian	200 m	Dist./reflect. 8k points/s	40°−(0.1°) 2°−(1°)	0.2 kg	±6 cm
DENSO Lane	120 m	Dist./reflect. 16k points/s	36°−(0.1°) −2/+2°−(1°)	0.2 kg	±6 cm
SRL1 Obstacle	10 m	Distance 2.3k points/s	27°−(1°) 11°−(1°)	<0.1 kg	±10 cm

Table 8.3 Two motion models

	Variables	Challenges	Tools
Physics-based	Kinematic and dynamic data	State estimation from noisy sensors Sensitivity to initial conditions	Kalman filters Monte Carlo sampling
Maneuver-based	Intentions Perception Surrounding objects and places	Complexity of intentional behavior	Clustering Planning as prediction Hidden Markov models Goal oriented models Learning

8.2.3 Decision Making

Human driving capacity requires not only the ability to properly maneuver the steering wheel, brake, and accelerator according to a set of traffic rules, but also must assess social risks, health, legal consequences, or the life-threatening results of driving actions (e.g., What should the vehicle do if a pedestrian does not stop at a red light?). The resolution of driving requires high levels of human knowledge, for this reason science uses complex artificial intelligence systems to emulate them. The decision-making system has the role of interpreting the abstract information supplied for the perception system of the vehicle and generates actions to carry out sustainable and safe driving.

To operate reliably in the real world, an autonomous vehicle must evaluate and make decisions about the consequences of its potential actions by anticipating the intentions of other traffic participants. The first decision-making systems in autonomous vehicles appeared in 2007 in DARPA Urban Challenge (Urmson et al., 2008). These systems allowed the vehicles to operate in the urban scenarios in which they were involved, U-turns, intersection, parking areas, and real traffic among others. These early decision systems used common elements such as planners where systems were implemented using finite state machines, decision trees, and heuristics. More recent approaches have addressed the decision-making problem for autonomous driving through the lens of trajectory optimization. However, these methods do not model the closed-loop interactions between vehicles, failing to reason about their potential outcomes (Galceran et al., 2015). Nowadays, there are no real systems that outperform a human driver. Advances in decision making are aimed at increasing the intelligence of the systems involved in decision making. Cognitive systems (Czubenko et al., 2015), agents systems (Bo and Cheng, 2010), fuzzy systems (Abdullah et al., 2008), neural networks (Belker et al., 2002), evolutionary algorithms (Chakraborty et al., 2015), or rule-based methods (Cunningham et al., 2015) compose the Intelligent-Decision-Making Systems (IDMS) (Czubenko et al., 2015). Fig. 8.6 shows the location of IDMS from a point of view of the functional architecture for autonomous driving (Behere and Törngren, 2015).

A general intelligent system in the autonomous driving functional architecture will contain processing units of sensory processing, a world model, behavior generation, and value judgement with information flow as shown in Fig. 8.7 (Meystel and Albus, 2002).

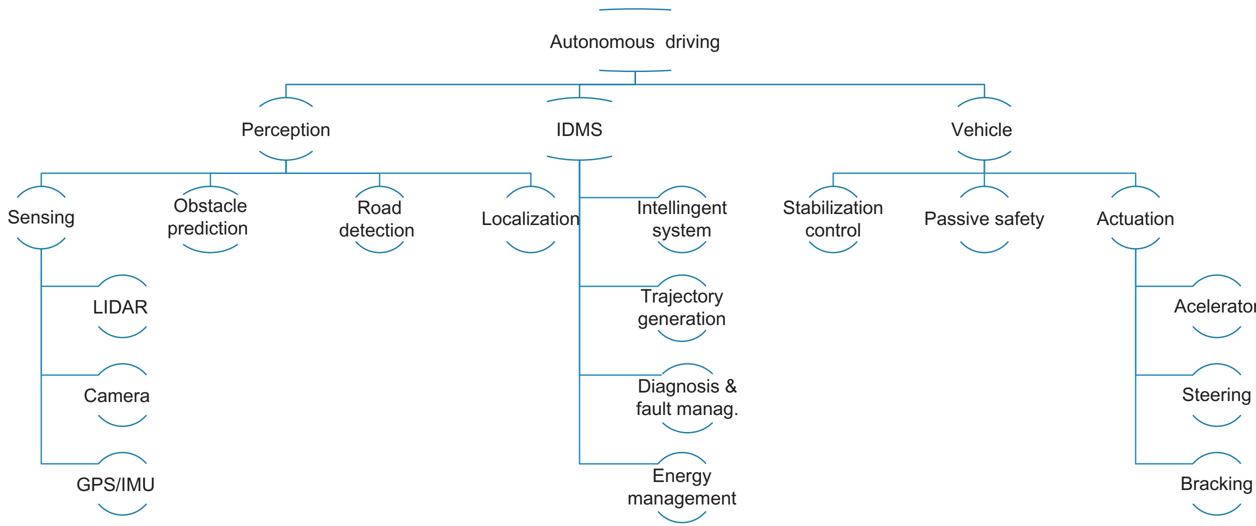

Figure 8.6 Autonomous driving functional architecture.

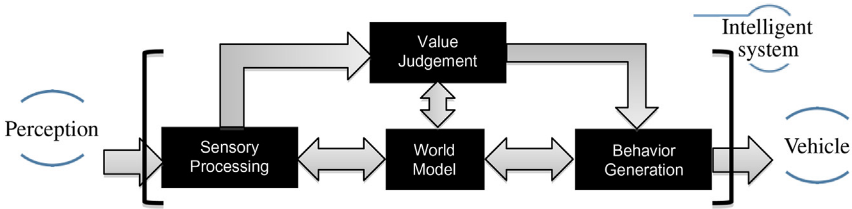

Figure 8.7 General ANSI intelligent system description.

Autonomous driving in complex scenarios where multiple vehicles are involved (e.g., urban areas) with inherent risk to the integrity of people, require real-time solutions. In this case decision making demands reliability, safety, and a fault tolerance system. In this direction it is necessary to mention, a real-time motion planning algorithm, based on the rapidly-exploring random tree (RRT) approach proposed by (Kuwata et al., 2009). The proposed algorithm was at the core of the planning and control software for Team MIT's entry for the 2007 DARPA Urban Challenge, where the vehicle demonstrated the ability to complete a 60 mile simulated military supply mission, while safely interacting with other autonomous and human driven vehicles.

A Multicriteria Decision Making (MCDM; Furda and Vlacic, 2010) and Petri nets (Furda and Vlacic, 2011) are proposed for solving the problem of the real-time autonomous driving. MCDM offers a variety of benefits such as:

- The hierarchy of objectives allows a systematic and complete specification of goals to be achieved by the vehicle.
- The utility functions can be defined heuristically to reflect the choices of a human driver, or, alternatively, learning algorithms can be applied.
- MCDM allows the integration and evaluation of a very large number of driving alternatives.
- Decision flexibility can be achieved by defining the set of attribute weights depending on the road conditions.
- Additional objectives, attributes, and alternatives can be added without the need for major changes.
- The driving maneuvers are modeled as deterministic finite automata.
- The decision making unit is modeled as a Petri net.

The author concluded that the method was highly based on heuristics, but the application of MCDM in this new research area offers a variety of benefits with respect to the problem specification, decision flexibility, and scalability.

8.2.3.1 Simulation and Software tools for IDMS

The first tasks of testing and implementation of IDMS are strongly relational with software simulation and implementation. There are numerous works about framework and middleware capable of the testing and development of IDMS (Veres et al., 2011; Behere, 2013). Table 8.4 shows a set of software environments and tools capable of participating in the process of modeling, simulation, and implementation of a decision-making system. The table shows the name of the environment, a short description, its purpose and features, and a website where the software packages can be found.

8.2.4 Driver–Vehicle Interaction

As shown in Fig. 8.8, a car's cockpit has constantly increased its complexity and evolved, according to the implemented systems, available technologies, and driver needs present at each specific moment in automotive history.

At the end of the 1990s and the beginning of 21st century the increase of in-vehicle implemented functions and the associated human interaction complexity, obliged OEMs to pay more attention to the proper design of the driving place and the driver–vehicle interaction. At that time, new transversal HMI departments were introduced in most R&D automotive manufacturing organizations to manage this new challenge. The role of this department was to take the leadership to develop, from a global vehicle perspective, the complete driving place in terms of the driver–vehicle interaction—of course, with the support of all other technical areas involved.

In recent years, the massive introduction of new driver assistance systems and infotainment and telematics applications has given even more relevance to the HMI design activities.

Nowadays, the introduction of automated driving functions in the next-generation vehicles has HMI and driver–vehicle interaction as one of the key challenges for its success in the market. In this sense, there is a need to rethink ant to redesign the driving place and the way the "driver" or "vehicle supervisor" is going to interact with the vehicle and the automated and autonomous functions.

Some of the key questions related to the HMI design for automated vehicles are the following:

- How can two different driving modes coexist at the same time?

Table 8.4 Analysis, simulation, and controller development tools for developing of IDMS

Software environment/tools	Short description, purpose, or features	Website
Charon	Charon HS modeling language, supports hierarchy and concurrency, has simulator and interfaces to Java.	www.cis.upenn.edu/mobies/charon
Modelica/Dymola	OO HS modeling language for multi-domain physics, has simulator, has object libraries.	http://www.modelica.org/ http://www.3ds.com/products/catia/portfolio/dymola
HyTech	Modeling and verification of hybrid automata, has symbolic model checker.	http://embedded.eecs.berkeley.edu/research/hytech/
HyVisual	Visual modeling (Ptolemy II) and simulation of HS, supports hierarchies.	http://Ptolemy.berkeley.edu/hyvisual/
Scicos/Syndex	Modeling and simulation of HS, has toolbox, real-time code generation, provides formal verification tools.	http://www.scicos.org/
Shift	Has its own programming language for modeling of dynamic networks of hybrid automata, has extension to real-time control and C-code generator.	http://path.berkeley.edu/shift/
Simulink/Stateflow	Has analysis, simulation, has libraries and domain specific block-sets, can compile C-code for embedded applications via the use of embedded MATLAB™, and Real Time Workshop™.	http://www.mathworks.com/
OROCOS	Portable C++ libraries for advanced machine and robot control. Including kinematics chains, EKF, Particle Filters, RT software components, state machine, etc.	http://www.orocos.org/
ROS	It is a set of software libraries and tools that help you build robot applications. From drivers to state-of-the-art algorithms, and with powerful developer tools. It is NOT RT framework.	http://www.ros.org/

(*Continued*)

Table 8.4 (Continued)

Software environment/ tools	**Short description, purpose, or features**	**Website**
YARP	It is a communication middleware, or "plumbing," for robotic systems. It supports many forms of communication (tcp, udp, multicast, local, MPI, mjpg-over-http, XML/RPC, tcpros, . . .).	http://www.yarp.it/
CLARATy	It consists of a Functional Layer that provides abstractions for various subsystems and a Decision Layer that can do high level reasoning about global resources and mission constraints.	http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1249234
BALT & CAST	It is a middleware for cognitive Robotics development.	http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4415228; http://ieeexplore.ieee. org/lpdocs/epic03/wrapper.htm?arnumber=4415228

Figure 8.8 Evolution of car's driving place.

- How would the driver select the automated mode?
- Which functions should be controlled during automated mode and how to manage them?
- How should the vehicle give back the control to the driver?
- How should be the transitions between different automation levels be achieved?
- Which kind of information should the vehicle provide to the driver in automated mode to make him/her feel safe and comfortable?
- Which present and future technologies can be used and are more appropriate to make all this happen?

Figure 8.9 Example of an augmented reality evaluation study for an automated driving prototype (carried out at CTAG's driving simulator).

To approach in a consistent manner all these questions, there are several issues which are very relevant and that must be taken into account:

- To follow a systematic methodology that allows to identify relevant functions and HMI parameters for automated driving, to design innovative HMI solutions, and to evaluate the developed concepts from a human factors perspective.
- To implement a multidisciplinary approach that takes into account, from the very first moment, the opinion of engineers, specialists, and technicians coming from different disciplines.
- To prepare and to adapt new methods and new development and evaluation tools (driving simulators, mock-ups, vehicle prototypes, ...) that allows to perform clinics with users to measure in detail ergonomics and technical aspects.
- To test the driver—vehicle interaction proposed solutions in very early development stages to get the voice of the user at the beginning of the development process.
- To use, in an intelligent manner, the future possibilities of new HMI technologies such as augmented reality (see Fig. 8.9), state-of-the art displays, new control elements, gesture recognition, interior lighting, reconfiguration possibilities, etc.
- To improve, in a very safely manner, the in-vehicle onboard user experience.

8.2.5 Motion Planning

Motion planning first for mobile robots and then for autonomous vehicles has been extensively studied over the last few decades. The resulting strategies were designed to meet, under different hypotheses, a variety of kinematic, dynamic, and environmental constraints. In this section, path and speed planning are presented under a specific approach, but different strategies could be alternatively considered (c.f. Paden et al., 2016 or Katrakazas et al., 2015 for more details)

8.2.5.1 Path Planning

General techniques to obtain optimal paths can be grouped into two categories: indirect and direct. Indirect techniques discretize the state/control variables, and convert the path planning problem into one of parameter optimization which is solved via nonlinear programming (Dolgov et al., 2010) or by stochastic techniques (Haddad et al., 2007). The latter use Pontryagin's maximum principle and reexpress the optimality conditions as a boundary value problem, whose approximate solutions have been investigated under a large set of possibilities and constraints. The subsequent description is based on the latter family, where a local planner is encapsulated into a generic procedure to cope with complex scenario topologies and obstacle avoidance.

The path planning system is composed of several subsystems that operate separately as standalone processes depending on one another. These subsystems are: *Costmap Generation, Global Planner*, and *Local Planner*. The former is in charge of the computation of the costmap that will be used by the other two methods in order to compute the trajectories, considering the safety existing for the different possibilities (attending to the obstacles in the environment, as well as to the estimations of the expected changes in the near future); the second is used for the computation of a trajectory that allows the vehicle to travel between the current position and the goal in the unstructured map. The third provides the system with the mechanisms needed to follow it, computing the commands required by the low-level controller to move the prototype.

