8.1 Introduction

Accompanying the introduction of the European navigation satellite system “Galileo,” possibilities of Global Navigation Satellite Systems (GNSS) for advanced applications in the field of driver assistance systems are currently discussed.

GNSS-based applications have gained increasing relevance in the field of modern Advanced Driver Assistance Systems (ADAS), as they contribute to higher safety and comfort in road traffic. While classic driver assistance systems such as antilock braking systems (ABS) and electronic stability control (ESC) use onboard sensors, future ADAS developments take the environment into account. In this context, the effective detection of stationary or moving vehicles in the environment is essential.

Classic driver assistance systems actively support the driver. Nevertheless, current systems are restricted by the limited detection range of the sensors employed and are subject to varying environmental conditions. In this context, a significant improvement can be achieved by the additional use of GNSS in combination with vehicle-to-vehicle (V2V) communication, to enable long range detection of the environment.

The European navigation satellite system “Galileo” will have better integrity and availability than current systems. In addition, it will offer improved position accuracy. In this respect, GNSS-based ADAS are subject of current research at the Institute of Automatic Control of RWTH Aachen University.

This chapter is structured as follows: Section 8.2 gives an overview of the test environment in use. Section 8.3 presents an introduction into sensor data fusion of Galileo and onboard sensors using Kalman filters. Subsequently, Sect. 8.4 presents...
two applications of Galileo in ADAS, a Cooperative Adaptive Cruise Control and a Collision Avoidance System.

8.2 Test Environment Aldenhoven Testing Center and automotiveGATE

For the testing of safety-relevant advanced driver assistance systems, which can influence the driving behavior of the vehicle, a dedicated automotive testing center is a very important development component. Due to the fact that the automotive manufacturers themselves operate most of the European automotive testing centers, it is very difficult for small and medium enterprises or research facilities to get access to these testing centers. Furthermore, there exists no possibility to test GNSS-based systems in real vehicles under controllable conditions. For these reasons, RWTH Aachen University developed the idea of a manufacturer-neutral automotive testing center in combination with a Galileo testing environment.

The automotiveGATE consists of six terrestrial transmitting antennas, so-called pseudolites. These pseudolites simulate the signals of the European satellite navigation system Galileo within the area of the Aldenhoven Testing Center. The precision of position measurement is up to 0.8 m. The automotiveGATE offers the possibility to test Galileo-based applications independently of the real Galileo satellites. The signals of the automotiveGATE can be manipulated. It is, for example, possible to investigate the influence of different levels of position accuracy on the newly developed Galileo-based driver assistance systems. One additional advantage of this center is that it allows for testing applications under preassigned and reproducible conditions.

In Fig. 8.1, the different track elements of the Aldenhoven Testing Center (ATC) and the positions of the pseudolites of the automotiveGATE are depicted. The ATC provides all necessary track elements for automotive tests. In detail, these are the oval circuit, the handling track, the braking test track, the vehicle dynamics area, the rough road, the hill section, and the highway. For example, the oval circuit has a length of 2 km and the vehicle dynamics area has a diameter of 210 m. Especially, the vehicle dynamics area is ideal for the testing of advanced driver assistance systems, as it is possible to set up arbitrary traffic scenarios.

The combination of the ATC and the automotiveGATE is unique and gives the possibility to test not only standard driver assistance systems but also advanced driver assistance systems, which use position, velocity, and time information (PVT) of different road users under controllable conditions.

The automotiveGATE makes it possible to develop Galileo-based applications before the satellite system is in full operation. The Aldenhoven Testing Center in combination with the automotiveGATE provides ideal conditions for the development of Galileo-based driver assistance systems, which are described in the following sections.
8.3 Galileo-Based Sensor Fusion

In this chapter, certain fundamental characteristics of Galileo-based (or more generally GNSS-based) driver assistance systems are described. Also, an introduction is given on how these characteristics need to be considered when implementing a GNSS-based control system. More specifically, aspects that will be addressed are the update rates of typical GNSS sensors, delayed input data, and the need for sensor fusion.

