Chapter 3
Perception and Control for Connected and Automated Vehicles

In this book, by connected vehicles we are referring to vehicles that use communication technologies such as DSRC and cellular for vehicle-to-everything (V2X) communication. The U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTSA) defines fully automated vehicles as those in which operation of the vehicle occurs without direct driver input to control the steering, acceleration, and braking and are designed so that the driver is not expected to constantly monitor the roadway while operating in self-driving mode [1]. In categorizing partial automation, NHTSA's federal automated vehicles policy adopts that of Society of Automotive Engineers (SAE) definitions for levels of vehicle automation as shown in the reproduced Table 3.1. Automation levels range from no automation with full driver control (Level 0) to full automation with no driver control (Level 5). Many of the benefits discussed in this book are realizable with partial level 2 or 3 automation as they mostly rely on automated speed and steering control which can be overseen and overridden by a human driver.

In this chapter, after a review of V2X technologies for connected vehicles in Sect. 3.1, we provide a brief overview of automated vehicle localization and perception in Sect. 3.2 and planning and control in Sect. 3.3. A schematic overview is shown in Fig. 3.1.

3.1 V2X Communication

Connected vehicles could ideally benefit from Vehicle-to-everything (V2X) communication channels and protocols to exchange data and information with a wide variety of entities. Some of the main benefits are increased road safety, harmonized traffic flow, and energy savings. For instance Vehicle-to-Vehicle (V2V) communication allows equipped vehicles to exchange their coordinates and intentions to prevent

A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles*,
Lecture Notes in Intelligent Transportation and Infrastructure,
https://doi.org/10.1007/978-3-030-24127-8_3

Table 3.1 Society of Automotive Engineering Vehicle automation levels reproduced from SAE Standard J3016 [2]. Copyright © 2018 SAE International. The following abbreviations are carried from J3016: Dynamic Driving Task (DDT), Object and Event Detection and Response (OEDR), Operational Design Domain (ODD), and Automated Driving System (ADS)

Level	Name	Narrative definition	DDT		DDT fallback	ODD
			Sustained lateral and longitudinal vehicle motion control	OEDR		
Driver performs part or all of the DDT						
0	No driving automation	The performance by the driver of the entire DDT, even when enhanced by active safety systems	Driver	Driver	Driver	n/a
1	Driver assistance	The sustained and ODD-specific execution by a driving automation system of either the lateral or the longitudinal vehicle motion control subtask of the DDT (but not both simultaneously) with the expectation that the driver performs the remainder of the DDT	Driver and system	Driver	Driver	Limited
2	Partial driving automation	The sustained and ODD-specific execution by a driving automation system of both the lateral and longitudinal vehicle motion control subtasks of the DDT with the expectation that the driver completes the OEDR subtask and supervises the driving automation system	System	Driver	Driver	Limited
ADS ("System") performs the entire DDT (while engaged)						
3	Conditional driving automation	The sustained and ODD-specific performance by an ADS of the entire DDT with the expectation that the DDT fallback-ready user is receptive to ADS-issued requests to intervene, as well as to DDT performance- relevant system failures in other vehicle systems, and will respond appropriately	System	System	Fallback-ready user (becomes the driver during fallback)	Limited
4	High driving automation	The sustained and ODD-specific performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene	System	System	System	Limited
5	Full driving automation	The sustained and unconditional (i.e., not ODD- specific) performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene	System	System	System	Unlimited

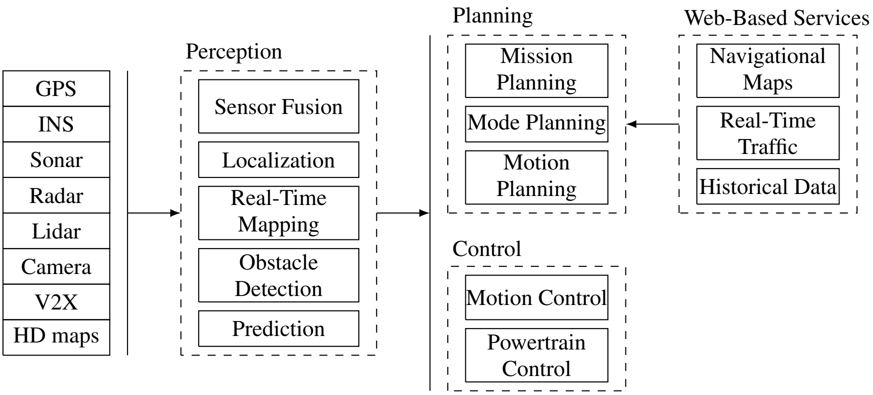


Fig. 3.1 Sensing, perception, planning, and control in CAVs

collision or to move in coordination. Vehicle-to-Infrastructure (V2I) communication allows vehicles to communicate with roadside units and infrastructure such as traffic signals enabling better coordination between them. A few other communication modes to name are Vehicle-to-Pedestrian (V2P), Vehicle-to-Device (V2D), Vehicle-to-Network (V2N), Vehicle-to-Cloud (V2C) and Vehicle-to-Grid (V2G) communication. In this book our main results only require V2V or V2I.

Today there exists two main communication technologies for V2X: (i) Wireless Local Area Network (WLAN) and (ii) Cellular Network.

WLAN technology allows vehicles moving at high speeds to establish ad-hoc and direct communication channels with neighboring vehicles and roadside traffic units without the need for additional communication infrastructure. Several countries have allocated a spectrum for Intelligent Transportation Systems communication that enables WLAN V2X. For instance in the United States, a 75 MHz band in the spectrum of 5.850–5.925 GHz has been set by the US Federal Communication Commission (FCC) since 1999. In Europe 30 MHz has been assigned for the same purpose. Currently the IEEE 1609 family, IEEE 802.11p, and the Society of Automotive Engineers (SAE) J2735 [3] form the key parts of the currently proposed Wireless Access in Vehicular Environments (WAVE) protocols. [4]. The architecture, communications model, management structure, security mechanisms and physical access for high speed (up to 27 Mb/s) short range (up to 1000 m) low latency wireless communications in the vehicular environment is defined by the IEEE 1609 Family of Standards [5]. Society of Automotive Engineers (SAE) uses the term Dedicated Short Range Radar Communication (DSRC) for the WAVE technology with J2735 set of standards which define the message payload at the physical layer. The SAE J2735 [6] supports interoperability among DSRC applications through the use of standardized message sets, data frames, and data elements.