8.2.5.1.1 Costmap Generation

The costmap maintains information about occupied/free areas in the map in the form of an occupancy grid. It uses sensor data and information from the static map to store and update information about obstacles in the world, which are marked in the map (or cleared, if they are no longer there). Costmap computation is supported on a layered costmap, which will be used for the integration of the different information sources into a single-monolithic costmap. At each layer, information about occupied/

free areas in the surroundings of the vehicle is maintained in the form of an occupancy grid, using the different observation sources as input. Using this information, both dynamic and static obstacles are marked in the map. For example, let us suppose each cell in the map can have 255 different cost values. Then, at each layer, costmap is represented as follows:

- A value of 255 means that there is no information available about a specific cell in the map.
- 254 means that a sensor has marked this specific cell as occupied. This is considered as a lethal cell, so the vehicle should never enter there.
- The rest of cells are considered as free, but with different cost levels depending on an inflation method relative to the size of the vehicle and its distance to the obstacle.

Cost values decrease with the distance to the nearest occupied cell using the following expression:

$$C(i,j) = \exp\left(-1.0 \cdot \alpha \cdot \left(\left\| c_{ij} - \vec{o} \right\| - \rho_{\text{inscribed}}\right)\right) \cdot 253 \tag{8.1}$$

In this expression, α is a scaling factor that allows increasing or decreasing the decay rate of the cost of the obstacle. $\left\| c_{ij} - \vec{o} \right\|$ is the distance between the cell $c_{ij} \in C$ (where C is the set of cells in the costmap) and the obstacle. Finally, $\rho_{\text{inscribed}}$ is the inscribed radius, which is the inner circle of the limits of the car.

Despite all of them being free cells, normally different distance thresholds are defined in order to set different danger levels in the map. For example, it is possible to define four thresholds:

- ζ_{lethal}: There is an obstacle in this cell, so the vehicle is in collision. It would be represented by the cost level 254.
- $\zeta_{\text{inscribed}}$: Cell distance to the nearest obstacle is below $\rho_{\text{inscribed}}$. If the center of the vehicle is in this cell, it is also in collision, so areas below this distance threshold should be avoided. Cost level would be 253.
- $\zeta_{\text{circumscribed}}$: If the vehicle center is on this cell, it is very likely that the car is in collision with an obstacle, depending on its orientation. A cell with a distance to an obstacle below this threshold should be avoided, but there are still chances of being in one of them without colliding an obstacle.
- The rest of cells are assumed to be safe (except from those with unknown cost, for which it is not known if they are occupied or not, being considered as lethal).

In the presented approach, just those paths passing through cells with a cost below $\zeta_{\text{circumscribed}}$ are considered. This cost is obtained using the

Eq. 8.1 and other cost factors that will be explained later. Paths passing through the cells over this threshold will be truncated at the last safe point.

For the computation of the costmap and the costs associated to each cell, ROS plugin costmap 2d (http://wiki.ros.org/costmap_2d) could be used, which implements some of the functionalities described in this section.

Layered Costmap

Nowadays, in the layered costmaps, four different layers are usually being considered:

- A first layer represents the obstacles in a static map previously captured. This map represents the static obstacles in the whole area in which the vehicle will move. This layer is the only one used by the nonprimitive-based global planner, since nonstatic obstacles are not being considered for nonprimitive-based trajectory generation (meaning by nonstatic those obstacles that are not already included into the map). These are supposed to be avoided at local planning level.
- A second layer, also based on a static map, is included. For optimization reasons, in this and following layers, the costmap is not computed for the whole map at each iteration. Instead, just the cells in an area centered into the current car position are updated. The goal is not to update the whole map, since these layers are just used for local planning or local maneuvering. Static obstacles are also included for local planning, since the vehicle is not desired to pass along restricted areas while avoiding obstacles. This allows the vehicle to know which areas are forbidden, also at local planning level.
- A third layer is used to represent the dynamic obstacles detected by the different sensors. Using this input, ground is detected and removed, extracting just the vertical obstacles to which the vehicle could collide. Parameters in this layer are chosen so the obstacle inflation is stronger than the one computed for the second layer. This gives more priority to the obstacles being detected in real time over those in the static map.
- The last layer provides an estimation of the future motion of the dynamic obstacles. To do so, input point clouds are segmented using a voxel grid, in order to reduce dimensionality. The world surrounding the vehicle is divided into a discrete number of voxels of equal size. For each voxel, an occupancy probability is assigned, based on the number of points from the input point cloud in its neighborhood.

Using this probability, valid voxels (with a higher occupancy probability) are distinguished from the noisy ones (with a smaller probability).

All these layers are combined into a single costmap. Note that it is interesting to include the motion of the obstacles in the costmap because then the vehicle tries to avoid the obstacle by the side in which it is not crossing its trajectory

8.2.5.1.2 Global Planner

Usually there are two global planners in use in autonomous vehicles: the primitive-based planner and the nonprimitive global planner. These planners are intended to obtain a feasible path going from the vehicle's current position to a determined goal.

Although both methods are included in this section, their aims in the system are completely different. The nonprimitive-based global planner is used for regular navigation, while the primitive-based global planner, is used for recovering the vehicle in situations in which there is an obstacle in the way for a long time, or the vehicle is performing some complex maneuver, like parking.

Primitive-Based Global Planner

The primitive-based global planner constructs a path from the vehicle's position to a desired goal. The path is generated by combining "motion primitives," which are short, kinematically feasible motions. These motion primitives are generated using a model of the vehicle in order to comply with the curvature restrictions of the vehicle.

The computation of these primitives is performed as follows: a set of predefined orientations is considered. For each orientation, the model is evolved until it reaches one of the predefined orientations, at different speeds. This process is done both forwards and backwards. After this process, a set of small trajectories that fulfill vehicle restrictions is obtained, which will be used as the building blocks for the planner.

Having these, an ARA algorithm is used for the search of a feasible path. At each node expansion, a new x, y, and θ position is explored, until the best path is found or the exploration time finishes (if so, the best path found until then is used). During this search, the cost of backward primitives is set higher than the cost of forward ones to prevent the vehicle from using backward paths as much as possible, without decreasing the performance. Also the original search algorithm can be improved by adding a new cost that penalizes the concatenation of forward and backward primitives. This is done with the intention of planning more natural paths.

Nonprimitive-Based Global Planner

The nonprimitive based global planner computes the minimum cost path from the vehicle's position to the goal using, e.g., the Dijkstra's algorithm. Given the speed of the search algorithm to obtain the global plan, this planner is being used as a rough estimate of the route that the vehicle is going to follow. The static obstacles of the costmap are then overinflated in order to make the planner construct smooth paths, feasible for being followed by an Ackermann vehicle.

If the generated routes are not constructed bearing in mind the nonholonomic restrictions of the vehicle, it is frequent that the initial angle between the vehicle's orientation and the orientation of the global plan is larger than the maximum angle required by the local planner to generate feasible paths. That is the reason why the nonprimitive-based planner is used in combination with a local planner state machine that takes into account this circumstance and reorients the vehicle properly before using the Frenet-based local planner.

8.2.5.1.3 Local Planner

Once the global path is defined, a method is required that is able to compute the steering and speed commands needed to control the vehicle, in order to follow that path. This method should be also able to avoid the obstacles present in the road. This must be done in a safe and efficient way.

The basic idea of the local path generation is to define a set of feasible paths and choose the best option in terms of their cost. The winner path defines the steering and speed commands that the vehicle will use. Having options among local paths is useful to overcome the presence of unforeseen obstacles in the road.

Usually the current Euclidean coordinate system is transformed into a new system based on the Frenet space. This space is computed as follows: the global path is considered as the base frame of a curvilinear coordinate system. The feasible local paths are defined in terms of this base frame in the following way:

- The nearest point (where the distance is computed perpendicular to the global path) to the main trajectory will be the origin of the curvilinear coordinate system.
- The horizontal axis will be represented by the distance over the global path, in its direction.
- The vertical axis is represented by the vector perpendicular to the origin point, which is pointing to the left with respect to the path direction.

In this schema, trajectories can be computed easily in the curvilinear space (that is, maneuvering information is generated). These are then transformed to the original Euclidean space, in which the obstacles information is added by assigning costs to each path.

Based on this idea, the method can be divided in five stages:

1. Generation of the costmap. Using the information generated by the sensors or by the methods described in previous sections, the system constructs a costmap in which costs are related to the distance to obstacles.
2. Base frame construction. Based on the global path constructed in the previous section, the base frame of the curvilinear coordinate system is generated.
3. Candidate paths generation. Candidate paths are generated into the curvilinear space. Then, they are transformed to the Euclidean space.
4. Selection of the winner path. Costs for all the paths are assigned, and the one with the lowest value is selected.
5. Computation of the vehicle commands. Vehicle speed and steering angles are computed based on the characteristics of the winner path.

Base Frame Construction

In this stage, the base frame of the curvilinear coordinate system is defined, so the algorithm is able to compute the trajectories in this space as if the global plan was a rectilinear trajectory. At this point, the potential presence of obstacles or the restrictions associated to the vehicle's motion model are not considered, limiting this stage to the generation of trajectories.

The origin of coordinates of the base frame is the nearest point in the global plan to the vehicle's position.

The base frame's arc length is obtained as the distance of each point along the global plan (represented as a green line) to the origin of coordinates. This distance is represented in the x-axis of the curvilinear system. y-axis, q, represents the perpendicular lateral distance respect to the path. The left side is represented by positive values and the right by negative values.

For the computation of the transformation between the Euclidean and the curvilinear coordinate system, the path curvature κ is needed:

$$\kappa = \frac{S}{Q} \cdot \left(\kappa_b \cdot \frac{(1 - q \cdot \kappa_b) \cdot \left(\partial^2 q / \partial s^2\right) + \kappa_b \cdot \left(\partial q / \partial s\right)^2}{Q^2} \right) \tag{8.2}$$

where

$$\begin{cases} S = \text{sign}(1 - q \cdot \kappa_b) \\ Q = \sqrt{\left(\dfrac{\partial q}{\partial s}\right)^2 + (1 - q \cdot \kappa_b)^2} \end{cases} \tag{8.3}$$

A generated path will be rejected if $q > \frac{1}{\kappa_b}$. In this case, the path curvature and sense is opposed to that of the base frame. The path violates the nonholonomic condition of the movement of the vehicle, so the vehicle enters in a recovery state.

Only paths with a lateral offset q equal or smaller to the curvature radius of the base frame $\frac{1}{\kappa_b}$ are accepted.

If $q = \frac{1}{\kappa_b}$, that means that the path passes through the center of curvature of the base frame. Also, the maximum curvature a path can have in order to be feasible by the vehicle is limited by the maximum steering angle. If this restriction is violated, the corresponding path is rejected. Curvature is directly related to the movement of the vehicle, which can be described through several models.

Candidate Paths

As seen, path generation is performed in the curvilinear space, without considering the obstacles in the environment. These will be taken into account later, once the tentative trajectories are transformed to the Euclidean space.

Maneuvering paths generation. The curvature of the generated paths is defined by the lateral offset q with respect to the base frame. First and second order derivatives of q are needed for the computation of κ (see Eqs. 8.2 and 8.3), so a function dependent on the lateral offset is needed to compute a smooth lateral change.

Candidate paths generation. Once the paths in the curvilinear coordinate system are computed, they are transformed to the Euclidean space. In this new space, their associated costs will be evaluated. Now the paths are in Euclidean coordinates, the maximum distance they can reach individually (if obstacles are considered) can be calculated. To do that, the cells C_{ij} of the costmap associated to the points of the trajectory are checked. If this cost is over the value associated to the threshold circumscribed, the path is truncated at this point.

When a path collides with an obstacle, it is not completely removed. The reason is that there are certain situations in which the maximum length cannot be reached with any path. However, it is still desirable to approach slowly towards the maximum reachable point, with the hope that the obstacles that are blocking the way will disappear in the next iterations. In crowded areas with many pedestrians this is a typical situation: the way is blocked, but when pedestrians see a vehicle that is approaching, they move away. However, if the vehicle reaches a point in which it can not move for a long time, the recovery behavior is triggered. The problem with this strategy is that one of the colliding paths could win even if there is a path able to go through a clear area. In order to avoid that, a weighted cost function based schema is implemented. This schema, which permits a smart selection of the winner path, is explained in the next section.

Winner Path

The winner path is selected through the use of a linear combination $J[i]$ of weighted cost functions, related to the following parameters: occlusion, length, distance to the global path, curvature and consistency of the path. $J[i]$ is evaluated as follows:

$$J[i] = \omega_0 C_0[i] + \omega_l C_l[i] + \omega_d C_d[i] + \omega_\kappa C_\kappa[i] + \omega_c C_c[i] \tag{8.4}$$

Here, i is the path index, and C_0, C_l, C_d, C_κ, and C_c are the costs of occlusion, length, distance to the global path, curvature, and consistency, respectively. Their relatives ω_i, $i \in \{o, l, d, \kappa, c\}$ are the associate weights that allow to adjust the influence of each of the costs to the final cost value. All these costs are normalized to 1.0, and

$$\sum_{i \in \{o,l,d,\kappa,c\}} \omega_i = 1.0 \tag{8.5}$$

so it is easy to determine the proportional influence of each weight

Occlusion. The occlusion cost is related to the safety of the path. This cost estimates the goodness of a path, with the bests paths being those passing far enough from the obstacles. To do that, the method iterates along the path, simulating the footprint of the car at each position. The occlusion cost corresponding to the trajectory point i will be the maximum cost of each of the cells $c_{ij} \epsilon C$ under the footprint of the vehicle at that position. Based on this, the occlusion cost will be:

$$C_0 = \frac{\max\{c_i\}}{255}, \quad i = 1, ..., L \tag{8.6}$$

In this expression, L is the length of the current path being evaluated. $\max\{c_i\}$ is the maximum value of all the costs, associated to a point in the path. If the maximum value of each cost is 255, so C_0 is divided by this value, in order to normalize it to 1.