8.3.1 GNSS Characteristics

One aspect that almost inevitably needs to be addressed when using GNSS data in control systems is the need for sensor fusion. Unlike other sensors used in automotive control systems, a GNSS sensor needs to be treated as unreliable. Although the position of highly specialized GNSS sensors can reach the order of centimeters or even less, the achievable accuracy is highly situation-dependent. The accuracy can degrade, for example, due to an insufficient number of satellites in view or multipath effects. Also, situations in which no or only imprecise GNSS information is available are manifold, such as tunnel or car-park driving. Obstacles along the road, especially high buildings (“urban canyon”), can also dramatically reduce accuracy. Even in best conditions, a GNSS sensor is subject to a startup acquisition time; therefore, one cannot rely on the signal to be readily available at system start. Once in proper operations, two other aspects are very common for GNSS sensors. For one, data is typically output at a relatively low sample rate of approximately 5–10 Hz. Also, GNSS sensors need a certain computation time in order to calculate a position from the acquired satellite signals. This time delay is very often not negligible and can be on the order of tenths of a second.
8.3.2 Sensor Fusion

Sensor fusion is a very common and useful means in order to overcome restrictions arising from the characteristics mentioned above. For this, the GNSS signal is augmented using onboard sensors, such as accelerometers, gyroscopes, or wheel speed sensors. These sensors are highly reliable and provide information at a high update rate. However, this information is only incremental, such that no absolute position can be calculated from onboard sensors alone. Also, due to error integration, navigation results based on onboard sensors alone are prone to drift. In a typical sensor fusion implementation for navigation applications, GNSS data is used to provide an absolute position measurement whereas the onboard sensor data is used for “interpolation.” That way, higher update rates can be achieved. Also, the aforementioned sensor delay can be accounted for. Furthermore, it is possible to provide valid position information for short outages of the GNSS sensor.

From an algorithmic point of view, a predictor–corrector structure is well suited to implement sensor fusion tasks. For this, a dynamic model, e.g., in state space form, can be used. In the following, the indices \( k \) and \( k-1 \) are used to denote the time step. For example, \( x_k \) and \( x_{k-1} \) denote the system states at time steps \( k \) and \( k-1 \). In analogy, \( u_{k-1} \) denotes the input at time step \( k-1 \) and \( y_k \) the (measurement) output at time step \( k \). \( A_{k-1} \), \( B_{k-1} \), and \( C_k \) describe the system dynamics as well as input and output behavior; no feedthrough is considered:

\[
\begin{align*}
    x_k &= A_{k-1} x_{k-1} + B_{k-1} u_{k-1} \\
    y_k &= C_k x_k
\end{align*}
\]  

(8.1)

The functionality of the predictor–corrector structure can be outlined as follows. First, the model is used to predict the current state and outputs of the system based on the state of the system at the last time step as well as on known inputs (prediction step).

\[
\begin{align*}
    \hat{x}_k^- &= A_{k-1} \hat{x}_{k-1}^+ + B_{k-1} u_{k-1} \\
    \hat{y}_k^- &= C_k \hat{x}_k^-
\end{align*}
\]  

(8.2)

Then, the outputs of the system model are compared to measured system outputs, feeding back the difference in order to correct the estimated model state (correction step):

\[
\begin{align*}
    \hat{x}_k^+ &= \hat{x}_k^- + K_k (\hat{y}_k - \hat{y}_k^-)
\end{align*}
\]  

(8.3)

Here, \( \hat{y}_k \) denotes the measured system outputs and \( K_k \) the feedback gain for the correction. Please note that the nomenclature does not use the “actual” state \( x_k \). Instead, an “estimated” state \( \hat{x}_k \) is used. Furthermore, there is a distinction between the estimated state \( \hat{x}_k^- \) and output \( \hat{y}_k^- \) before (“a-priori”) and the estimated state \( \hat{x}_k^+ \) and output \( \hat{y}_k^+ \) after (“a-posteriori”) the most current measurement values have been used to correct it.
8.3.3 Kalman Filter, Extended Kalman Filter

Many methods and variations exist on how the actual prediction and correction steps are implemented. Among the most popular methods are the Kalman Filter [1] and an extension of it, the Extended Kalman Filter (EKF).