Cellular V2X or in short C-V2X technology was initially defined as LTE in the Third Generation Partnership Project (3GPP) Release 14 [7] and is designed to operate in several modes: (1) Device-to-device and (2) Device-to-cell-tower. The device-to-device mode allows direct communication without necessarily relying on cellular network involvement. On the other hand, device-to-cell-tower relies on existing cell towers, network resources, and scheduling. Direct device-to-device communication improves latency and supports operation in areas without cellular network coverage.

3.2 Localization and Perception for Automated Driving

A key to successful automated driving is effective localization, obstacle detection, and perception. The vehicle must not only determine with high precision its location in the world and on the road but it should perceive accurately its surrounding environment such as neighboring vehicles, pedestrians, animals crossings, lane markings, traffic signs and signals, street signs, curbs and shoulders, buildings and trees, etc, and measure their relative distance and speed. These are perhaps the hardest technical challenges to overcome for highly automated driving. Here we present a brief overview of sensors and algorithms that are currently used for localization and perception.

3.2.1 Sensors for Perception and Localization

An overview of sensors for perception and localization is provided in Fig. 3.2. Self- or proprioceptive-sensors measure the ego vehicle internal states such as its velocity,

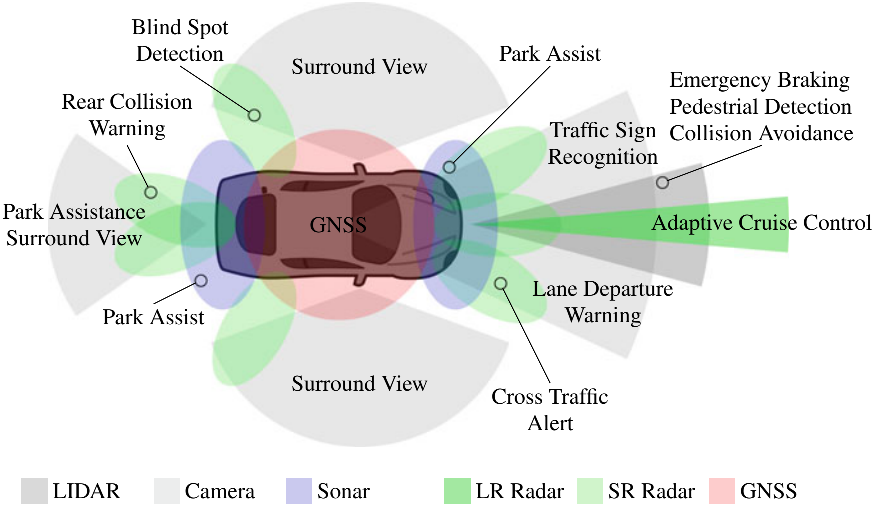


Fig. 3.2 Schematic of vehicle sensors for perception and localization. Adapted from [8]

acceleration, wheel speed, yaw, steering angle, engine speed, and engine torque. Odometers, accelerometers, Inertial Measurement Units (IMU), and information from Control Area Network (CAN) bus are used for proprioceptive-sensing and are not limited only to automated vehicles; many modern human driven vehicles rely on them for state estimation and advanced control functions. For example, IMUs contain gyroscopes, accelerometers, and sometimes magnetometers along each axis that provide dead reckoning capability in combination with the vehicle's wheel speed sensors. Since an IMU relies on integrating acceleration to determine positions, they are prone to drift and may require GPS-fusion (or camera-fusion when indoor) for more accurate localization.

Global Navigation Satellite Systems (GNSS) sensors also known commonly as Global Positioning System (GPS) are becoming standard on modern vehicles for navigation and localization. Other regional GNSS systems are Russia's GLONASS, Europe's Galileo, and China's Beidou. While current GNSS may not provide the needed sub-meter precision for localization of automated vehicles, filtering algorithms that fuse GNSS and IMU readings could offer more precise localization. Still centimeter precision levels needed for some automated vehicle functions, such as lane determination, may benefit from more precise positioning systems. Reduction in cost of highly accurate GNSS is expected in near future making it available to the mass market [9]. Today Real-Time Kinematic (RTK) GPS technology is available and relies on a roadside base station to correct GPS readings to within centimeter accuracy. Simultaneous Localization And Mapping (SLAM), which we discuss in more detail later under localization algorithms, are used by many autonomous vehicle developers to localize the vehicle with respect to the surrounding environment.

Extroceptive sensors such as sonar, radar, LIght Detection And Ranging (LIDAR) and cameras are used for sensing the surrounding environment and objects as summerized in Table 3.2. Sonar, radar, and Lidar are called active sensors because they emit energy in the form of sound and electromagnetic waves and measure the return to map the surrounding environment, e.g. distance to nearby objects. On the other hand light and infrared cameras are called passive sensors since they do not emit energy and only measure the light/electromagnetic waves in the environment [10].

Sonar can measure distance to nearby objects but only has a very limited range ($<$2m) and has low angular resolution. Radars rely on reflection of radio waves that they emit to measure distance to and velocity of moving objects and have a much higher range than sonar but are weak in classification, pedestrian detection, and in detecting static objects. Also radars may suffer from interference from other radars and create false alarms. Lidar works similarly to radar but relies on infrared light (laser) instead of radio waves. Lidars emit laser at wavelengths beyond the visual light spectrum at typical scan frequency of 10–15 Hz. They emit millions of pulses per second giving them high resolution, a large field of view, and the capability to create a 3D point cloud of surrounding environment. This has made them an essential sensor for most automated vehicle developers. Nevertheless, Lidar cannot directly measure velocity, may have difficulty with detecting highly reflective objects and has degraded performance in fog, rain, or snow. Segmentation, classification, and sometimes time integration algorithms are still needed to convert the 3D raw data

Table 3.2 Comparison of different extroceptive sensor technologies for automated driving. The results are compiled from [10, 12–14]