Length. This cost represents the length of the current path. By iterating along the points in the path, the distance between them is accumulated, so the real distance traveled in Euclidean coordinates is known. The longer a path is, the better, as it is assumed that it will traverse an obstacle-free zone. Thus, long paths should produce low cost values. This is done through the expression:

$$C_l = 1 - \frac{\sum_{i=1}^{L} \left\| p_i - p_{i-1} \right\|}{q_{f_{\max}} + s_f} \tag{8.7}$$

Here, p_i is a certain point inside the evaluated path. $q_{f_{\max}}$ is the maximum value that a q_f can have for a certain path. Lengths are normalized to a value that a path will never reach. This cost is subtracted from 1.0, in order to make it comparable to the rest of costs (as said, lower values are preferred respect to the higher ones).

Distance to the global path. In the presented implementation, information about the average lateral o_{set} with respect to the global path has been also considered. The use of this cost will benefit the choice of those paths that are allowed to come back to the global path after an occasional obstacle is avoided. It is computed as follows:

$$C_d = \frac{\sum_{i=1}^{L} \left\| p_i - \text{nearest}(p_i, g) \right\|}{L \cdot q_{f_{\max}}} \tag{8.8}$$

where nearest(p,g) is the nearest point in the global path g to the point p. This cost is normalized with respect to the maximum expected offset $q_{f_{\max}}$.

Curvature. This cost gives priority to the smoother paths. Let $p(x_i, y_i)$, $i = 1, \ldots, L$, be a point in the path. Then,

$$C_\kappa = \max\left\{\frac{\dot{x}_i \cdot \ddot{y}_i - \ddot{x}_i \cdot \dot{y}_i}{\left(\dot{x}_i + \dot{y}_i\right)^{3/2}}\right\}, i = 1, \dots, L \tag{8.9}$$

Consistency. This cost avoids the continuous changes in the winner paths between iterations. Once the vehicle starts a maneuver, the idea is to keep this behavior in the following iterations. This is done through the following expression:

$$C_c = \frac{1}{s_2 - s_1} \int_{s_1}^{s_2} l_i ds \tag{8.10}$$

The lateral cost $l_i(s)$ is the distance between the current and the previous winner path at the same longitudinal position s.

Selection of the winner path. Once all costs are computed, the expression described in Eq. (8.4) is applied. In those paths for which it is impossible to advance due to the presence of a nearby obstacle or because the car is incorrectly aligned to the global path (meaning that no valid paths can be generated in this situation), the cost will be negative (invalid path). From all paths, that path with the smallest cost (winner path W) is selected. If for any reason there are no valid paths, a recovery maneuver is initiated.

8.2.5.2 Speed Planning

The speed reference is commonly assumed to be continuously differentiable, and is often designed by optimizing an appropriate performance index (minimum time is the commonest criterion, but minimum acceleration and/or jerk has also been used). For most of them, as topological semantic maps are not used, iterative or optimization processes are required to satisfy a certain number of driving comfort constraints—maximum speed, longitudinal and lateral acceleration, and jerk. The strategy described below tackles this problem, approximating the considered path by well-known primitives, from which a speed profile can be derived.

Indeed, any path in an unstructured environment can be decomposed, with the help of a path planning algorithm, into a succession of turns—composed of clothoids and arcs of circles—and straight lines. The clothoid is chosen because an arc of a clothoid has variable curvature, in every point proportional to the arc length, and it provides the smoothest link between a straight line and a circular curve. It is used in roads and railroads design: the centrifugal force actually varies in proportion to the time, at a constant rate, from zero value (along the straight line) to the maximum value (along the curve) and back again.

This decomposition is extremely useful in finding closed-form optimal speed profiles because both straight line segments and circle arcs can be associated with constant speeds. More precisely, when a turn is initiated the maximum velocity will be constrained by the comfort lateral acceleration threshold, and when a straight segment is being tracked, the maximum longitudinal speed, acceleration, and jerk will be the limits imposed on the reference speed.

The speed profile can be defined as follows:

- Constant speed curves at a minimum value $V_{\min}$ when the curvature profile is a circular arc or its preceding clothoid.
- A smooth transition from the minimum value $V_{\min}$ to a maximum allowed speed $V_{\max}$ and back again to $V_{\min}$ that fulfills the acceleration and jerk constraints.
- A set of one or two smooth transition curves (of type 2 above) that go from zero to the maximum speed, and vice versa.

In order to obtain closed-form expressions for the second type of curve, the speed trajectory is divided into a number of intervals. Let us suppose seven intervals $[t_{i-1}, t_i]$, $i = 1\ldots7$ and represented in terms of the arc length s_r as follows:

$$
\begin{aligned}
&\dddot{s}_r(t) = \begin{cases} \dddot{s}_{r_{\max}}, \ t \in [t_0, t_1] \ \text{ or } \ t \in [t_6, t_7] \\ 0, \ t \in [t_1, t_2] \ \text{or} \ t \in [t_3, t_4] \ \text{ or } \ t \in [t_5, t_6] \\ -\dddot{s}_{r_{\max}}, \ t \in [t_2, t_3] \ \text{ or } \ t \in [t_4, t_5] \end{cases} \\
&\ddot{s}_r(t) = \ddot{s}_r(t_{i-1}) + \dddot{s}(t)_r \cdot (t - t_{i-1}) \\
&\dot{s}_r(t) = \dot{s}_r(t_{i-1}) + \ddot{s}_r(t_{i-1}) \cdot (t - t_{i-1}) + \frac{1}{2}\dddot{s}_r(t_{i-1})(t - t_{i-1})^2 \\
&s_r(t) = s_r(t_{i-1}) + \dot{s}_r(t_{i-1}) \cdot (t - t_{i-1}) + \frac{1}{2!}\ddot{s}_r(t_{i-1})(t - t_{i-1})^2 + \frac{1}{3!}\dddot{s}_r(t_{i-1})(t - t_{i-1})^3
\end{aligned}
\tag{8.11}
$$

The arc length will go from the initial point of the closing clothoid in a turn $(s_r(t_0))$ to the final point of a straight line segment $(s_r(t_7))$, the initial and final speeds $(\dot{s}_r(t_0), \dot{s}_r(t_7))$ will be set by the minimum speed $V_{\min}$, and the initial and final accelerations $(\ddot{s}_r(t_0), \ddot{s}_r(t_7))$ and jerks $(\dddot{s}_r(t_0), \dddot{s}_r(t_7))$ will be both equal to zero. Concerning the comfort constraints, the maximum speed will be $V^*_{\max}$ and the maximum speed and acceleration will be determined by design parameters $\gamma_{\max}$ and $J_{\max}$.

Note that the value of $V^*_{\max}$ corresponds to the $V_{\max}$ previously defined if there is enough distance to reach the target. If the available arc length is less than some critical value, the maximum speed will be set equal to the initial speed V_0 resulting in the generation of a constant speed profile. Otherwise, a maximum speed between V_0 and $V_{\max}$ will be computed. The closed form polynomial expression of equations (8.11) permits the maximum speed to be computed as follows:

$$V^*_{\max} = \begin{cases} V_{\max}; & \text{if condition1 is satisfied} \\ V_0; & \text{if condition2 is satisfied} \end{cases} \tag{8.12}$$

condition1

$$\Delta_s \geq (V_{\max} + V_0)\sqrt{\frac{V_{\max} - V_0}{J_{\max}}} + (V_{\max} + V_{\min})\sqrt{\frac{V_{\max} - V_{\min}}{J_{\max}}} \tag{8.13}$$

condition2

$$\begin{aligned} \Delta_s < {} & \frac{V_0}{2} \cdot \left(\frac{V_0}{\gamma_{\max}} + \frac{\gamma_{\max}}{J_{\max}}\right) \\ & - \frac{1}{2J_{\max}}\left(\gamma^2_{\max} - \sqrt{\gamma^4_{\max} + 8J^2_{\max}\gamma_{\max}\Delta_s + 4J^2_{\max}V_0^2 - 4\gamma^2_{\max}J_{\max}V_0}\right) \end{aligned} \tag{8.14}$$

where $\Delta_s = s(t_7) - s(t_0)$

An alternative algorithm can be implemented to reduce the overall time needed to cover the path by slightly compromising the passenger comfort. Instead of reducing the speed to $V_{\min}$ in each turn, only the nondegenerate turns are taken into account for this purpose.

8.2.6 Vehicle Control

Mathematical models are of great importance in the analysis and control of automotive vehicle dynamics. Several mathematical models are available in the literature with different levels of complexity and accuracy according to the physical phenomena captured. Usually the motion of the vehicle is considered in the yaw plane, mainly describing the longitudinal and lateral vehicle motion. In the description of the vehicle motion, different longitudinal and lateral dynamic couplings must be considered:

- Dynamic and kinematic couplings are due to the motion in the yaw plane caused by wheel steering.
- The interaction between tire and road is at the origin of another important coupling.
- The longitudinal and lateral accelerations cause a load transfer between the front and rear axles as well as the right and left wheels.

The complexity degree is used to obtain a trade-off between complexity and accuracy. A complexity model can provide a good accuracy level but remains too complex for controller synthesis. For this reason, usually a nonlinear bicycle model is used for lateral control and a one-wheel vehicle model for longitudinal control design.

A nonlinear bicycle model considers the longitudinal (x), lateral (y), and yaw motion (θ). For this model, it is assumed that the mass of the vehicle is entirely in the rigid base of the vehicle, and it considers the pitch load transfers while neglecting the lateral load transfer caused by roll motion.

In the Fig. 8.10, α is the steer angle, and a and b represent the distance between wheels and the gravity center of the vehicle. The indexes f and r indicate front and rear.

The dynamic equations are:

$$\begin{aligned} m\left(\ddot{x} - \dot{y}\dot{\theta}\right) &= \textstyle\sum_{i=f,r} F_{xi} + F_r \\ m\left(\ddot{y} + \dot{x}\dot{\theta}\right) &= \textstyle\sum_{i=f,r} F_{yi} \\ I_z\ddot{\theta} &= F_{yf}\cdot a - F_{yr}\cdot b \end{aligned} \tag{8.15}$$

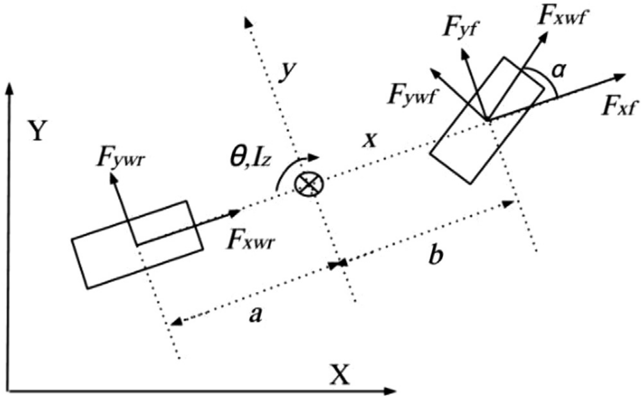

Figure 8.10 Nonlinear bicycle model.

where m is the vehicle mass, I_z is the yaw moment of inertia, F_r is the sum of resisting forces, and F_{xi}, F_{yi} are longitudinal and lateral tire forces along the x-axis and y-axis, respectively. These forces can be relationed with the longitudinal tire force F_{xwi}, lateral force F_{ywi}, and the wheel steer angle α as:

$$\begin{cases} F_{xf} = F_{xwf}\cos(\alpha) - F_{ywf}\sin(\alpha) \\ F_{yf} = F_{xwf}\sin(\alpha) - F_{ywf}\cos(\alpha) \end{cases} \tag{8.16}$$

When a driving torque T_d and a braking torque T_b are applied the rotational motion can be derived as:

$$I_z\dot{\omega}_{wi} = T_{di} - F_{xi}R - T_{bi}(i = f, r) \tag{8.17}$$

where R is the radius of the wheel and ω_{wi} is the yaw angular velocity.

The trajectory of vehicle center of gravity in an absolute inertial coordinate system is given by:

$$\begin{cases} \dot{X} = \dot{x}\cos(\theta) - \dot{y}\sin(\theta) \\ \dot{Y} = \dot{x}\sin(\theta) + \dot{y}\cos(\theta) \end{cases} \tag{8.18}$$

8.2.6.1 Longitudinal Motion Control

For controller synthesis, usually the longitudinal model is based on a one-wheel vehicle model. So, the sum of the longitudinal forces acting on the vehicle center of gravity is:

$$m\dot{v} = F_p - F_r \tag{8.19}$$

where $v = \dot{x}$ is the vehicle speed, F_p is the propelling force, and F_r is the sum of resisting forces. The propelling force is the controlled input resulting from brake and driving actions.

The equation describing the wheel dynamics is:

$$I_z\dot{\omega} = T_d - F_xR - T_b \tag{8.20}$$

For longitudinal controller synthesis, a nonslip rolling is assumed. Then

$$v = R\omega; \quad F_p = F_x \tag{8.21}$$

So, the longitudinal dynamics is:

$$\left(m + \frac{\dot{I}_z}{R^2}\right)\dot{v} = \frac{T_d - T_b}{R} - F_r \tag{8.22}$$

A Lyapunov-based approach is frequently used to synthetize the longitudinal control. Consider the speed tracking error given by:

$$e = v_{\text{ref}} - v \tag{8.23}$$

where v and v_{ref} are the actual and reference speeds. The derivative of the error is:

$$\dot{e} = \dot{v}_{\text{ref}} - \dot{v} = \dot{v}_{\text{ref}} - \frac{1}{M_t}(T_d - (T_b + RF_r)) \tag{8.24}$$

where $M_t = \left(mR^2 + I_w\right)/R$, using the expression of $\dot{v}$ given by the nonlinear longitudinal model. Note that T_b can be considered zero, since that when throttle is active the brake is inactive.

As is known, in the Lyapunov methodology to ensure the convergence of the tracking error towards zero it is neccesary to propose a Lyapunov candidate function, which verifies two conditions: it must be definite positive and its derivative with respect to time must be negative.

Usually the following function is proposed:

$$V = \frac{1}{2}e^2 \tag{8.25}$$

Its time-derivative will be:

$$\dot{V} = e\dot{e} \tag{8.26}$$

To ensure the convergence to zero, the following condition is imposed, where $c > 0$:

$$\dot{V} = -cV \tag{8.27}$$

Substituting the value of $\dot{e}$, the following expression can be obtained for $\dot{V}$

$$\dot{V} = e\left(\dot{v}_{ref} - \frac{1}{M_t}(T_d - RF_r)\right) \tag{8.28}$$

Then the control law is:

$$\hat{T}_d = M_t(ce + \dot{v}_{ref}) + RF_r \tag{8.29}$$

with the parameter $c > 0$.

