IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 May 2024, accepted 20 July 2024, date of publication 30 July 2024, date of current version 8 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3435668

RESEARCH ARTICLE

Optimizing Lane Change Precision in Autonomous Vehicles: Smooth Trajectory Execution With MPC-Based State Estimation

HEYI MULUNEH HAILU[1], SALIM HIMA[2], AND AHMED CHAIBET[1]
[1]DRIVE Laboratory, Université de Bourgogne, 58027 Nevers, France
[2]ESME Research Lab, 38 rue Molière, 94200 Ivry-sur-Seine, France

Corresponding author: Heyi Muluneh Hailu (Muluneh-Hailu_Heyi@etu.u-bourgogne.fr)

This work was supported in part by French Government Scholarship for Higher Education and Research and DRIVE Laboratory, Université de Bourgogne.

ABSTRACT Autonomous vehicle lane change maneuvers are an intricate task that requires multiple subsystems to work together to ensure the maneuver's safety and efficiency. This research paper investigates lane change maneuvers of autonomous vehicles, utilizing a trajectory generated using a sinusoidal function considering the ISO 3888 standards. The driving maneuver steering angle input is derived from the generated trajectory, which is adjusted using model predictive control (MPC) to calculate an optimal steering angle for lateral movement and manage throttle input to maintain longitudinal stability. Throughout lane change maneuvers, the vehicle's states are estimated using model-based Kalman filters, relying on input and measurement data from inertial sensors. The paper compares four state estimator filters: Standard Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter, and Adaptive Unscented Kalman Filter. The implementation of the lane change trajectory generator, MPC algorithm, and state estimators within MATLAB/Simulink, validated through IPG CarMaker, highlights the Adaptive Unscented Kalman Filter as the optimal choice for lane change state estimation. Its adaptive covariance adjustment sets it apart from the other filters under examination.

INDEX TERMS Autonomous vehicle, Kalman filter, lane change, lateral control, longitudinal control, state estimation, trajectory generation.

I. INTRODUCTION

In recent years, technological advancements in autonomous vehicles have shown tremendous growth in bringing fully autonomous vehicles into reality. As the achievements indicate, fully automated self-driving cars will become pervasive in the coming few years [1], [2]. The main priority when navigating with autonomous vehicles is to guarantee passenger safety and ensure collision-free maneuvering toward the destination. Even today, the majority of vehicles are operated by humans, who are tasked with making decisions to prevent collisions while adhering to established standards.

However, humans have limitations in making immediate and accurate decisions to avoid collisions and efficiently utilize resources, such as time, energy, and managing traffic congestion. The imperfections inherent in human driving behaviors result in a tragic loss of life and valuable resources. Transitioning to fully autonomous vehicles offers a solution by mitigating accidents stemming from human error.

The associate editor coordinating the review of this manuscript and approving it for publication was Wonhee Kim.

By replacing fallible human drivers with Advanced Driver Assistance Systems (ADAS) functions, and computerized self-driving technology, we can dramatically reduce both resource damage and fatalities.

Furthermore, autonomous vehicles have the potential to alleviate traffic congestion and curb pollutant emissions by precisely controlling speed, optimizing routes, and maintaining safe distances from neighboring vehicles [3], [4].

In the realm of self-driving cars, ensuring passenger safety by preventing collisions remains a critical concern. Lane changing frequently occurs in both highway and urban driving, which presents a significant challenge.

Vehicles often change lanes for various reasons, including overtaking slower vehicles, yielding to faster ones, or avoiding obstacles [5]. Thus, making a perfect lane change increases the safety of the passengers. Making accurate lane changes depends on the accuracy of the basic components of autonomous vehicles. Localization, trajectory planning, and vehicle control are some of the essential subsystems that affect the decision-making process to put the vehicle on the right track by avoiding collisions.

The localization component estimates the vehicle's state using input from the sensors and the vehicle model. The state estimator helps the vehicle accurately estimate the next states, such as the vehicle position, orientation, speed, and yaw rate, based on the current state, the noisy measured input from sensors, the control inputs, and the vehicle model.

Accurate estimation of the vehicle's state is an important factor in making perfect decisions for taking further action and maintaining the safety of passengers. In [6], state estimators are classified into two methodologies: the model-based estimation approach and the data-driven estimation approach. The model-based technique relies on the vehicle's kinematic or dynamic model to estimate its states, establishing direct correlations between measurements and estimated states. In contrast, the data-driven estimation approach employs artificial neural networks to estimate the vehicle's state. The main contribution of this work is to develop a model-based state estimation algorithm. These algorithms aim to accurately estimate the state of an autonomous vehicle during the lane change process by using control inputs managed by MPC, enabling the vehicle to follow the generated lane change trajectory.

Additionally, the proposed approach relies on advanced vehicle state estimators, primarily based on Kalman filter estimation techniques. These estimators predict and update the system's state and error covariance matrix recursively, allowing for the most precise estimations possible. Additionally, to figure out the best state estimator algorithm for lane change in autonomous vehicles, a comparative study of four state estimators (KF, EKF, UKF, AUKF) is conducted to determine the best state estimation algorithm for lane change in autonomous vehicles, taking into account their accuracy and computational complexity.