	Sonar	Radar	Lidar	Single vision camera
Perceived Energy	Sound waves	Millimeter wave radio signal	600–1000 nanometer wave laser signal	Visible light
Range [m]	2–5	0.15–250	2–100	250
Vehicle recognition versus other objects	Tracking	Tracking	Spatial segmentation, motion	Appearance, motion
Resolution	–	o	+	++
Field of view	–	+ (short range radar)- (long range radar)	++	++
Distance measurement	+	++	++	–
Velocity measurement	−−	++	+	–
Operation in poor weather	++	++	o	–
Poor lighting performance	++	++	++	−−
Other challenges	Poor classification low range low resolution	Poor classification, poor pedestrian detection, poor static object detection, prone to interference	Poor classification compared to vision, difficulty with highly reflective objects	high computational cost
Cost (US $)	50	50–200	7,000–70,000	100–200

++Very good +good o average – poor −−very poor

to classified objects [11]. While laser emitting detection technology is not new, it was not till 2005 that Velodyne put 64 rotating lasers in one compact package for 360° detection needed in automated driving. Since then Lidar technology has been adopted by almost all autonomous vehicle teams. Still current Lidars are not designed to withstand many years of harsh conditions in open road driving. Both radar and Lidar also are weak in detecting very near objects (<2m) where sonar performs well [10].

Cameras provide high field of vision and high resolution, and capture information that Lidar cannot such as color and texture which helps object classification. However with monocular camera vision it is more difficult to measure depth; this can be overcome with stereo vision provided by two cameras. Computationally, camera vision is more demanding than Lidar. Converting 2D images to 3D understanding of the environment requires computationally demanding software and machine learning algorithms. Camera vision is sensitive to lighting conditions and its performance degrades in bad weather [12].

The algorithmic aspects of perception and localization are briefly discussed next.

3.2.2 Algorithms for Perception and Localization

Given the pros and cons of extroceptive sensors, in particular camera and Lidar, it is common to use both and rely on filtering and data fusion algorithms to increase accuracy and robustness. Measurement error covariance when using two sensors is always smaller than the error covariance achieved by each individual sensor. So it often makes sense to fuse data from two inexpensive sensors and achieve similar accuracy of a single high end sensor [11]. V2X communication can provide additional information from other vehicles and roadside units for higher accuracy perception and localization.

3.2.2.1 Perception Algorithms

Perception algorithms could be vision-based relying on camera data, or rely on active sensors which capture objects by a large number of points on their surface, also called point clouds. Camera and active sensors can be employed together to detect and perceive the surrounding environment and objects (such as vehicles, pedestrians, animals, curbs) more precisely. While there are mature machine vision and statistical learning and classification algorithms for parsing information embedded in an image or point cloud, recent advances in deep learning and artificial intelligence provide new supervised learning methods for real-time object detection. Rapidly growing training datasets, increased computing power, cheaper storage, and widely available open-source algorithms seem to be bringing about revolutionary advances. For instance an open source real-time object detection algorithm presented recently in [15] based on convolutional neural networks has the ability to process 45–150 frames per second, label objects in it with a bounding box, and assign a confidence score to each as illustrated in Fig. 3.3.

In automated driving three paradigms have been proposed for perception: (i) mediated perception, (ii) behavior reflex perception, and (iii) direct perception [16]. In the more common mediated perception, a detailed map and distance to relevant objects around the ego vehicle including other vehicles, pedestrians, trees, and road markings are extracted first using standard machine vision or deep learning algorithms. Planning and control algorithms will then use this map to plan the motion of the vehicle considering the constraints imposed by the road and stationary and dynamic obstacles. Quite differently, behavior reflex perception algorithms use artificial intelligence to construct a direct mapping from the sensory input to a driving action thus bypassing intermediate layers such as localization, path planning, decision making

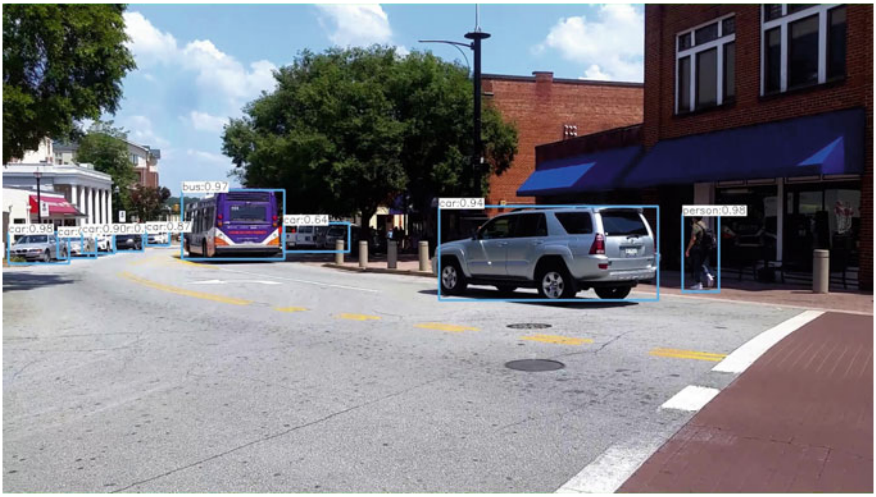


Fig. 3.3 An example of application of YOLO real-time object detection [15] to a driving scene. The numbers next to each label show the confidence in that label. Picture courtesy of Austin Dollar and Tyler Ard of Clemson University

and control [17]. While they reduce complexity, such end-to-end solutions, lack transparency, are too low-level missing the big picture, and sometimes may be ill-posed in training. For instance in [18] it is shown that stability can be lost when applying supervised learning to a training set of locally exponentially stable controllers. Direct perception methods proposed in [16] aim to strike a balance between the former two approaches. They abstract an image to a selected and meaningful set of indicators of the road situation, such as the angle of the car relative to the road, the distance to the lane markings, and the distance to cars in the current and adjacent lanes. The outcome is much more compact than what a mediated perception approach would generate and only contains the most relevant information to the the planning and control layers which could now be simplified according to [16].