It is important to highlight that the stability condition assumes that the model matches with the real system. This is a very strong assumption. The uncertainties in the real parameters of the system must be considered in the controller synthesis. A robustification term must be added to the control law to ensure the robust convergence of the tracking error.

8.2.6.2 Lateral Motion Control

The lateral control problem is complex due to the longitudinal and lateral coupled dynamics as well as the tire behavior. These phenomena are well captured in a simplified way by the nonlinear bicycle model.

An algorithm chosen to perform the steering control tasks of the vehicle is fuzzy logic. Another algorithm frequently chosen to perform the steering control tasks of the vehicle is the predictive controller. When the reference trajectory is a priori known, a predictive algorithm has important advantages compared with other algorithms and is simpler to implement as PID controllers.

The precepts contained in the control strategies included under the term predictive control are:

- This kind of algorithm uses an explicit plant model that is able to predict the system output until a given time (prediction horizon).
- The future control signals obtained by the controller are calculated minimizing an objective function to a certain number of steps (control horizon).
- Sliding horizon concept. The prediction is carried out and the objective function is minimized in order to obtain input commands to the plant. The first control command obtained in the minimization is applied, discarding the rest, and slides the horizon into the future, repeating this steps in every sampling period.

The different predictive control algorithms differ in the models used to describe the system and in the cost function to be minimized. Fig. 8.11 shows the general structure of Model Predictive Controller. Many successful implementations of predictive controllers have been proposed in the literature. In particular, and for simplicity reasons we show the Dynamic Matrix Control (DMC) algorithm. The mathematical model used in this method to represent the system is the step response of the piecewise linearized system. The cost function used is intended to minimize future errors and control efforts. The name of the algorithm comes from the fact that the dynamic of the system is represented in a single matrix formed by the step response elements.

Mathematical expressions for the prediction and the cost function are:

$$\hat{y} = Gu + f \tag{8.30}$$

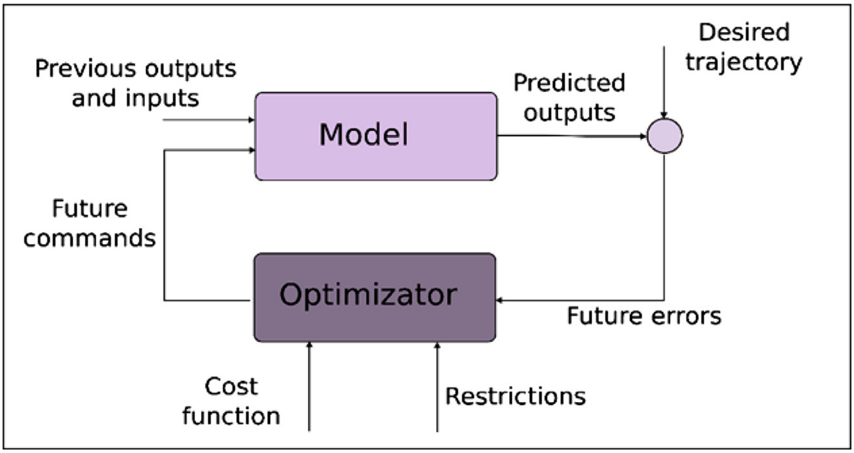

Figure 8.11 MPC structure.

with u, proportional to the lateral torque.

$$J = \sum_{j=1}^{p} \left[\hat{y}(t+j|t) - w(t+j)\right]^2 + \sum_{j=1}^{m} \lambda\left[\Delta u(t+j-1)\right]^2 \tag{8.31}$$

where ŷ is a vector with dimension equal to the prediction horizon containing the predicted outputs until the prediction horizon p, w is the future output desired value, u is a vector with dimension equal to the control horizon m containing the future control actions, G is the dynamic matrix control of the locally linearized system, and f is the vector of free response, with dimension equal to the prediction horizon. The free response is the prediction of how the system will behave if the command keeps constant and equal to the last command calculated. The λ parameter allows to carry out the weighing of the path tracking errors and the control efforts separately, in the way that we could design a controller to try to adjust to the desired trajectory, regardless of usage command, or on the other hand the controller could be more permissive with the path tracking errors and has a more soft use of command, saving energy in the control.

The main methodology of a predictive controller could be summarized as: the model of the process is used to predict the future outputs using the information of the past input signals, past control commands, as well as the future control actions calculated by an optimizer. To calculate the future control signals, the optimizer uses the cost function mentioned previously. Keeping in mind this explanation, the model process is fundamental to the correct functioning of the system.

Note that if in the optimization process it doesn't include the restrictions of the physical model, it is possible to obtain the minimization of the next cost function analytically:

$$J = ee^T + \lambda uu^T \tag{8.32}$$

where e is the vector of predicted errors until the prediction horizon and u is the vector of future signal control increments until the control horizon. The mathematical expression to calculate the future commands is obtained taking the derivative of J and equating to zero:

$$u = \left(G^T G + \lambda I\right)^{-1} G^T (w - f) \tag{8.33}$$

The optimizer will be able to calculate the steer angle in the way to minimize the differences between the free response and the desired trajectory. In other words, the optimizer will calculate the steer angle in order to produce the best path tracking.

The software reads the sensors and sets the values of the internal states of the system in each iteration. These states are the position, the orientation, and the velocity of the vehicle. With these values the step response of the system is calculated. The parameters of the step response form the dynamic matrix G. The prediction of vehicle behavior is calculated using the read values of the sensors at the beginning of the iteration. The prediction of vehicle movement is compared with the desired trajectory from the point closest to the prototype. The future errors vector is the result of the previous comparison. The future commands are obtained using the equation, but only the first term of future commands vector is applied, keeping in mind the concept of sliding horizon. Finally, the variables of the algorithm are updated.

8.3 COOPERATIVE AUTOMATED DRIVING

To tackle the current traffic congestion problems it is essential to improve road capacity and safety while reducing travel time. With a cooperative approach, individual vehicles relate to the environment communicating with other individual vehicles or road infrastructures. Indeed, using wireless communication, potential risk situations can be detected earlier to help avoiding crashes and more extensive information about other vehicles' motions can help to improve traffic throughput.

The extension of the commercially available Adaptive Cruise Control (ACC) system toward the Cooperative ACC (CACC) system has a high potential to be the first cooperative system to be deployed in the market. By introducing V2V communications, the vehicle gets information not only from its preceding vehicle—as occurs in ACC—but also from the vehicles in front of the preceding one.

This section pays particular attention, on the one hand, to platooning, based on the CACC concepts, and on the other hand, to the early-stage developments on Urban Road Transport.

8.3.1 Platooning

Platooning is a particular example of connected (and cooperative) adaptive cruise control (CACC), where a single driver may be in control of an entire "road train," potentially also including their lateral (lane changing) behavior. It combines the use of exteroceptive sensors (mainly 76 Ghz radars), the automation of pedals (or even steering wheel) with the use of a V2V secure wireless communication (using mostly 5.8 GHz DSRC) between the involved vehicles to synchronize braking and acceleration, obtaining a reaction time that is unattainable by humans.

Platoons, deeply investigated for freight transportation systems using heavy-duty trucks, are usually managed following a dynamic assignment for the interested vehicles before the trip. Then, a platoon formation stage leads to the nominal platoon operation mode. A machine state usually handles different events, such as an emergency break, the interference introduced by an intermediate vehicle, the special management of a motorway entrance or exit. The latest advance in truck platooning proposes the use of fault-tolerant systems (Companion, 2016) to perform an eventual recalculation of the assignment when significant deviations from the original plan are detected (Fig. 8.12).

Truck platooning is likely to be one of the earliest applications of road vehicle automation to be commercially viable. It is highly likely that it would mostly materialize on highways, where traffic is less turbulent than on city streets, with a deployment where automation would gradually increase, going from driver assistance up to highly (or even fully) automated vehicles. To make this happen, regulations governing how long

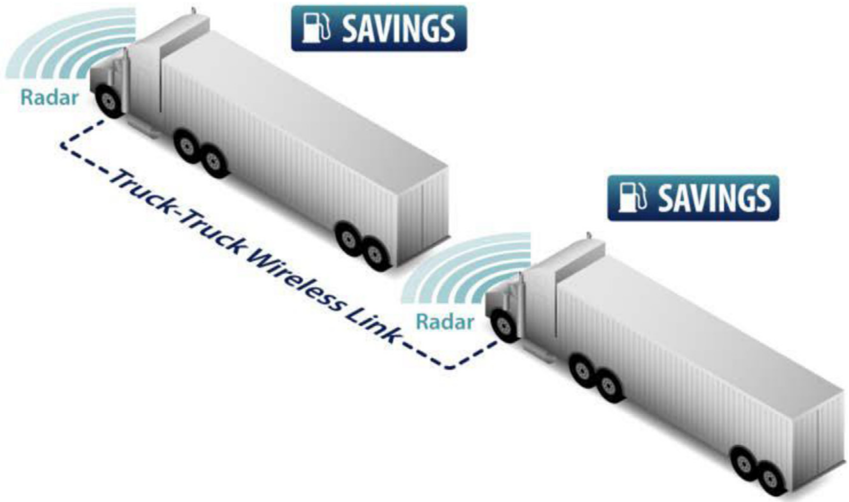

Figure 8.12 Platoon scheme (from Peloton Tecnology).

truckers can drive (or supervise) before taking breaks may have to be modified to consider situations where drivers are in the sleeper berth while an autonomous truck is in operation.

Although the different studies show certain variability in their conclusions, it is widely accepted that the deployment of platooning might have a positive impact on the following aspects:

Capacity: The accordion effect that generates traffic jams could be significantly reduced using constant spacing, increasing thus the capacity of roads by closer spacing of vehicles, narrower lanes, reduction in the wave effect of braking, faster average speeds, and fewer accidents (Swaroop and Hedrick, 1996; Rajamani, 2011). In Fig. 8.13A, a research work (Fernandes, 2012) shows how the maximal flow can be increased up to five times with respect to the peak of the flow/density curve in the classical traffic model. However, to be effective with respect to traffic flow, platooning should be performed with vehicles evolving on dedicated tracks and operating on a nonstop basis from origin to destination (Anderson, 2009). As such, by eliminating the stop-and-go problem of common car and transit systems, platooning could contribute to a faster and more comfortable mobility with higher energy efficiency.

Conversely, other research has suggested that cooperation may negatively impact capacity in merge or lane-drop situations, creating bottlenecks. In this connection, some regulatory bodies are likely to require

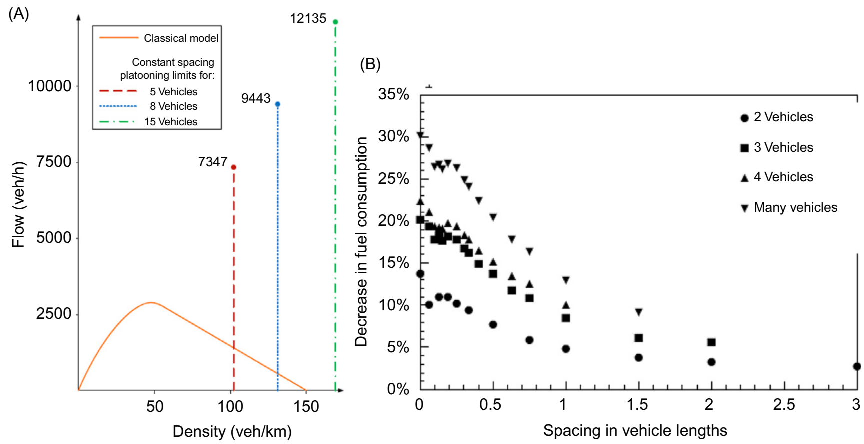

Figure 8.13 (A) Road capacity with different number of vehicle trains (Fernandes, 2012), (B) Decrease in fuel consumption (Zabat et al., 1995).

dedicated truck lanes. The main reason behind this idea is that even a three-truck platoon will function as a visual and physical barrier for cars needing to get on or off the road, and cars attempting to dart between trucks in a platoon represent a new safety hazard.

Fuel consumption: For heavy-duty trucks, the potential fuel savings obtained by platooning are particularly large, ranging from 4%—10% with conservative gaps (Al Alam et al., 2010; Janssen et al., 2015) up to over 20% when spacing is 1/10 vehicle length (see Fig. 8.13B). Thus, 2 trucks driving 100,000 miles annually can save €6,000 on fuel by platooning, compared to driving on cruise control.

Employment: In the long term, fully automated trucks may provide a solution to the growing driver shortage. The American Trucking Associations estimates that the industry will need more than 96,000 new drivers annually for the next 10 years to keep pace with current consumer spending rates (Costello and Suarez, 2015).

Vehicle platooning is an active research area and many contributions have been reported over the last decades. Early theoretical results on the control of platoons were presented in Levine and Athans (1966) and Melzer and Kuo (1971), focusing on a centralized optimal control scheme. Safety is typically addressed by the concept of string stability (Swaroop and Hedrick, 1996; Ploeg et al., 2014), which is related to the suppression of disturbances in vehicle position, velocity, or acceleration, as they propagate through the platoon. More recently, research on implementation aspects has been emerging, herein analyzing the aspects of heterogeneous vehicle strings (Shaw and Hedrick, 2007), intervehicular communication constraints (Al Alam et al., 2010), and implementation issues (Naus et al., 2010).

Although there are still some technological barriers to overcome (e.g., V2V safety and security or the stable platoon control under any circumstances), the main risks for the soft deployment of this type of cooperative systems come from the legal, business, deployment/timing, and user acceptance aspects. The interoperability between service providers, the absence of commitment and corresponding deficient market take up from stakeholders, or the potential boycott by driver-representation lobbies are some of the most significant risks in the exhaustive list of barriers and risks towards platooning (Janssen et al., 2015).

8.3.2 Urban Road Transport

One of the most difficult and challenging scenarios in implementing automated driving is the urban environment, since there are many complex and changing situations with different moving actors and infrastructure elements (vehicles, pedestrians, bikes, intersections and crossing areas, traffic lights, etc.) that must be taken into account at any moment. Considering this complexity, cooperative communication technologies have a very high potential to support and to enhance automated driving strategies, in its different automation levels, in a holistic approach that could cover vehicles, vulnerable road users, traffic infrastructure, shared digital data, and mobility management centers.