A Kalman filter does not only estimate the system state itself but also keeps track of the uncertainty of these estimates in form of an (estimated) covariance matrix \( P_k \) of the system state:

\[
P_K = E \left[ (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \right]
\]  

(8.4)

\( P_K \) describes the expected probability distribution of the estimation error. This covariance increases every time a prediction is performed (as more insecurity is introduced through the prediction) and drops every time a new measurement value is used to correct the system state. A thorough introduction into Kalman filtering can be found in [2].

For the design of a Kalman Filter, a system as well as a measurement model of the form

\[
\begin{align*}
    x_k &= A_{k-1} x_{k-1} + B_{k-1} u_{k-1} + w_{k-1} \\
    y_k &= C_k x_k + v_k
\end{align*}
\]  

(8.5)

is used. Here, \( w_{k-1} \) and \( v_k \) are white noise disturbances that are assumed to be acting on the process and its measurement output. Their noise levels are quantified using the Process Noise Covariance Matrix \( Q_k \) and the Measurement Noise Covariance Matrix \( R_k \):

\[
E \left( w_i, w_j^T \right) = \begin{cases} Q_k, & i = j \\ 0, & i \neq j \end{cases} \]  

(8.6)

\[
E \left( v_i, v_j^T \right) = \begin{cases} R_k, & i = j \\ 0, & i \neq j \end{cases} \]  

(8.7)

The Kalman Filter Gain \( K_k \) to be used in the correction step of the filter according to Eq. (8.3) is then given as

\[
K_k = P_{k}^+ C_k^T \left( C_k P_{k}^- C_k^T + R_k \right)^{-1}
\]  

(8.8)

The update of the covariance estimation is performed as

\[
P_{k}^- = A_{k-1} P_{k-1}^+ A_{k-1}^T + Q_{k-1}
\]  

(8.9)
and the correction as

\[ P_k^+ = (I - K_k C_k) P_k^- \] (8.10)

Given the assumptions of a linear system and that the process and measurement noise are correctly quantified using the white noise assumption, the Kalman filter is an optimal iterative estimator. That means it is not possible to find an iterative filter that can provide a better approximation of the system state [3].

Of course, it is seldom possible to perfectly describe a practical system using a linear model as described above. Furthermore, often neither the white noise assumption holds nor is it always possible to obtain a precise quantification of the disturbances. Still, this type of filter is very powerful and therefore used in many applications. Many extensions and variants of the Kalman Filter and other observer algorithms exist that use the predictor–corrector structure. For example, the EKF uses a nonlinear model within the prediction step and a linearization of it within the correction step [4], allowing to consider certain nonlinear systems. Other filters, such as the Sigma Point filter [5], use a nonlinear model and a sampled probability distribution to improve the covariance update when the system is subject to nonlinearities.

### 8.3.4 Example: Simple 2D Case

In the following, an example of a model for the implementation of a simple filter is used. The filter is based on measurements of the yaw rate, wheel speeds, and a GNSS sensor. Despite being relatively simple, this filter can already be a substantial improvement over using a raw GNSS signal.

A model widely used in vehicle dynamics is the two-track model (see Fig. 8.2). Here, the vehicle is modeled as a rigid body with the four wheels attached to it. The forces \( F_{X,x} \) and \( F_{Y,x} \) acting on each tire are modeled through lateral and longitudinal slip that arises from non-holonomous movement. This model is able to consider effects in lateral dynamics such as over- and understeer as the vehicle heading is uncoupled from the direction of movement of the center of gravity. For applications that do not need to consider lateral dynamics in such detail, a simpler model, such as the simplified Kinematic Single-Track Model (see Fig. 8.3), can be sufficient. Here, perfect holonomous movement is assumed; therefore kinetics as well as tire slip are completely neglected. If the reference point (and therefore ideally the mounting point of the GNSS receiver) is selected as the center of the rear axle, its movement can be described as a circular motion.