3.2.2.2 Prediction

While perception by itself is an important and challenging step, predicting the motion of neighboring vehicles or pedestrians based on perceived current and historical information may be as important for CAV planning purposes. This is a difficult topic and still an open problem. In Sect. 1.3.3 we discussed relevant prediction literature in the context of anticipative car following where probabilistic prediction was a common theme for predicting the longitudinal motion of a preceding vehicle. Other examples are assuming a constant speed in [19], a speed-dependent acceleration in [20], probabilistic trajectory prediction in horizontal plane using a variational mixture model in [21], a Gaussian mixture model in [22], or classification and particle filtering in [23]. Most of these prediction methods target a 1–3 second prediction window which may be limited. With V2X connectivity the opportunity exists for receiving future intentions of neighboring vehicles and nearby traffic controllers which should enable predicting with more accuracy over longer horizons. We will come back to this topic in Sect. 8.2.3 in the context of energy efficient driving.

3.2.2.3 Localization and Mapping

CAVs require rather precise localization not only for navigational purposes but also to situate themselves within the road and the lane, with respect to other (connected) vehicles, and for use of mapped information such as location of traffic signals, upcoming hills, curves, dynamic congestion tail, etc. While localization and mapping is a well established topic in indoor robot navigation and mature algorithms exist [24], outdoor, dynamically changing, and high speed road environments present extra challenges for CAV localization.

Fusing GPS, IMU, and wheel odometer readings could provide meter-level precision in determining the position of the vehicle on the road. The raw coordinates determined by GPS may not match a logical model of the world where vehicles are expected to be on a road. Established map matching methods [25] are commonly used to correct the raw GPS recordings to a logical position on a road. The (corrected) GPS data can be fused with IMU and odometry readings via Extended Kalman Filtering (EKF) methods that rely on a model of vehicle kinematics or dynamics. Velocity of the vehicle could be determined as a by-product. Accurately determining vehicle heading is more difficult due to reliance on IMU readings which are subject to drift.

Algorithms relying on GPS inertial navigation could be challenged in urban canyons with tall buildings due to loss of GPS signals [26]. Also autonomous vehicle control may require centimeter level position accuracy not provided by conventional GPS/IMU fusion. While RTK GPS provides high level of position accuracy, its reliance on additional roadside stations makes it impractical on today’s roads. To overcome this challenge many automated driving vehicles such as Waymo’s and Uber’s rely on a priori mapped roads. Instrumented mapping vehicles drive roads of interest and collect detailed 3D image or Lidar data linked to highly accurate GPS information, process and store them in large databases. Subsequent CAVs can localize by comparing their sensor readings against these a priori maps and triangulating their position with the aid of fixed objects. Moreover they can more easily distinguish dynamic objects absent from a priori maps. An early successful implementation can be found in [27]. Such a method works as long as the mapped roads remain unchanged. Construction zones, changes in lane markings or road geometry could render parts of these maps obsolete.

This problem can be overcome by High Definition (HD) mapping where a priori maps are dynamically updated in the cloud based on latest sensory information communicated from CAVs traversing these roads [28]. For instance a consortium formed by BMW, HERE, and Mobileye aims to crowdsource HD maps relying on accurate prior maps from HERE, BMW connected fleet, and Mobileye REM technology that transmits changes detected with respect to prior map to cloud servers to update the maps. The dynamically updated maps become then accessible to the connected fleet in real-time via HERE servers.

In this context Simultaneous Localization And Mapping (SLAM) arises when the vehicle has to simultaneously localize and map the environment and obviously are more difficult than only localization or only mapping. SLAM is well established in indoor robotic navigation [24] often in well-structured and well-lit environments. SLAM is more challenging for automated vehicles due to variable lighting, less structured road environments and higher speeds that require faster computations [10].

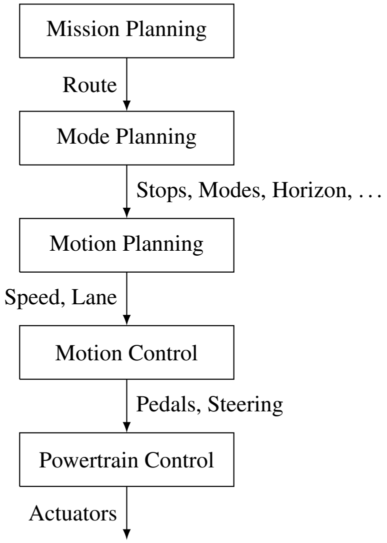
3.2.3 Web Services

Connected vehicles can query web-based Application Programming Interfaces (API) to retrieve map, traffic congestion, and weather information in real-time. For instance the cloud based Google Map Platform [29] provides several APIs for retrieving maps, elevation, traffic, directions, travel times and distances, and places in real-time. Similar services are provided by HERE APIs [30]. Inrix offers a traffic and a parking API [31]. There are several weather information APIs such as Yahoo Weather API [32]. Today computational clouds such as Amazon Web Services (AWS) offer their computing and machine learning tools to connected [33] and automated [34] vehicle developers. The idea is to offload the onboard computations and data analytics partially to the cloud.

3.3 Planning and Control

Once an automated vehicle localizes itself with respect to a 3D map of the environment and identifies constraints imposed by the surrounding stationary and moving objects, traffic rules, traffic control infrastructure, and road geometry, it can plan its long- and short-term moves. This plan is then executed by a hierarchy of motion planners and controllers in the longitudinal and lateral directions. Both planning and control layers can benefit from the extended preview of the upcoming road and traffic scene provided by V2X connectivity to make longer term judicious decisions. Here we provide a brief overview of the planning and control layers as shown schematically in Fig. 3.4.

Fig. 3.4 Logical scheme of planning and control layers in CAVs



3.3.1 Mission Planning

At the highest planning layer, the route is decided, for instance to minimize trip distance, time, delay, or energy. The road network is often modeled as a directed graph with its edge weights reflecting the relevant cost of travel on that link. The minimum cost path can then be found via optimization which can be executed very efficiently today as explained in [35]. For electric vehicles, visits to charging stations may also be planned at this stage. The mission planning layer can then set waypoints along the chosen route as targets for the lower level motion planning layer. More details of algorithms employed in the mission planning layer in the context of eco-routing are described in Chap. 5.