Through the exchange of cooperative information among all actors involved, it could be possible, for example, to extend the sensing capabilities of automated and autonomous vehicles beyond the perception of their own physical sensors. Moreover, urban traffic authorities can also enhance the information gathered at their mobility management centers with the one coming from automated vehicles that would be acting as moving sensors, increasing therefore its capacity to implement new traffic management strategies. In this connection, some interesting applications of cooperative urban road transport are listed in the next section.

Nevertheless, due to its specific high complexity, urban automation deserves still a big R&D joint effort among all involved stakeholders. Specific research in the domains of environmental sensing, Internet of Things, cloud computing, Big Data, or artificial intelligence will significantly contribute to the progress of automation in urban environments.

8.4 VERIFICATION AND VALIDATION

While Automated Driving is becoming nowadays a key topic for the future of the automotive industry, the technology behind it has been evolving since some years ago and it is reaching maturity in some of the Advanced Driver Assistance Systems (ADAS) that can be found right now in serial vehicles (e.g., ACC, AEB, or LKA).

In parallel to this evolution, the processes and procedures for testing ADAS functions have been also developed and established during the previous decades, according to the functions' requirements and the normative established. Therefore, today, it is possible to find standard procedures for testing such functions, e.g., Euro NCAP procedures (Fig. 8.14).

In the case of the Automated Driving functions, the work is still to be done. Several research projects in various stages of development can be found in this field, including extended on-road testing with vehicle fleets, but these demonstrations are only the beginning of the Verification and Validation steps, and the challenges are still not solved.

Among others, some of the main questions appearing that have to be clarified are the following:

- How Automated Driving functions should be tested (methods and tools) to achieve the levels of safety and confidence required?
- How much testing will be needed at each development stage?

Test approaches supporting massive and specific tests in the different technologies and at the different levels and points of the lifecycle are needed, and should cover concepts and algorithms, software units and physical components, integrated systems, and in-vehicle functions.

The V-Model development cycle (see Fig. 8.15) has been used in the development of vehicle functions already for some time. More recently, it has been adopted as the reference model that can be used for the ISO

Figure 8.14 AEB testing following EuroNCAP procedures.

Figure 8.15 Traditional V-Model development cycle.

26262, for functional safety critical systems. But, although ISO 26262 and the V-Model provide a generic methodological framework for assuring automotive safety and also a reference for development of automated driving functions, automated driving presents unique challenges for the application of both approaches, that are now under discussions in the different working groups and fora dealing with both topics.

With regards to the performance of tests at early phases, the use of modeling and simulations presents a clear advantage for achieving acceptable levels of safety and assurance for an autonomous vehicle. Virtual testing allows performing many test cases with variation of parameters in order to assure a correct performance of the algorithms, bringing the system into a good maturity level.

However, there is still a need for extensive testing of the overall system in the real environment, on first the proving ground and as a second step in real world campaigns. In this sense, the development of new specific proving grounds devoted to testing this kind of vehicle will have a major relevance in the coming years.

In-vehicle quantitative validation also requires a set of tools that allows to control a wide range of different parameters (related to the adjustment of the system to be tested, but also of the other targets in the scenario). On the other side, systems for repetitiveness of the maneuvers instead of using human drivers, in addition to the minimization of risks, allows being more efficient in the testing process. With this aim, several systems

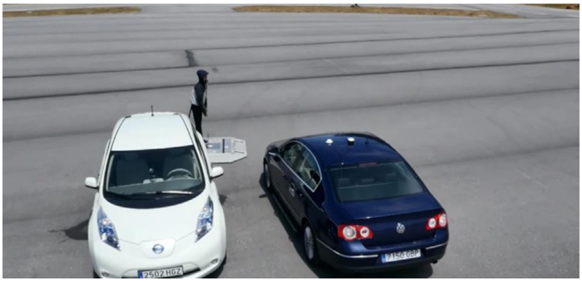

Figure 8.16 Example of new tools for automated driving testing.

have been proposed in the recent years to bring the vehicle under test in exactly the same test conditions while guaranteeing safety. In this case, it is possible to find in the market reference systems with enough precision, driving robots, or new movement platforms and dummies to support the new needs appearing in the market (Fig. 8.16).

The amount of testing needed for covering the needs of the Automated Driving functions is the other key topic, and subject to much debate. On the one hand, the discussion is focused on the number of kilometers that must be driven in order to establish a safe operation of an Automated Driving function, and, on the other hand, to arrive at a reasonable and efficient testing cost to meet the time to market in an affordable manner.

As a main conclusion, the verification and validation processes for the Automated Driving functions should meet, in an efficient way, the strong safety and acceptance objectives required to deploy the systems in millions of vehicles around the world and, in parallel, should be efficient enough in terms of costs and time to arrive to the market with the right product. Both facts suppose the modification of current frameworks and development processes, the increase on the usage of virtual testing and modeling, and finally, to be able to perform real environment tests in the most appropriate and repetitive conditions to evolve and fine-tune the correct system performances.

8.5 MAIN INITIATIVES AND APPLICATIONS

8.5.1 Prototypes

8.5.1.1 Relevant Prototypes at International Level

Besides some pioneer works at the University of Ohio in the 1970s, at the Carnegie Mellon University in the 1980s, in specific Programmes in

Europe and Japan (i.e., Prometheus and Advanced Safety Vehicles projects, respectively), the turning point of automated vehicles are the DARPA Grand and Urban Challenges (2005 and 2007).

Although more than 20 teams designed and built automated vehicles to meet the challenges posed by the organizers, the most successful prototypes were the following:

- *Stanley, Stanford University* (Thrun et al., 2006): it won the 2005 Grand Challenge. The vehicle is a Volkswagen Touareg, where the native drive-by-wire control system was adapted to be run directly from an onboard computer without the use of actuators or servo motors. It used five roof mounted Lidars to build a 3-D map of the environment, supplementing the position-sensing GPS system. As in many other prototypes, an internal guidance system utilizing gyroscopes and accelerometers monitored the orientation of the vehicle and also served to supplement GPS and other exteroceptive sensor data. Additional guidance data was provided by a video camera used to observe driving conditions out to 80 m (beyond the range of the LIDAR) and to ensure room enough for acceleration. Includes a Planning and Control layer with a local path planner, obstacle detection, health monitor module in the case of critical system or GPS failures, emergency stop remote control. The sequel of Stanley, Junior, obtained second place in the Urban Grand Challenge.
- *SandStorm, Carnegie Mellon University* (Buehler et al., 2007): the 2005 version of Sandstorm, mounted in a Hummer, used six fixed Lidars, a steerable LIDAR, and short- and long-range radar. It implements a global route planning, path planning, satellite images, and other topography data to generate the global path. The A* heuristic function together with cubic B-splines were used to smooth a reference path. Velocity and steering were controlled using classic PID controllers derived from Simulink models.
- *Boss, Carnegie Mellon University* (Urmson et al., 2008): winner of the Urban Grand Challenge. It is a Chevrolet Tahoe that uses perception, planning, and behavioral software to reason about traffic and take appropriate actions while proceeding safely to a destination. It is equipped with more than a dozen lasers, cameras, and radars to view the world. It allows the tracking of other vehicles, detecting static obstacles, and localizing itself relative to a road model. Planning system combines mission, behavioral, and motion planning to drive in urban environments

More recently, other relevant prototype examples from Universities have achieved very significant milestones:

- *The CMU autonomous vehicle research platform* (Wei et al., 2013): based on a Cadillac SRX. Since the car is not equipped with drive-by-wire controls for operation, several mechanical and electrical modifications were necessary to enable computer control of the required inputs that operate the vehicle. The vehicle is capable of a wide range of autonomous and intelligent behaviors, including smooth and comfortable trajectory generation and following; lane keeping and lane changing; intersection handling with or without V2I and V2V; and pedestrian, bicyclist, and work-zone detection. Safety and reliability features include a fault-tolerant computing system and smooth and intuitive autonomous-manual switching
- *Shelley, Stanford's self-driving Audi TTS* (Funke et al., 2012): it managed to autonomously ascend Pikes Peak in 2010. More recently, they took the vehicle to Thunderhill Raceway Park, and let it go on track without anyone inside, hitting over 120 miles per hour. The goal of this prototype was to push autonomous driving to the vehicle's handling limits. To that end, a high speed, consistent control signal is used in combination with numerous safety features capable of monitoring and stopping the vehicle. The high level controller uses a highly accurate differential GPS and known friction values to drive a precomputed path at the friction limits of the vehicle
- Vislab (Broggi et al., 2012), from the University of Parma, with which the International Autonomous Challenge was accomplished. The platforms are small and electric vehicles produced by Piaggio. The automated driving technology did not affect its performance since the sensors, the processing systems, and the actuation devices are all powered by solar energy, thus they do not drain anything from the original batteries. The vehicle managed to run almost 16.000 km on a 100-day trip, combining automated and manual mode in very challenging driving zones.
- Spirit of Berlin and Made in Germany, prototypes from the Freie Universität Berlin (Berlin, 2007). They have a modular sensor setup with most of its sensors mounted on top of the car on a flexible rack. Obstacle processing is done by a combination of rotating and fixed Lidars with stereo camera systems. In addition, the car localizes itself with an integrated GPS/INS unit and RTK correction signals.

In addition to these representative prototypes, many others have appeared in recent years from OEMS (Daimler, BMW, Audi, GM, Ford, Volvo, ...), Tier 1 providers (Bosch, Delphi, Valeo, Continental ...), and new incomers providing either the embedded intelligence or the whole automated vehicle (Waymo, Tesla, Peloton Technologies, EasyMile, Navya, Otto, Cruise, Zoox, Baidu, Aimotive). In addition, big players working on new solutions for the mobility as a Service paradigm are intensively working on highly/fully automated driving solutions (Lyft, Uber, Nutonomy, Didi Chuxing).

8.5.1.2 Relevant Prototypes in Spain

There are some R&D centers and universities in Spain working since several years intensively in the domain of automated driving, such as CSIC, UPCT, CTAG, and INSIA. Some of them have prepared real functional automated driving prototypes to test the progress of their different developments.

In 2012 the AUTOPIA Program, from CSIC, showcased their technology in communications and control performing a widely publicized demonstration, from El Escorial up to Arganda, in Madrid (see Fig. 8.17A). One automated prototype vehicle (Platero, Citroën C3) ran driverless for 100 km following a leading manual car (Clavileño, Citroën C3 Pluriel), with sensing and communicating devices, which dynamically generated a high precision map to be tracked by the automated following car. The journey covered a wide range of driving scenarios, including urban zones, secondary roads and highways, in standard traffic conditions. To that end, V2X communications were combined with onboard sensors to achieve a centimetric localization and a safe and smooth motion planning (Godoy et al., 2015).

Probably the most relevant activity performed so far in Spain was carried out by PSA, in collaboration with CTAG, and took place in

(A) (B)

Figure 8.17 (A) Platero in the El Escorial demonstration, (B) PSA autonomous trial Vigo-Madrid.

November 2015. A level 2 and level 3 PSA C4 Picasso prototype (see Fig. 8.17B), equipped with different sensors and enriched digital maps, covered in automated mode the distance of 599 km from Vigo to Madrid, showing the feasibility to deploy automated driving vehicles in the coming years. DGT was also deeply involved in this trial, facilitating all required authorizations to perform this test in open roads.

Other relevant prototypes have been built in Spain, such as the Renault Twizy automated by the Technical University of Cartagena (Navarro et al., 2016). It has a large set of exteroceptive sensors connected to three different computing platforms running on VxWorks. It includes a global route planning based on maps, a machine learning (SVM) for detecting pedestrian and vehicles, local path planning, and an obstacle avoiding system based on Bezier curves trajectories.

CTAG has also developed some automated driving prototype vehicles in the framework of European, national, or bilateral cooperative R&D projects. To mention two interesting examples (see Fig. 8.18), it is possible to mention a prototype developed in the framework of the project $CO^2PERAUTOS^2$ (INNTERCONECTA Spanish R&D Program), implementing some cooperative automated driving functions such as cooperative highway chauffeur and cooperative urban chauffeur, including also some cooperative sensing use cases (Sánchez et al., 2015). A second relevant example, in this case in cooperation with PSA, is the MobilLab automated driving prototype, devoted to explore the challenges of HMI and driver-interaction for automated and autonomous vehicles.

INSIA (University Institute for Automobile Research of Technical University of Madrid) is also involved in several projects regarding intelligent vehicles, specifically in automated and connected experiences, including cooperative systems, like AUTOCITS (Regulation Study in the Adoption of the autonomous driving in the European Urban Nodes

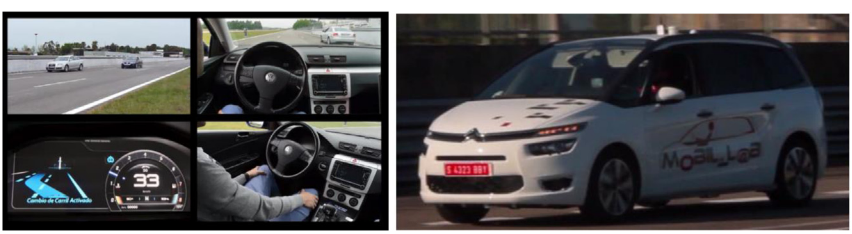

Figure 8.18 Examples of CTAG prototypes: Cooperative Automated Driving prototype, Mobil-Lab HMI prototype.

Figure 8.19 INSIA's autonomous vehicles prototypes.

funded by European Commission), focused on the deployment of pilots of Cooperative Systems and Autonomous Vehicles in the cities of Madrid, Paris, and Lisbon. The first projects on autonomous driving began with SAMPLER and ADAS-ROAD projects in which automatic evasive maneuvers were performed when risky situations were detected. Then, the Cooperative and Autonomous Vehicles (CAV) project was focused on the integration of autonomous vehicles with C-ITS in critical environments such as complex crossings, roundabouts, and tunnels. AUTOMOST (Automated Driving for dual transport systems) project is aimed to the development of autonomous city buses. INSIA also have strong links with the Spanish automotive sector and with other important actors such as the Spanish Ministry of Defence with the project REMOTE-DRIVE (drive-by-wire for tactic vehicles in surveillance missions) where a military tactic vehicle has been automated to act in emergency and defence missions. This center includes a testbed private circuit, two fully automated vehicles (see Fig. 8.19), two instrumented vehicles, one instrumented motorcycle, electronics and instrumentation lab, V2X proprietary communication technology, and a granted patent of a device to automatically control the steering of a vehicle from a computer.