The turning radius is determined by the vehicle length \( L \) and the steering angle \( \delta \):

\[ R = \frac{L}{\tan(\delta)} \] (8.11)
The yaw rate $\dot{\psi}$ then results as

$$
\dot{\psi} = \frac{V}{R} 
$$

(8.12)
If a small enough time step $T$ is assumed, even a stepwise linear movement can be used to finally describe the model in a state space form:

$$x_{k+1} = \begin{bmatrix} X \\ Y \\ \psi \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \cos(\psi) \cdot T \\ 0 \\ \sin(\psi) \cdot T \end{bmatrix} u_k$$

$$y_k = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_k \text{ with } x = \begin{bmatrix} X \\ Y \\ \psi \end{bmatrix} \text{ and } u_k = \begin{bmatrix} V \\ \psi \end{bmatrix}$$

The velocity $V$ of the reference point can be taken as the mean speed of the rear wheels whereas a measurement of the yaw rate can be obtained from a (bias-compensated) inertial sensor.

Even if the application does not have high demands on the accuracy, it is advisable to consider the time delay of the GNSS measurements. Although for GNSS sensors, the time stamp of a measurement is known very precisely, it is only available after a (varying) processing time, which can be on the order of tenths of a second. If the measurements are simply used for state correction with no further processing, the system state will be corrected using an up-to-date estimate of the output and a measurement that is several time steps old. An intuitive but effective possibility to handle the delay is to use a ring buffer to save the old system state estimates as calculated by the filter. Then, when a new measurement becomes available, the state estimate corresponding to the actual time of measurement can be obtained from the buffer. More sophisticated approaches to handle the delay are available, for example, as described in [6], where closer attention is paid to not only consider the delayed measurement but also to correctly propagate the covariance estimates.

### 8.3.5 Example: 3D Case

In case of highly dynamic maneuvers, a simple single-track representation does not capture the important dynamics of the vehicle. Thus, a more complex representation has to be chosen.

For a complete description of the vehicle, the position, the velocity, and the attitude of the vehicle have to be estimated in all three dimensions. An estimate can be calculated using the signals provided by an inertial measurement unit (IMU). This IMU uses three accelerometers and three gyroscopes to measure the accelerations in $x$-, $y$-, and $z$-direction and the roll, pitch, and yaw angular rates. By integration of these three accelerations and three rotational speeds, it is possible to predict position, velocity, and orientation. This integration is able to provide calculation results at a high data rate. However, the prediction is only accurate for a short time being prone to drift.
A GNSS receiver provides long-time stable position and velocity information but at a low data rate. This information can be used to estimate the errors $\delta x$ resulting from the integration of the IMU sensor data. Therefore, a Kalman filter in error state space formulation is used. This filter estimates the error covariance matrix in the prediction step and computes the estimation errors $\delta x$ once a new measurement from the GNSS is available. Besides the estimation of the error of position $\delta p$, velocity $\delta v$, and attitude $\delta \epsilon$, the bias of the accelerometers $\delta b_a$ and gyroscopes $\delta b_\omega$ can be estimated in order to correct them online. All in all, the resulting model consists of the 15 state variables

$$
\delta x = \begin{bmatrix}
\delta x_n & \delta x_e & \delta x_d \\
\delta v_n & \delta v_e & \delta v_d \\
\delta \phi & \delta \theta & \delta \psi \\
\delta b_{a,x} & \delta b_{a,y} & \delta b_{a,z} \\
\delta b_{\omega,x} & \delta b_{\omega,y} & \delta b_{\omega,z}
\end{bmatrix}
$$

(8.14)

Linearization of the model leads to a state space model in the form:

$$
\frac{d}{dt} \begin{bmatrix}
\delta p \\
\delta v \\
\delta \epsilon \\
\delta b_a \\
\delta b_w
\end{bmatrix} = F \begin{bmatrix}
\delta p \\
\delta v \\
\delta \epsilon \\
\delta b_a \\
\delta b_w
\end{bmatrix} + G \begin{bmatrix}
n_a \\
n_\omega \\
n_{b,a} \\
n_{b,w}
\end{bmatrix}
$$