3.3.2 Mode Planning

Another distinct planning layer may exist that chooses between a finite set of driving modes in consideration of mission waypoints, road rules, and traffic conditions. For instance the vehicle may choose lane keeping, a lane change, (adaptive) cruise control, stopping at a stop sign, or emergency braking. This will be a finite set of modes that can be handled in a finite state machine framework or via decision trees. We refer to this layer as Mode Planning, but in the literature other terms such as driving strategy [36], maneuver planning [37] and behavioral decision making [35] are also used.

We will show in Chaps. 6 and 7 that optimal eco-driving in a trip could consist of several modes for example maximum acceleration, constant speed cruising, coasting, and maximal braking between two stopping intervals.

3.3.3 Motion Planning

After a driving mode is selected, the motion planning layer generates legal, collision-free, smooth, comfortable, and efficient paths or trajectories for longitudinal and lateral motion of the vehicle. The literature distinguishes a trajectory from a path in that a path is in the spatial configuration space of the vehicle while a trajectory has a temporal component as well [38]. For instance in the longitudinal direction s, usually the velocity trajectory $\dot{s}(t)$ is planned with safety, ride comfort, travel time, and energy efficiency considerations while respecting constraints imposed by speed limits, traffic lights and stop signs, surrounding vehicles, road curvature, and longitudinal vehicle dynamics.

For example in Cruise Control (CC) mode the vehicle tracks a constant reference speed while Adaptive Cruise Control (ACC) adjusts the velocity to maintain a safe time or distance headway to the preceding vehicle. More details on that are discussed in Sect. 4.2.2. In Predictive Cruise Control (PCC) mode, the velocity is adjusted relying on V2I communication and in anticipation of future events such as changes in road slope or traffic signal phase and timing. Cooperative Adaptive Cruise Control (CACC) mode relies on V2V communication to allow vehicles cruise in coordination with neighboring vehicles. In emergency braking mode, the vehicle could apply maximal braking to avoid a collision.

Lane change, merge, and collision avoidance need to determine a feasible path in the 2D x-y plane which is by itself complex due to many choices in a 2 dimensional space and the non-convex drivable regions. Furthermore due to velocity and time dependent constraints arising from vehicle dynamics and movement of surrounding vehicles, the motion planning algorithms should also determine safe and comfortable acceleration and velocity profiles on these paths; thus a trajectory planning problem.

In Sect. 3.3.6 we discuss optimal planning algorithms applicable to motion planning.

3.3.4 Motion Control

The trajectory or path planned at the motion planning layer is issued as a reference to the vehicle longitudinal and lateral controllers for feedforward and feedback tracking. In the longitudinal direction, throttle and braking control adjusts acceleration and velocity. Lateral control relies mainly on steering and sometimes on differential braking to control lateral acceleration, velocity, and vehicle yaw rate.

3.3.4.1 Longitudinal Control

When the reference speed is determined at the planning layer, well-established classical or modern control techniques can be used at the motion control layer, to follow the planned reference by accelerator or brake actuation. For instance standard fixed gain or gain-scheduled PID type controllers can actuate accelerator and brakes for velocity reference tracking [39]. An integrator anti-windup mechanism [40] must be added to properly handle actuator saturation. Logical checks should be in place to ensure safe operations under all perceivable circumstances. Switching between accelerating and braking modes needs to be handled with care for smooth performance [41].

For instance [42] proposed the following PID type controller with an added nonlinear term shown below:

$$u(s) = -k_p e_v - k_i \frac{1}{s}\left(e_v - \frac{1}{T_t}[u - \mathrm{sat}(u)]\right) - k_d \frac{\tau_d s}{\frac{1}{N}\tau_d s + 1} e_v - k_q e_v |e_v| \,, \quad (3.1)$$

where the equation should be read in the Laplace domain with s denoting the Laplace variable. Here u commands accelerator or braking, and e_v is the velocity tracking error. Tunable proportional, integral, and derivative gains are denoted by k_p, k_i, k_d respectively, while k_q is a tunable gain for the last nonlinear term. The term $\frac{1}{T_t}[u - \mathrm{sat}(u)]$ prevents integrator windup where the $\mathrm{sat}(u)$ function saturates at actuator limits and T_t is a time constant that determines how fast the integrator is reset. Because a pure derivative term would be non-causal and prone to noise, a pseudo-derivative term is employed by augmenting a first order lag, wherein parameter N determines the amount of filtering on the derivative term. The last nonlinear term $k_q e_v |e_v|$ is termed the quadratic component in [42] and is intended to achieve fast tracking while limiting the overshoot. Asymptotic convergence of tracking error to zero is established in [42] via a Lyapunov analysis.

Feedforward control along with feedback control can enhance the responsiveness of the longitudinal control loop. For instance when the planning layer commands an acceleration profile, a feedforward pedal/braking input can be issued [43] based on pedal-to-acceleration and braking-to-deceleration response mappings along with a feedback controller.

Input saturation, vehicle state constraints, and toggling between accelerator and braking actuators can be more systematically handled in a constrained control framework. For heavy vehicles sensitivity to often unknown mass of the truck can also be handled by adaptive control techniques as shown in [44].