8.5.2 Projects

Automated and connected driving has become one of the technological mega-trends, recognized by several reports of consulting firms (Manyika et al., 2013). In addition to that, strategic roadmaps from different international organizations merely confirm the importance of these technologies:

- The Amsterdam Declaration (European Union, 2016), signed on April 2016 by all 28 EU member states during the informal meeting of the Transport Council, lays down agreements on the steps necessary for the development of self-driving technology in the EU. In this document the EU member states and the transport industry pledge to draw up rules and regulations that will allow autonomous vehicles to be used on the roads.

- La Nouvelle France Industrielle (NFI, 2013) is a French strategic document whose aim is to focus economic and industrial stakeholders around common goals, to align government means more effectively to these goals, and to harness local ecosystems to build a new, competitive French industrial. It has borne fruit with the presentation of 34 industrial renewal initiatives, among which driverless vehicles, pushing the French automotive sector to be a pioneer in vehicle automation, notably by removing regulatory barriers to growth
- The German Association of the Automotive Industry (VDA) proves that the adaptation to legal provisions and parameters is required, since the corresponding regulations always assume that a driver is actively steering and controlling the vehicle at all times. However, this is not the case in higher automation levels. Their position paper (der Automobilindustrie eV, 2015) on automated paper argues why The Vienna Convention of 1968 must be amended accordingly in order to create a basis for compliance with the national road traffic regulations of the respective signatories.
- NHTSA (National Highway Traffic Safety Administration) released a policy document for Automated Vehicles in 2016 (U.S. Department of Transportation, 2016), where it recognizes three realities that necessitate some sort of guidance: (1) the rise of new technology is inevitable; (2) more significant safety improvements will be achieved by establishing an approach that translates knowledge and aspirations into early guidance; (3) as this area evolves, the "unknowns" of today will become "knowns" tomorrow. The overall intention is therefore to establish a foundation and a framework upon which future Agency action will occur.

Supported by this trend and guidelines, the European Commission and public authorities of the EU Members States have already funded an important number of research and innovation projects (see Fig. 8.20) that seek to set the basis for a sustainable and competitive development of automated driving technologies in Europe.

The first attempts in autonomous driving permitted to see that taking the driver out of the loop in the evolution process from automated to autonomous driving will not happen easily and probably will not happen

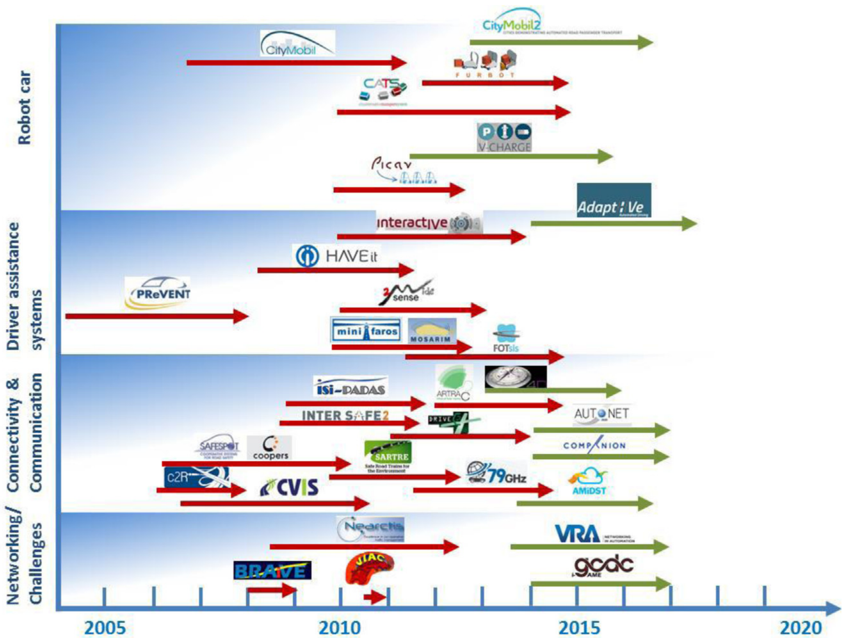

Figure 8.20 Overview of EU-funded Project on automated and connected driving (Dokic et al., 2015).

at all. As a result of this, some initiatives guided his steps through highly automated vehicles. Among them, the project HAVEit, Highly Automated Vehicles for Intelligent Transport (2008–11), aimed at a higher level of automation to be used on existing public roads in mixed traffic.

Following this path, the project interactIVe (2010–13) involved 29 entities from 10 countries working together towards the increase of an accident-free traffic in Europe and developed advanced assistance systems for safer and more efficient driving. The driver is continuously supported by these systems, that warn the driver in potentially dangerous situations. The systems do not only react to driving situations, but are also able to actively intervene in order to protect occupants and vulnerable road users. The continuation of interactIVe was AdaptIVe, an European project launched in 2014 that is about to conclude. It is a lighthouse project on automated driving, involving most of the major OEMs and Tier ones. Its main goal is to demonstrate automated driving in complex traffic environments, taking into account the full range of automation levels. In addition, it is working on (1) providing guidelines for the implementation of cooperative controls involving both drivers and automation; (2) defining and validating specific evaluation methodologies; (3) assessing the impact of automated driving on European road transport; and (4) evaluating the legal framework with regards to existing implementation barriers.

Another example for the successful integration of driverless intelligent vehicles (level 5) in urban environments is the project of CityMobil2. As a successor of CityMobil, the project is implementing intelligent transportation systems (ITS) for automated transport in protected environments. The applied vehicles are based on the "CyberCars" concept defined and promoted by INRIA in France, but extended to new marketable platforms (Easymile, Navya).

As already mentioned, cooperative driving addresses automotive and road traffic systems that make use of information and communication technologies (ICT), in conjunction with automated or nonautomated driving vehicles. These technologies are used to exchange specific information between vehicles (vehicle-to-vehicle communication, or V2V) and between vehicles and road infrastructure (V2I).

In the last years, several European projects and initiatives have marked important milestones in the deployment of these technologies. SAFESPOT (2006–10) developed dynamic cooperative networks where the vehicles and the road infrastructure communicated to share information gathered onboard and at the roadside, to enhance the driver's perception of the vehicle surroundings. Intersafe 2 (2008–11) aimed to develop and demonstrate a cooperative Intersection safety system able to significantly reduce injury and fatal accidents at intersections. The Grand Cooperative Driving Challenge (GCDC, 2011), and its sequel i-Game (2013–16), aims at accelerating the development and implementation of cooperative driving technologies, by means of a competition between international teams.

Several experimental case studies have shown the feasibility of platooning. One of the earliest demonstrations was given in the US in 1997 by the PATH project at UC Berkeley using a platoon of eight cars, followed then in the 2000s by experiments with trucks that had only automated longitudinal control (Fig 8.21A). Regarding truck automation, the first studies were "Chauffeur" within the EU project T-TAP from the mid-1990s to the beginning of 2000 (Fritz et al., 2004), where driving experiments were conducted with three heavy trucks along the Brenner Pass through the Alps between Austria and Italy.

Figure 8.21 From top left to bottom right: (A) PATH Program (PATH, 2017); (B) Konvoi Project (Kunze et al., 2009); (C) SARTRE Project (Sartre, 2016); (D) European Truck Platooning Challenge (Eckhardt, 2015); (E, F) COMPANION Project (Companion, 2016).

From 2005 to 2009 the Aachen University developed a platoon of four heavy trucks (Fig. 8.21B) in their project KONVOI (Kunze et al., 2009) with the objective of increasing transportation capacity while reducing fuel consumption. In 2008 Japan started a 5-year project "Energy ITS" aiming at reducing energy consumption by truck platooning (Tsugawa, 2014).

The most recent European projects SARTRE (Sartre, 2016) and COMPANION (Companion, 2016) have showed significant advances in platooning. The former (Fig. 8.21C) developed strategies and technologies to allow heterogeneous vehicle platoons (car and trucks) to operate on normal public highways, showing its potential with a five vehicle demonstration. The latter (Fig. 8.21E,F) provided solutions for handling the lack of holistic solutions including the creation, coordination, and operation of platoons. As a result, a real-time coordination system can dynamically create, maintain, and dissolve platoons, according to a decision-making mechanism, taking into account historical and real-time information about the state of the infrastructure. More recently, in 2016 about a dozen trucks from major European manufacturers completed a week of largely autonomous driving across Europe in the European Truck Platooning Challenge (Eckhardt, 2015), the first major exercise involving multibrand platooning on the continent (see Fig. 8.21D). In parallel in the USA, Peloton Technologies is the first company to provide automated vehicle technology for commercial truck platooning.

Another interesting and completely different project on cooperative and automated vehicle technology is AutoNet2030. It has worked towards a decentralized decision-making strategy which is enabled by mutual information sharing among nearby vehicles. It considers the gradual introduction of fully automated driving systems, which makes the best use of the widespread existence of cooperative systems in the near-term and makes the deployment of fully automated driving systems beneficial for all drivers already from its initial stages.

8.5.3 Special Applications

The potential of autonomous driving goes beyond standard road applications. There are several areas where the use of this technology can be applied. Researchers, scientists, universities, and R&D departments of the best automotive companies have explored this idea and other possible special applications based on this type of vehicles have been implemented along years. In this way, there are specific applications for off-road environments, nonurbanized scenes, where the useful information for the interpretation of the surroundings is much more limited: military missions, rescue, supervision and surveillance, land exploration, agricultural applications, among others. Furthermore, an additional field of broad development is the specific applications for public passenger transportation, taxi services, car-sharing, or for freight transport and, in a general way, any application that either cannot be performed by a human operator due to imminent danger exposure or provides a new solution for a specific service.

The military environment is one of the main promoters of this type of applied technology. The first military applications were carried out in 1930 with the so-called "Nagayama tank," a tank that received the orders of movement via radio. Subsequently, and more famously, the Soviet Union developed one during the late 1930s and early 1940s, the so-called "Teletank." These military platforms used in the well-known "Winter War," at the beginning of World War II, consisted of tanks controlled by radio without any operator, carrying out missions to approach the enemy while they were guided from another tank at a safe distance. Another military platform that became popular, not for the technology used, but for laying the groundwork for post-World War developments in tele-operation technologies, was "The Goliath," a small-tracked tank, used by the German army during World War II, in battles such as the Normandy landings. "The Goliath" was controlled remotely by wires up to 650 meters and its mission was to demolish buildings and infrastructures of the enemy through the explosive charge that was carried.

More recently, the American organization DARPA, founded the "DARPA Grand Challenge," an autonomous vehicles' competition where they had to travel long distances in a totally autonomous mode. In first instance, it was intended to promote driving and vehicles with a high level of automation, with new developments in technology, but the final goal had a strong military character. In 2004, the first edition of this competition was celebrated, in which vehicles had to complete a route of 240 km along an off-road environment. No participant completed the route. In 2005, the second edition was held, where five vehicles successfully completed the test. This edition continued to consist of an off-road environment, where the participating vehicles had to go through different complex situations like narrow tunnels, roads where it is difficult to delimit the boundaries, or perform complicated turns. The third edition of the DARPA Grand Challenge took place in 2007, better known as "Urban Grand Challenge"; in this case, the environment was urban and 96 km long, respecting traffic rules and the other vehicles.

Since then, different armies around the world have been developing their own Unmanned Ground Vehicles (UGV's). For example, UGVs have been developed such as the "Guardium-LS UGV," a military platform used by Israel equipped with a large amount of ballistic material, capable of being tele-operated from another mobile platform and able to detect and avoid obstacles that appear in their way. Another interesting example is the Tank Automotive Research Development and Engineering Center (TARDEC) group of U.S army that develops UGVs for military applications whose purpose is to be able to be tele-operated from anywhere in the world. On the other hand, in this type of application, unlike when urban environment are considered, it is difficult to predict the state of the road through which the vehicle circulates, besides not being able to rely on elements that serve as reference, such as lane lines, curbs, buildings, crossings, etc. With the LiDAR technology, obstacles that are above the ground level could be easily recognized; however, in these off-road conditions it cannot be assumed that the condition of the terrain is always in good condition and the identification of negative obstacles is of great importance. That is why research projects are being developed especially dedicated to the identification of such obstacles, such as the case of Shang et al. (2015), where a different set up for the LiDAR sensors is presented in order to identify the negative obstacles in nonurbanized environments. They were the winners of the "Overcome Danger 2014," a ground vehicle challenge supported by the Chinese army, similar to the DARPA Grand Challenge of the USA.

Another interesting application of autonomous vehicles is their use for space missions. For example, the vehicle Mars Rover can be guided by a human operator remotely but the obstacles avoidance and best trajectory finding for reaching the destination are tasks that the vehicle performs autonomously.

Apart from these projects supported by governments or state administrations, some companies from the private sector are also paying special attention to developing new solutions for special off-road applications such as Jaguar Land Rover. Throughout 2016, they have been working on a self-driving off-road connected vehicle. On the one hand, the off-road vehicle can identify the terrain where it circulates based on recognition of the environment and it can offer semi-autonomous driving. On the other hand, it is intended that these vehicles can talk to each other and communicate, in such a way that, if several vehicles circulate in a convoy, the first one can communicate to the other vehicles located behind it the state of the terrain, the speed, and location, etc., through DSRC communication modules.

In the agricultural sector, the autonomous navigation of the industrial machinery is promoting a great increase of the productivity in tasks of plowing, mowing, harvesting, etc. All these tasks become more efficient when a fleet of tractors works at the same time throughout the day and the operator manages all the work in remote. There are several developments in this sector. One of the most outstanding is the manufacturer of agricultural machinery Case IH with its concept of autonomous vehicles, capable of following a preloaded route and calculating the most optimal paths to perform. Furthermore, since the vehicle has sensors for the recognition of the environment, it can identify obstacles and make the decision to stop and send a warning signal if necessary. In a similar line, other agriculture machinery manufacturers have developed vehicles that can follow a predefined trajectory.