(8.15)

with the process noise originating from accelerometer and gyroscope measurements ($n_a$ and $n_\omega$) and sensor biases ($n_{b,a}$ and $n_{b,w}$), respectively. The sensor biases are assumed to follow a random walk process, leading to a zero-mean Gaussian white noise distribution. The complete description of the linearized state space model is omitted due to its complexity. For the complete mathematical description and the derivation, the interested reader is referred to [7] or [6].

### 8.4 Applications Examples

In the following, two examples for systems using of the filters and filter models described in Sect. 8.3 are presented. Both examples describe systems that use fused data originating from a GNSS sensor and onboard sensors. Whereas in the first example (Cooperative Adaptive Cruise Control), the sensor fusion is based on the simpler 2D case, the system in the second example (Collision Avoidance System) is based on the 3D case.
8.4.1 Application 1: Cooperative Adaptive Cruise Control

A possible application of GNSS in ADAS is the extension of Adaptive Cruise Control (ACC) systems for road vehicles to situations where the vehicles cannot locate each other using only onboard sensors. This can, for example, be driving through tight corners and on hilly roads, where radar- or lidar-based sensors reach their geometric limitations or situations where vehicles located far ahead have to be accounted for (e.g., at the end of a traffic jam). This section describes an extension of an ACC system to a GNSS- and Map-Based Cooperative Adaptive Cruise Control (CACC) system for road vehicles. The CACC system implements a distance control based on position data acquired from a GNSS sensor fused with onboard sensors, a digital road map, and vehicle-to-vehicle (V2V) communication. The system is validated in experiments.

Figure 8.4 describes the principle structure of the CACC system. First, each vehicle has to be located on a digital road map. Then, the leading vehicle provides its position data to the following vehicle using V2V communication, enabling it to determine the distance between both vehicles. Additionally, the CACC system uses vehicle data from both the following and the leading vehicle. In the following, we give an overview of the main components distance determination and controller and show experimental results.

8.4.2 Distance Determination Between Two Vehicles

In order to keep a reference distance between two vehicles, the actual distance between them should be determined. However, the Euclidean distance between two vehicles, which is easy to calculate, does not represent the real route distance in between them. The route distance is the actual distance to be travelled on a road network. Therefore, it is the relevant distance concerning the CACC system.

In order to use the map data, the vehicles first need to find the logical equivalent of their measured, physical position on the map. It has to be determined on which map segment and where within this segment the vehicle is located. This process is called map matching. In [9], a map matching algorithm for application in GNSS-based ADAS has been developed that is also applied here.

In the first step of distance determination, the map is converted to a directed line graph $G = (V, E)$. The vertices of the directed line graph represent the map segments. Its edges contain the connection information of the map segments. Two vertices are connected by an edge if their respective map segments are connected and if it is possible to drive from the first segment to the second. It is not possible to drive between two segments if the second segment represents a one-way road in the opposite direction of travel. Each vertex has a cost value, which is assumed here to be its length. The graph is saved in an adjacency matrix.

The shortest route distance between two vertices (map segments) can be determined by the generated directed line graph. This is done using the Dijkstra graph
search algorithm that solves the shortest path problem \[10\]. In addition, the Dijkstra algorithm is modified to apply to directed line graphs and to fulfill the real time requirements of the controller device. The returned cost from the Dijkstra algorithm of the shortest path is the sum of all vertex costs on it. The route distance between two vehicles is determined as

\[
d_{\text{map}} = d_{\text{dijkstra}} - d_{\text{follower}} - d_{\text{leader}} - \frac{t_{\text{follower}}}{2} - \frac{t_{\text{leader}}}{2} \quad (8.16)
\]
where \( d_{\text{dijkstra}} \) is the route distance computed by Dijkstra algorithm, \( d_{\text{follower}} \) is the route distance between the current position of the following vehicle and the start point of its map segment, \( d_{\text{leader}} \) is the route distance between the current position of the leading vehicle and the end point of its map segment, and \( l_{\text{follower}}, l_{\text{leader}} \) are the following vehicle and leading vehicle length, respectively.