3.3.4.2 Lateral Control

Lateral control engages steering and sometimes differential braking to control the vehicle in scenarios such as lane changing, merging, turning, and parallel parking. The assumption is that an appropriate reference path or trajectory is already deter-

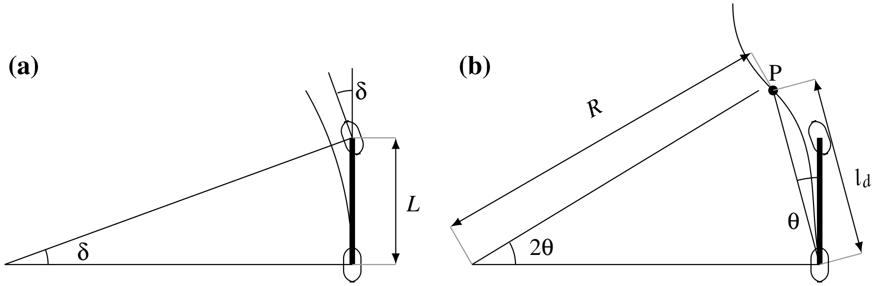


Fig. 3.5 A simplified bicycle model of a 4 wheeled vehicle: geometric bicycle model (**a**) and pure pursuit geometry (**b**). Adapted from [46]

mined in the motion planning layer. A widely used approach for path tracking with mobile robots and autonomous vehicles is pure pursuit control that was first introduced in [45] and is relatively simple to implement. The pure pursuit algorithm has a simple formula for choice of the steering angle that steers the rear axle on a circular arc to the center of the path. If a bicycle model, as shown in Fig. 3.5, is used to relate the steering angle of the front wheels $\delta(t)$ to vehicle heading $\theta(t)$, the pure pursuit algorithm formula is

$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\theta(t))}{l_d}\right), \tag{3.2}$$

where L is the wheelbase, l_d is the the distance from the rear wheel to a look-ahead point on the center of the path, and θ is angle between the heading vector and the look-ahead vector pointing to the center of the path l_d units ahead. In practice the look-ahead distance is chosen as a function of vehicle speed [46].

Another method [47], adjusts the steering as a function of vehicle heading misalignment with the path and a nonlinear function of cross track error,

$$\delta(t) = \theta(t) - \theta_p(t) + \tan^{-1}\left(\frac{ke_y}{v_x(t)}\right), \tag{3.3}$$

where θ is heading angle of the vehicle, θ_p is path heading at the point nearest to the front wheel, e_y is the cross-track error measured from the center of the front wheels to the nearest point on the track, v_x is vehicle's forward velocity, and k is a gain parameter. Using an idealized bicycle model, the cross-tracking error is shown to be monotonically convergent to zero.

The above lateral control methods are easy to implement, but rely on feedback from a single point of the lane at each time. For smoother performance, the lane tracking problem can be formulated as a finite horizon optimal control problem with full horizon preview of lane reference trajectory. The optimal steering control action will not only be a function of instantaneous vehicle state but also will include a feedforward term that integrates the entire lane preview. An analytical solution to this preview optimal control problem exists when the vehicle model is linear, the tracking cost is quadratic, and input and states are unconstrained as shown in [48]. Input and state constraints must be considered for aggressive or emergency maneuvers or when driving on slippery roads with the tires at their traction limit. In such scenarios the trajectory tracking problem can be formulated in a model predictive control framework with higher fidelity vehicle models and with explicit consideration of traction constraints. Successive model linearizion results in a Linear Time-Varying (LTV) MPC problem as shown in [49] along with experimental results that demonstrates the feasibility of real-time implementation.

Planning and control algorithms that can handle more sophisticated conditions than the relatively simple longitudinal and lateral control methods described above, are described later in Sect. 3.3.6.

3.3.5 Powertrain Control

The powertrain control modules of a CAV can be programmed to take advantage of extra information that is available to them due to connectivity and increased certainty in that information due to driving automation as highlighted in Sect. 1.3.1. Depending on the powertrain type as described in Chap. 2, we have several actuators to coordinate such as throttle, braking, ignition, injection, cam phasing, wastegate, valve lift, cylinder deactivation, and transmission for SI ICEVs, battery utilization in HEVs, and vehicle-level actuators for accessory loads. Anticipated future velocity and road grade profile provide an estimate of the future power demands. This anticipated power demand profile can be used to better schedule choice of gears, battery utilization in hybrid vehicles, thermal load management, and handling of the powertrain auxiliary loads such as air-conditioning load.

The powertrain controllers can benefit from longer term plans of the mission planning and mode planning layers as well as more imminent intentions of motion planning and motion control layers. For example scheduling a hybrid vehicle's battery utilization can benefit from the long term mission plan due to the slow dynamics associated with the battery state of charge; so is thermal management due to relatively slow thermal dynamics as discussed respectively in Sects. 2.4.2 and 4.4.4. On the other hand, shorter term decisions at motion planning and control layers could be beneficial to functions with faster dynamics such as anticipative gear shift, fuel cut-off, engine start/stop, and cylinder deactivation.

3.3.6 Algorithms for Planning and Control

Two main schools of thoughts dominate the planning and control literature and practice. One approach guided by the robotics and computer science community employs (model-free) learning methods that aim to emulate human drivers, leveraging abundant training data and advances in deep learning and reinforcement learning algorithms. The second approach spearheaded by the automatic control community casts planning in a (model-based) optimal control framework aiming to minimize a mathematical cost of the motion (be it time, discomfort, energy, risk, etc.) while respecting all motion constraints. For instance, in a reinforcement learning approach to lane changing, the motion planning layer gradually learns a lane change policy that maximizes a cumulative reward function. The policy defines what action to take given the state of the road and neighboring vehicles and associates a reward to a successful lane change and gets penalized for a collision. The algorithm goes through a systematic trial and error process in a realistic simulation or real-world environment until it is "sufficiently" trained. It can then employ its learned policy in real-world driving.

An alternative to learning from training scenarios is optimal control that relies on models of the vehicle and its surrounding environment, a carefully designed objective function, and well characterized motion constraints. The plan can then be determined by solving a dynamic constrained optimization problem. For example in an optimal control approach to lane selection, the objective could be to balance a trade off between deviation from a desired velocity and a deviation from a desired lane. The predicted path of the surrounding vehicles can be imposed as motion constraints and a bicycle model of the vehicle can approximate the ego vehicle motion under candidate input sequences [50].

Closed-form analytic solutions for optimal control and planning problems rarely exist. Exact numerical solutions are often NP hard and not solvable in polynomial time. But one can often find approximations that simplify the problem. For example discretization, linearizing the models and constraints, and using a quadratic cost are common and reduce an optimal control problem to a quadratic program for which computationally efficient solution methods exist and enable its real-time implementation. The planning problem can be solved over a receding temporal or spatial horizon using feedback from the current state of the vehicle to update its plan at each optimization stage in what is referred to as Model Predictive Control (MPC) [51]. More details of numerically solving an planning problem in an MPC framework are described later in Sect. 8.2.5.