In addition, autonomous navigation serves as a catalyst for other services that are carried out in an urban or interurban scene. One of the most successful special applications among the manufacturers and software companies is the taxi service. In fact, the first self-driving taxi was tested in Singapore in 2016 by nuTonomy. This software startup for autonomous vehicles has launched a taxi service that currently operates in a specific area of Singapore with specific destinations, using electric cars, providing, therefore, a solution to decongesting the cities of traffic and pollution. On the other hand, Uber is currently developing its autonomous taxi service, conducting tests in the city of Pittsburgh, through the sensor fusion of different LiDARs, Radars, stereovision, and computer vision for the recognition of the environment and making use of machine learning in their control algorithms.

Besides the specific aforementioned initiative, there are many potential interesting cooperative applications related to urban automation. To summarize some of them, the following examples could be found in the future:

- Automated parking and valet parking cooperative services.
- Cooperative services to support and to manage autonomous vehicles in dedicated lanes, such as last-mile autonomous vehicles.
- High precision positioning cooperative services.
- Cooperative services for urban chauffeur and urban autopilot, for example, to manage intersection scenarios in a safely and efficient manner.
- Cooperative services for robotaxis and autonomous car sharing fleets.
- Cooperative sensing services to extend perception capabilities of AVs.

Parking scenarios usually represent a kind of relatively stable - environment where vehicle automation is already happening. With the support of cooperative communication technologies, the next step will be in the direction of autonomous valet parking applications. In this case, cars and garages will cooperate to park autonomously the vehicle, without the presence of the driver, which will interact with the car through his/her smart device. All vehicles manufactures and main Tier 1 suppliers in this domain are in a development phase to bring this functionality into the

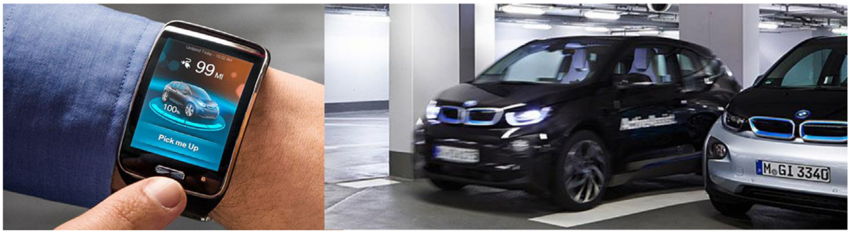

Figure 8.22 Remote valet parking assistance demonstration (BMW).

Figure 8.23 Last-mile driverless shuttles testing trial that will run from 23rd January 2017 until 7th April 2017 in a dedicated lane in Paris (Easymile).

market, with the involvement also of parking management companies (Fig. 8.22).

Other important area, very much related to connectivity and cooperative services, which have an interesting potential to make urban mobility more sustainable, is linked to the implementation of electric autonomous last-mile vehicles. There are many R&D projects and predeployment activities running at this moment to support the future introduction of these kind of solutions (Fig. 8.23).

Talking about urban scenarios that automated vehicles will have to cover, intersections are very complex situations where cooperative services can really provide support. As an example, cooperative traffic lights can exchange Green Light Optimal Speed Advisor (GLOSA) information with automated vehicles to let them adapt their speed to pass the intersection in the safest and most efficient manner (Fig. 8.24).

Moreover, V2V, V2I, V2VRU, or V2Cloud cooperative sensing strategies can enhance the perception capabilities of automated vehicles, allowing them to manage better these complex urban scenarios (Fig. 8.25).

Figure 8.24 Example of automated GLOSA cooperative service (CTAG).

Figure 8.25 C2C pedestrian detection demonstration at Bordeaux ITS World Congress (PSA-CTAG).

8.6 SOCIOREGULATORY ASPECTS

8.6.1 Legal Pathways

There are important questions to be answered regarding autonomous vehicles, like security, ethics, use of data, and coexistence of autonomous technology with conventional vehicles (manually controlled vehicles).

Firstly, the system must be safe for both the driver and the other users of public roads. In addition, autonomous vehicles must comply with the traffic laws of the region in which they operate, in the same way as all other vehicles in circulation.

Secondly, it is necessary that the regulation on autonomous vehicles is as homogeneous as possible. So that there should not be differences between states that prevent the same model of vehicle from operating in all of them (which would be an important obstacle to the deployment of this technology).

Then, appropriate education and training are essential to ensure the safe deployment of automated vehicles. Therefore, manufacturers and other entities should develop, document, and maintain some form of training for employees, distributors, and users. The differences between the use and operation of autonomous vehicles and conventional vehicles should be addressed. In addition, these programs should be designed to provide users with the level of understanding necessary to use them properly and safely.

A critical aspect in this type of technology is the changing between manual control and autonomous vehicle control. Adequate mechanisms and procedures must be provided to ensure the change is made safely, comfortably, and efficiently.

In addition, the data generated by the use of connected and/or autonomous vehicles may be useful, for example in case of accidents, to analyze the causes of them. It is necessary to clarify the conditions and availability for the use and exchange of data generated by connected and automated vehicles, as well as the responsibility of each of the parties involved.

Moreover, manufacturers and other entities are expected to develop software upgrades for automated vehicles or new vehicle versions incorporating different and/or upgraded hardware. If these software or hardware upgrades substantially change the operation of the vehicle, an evaluation or additional certification process may be necessary. The purpose of these updates may be to improve performance, security, or other aspects of the system. In addition, in case of changes in the software, the download of these updates or patches could be done through "over-the-air updates" or other methods that should be regulated.

Finally, it is necessary to contemplate the possibility of when two situations of risk happen simultaneously, requiring a consideration of "dilemma situations." It may be necessary to define the procedure in these kinds of situations.

8.6.1.1 General Framework: Vienna and Amsterdam

There are two traditional international framework agreements on Road Traffic: the Geneva Convention and the Vienna Convention. Firstly, the Geneva Convention on Road Traffic was signed in 1949 by 95 states to promote the development and safety of international road traffic by establishing certain uniform rules. Secondly, the Vienna Convention on Road Traffic from 1968 is an international treaty ratified by 74 countries. It was designed to facilitate international road traffic and to increase road safety through the adoption of uniform traffic rules. At European level, the need to modify the Vienna Convention to promote the use of autonomous vehicles in road traffic has intensified, mainly due to the need of a "driver" controlling the vehicle at all time. Article 1, paragraph (v): "Driver means any person who drives a motor vehicle or other vehicle (including a cycle), or who guides cattle, singly or in herds, or flocks, or draught, pack or saddle animals on a road." Article 8, paragraph 5: "Every driver shall at all times be able to control his vehicle or to guide his animals."

Then, on April 2016, European Union transport ministers, as well as a number of car manufacturers, signed the European Declaration of Amsterdam. The main objective of this initiative is the cooperation between governments and industry to develop a legal framework and to boost research and development on connected and automated driving.

In addition, there are currently informal discussions on the analysis and development of regulations for the autonomous vehicles at the United Nations Economic Commission for Europe's "Working Party on Road Traffic (WP.1)."

8.6.1.2 Legal Framework and Regulation About Autonomous Vehicles

This section summarizes the state of the regulation of autonomous vehicle driving in early 2016. Initiatives in USA and Europe are mainly cited, but others are also discussed.

The US Federal Government released in September 2016 an autonomous vehicle policy designed to help the safe development of driverless technology, while also allowing enough flexibility so development of the technology can continue.

In addition, in the United States the number of states working on legislation related to autonomous vehicles has gradually increased. The enacted autonomous vehicles legislations in the USA are listed below.

Nevada was the first state to authorize the operation of autonomous vehicles in 2011. AB 511 authorizes operation of autonomous vehicles it also defines "autonomous vehicle" and directs state Department of Motor Vehicles to adopt rules for license endorsement and for operation (including insurance, safety standards, and testing). SB 140 permits use of cell phones or other handheld wireless communications devices for persons in a legally operating autonomous vehicle (these persons are deemed not to be operating the vehicle for the purposes of this law). SB 313 requires certain conditions that human operators and autonomous vehicles must meet in order to being registered, or tested or operated on a highway within the state.

California, in 2012 (SB 1298), permits the operation and testing of autonomous vehicles pending the adoption of safety standards and performance requirements that would be adopted under this bill. In 2016 (AB 1592) California authorized the Contra Costa Transportation Authority to conduct a pilot project (only at specified locations and speeds) "for the testing of autonomous vehicles that do not have a driver seated in the driver's seat and are not equipped with a steering wheel, a brake pedal, or an accelerator."

Florida declared in 2012 (HB 1207 and HB 599) "desires to encourage the current and future development, testing, and operation of autonomous vehicles on the public roads of the state" and found that it "presently does not prohibit or specifically regulate the operation of autonomous vehicles." In 2016 (HB 7027) legislation expands the allowed operation of autonomous vehicles on public roads and eliminates the requirement that the vehicle operation is being done for testing purposes and the presence of a driver in the vehicle.

Florida House Bill 7061 (2016) defined driver-assistive truck platooning technology and required a study on the use and safe operation of this kind of technology and allows for a pilot project upon conclusion of the study.

Through House Bill 1143 (2016), Louisiana defined "autonomous technology" for purposes of highway regulatory provisions and related matters.

In the Michigan Senate Bills 169 and 663 (2013), issues like "automated technology," "automated vehicle" and "automated mode" were defined. Automated vehicles were allowed to be tested by certain parties under certain conditions. By 2016 Michigan Senate Bills 995, 996, 997, and 998, modified aspects such provide immunity from liability that arises out of any modification made by another person without the autonomous technology manufacturer's consent.

Through House Bill 1065 (2015), North Dakota provided for a legislative management study of automated motor vehicles. The study might include research into the degree that automated motor vehicles could reduce traffic fatalities, crashes and congestion.

In 2015 Tennessee prohibited (SB 598) local governments from banning the use of motor vehicles equipped with autonomous technology if the motor vehicle otherwise complies with all safety regulations. SB 2333 (2016) allows an operator to use an electronic display (integrated with the vehicle) for communication, information, and other uses enabled by the display only if the autonomous technology isn't disengaged. SB 1561 (2016) establishes certification program through department of safety for manufacturers of autonomous vehicles before such vehicles may be tested, operated, or sold; creates a per mile tax structure for autonomous vehicles.

The Utah House Bill 373 (2015), modified the Motor Vehicles Act by authorizing the Department of Transportation to deploy connected vehicle technology tests.

The HB 280 (2016) requires a study related to autonomous vehicles, including evaluation of the different standards, best practices, regulatory strategies, and schemes implemented by other states.

Washington, D.C. through DC B 19−0931 (2013) defined "autonomous vehicle" as a "vehicle capable of navigating District roadways and interpreting traffic control devices without a driver actively operating any of the vehicle's systems." It authorizes autonomous vehicles to operate on public roadways if a driver can assume control of the autonomous vehicle at any time.

At European level, there are different countries that have carried out initiatives for the development of regulation related to autonomous driving. The most significant, are listed below.

The Ministry of Transport and Communications of Finland is preparing an amendment to the Road Traffic Act that would allow for driverless robotic cars to drive within a restricted area on public roads. The act in question would constitute experimental legislation that would be in force for five years starting at the beginning of 2015.

France (*L'Etat)* announced in July 2014 that the necessary regulations to guarantee road safety in the first experiments of autonomous vehicles on public roads should be developed. On August 3, 2016, the *Conseil des ministres* of France announced an Ordinance (Ordonnance: *experimentation de vehicules a delegation de conduite sur les voies publiques*), which allows the deployment of (partially or completely) autonomous vehicles tests, but only if safety is ensured.

Germany does not have specific (ad hoc) legislation on autonomous vehicles due to the strict interpretation of the Vienna Convention followed in Germany. In addition, Germany's Minister of Transport has announced that a section of the A9 autobahn that connects Berlin and Munich is to be set up for testing autonomous vehicles and connected vehicles (V2V and V2I).

In the Netherlands, the Ministry of Infrastructure and the Environment amended the Dutch regulation to allow large-scale road tests. Companies that wish to test autonomous vehicles must submit an application for admission to the RDW (Dutch Vehicle Authority) and demonstrate that the tests will be conducted in a safe manner.

In Spain, by means of instruction 15/V-113 of November 2015, the regulation for the authorization of tests with autonomous vehicles on open roads to traffic in general was published. In addition, in January 2016, the regulation on assisted parking of motor vehicles (INSTRUCTION 16 TV/89) was made public.

Sweden started in 2014 a project (Drive Me) which has given Volvo permission to test 100 autonomous vehicles in the city of Gothenburg by 2017−2018. It will be the world's first large-scale autonomous driving project. This initiative, which is the result of the collaboration between Volvo, the Swedish Department of Transport, the Swedish Transport Agency, the Lindholmen's Science Park, and the city of Göteborg, is supported by the Swedish Government.

UK Department for Transport released in February 2015 a regulatory review. Testing automated vehicles is allowed on any road in the UK without needing to seek permission from a network operator, report any data to a central authority, or put up a surety bond. In July 2015 a Code of Practice for testing autonomous vehicles was published. It explains to testers how to comply with the UK laws: "testers must obey all relevant road traffic laws; test vehicles must be roadworthy; a suitably trained driver or operator must be ready, able, and willing to take control if necessary; and appropriate insurance must be in place."

Finally, some of the initiatives developed in other countries are indicated. In Australia, there is an initiative called Australian Driverless Vehicle Initiative (ADVI) that includes different companies, government bodies, and research centers. The main objective of this initiative is to "build momentum by rapidly exploring the impacts and requirements of this new technology in a truly Australian context and making recommendations on ways to safely and successfully bring self-driving vehicles to Australian roads."