8.4.3 Design of the Distance Controller

The distance controller computes a reference acceleration \( a_{\text{ref}} \) using a cascaded controller [11]. The inner loop controller is a proportional controller controlling the velocity, and the outer loop controller is a proportional-integral controller with disturbance feedback controlling the route distance. As the velocity error acts as ramp disturbance on the controlled output \( d_{\text{map}} \), the integral controller is required to achieve steady-state offset-free tracking in the closed loop setup. The CACC system keeps a reference route distance \( d_{\text{ref}} \) according to a constant time gap [12].

8.4.4 Experimental Results

The results are based on measurement data that has been recorded on the test track (Aldenhoven Testing Center) using a Volkswagen Passat CC as the following vehicle and a BMW 7 Series as the leading vehicle. The performance of the proposed CACC system is evaluated using a sensor data fusion of radar and a camera as a reference system in the following vehicle.

The distance determination is evaluated by comparing the GNSS- and map-based route distance with the radar-based distance from the reference system on a straight line (1st plot in Fig. 8.5). Thereby, the radar-based distance and relative velocity are 0 if no target object is selected. The 2nd plot shows the difference between several measurement methods of the relative velocity \( \Delta v = v_{\text{leader}} - v_{\text{follower}} \). Thereby, \( \Delta v_{\text{wheel}} \) (the difference of the raw measurements of the mean wheel velocities) delivers the best availability. Hence, it is used in the CACC system although its accuracy is lower than \( \Delta v_{\text{GNSS}} \) (the difference of the raw values of the velocities measured using GNSS).

To validate the CACC system, the following test scenario is chosen. The leading vehicle initially drives with approximately constant velocity (as far as possible for the driver), since the leading vehicle has no cruise control system. The driver of the following vehicle tries to follow it as possible with the same initial velocity and an initial route distance \( d_{\text{map}} > d_{\text{ref}} \) without control. The CACC system is enabled at \( t = 9 \) s (5th plot). Subsequently, the following vehicle accelerates and joins up to the leading vehicle keeping the reference route distance (3rd and 4th plot). The reference route distance is computed using a time gap of 1.8 s and an additional safety distance of 5 m. The experimental results show the applicability of
the proposed CACC system. Note that the control could also be conducted at times where geometric limitations did not allow for valid distance measurement using the onboard sensors.

### 8.4.5 Application 2: Collision Avoidance System

The task of a Collision Avoidance System (CAS) is to observe the surroundings of the vehicle and to perform an autonomous emergency braking or evasion maneuver in case of an imminent collision in order to avoid the collision or mitigate collision damage. The emergency maneuver is started once it is clear that the driver does not react appropriately in time. This leads to an intervention of the system at the last possible moment. Thus, the algorithm must be able to guide the vehicle around the obstacle while driving at the vehicle handling limits.

A collision avoidance system can generally be divided into several different components:

- **Sensor fusion**: In order to realize reliable collision avoidance, the position, the velocity, and attitude of the ego-vehicle is needed. Since the evasion maneuver will be highly dynamical, a 3D approach as described in Sect. 8.3.5 is needed.
Environment recognition and collision detection: The CAS needs proper information about the position of possible collision targets in order to avoid them reliably. One possibility to get that information is the use of environmental sensors such as camera, LiDAR, or radar sensors. Another possibility is the combination of GNSS systems with digital road maps and vehicle-to-vehicle (V2V) communication; see [13, 14]. The usage of such a combination can lead to a significant improvement of detection ranges.

Maneuver coordination: Once a collision is imminent, the CAS needs to decide on a possible maneuver to avoid the collision. This decision includes the choice between braking and evading and the best time of a driver warning.

Path/trajectory planning: For an evasion maneuver, a feasible and collision-free evasion path has to be found.