Numerical methods for optimal planning problems can be categorized to variational, graph search, and incremental search sample methods [38]. Under this categorization, Pontryagin Minimum Principle (PMP) is a variational approach that reduces the optimal control problem to a two point boundary value problem using variational calculus, the more details of which is described in Sect. 6.2.2.1. PMP is considered an *indirect* method because it is based on analytical construction of the necessary and sufficient conditions for optimality, and then discretizing these conditions and solving them numerically. *Direct* methods on the other hand discretize state and control trajectories and convert the optimal control problem to a nonlinear program [52], which is then solved using well-known optimization techniques. Pseudospectral optimal control methods [53] are among direct variational methods.

In graph search methods, the configuration space is discretized and represented by a graph consisting of vertices and edges. The graph is then explored to find the mini-

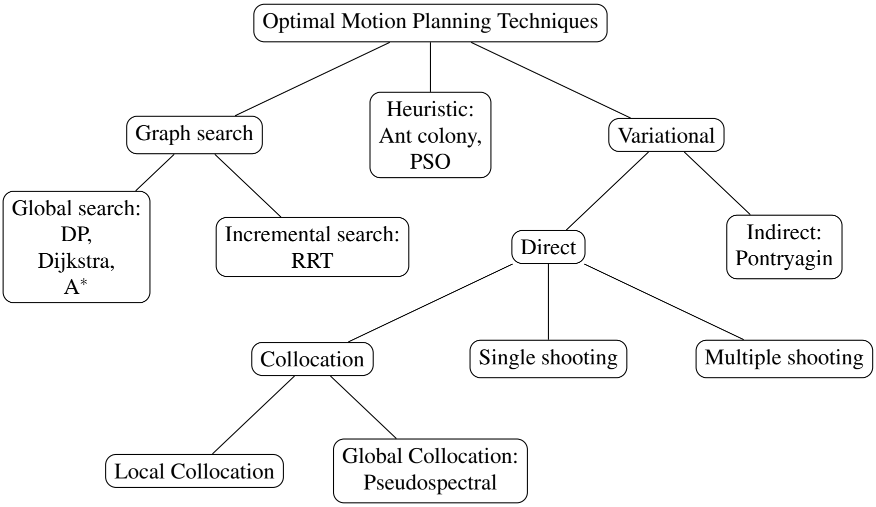


Fig. 3.6 Numerical methods for optimal motion planning

mum cost motion. Dijsktra [54], A* [55] and its variants, and Dynamic Programming (DP) [56] are among graph search methods. We will describe Dijkstra's algorithm in more detail in Sect. 5.1.2.1 in the context of eco-routing (mission planning) and DP in Sect. 6.2.2.2 in the context of eco-driving (motion planning).

A popular incremental search method is the Rapidly-exploring Random Tree (RRT) algorithm [57] designed to efficiently search nonconvex, high-dimensional spaces by randomly growing a space-filling tree in the reachable set of the vehicle. RRT algorithm is suited to problems with obstacles and differential constraints and is therefore widely used in robotic motion planning.

Heuristic methods such as ant colony optimization [58] and particle swarm optimization [59] have also been employed for path planning of autonomous agents and robots. A schematic of these categorizations is shown in Fig. 3.6.

*
* *

In the rest of this book, the main focus in on higher level decisions at Mission Planning, Mode Planning, and Motion Planning layers. Readers interested in Motion Control and Powertrain Control may refer to many articles and books that exist on vehicle control such as [60].

References

1. USDOT. U.S. (2013) Department of Transportation releases policy on automated vehicle development. https://www.transportation.gov/briefing-room/us-department-transportation-releases-policy-automated-vehicle-development. Accessed 30 May 2013
2. SAE International (2018) Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Surface vehicle information report J3016. Technical report, SAE International
3. Committee DSRC et al (2009) Dedicated short range communications (DSRC) message set dictionary. SAE Standard J 2735:2015
4. Li YJ (2010) An overview of the DSRC/WAVE technology. In: Proceedings of international conference on heterogeneous networking for quality, reliability, security and robustness, pp 544–558. Springer
5. USDOT. IEEE 1609 - family of standards for wireless access in vehicular environments (WAVE) (2009). https://www.standards.its.dot.gov/Factsheets/Factsheet/80
6. USDOT. SAE J2735 - Dedicated Short Range Communications (DSRC) message set dictionary (2009). https://www.standards.its.dot.gov/Factsheets/Factsheet/71
7. Araniti G, Campolo C, Condoluci M, Iera A, Molinaro A (2013) Lte for vehicular networking: a survey. IEEE Commun Mag 51(5):148–157
8. Energetics Incorporated and INC Z (2017) Study of the potential energy consumption impacts of connected and automated vehicles. https://www.eia.gov/analysis/studies/transportation/automated/pdf/automated_vehicles.pdf
9. Galileo GNSS. The path to high GNSS accuracy (2018). https://galileognss.eu/the-path-to-high-gnss-accuracy/
10. Van Brummelen J, O'Brien M, Gruyer D, Najjaran H (2018). Autonomous vehicle perception: the technology of today and tomorrow. Transp Res Part C: Emerg Technol 89:384–406
11. Pendleton SD, Andersen H, Du X, Shen X, Meghjani M, Eng YH, Rus D, Ang MH (2017) Perception, planning, control, and coordination for autonomous vehicles. Machines 5(1):6
12. Sivaraman S, Trivedi MM (2013) Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Trans Intell Transp Syst 14(4):1773–1795
13. Rasshofer RH, Gresser K (2005) Automotive radar and lidar systems for next generation driver assistance functions. Adv Radio Sci 3(B. 4):205–209
14. Dhar SP (2017) From google to tesla, it's a war of LiDAR or RADAR. https://www.unitedlex.com/news-and-insights/blog/2017/google-tesla-it%E2%80%99s-war-lidar-or-radar
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of conference on computer vision and pattern recognition, pp 779–788. IEEE
16. Chen C, Seff A, Kornhauser A, Xiao (2015) Deepdriving: learning affordance for direct perception in autonomous driving. In: Proceedings of international conference on computer vision, pp 2722–2730. IEEE
17. Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J et al (2016) End to end learning for self-driving cars. arXiv:1604.07316
18. Da X, Hartley R, Grizzle JW (2017) Supervised learning for stabilizing underactuated bipedal robot locomotion, with outdoor experiments on the wave field. In: Proceedings of international conference on robotics and automation (ICRA), pp 3476–3483. IEEE
19. Kamal MAS , Mukai M, Murata J, Kawabe T (2011) Ecological driving based on preceding vehicle prediction using MPC. IFAC Proc Vol 44(1):3843–3848
20. Kamal MAS, Mukai M, Murata J, Kawabe T (2013) Model predictive control of vehicles on urban roads for improved fuel economy. IEEE Trans Control Syst Technol 21(3):831–841
21. Wiest J, Höffken M, Kreßel U, Dietmayer K (2012). Probabilistic trajectory prediction with gaussian mixture models. In: Proceedings of intelligent vehicles symposium (IV), pp 141–146. IEEE