There are several initiatives developed in Singapore, related to the autonomous car. In 2014, the Committee for Autonomous Road Transport for Singapore (CARTS) was launched by the Singaporean Ministry of Transport. One of the tasks of this team is to investigate and create a framework for autonomous vehicles to work safely and efficiently on public roads. Besides this, the Land Transport Authority (LTA) jointly Agency for Science, Technology and Research (A*STAR) announced Singapore Autonomous Vehicle Initiative (SAVI), which one of their focus areas is to prepare technical and statutory requirements for future deployment of autonomous vehicles in Singapore. In addition, in August 2016, Singapore's government gave NuTonomy permission to test self-driving Taxis in a business park called "one-north." The tests began in the third quarter of 2016.

8.6.2 Ethical Aspects

Automated vehicles will be able to make precrash decisions, overcoming thus many of the limitations experienced by humans. However, there will be fatal car crashes that are unavoidable.

In these situations, a computer can quickly compute the best way to crash taking into consideration the likelihood of the outcome. One major disadvantage of automated vehicles is that, unlike a human driver who can decide how to crash in real time, an automated vehicle's decision of how to crash is a priori designed by a programmer ahead of time (Goodall, 2014). And there are many challenging driving situations where a dilemma may appear, requiring actions that are legally and ethically acceptable to humans.

To illustrate this complexity, consider for instance a sort of modified trolley problem, called the tunnel problem (Open Roboethics Initiative, 2016). A self-driving car just before entering a tunnel encounters a child that attempts to run across the road, but trips in the center of the lane, effectively blocking the entrance to the tunnel. The car has but two options: hit and kill the child, or swerve into the wall on either side of the tunnel, thus killing you. How should the car react? Or even more subtle. An autonomous car is facing an imminent crash, but it could select one of two targets to swerve into: either a motorcyclist who is wearing a helmet, or a motorcyclist who is not. What is the right way to program the car?.

Both outcomes will certainly result in harm as there is no obvious "correct" answer to these kind of dilemmas. If crash-optimization is considered as the most relevant criterion, the outcome may result in unfair actions, as the most responsible potential victim would be penalized, somehow awarding careless road actors and stakeholders.

An alternative and apparently elegant solution would be not to make a deliberate choice. However, such a random decision mimics human driving, which is completely against one of the key reasons to deploy autonomous cars: to avoid the human factor, responsible for 95% of accidents. Even worse, while human drivers may be forgiven for making a poor split-second reaction, robot cars will not enjoy that freedom, as such an action might be the difference between premeditated murder and involuntary manslaughter.

Some others argue that instead of assuming designers are the right people to decide in all circumstances how a driverless car should react, alternative methodologies may be explored allowing drivers to decide on the preferred unfortunate outcome.

In any case, to optimize crashes, designers/programmers would need to face an ethics problem, as they will need to design optimization algorithms that calculate the expected costs of various possible options, selecting the one with the lowest cost. It is therefore legitimate to ask the question of whether control algorithms of automated vehicles can be designed a priori to embody not only the laws but also the ethical principles of the society in which they operate (OCR Software blog, 2016). The basis for such complex decision system could be inspired from Isaac Asimov's three laws of robotics: (1) property damage takes always precedence of personal injury; (2) there must be no classification of people, for example, on size, age, and the like; (3) if something happens, the manufacturer is liable (Gerdes and Thornton, 2016).

Another very relevant ethical aspect derived from the pervasive connectivity of the new generation of vehicles is the challenge to preserve data security, and more in particular, privacy. Indeed, all the data that circulates within the transportation system will be subject to an intense reflection in order to regulate the data to be collected, owned, and shared; who will keep it, why, and for how long.

REFERENCES

Abdullah, R., Hussain, A., Warwick, K., Zayed, A., 2008. Autonomous intelligent cruise control using a novel multiple-controller framework incorporating fuzzy-logic-based switching and tuning. Neurocomputing. 71, 2727–2741.

Al Alam, A., Gattami, A., Johansson, K.H., 2010. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. s.l., s.n., pp. 306–311.

Anderson, J.E., 2009. An intelligent transportation network system: Rationale, attributes, status, economics, benefits, and courses of study for engineers and planners. PRT International.

Behere, S., Törngren, M., 2015. A Functional Architecture for Autonomous Driving. ACM, New York, NY, USA, pp. 3–10.

Behere, S., 2013. Architecting Autonomous Automotive Systems: With an Emphasis on Cooperative Driving, s.l.: s.n.

Belker, T., Beetz, M., Cremers, A.B., 2002. Learning action models for the improved execution of navigation plans. Rob. Auton. Syst. 38, 137–148.

Berlin, T., 2007. Spirit of Berlin: an autonomous car for the DARPA urban challenge hardware and software architecture. Retrieved Jan, 5, p. 2010.

Bhatt, D., Aggarwal, P., Devabhaktuni, V., Bhattacharya, P., 2014. A novel hybrid fusion algorithm to bridge the period of GPS outages using low-cost INS. Expert Syst. Appl. 41, 2166–2173.

Bo, C., Cheng, H.H., 2010. A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transp. Syst. 11, 485–497.

Broggi, A., et al., 2012. Autonomous vehicles control in the VisLab intercontinental autonomous challenge. Annu. Rev. Control 36, 161–171.

Buehler, M., Iagnemma, K., Singh, S., 2007. The 2005 DARPA grand challenge: the great robot race. s.l. Springer Science & Business Media.

Chakraborty, D., Vaz, W., Nandi, A.K., 2015. Optimal driving during electric vehicle acceleration using evolutionary algorithms. Appl. Soft Comput. 34, 217–235.

Companion, 2016. COMPANION - COoperative dynamic forMation of Platoons for sAfe and eNergy-optImized gOods transportatioN. [Online] Available at: <http://www.companion-project.eu/>.

Conner, M., 2011. Automobile sensors may usher in self-driving cars. EDN Mag.21–24.

Continental, A.G., 2012. SRL-CAM400 CMOS camera and infrared LIDAR in a compact unit. s.l.:s.n.

Costello, B., Suarez, R., 2015. Truck driver shortage analysis 2015. Am. Truck. Assoc. 206, 2015, Retrieved from http://www.trucking.org/ATA\%20Docs/News\%20and\%20Information/Reports\%20Trends\%20and\%20Statistics/10.

Cunningham, A.G., Galceran, E., Eustice, R.M., Olson, E., 2015. MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. s.l. IEEE1670–1677.

Czubenko, M., Kowalczuk, Z., Ordys, A., 2015. Autonomous driver based on an intelligent system of decision-making. Cognit. Comput. 7, 569–581.

Delphi Inc., 2015. Integrated Radar and Camera System. [Online].

der Automobilindustrie eV, V, 2015. Automation: From Driver Assistance Systems to Automated Driving. VDA Magazine-Automation.

Dokic, J., Müller, B., Meyer, G., 2015. European roadmap smart systems for automated driving. s.l.:European Technology Platform on Smart Systems Integration.

Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J., 2010. Path planning for autonomous vehicles in unknown semi-structured environments. Int. J. Robot. Res. 29, 485–501.

Eckhardt, J., 2015. European Truck Platooning Challenge 2016. The Hague, Delta3.

ERTRAC, 2015. ERTRAC Roadmap for Automated Driving. s.l., s.n.

European Union, 2016. Declaration of Amsterdam - Cooperation in the Field of Connected and Automated Driving. s.l.:s.n.

Fernandes, P., 2012. Platooning of IVC-Enabled Autonomous Vehicles: Information and Positioning Management Algorithms, for High Traffic Capacity and Urban Mobility Improvement, s.l.: s.n.

Fritz, H., Bonnet, C., Schiemenz, H., Seeberger, D., 2004. Electronic Tow-Bar Based Platoon Control of Heavy Duty Trucks Using Vehicle-Vehicle Communication: Practical Results of the CHAUFFEUR2 Project. s.l., s.n.

Funke, J. et al., 2012. Up to the Limits: Autonomous Audi TTS. s.l., s.n., pp. 541–547.

Furda, A., Vlacic, L., 2010. Multiple criteria-based real-time decision making by autonomous city vehicles. IFAC Proc. 43, 97–102.

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.

Galceran, E., Cunningham, A.G., Eustice, R.M., Olson, E., 2015. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. Robot. Sci. Syst.

Gerdes, J.C., Thornton, S.M., 2016. Implementable ethics for autonomous vehicles. Autonomous Driving. s.l. Springer, pp. 87–102.

Godoy, J., et al., 2015. A driverless vehicle demonstration on motorways and in urban environments. Transport 30, 253–263.

Goodall, N., 2014. Ethical decision making during automated vehicle crashes. Transp. Res. Rec. J. Transp. Res. Board 58–65.

Google Inc, 2015. The Google Self-Driving Car Project. [Online] Available at: <http://www.google.com/selfdrivingcar/>.

Haddad, M., Chettibi, T., Hanchi, S., Lehtihet, H.E., 2007. A random-profile approach for trajectory planning of wheeled mobile robots. Eur. J. Mech. A/Solids 26, 519–540.

Holguín, C., 2016. 4 misconceptions on "autonomous" cars that can lead cities to disaster, and how to remedy. [Online].

Ibañez-Guzmán, J., Laugier, C., Yoder, J.-D., Thrun, S., 2012. Autonomous driving: context and state-of-the-art. In: Eskandarian, A. (Ed.), Handbook of Intelligent Vehicles. Springer, London, pp. 1271–1310.

Janssen, R., Zwijnenberg, H., Blankers, I., de Kruijff, J., 2015. Truck Platooning: Driving the Future of Transportation.

Jenkins, A., 2007. Remote Sensing Technology for Automotive Safety. s.l.:s.n.

Katrakazas, C., Quddus, M., Chen, W.-H., Deka, L., 2015. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transp. Res. Part C: Emerg. Technol. 60, 416–442.

Kunze, R., Ramakers, R., Henning, K., Jeschke, S., 2009. Organization and Operation of Electronically Coupled Truck Platoons on German Motorways. s.l., s.n., pp. 135–146.

Kuwata, Y., et al., 2009. Real-Time Motion Planning With Applications to Autonomous Urban Driving. IEEE Trans. Control Syst. Technol. 17, 1105–1118.

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

Levine, W., Athans, M., 1966. On the optimal error regulation of a string of moving vehicles. IEEE Trans. Autom. Control 11, 355–361.

Manyika, J. et al., 2013. Disruptive technologies: Advances that will transform life, business, and the global economy. s.l.: McKinsey Global Institute San Francisco, CA.

Melzer, S.M., Kuo, B.C., 1971. Optimal regulation of systems described by a countably infinite number of objects. Automatica 7, 359–366.

Meystel, A., Albus, J.S., 2002. Intelligent Systems: Architecture, Design, and Control s.l.: Wiley.

Naus, G.J.L., et al., 2010. String-stable CACC design and experimental validation: A frequency-domain approach. IEEE Trans. Veh. Technol. 59, 4268–4279.

Navarro, P.J., Fernández, C., Borraz, R., Alonso, D., 2016. A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data. Sensors 17, 18.

NFI, 2013. The new face of industry in France. s.l.:s.n.

OCR Software blog, 2016. OCR Software blog. [Online] Available at: <https://ocr.space/blog/2016/09/liability-for-self-driving-car-accidents.html>.

Open Roboethics Initiative, 2016. Open Roboethics Initiative. [Online] Available at: <http://www.openroboethics.org> (Accessed 12.16).

Paden, B., et al., 2016. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans. Intell. Veh. 1, 33–55.

PATH, 2017. <http://www.path.berkeley.edu>. s.l., s.n.
Ploeg, J., Shukla, D.P., van de Wouw, N., Nijmeijer, H., 2014. Controller synthesis for string stability of vehicle platoons. IEEE Trans. Intell. Transp. Syst. 15, 854–865.
Rajamani, R., 2011. Vehicle Dynamics and Control. s.l.: Springer Science & Business Media.
SAE, 2016. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, s.l.: SAE International.
Sánchez, F., et al., 2015. CO2PERAUTOS2—Challenges for Cooperative Automated Driving. s.l., s.n.
Sartre, 2016. The SARTRE project. [Online] Available at: <http://sartre-project.eu/>.
Shang, E., et al., 2015. LiDAR based negative obstacle detection for field autonomous land vehicles. J. Field Robot.
Shaw, E., Hedrick, J.K., 2007. String Stability Analysis for Heterogeneous Vehicle Strings. s.l., s.n., pp. 3118–3125.
Skog, I., Handel, P., 2009. In-car positioning and navigation technologies—a survey. IEEE Trans. Intell. Transp. Syst. 10, 4–21.
Swaroop, D., Hedrick, J.K., 1996. String stability of interconnected systems. IEEE Trans. Autom. Control 41, 349–357.
Thrun, S., et al., 2006. Stanley: the robot that won the DARPA grand challenge, J. Field Robot., 23. pp. 661–692.
Tsugawa, S., 2014. Results and Issues of An Automated Truck Platoon Within the Energy ITS Project. s.l., s.n., pp. 642–647.
Tzoreff, E., Bobrovsky, B.-Z., 2012. A novel approach for modeling land vehicle kinematics to improve GPS performance under urban environment conditions. IEEE Trans. Intell. Transp. Syst. 13, 344–353.
U. S. Department of Transportation, 2016. Federal Automated Vehicles Policy – September 2016. s.l.:s.n.
Urmson, C., et al., 2008. Autonomous driving in urban environments: Boss and the urban challenge. J. Field Robot. 25, 425–466.
Urmson, C., et al., 2009. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. In: Buehler, M., Iagnemma, K., Singh, S. (Eds.), The DARPA Urban Challenge: Autonomous Vehicles in City Traffic. Springer, Berlin Heidelberg, pp. 1–59.
Urmson, C., Anhalt, J., Bagnell, D., Baker, C., 2008. Autonomous driving in urban environments: boss and the urban challenge. J. Field Robot.
Veres, S.M., Molnar, L., Lincoln, N.K., Morice, C.P., 2011. Autonomous vehicle control systems -- a review of decision making. Proc. Inst. Mech. Eng. Part I. J. Syst. Control Eng. 225, 155–195.
Wei, J. et al., 2013. Towards A Viable Autonomous Driving Research Platform. s.l., s.n., pp. 763–770.
Zabat, M., Stabile, N., Frascaroli, S., Browand, F., 1995. The Aerodynamic Performance of Platoons: Final Report. s.l. University of California, Berkeley.
Zhang, T., Xu, X., 2012. A new method of seamless land navigation for GPS/INS integrated system. Measurement 45, 691–701.