Vehicle Control: Once the evasion path is known, the task of the controller is the longitudinal and lateral guidance along the evasion path.

In the following, it is assumed that a collision is imminent. It is further assumed that an appropriate evasion path is computed in order to concentrate on the vehicle control.

Since the evasion path is computed directly and is thus known, a model predictive control scheme is chosen [15]. The main advantage of a model predictive controller is that limitations from the actuator dynamics and vehicle stability can directly be taken into account. A model predictive controller uses a mathematical plant model to predict the future outputs $y(k + i|k)$, $i = 1, \ldots, H_p$ of the plant over a finite prediction horizon $H_p$. The control inputs $u(k + j|k)$, $j = 1, \ldots, H_u$ are then optimized over a finite control horizon $H_u$ such that the deviation of the outputs from a specified reference trajectory $r(k + i|k)$, $i = 1, \ldots, H_p$ is minimized.

For the optimization, a quadratic cost function of the form

$$J(k) = \sum_{i=0}^{H_p-1} \left\| y(k + i|k) - r(k + i|k) \right\|^2 + \sum_{i=0}^{H_u-1} \left\| \Delta u(k + i|k) \right\|^2_{R(i)} \quad (8.17)$$

is used, where $Q(i)$ and $R(i)$ penalize deviations of the control outputs from the reference and changes in the control input, respectively. In this regard, the notation $(k + i|k)$ indicates that the future value of a variable is predicted for time $k + i$ at time $k$. The minimization of the cost function gives a sequence of optimal input steps $\Delta u_{opt}(k|k), \ldots, \Delta u_{opt}(k + H_u - 1|k)$. The control input $u_k = u_{k-1} + \Delta u_{opt}(k|k)$ is applied to the plant, and the optimization is repeated with a shifted prediction horizon. This principle is called receding horizon.

In order to apply the predictive control scheme, an appropriate plant model for the vehicle dynamics and the relative kinematics of the vehicle and the evasion path is needed. Figure 8.6 shows a principle sketch of the single-track model which is used.
All in all, the resulting nonlinear prediction model can be written in state space representation as

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t), z(t), \theta(t)) \\
y(t) &= g(x(t)) = \Delta y
\end{align*}
\] (8.18)

In that model, the state vector

\[
x^T = [v_x \ v_y \ \dot{\psi} \ \Delta \psi \ \Delta y \ d_{\Delta y}] 
\] (8.19)

consists of the lateral and longitudinal velocity \(v_x\) and \(v_y\), the yaw rate \(\dot{\psi}\), the actual steering angle \(\delta\), the relative yaw angle between the vehicle’s center of gravity and the evasion path \(\Delta \psi\), the lateral displacement \(\Delta y\) from the evasion path, and the lateral velocity disturbance \(d_{\Delta y}\). The control input \(u\) is the demanded steering angle \(\delta_{\text{ref}}\). The path’s curvature \(k\) is assumed as a known disturbance value. Additionally, an adaption parameter \(\theta(t)\) is added to account for unknown changes in tire-road contact.

Figure 8.7 shows the control results for a double lane change maneuver. This corresponds to the evasion of a standing obstacle in a scenario with oncoming traffic. The initial velocity of the vehicle is approximately 15.5 m/s.
It is visible that the controller follows the evasion path well with a maximum absolute lateral deviation of 0.33 m. Best control results were achieved with a prediction horizon of about 1 s and a control horizon of about 0.5 s. Furthermore, a sample time of 0.02 s is chosen. Simulations have shown that shorter horizons lead to a degraded control performance which may even lead to stability problems for very short horizons.

**8.5 Conclusion**

This contribution presented a test infrastructure that can be used to develop Galileo-based Advanced Driver Assistance Systems. The infrastructure consists of the Aldenhoven Testing Center (ATC) in combination with a pseudolite system that provides a Galileo Test and Development Environment (automotiveGATE). This infrastructure allows company-independent research on Galileo-based control systems. An overview is given on characteristics that are imminent to GNSS-based control systems and how some of the arising issues, such as measurement delay and low update rates, can be addressed using sensor fusion techniques.
Then, implementations of two GNSS-based ADAS applications developed using the presented infrastructure were presented. The first system uses navigation data in combination with a digital road map as well as V2V communication in order to extend the range and functionality of an Adaptive Cruise Control system. The second system performs an emergency Collision Avoidance maneuver based on GNSS and inertial data.