22. Havlak F, Campbell M (2014) Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. IEEE Trans Robot 30(2):461–474
23. Hermes C, Wohler C, Schenk K, Kummert F (2009) Long-term vehicle motion prediction. In: Proceedings of intelligent vehicles symposium, pp 652–657. IEEE
24. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. MIT Press
25. Quddus MA, Ochieng WY, Noland RB (2007) Current map-matching algorithms for transport applications: state-of-the art and future research directions. Transp Res Part C: Emerg Technol 15(5):312–328
26. Cui Y, Ge SS (2003) Autonomous vehicle positioning with GPS in urban canyon environments. IEEE Trans Robot Autom 19(1):15–25
27. Levinson J, Montemerlo M, Thrun S (2007) Map-based precision vehicle localization in urban environments. In: Proceedings of robotics: science and systems conference, vol 4, p 1
28. Seif HG, Hu X (2016) Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. Engineering 2(2):159–162
29. Google (2018) Google maps platform. https://cloud.google.com/maps-platform/
30. HERE (2019) HERE APIs. https://developer.here.com/develop/rest-apis
31. HERE (2018) INRIX API documentation. http://docs.inrix.com/
32. Yahoo (2018) Yahoo weather API. https://developer.yahoo.com/weather/?guccounter=1
33. Amazon (2018) Amazon Web Services for connected vehicles and mobility. https://aws.amazon.com/automotive/connected-vehicles/
34. Amazon (2018). Amazon Web Services for ADAS and autonomous driving. https://aws.amazon.com/automotive/autonomous-driving/
35. Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2016) Route planning in transportation networks. In: Algorithm engineering, pp 19–80. Springer
36. Aeberhard M, Rauch S, Bahram M, Tanzmeister G, Thomas J, Pilat Y, Homm F, Huber W, Kaempchen N (2015) Experience, results and lessons learned from automated driving on germany's highways. IEEE Intell Transp Syst Mag 7(1):42–57
37. 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
38. Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E (2016) A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans Intell Veh 1(1):33–55
39. Ioannou P, Xu Z (1994) Throttle and brake control systems for automatic vehicle following. IVHS J 1(4):345–377
40. Astrom KJ, Rundqwist L (1989) Integrator windup and how to avoid it. In: Proceedings of American control conference, pp 1693–1698. IEEE
41. Huang S, Ren W (1999) Vehicle longitudinal control using throttles and brakes. Robot Auton Syst 26(4):241–253
42. Yanakiev D, Kanellakopoulos I (1996) Speed tracking and vehicle follower control design for heavy-duty vehicles. Veh Syst Dyn 25(4):251–276
43. Kato S, Tsugawa S, Tokuda K, Matsui T, Fujii H (2002) Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. IEEE Trans Intell Transp Syst 3(3):155–161
44. Vahidi A, Stefanopoulou A, Peng H (2006) Adaptive model predictive control for co-ordination of compression and friction brakes in heavy duty vehicles. Int J Adapt Control Signal Process 20(10):581–598
45. Wallace RS, Stentz A, Thorpe CE, Moravec HP, Whittaker W, Kanade T (1985) First results in robot road-following. In: Proceedings IJCAI workshop, pp 1089–1095
46. Snider JM et al (2009) Automatic steering methods for autonomous automobile path tracking. Robotics Institute, Pittsburgh, PA, Technical Report CMU-RITR-09-08
47. Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G et al (2006) Stanley: the robot that won the DARPA grand challenge. J Field Robot 23(9):661–692

48. Peng H, Tomizuka M (1993) Preview control for vehicle lateral guidance in highway automation. J Dyn Syst Meas Control 115(4):679–686
49. Falcone P, Borrelli F, Asgari J, Tseng HE, Hrovat D (2007) Predictive active steering control for autonomous vehicle systems. IEEE Trans Control Syst Technol 15(3):566–580
50. Dollar RA, Vahidi A (2018) Predictively coordinated vehicle acceleration and lane selection using mixed integer programming. In: Proceedings of dynamic systems and control conference, pp V001T09A006–V001T09A006. American Society of Mechanical Engineers
51. Hrovat D, Di Cairano S, Tseng HE, Kolmanovsky IV (2012) The development of model predictive control in automotive industry: survey. In: Proceeding of international conference on control applications, pp 295–302. IEEE
52. Kelly M (2017) An introduction to trajectory optimization: how to do your own direct collocation. SIAM Rev 59(4):849–904
53. Ross IM, Karpenko M (2012) A review of pseudospectral optimal control: From theory to flight. Ann Rev Control 36(2):182–197
54. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271
55. Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 4(2):100–107
56. Bellman R (2013) Dynamic programming. Courier Corporation
57. LaValle SM (1998) Rapidly-exploring random trees: a new tool for path planning. Technical report
58. Porta Garcia MA, Montiel O, Castillo O, Sepúlveda R, Melin P (2009) Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. Appl Soft Comput 9(3):1102–1110
59. Zhang Y, Gong D-W, Zhang J-H (2013) Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing 103:172–185
60. Guzzella L, Sciarretta A et al (2007) Vehicle propulsion systems, vol 1. Springer