Acknowledgement

The presented research work has been performed within the projects “Galileo above” and “GALILEO-basierte Assistenzsysteme.”

The project “Galileo above” is sponsored by the Space Agency of the German Aerospace Centre (DLR) with funding by the Federal Ministry of Economics and Technology, in compliance with a resolution of the German Parliament (project/grant no. 50 NA 0902).

The project “GALILEO-basierte Assistenzsysteme” is funded by the Ministry of Economics, Energy, Industry, and Small Business of the State of North Rhine-Westphalia and the European Union with Financing of the European Regional Development Fund (ERDF).

The idea of a manufacturer neutral testing center (Aldenhoven Testing Center) was realized with the help of the county of Düren and fundings by the State Government of North Rhine-Westphalia and the European Union. The idea of the Galileo testing center for automotive applications (automotiveGATE) was implemented within the aforementioned project “Galileo above” by the funding of the DLR Space Administration with funds of the Federal Ministry for Economic Affairs and Energy. Both facilities explicitly guarantee the possibility to be used especially by small and medium enterprises and research facilities. All other customers are also welcome.

References

1. R.E. Kalman, A new approach to linear filtering and prediction problems. J. Basic Eng. 82(1), 34–45 (1960)
2. M.S. Grewal, A.P. Andrews, Kalman filtering: Theory and Practice with MATLAB (Wiley, Hoboken, NJ, 2014)
3. P.S. Maybeck, The Kalman filter: An introduction to concepts, in Autonomous Robot Vehicles, ed. by I.J. Cox, G.T. Wilfong (Springer, New York, 1990), pp. 194–204
4. H.W. Sorenson (ed.), Kalman Filtering: Theory and Application (IEEE Press, New York, 1985)
5. R. van Der Merwe, E.A. Wan, Sigma-Point Kalman Filters for Integrated Navigation, in Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION), pp. 641–654, 2004
6. J. Wendel, Integrierte Navigationssysteme (Oldenbourg Verlag, Munchen, 2011)
7. A. Katriniok, Optimal Vehicle Dynamics Control and State Estimation for a Low-Cost GNSS-based Collision Avoidance System, Ph.D. thesis, RWTH Aachen University, Aachen, Germany, 2014
8. B. Alrifaee, M. Reiter, J. P. Maschuw, F. Christen, L. Eckstein, D. Abel. Satellite- and Map-based Long Range Cooperative Adaptive Cruise Control System for Road Vehicles, in AAC 2013, 7th IFAC Symposium on Advances in Automotive Control, pp. 732–737, Tokyo, Japan, Sept 2013
9. M. Reiter, D. Abel, D. Gulyas, C. Schmidt. Konzept, Implementierung und Test eines echtzeitfähigen Map-Matching-Algorithmus für Anwendungen in GNSS-basierten Fahrerassistenzsystemen, in AUTOREG, 5, pp. 389–400, Baden-Baden, Nov 2011
10. E.W. Dijkstra, A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
11. R.C. Dorf, R.H. Bishop, Modern Control Systems, 9th edn. (Prentice-Hall, Upper Saddle River, NJ, 2000)
12. R. Rajamani (ed.), Vehicle Dynamics and Control (Springer, New York, 2005)
13. R. Schubert, M. Schlingelhof, H. Cramer, G. Wanielik, Accurate Positioning for Vehicular Safety Applications—The SAFESpot Approach, in IEEE Vehicular Technology Conference, pp. 2506–2510, 2007
14. P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, S. Cosenza, Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. IEEE Commun. Mag. 47(11), 84–95 (2009)
15. J. Maciejowski, Predictive Control with Constraints (Prentice Hall, Harlow, 2002)