The control inputs for the state estimator are derived from a lane change trajectory generator, which combines sinusoidal and linear functions to comply with ISO 3888 standards. The generated trajectory provided as a reference input to the controller to generate the control inputs for precise lane change maneuvering. In this research, the MPC method is specifically chosen due to its optimal control nature and its capability to handle multiple input and output controls effectively.

Integrated with the state estimators, the MPC system precisely manages steering angle and throttle inputs, ensuring accurate adherence to the designated route for safe lane changes. MPC offers the advantage of simultaneously controlling both lateral and longitudinal inputs, with changes in lateral motion directly influencing longitudinal motion parameters [7], [8], [9]. In the context of lane changes, MPC computes the appropriate steering angle and acceleration, ensuring stability in longitudinal motion.

In general, the overall lane change system architecture is illustrated in Figure 1 and comprises three main modules: the lane change trajectory generator, the state estimator, and the controller. The trajectory generator module is utilized to generate the reference lane change trajectory by combining sinusoidal and linear functions, aiming for an optimal lane change. Meanwhile, the state estimator module accurately predicts the vehicle's future trajectory state by leveraging the vehicle's kinematic model and control input.

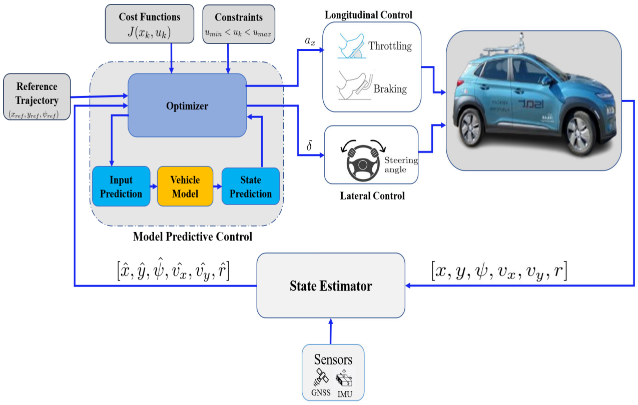

FIGURE 1. Overall system architecture.

In the subsequent sections, we provide a comprehensive overview of the trajectory generator for lane changes, the vehicle model, state estimators, and the controller design. Section II presents how the lane change trajectory is generated, while Section III addresses the vehicle model. Section IV examines state estimator algorithms, and Section V provides a comprehensive overview of MPC methodology. Results and discussion are presented in Section VI, followed by concluding remarks in Section VII.

II. LANE CHANGE TRAJECTORY GENERATOR

Generating an optimal lane change trajectory ensures an efficient and smooth lane change maneuver. Before initiating the lane change process, the autonomous vehicle plans its trajectory based on its current state as well as the state of the surrounding vehicle or potential obstacles. The trajectory set points generated from this unit will be used as reference paths in real-time motion planning of a lane change.

Existing literature illustrates various techniques employed to generate lane change trajectories. For instance, a quintic polynomial for trajectory planning was implemented by [10] to optimize jerk, and a 5th polynomial trajectory was utilized by [11] to determine the virtual reference lane. The polynomial describes the lateral position as a function of the vehicle's longitudinal position for a lane change. However, quintic polynomial has limitations in handling real-time, dynamic, and complex environments making them less suitable for autonomous vehicle lane change planning compared to more adaptive and flexible approaches. In addition, quintic polynomials require solving a system of equations to determine the coefficients, which can be computationally intensive. This complexity increases with real-time planning requirements.

An analysis of real driving data from various drivers during lane changes was conducted by [12], resulting in a set of simplified trajectories representing typical lane change behaviors, facilitating more efficient online computation. Moreover, the lane change process was integrated into a learning module, allowing for the generation of models used in online inference and planning. However, this approach has a limitation since the accuracy of the learning-based method depends on the quality and quantity of human driving data which can significantly impact its performance. Low-quality data or insufficient data can lead to less optimal or unsafe lane change maneuvers.

Dynamic lane change trajectory planning for an autonomous vehicle was conducted in [5], utilizing the Frenet reference frame and cubic polynomial to determine the path and speed adjustments of the autonomous vehicle during lane changes.

However, the reliance of the Frenet reference frame on a predetermined path poses challenges in accurately tracking dynamic targets. To address this limitation, an elliptic limit-cycle trajectory is employed for roadway navigation in the automatic lane change system presented in [13]. This approach involves extracting dynamic set points and defining a unique stable control law to mitigate errors and ensure smoother lane change maneuvers.

In this paper, a safe and optimal lane change trajectory is generated by using a combination of sinusoidal and linear functions to meet ISO 3888 standards for double lane change to overtake the preceding vehicle or avoid obstacles. During the double-lane change, the ego vehicle shifts from its current lane to the adjacent one and then reverts to its original lane after bypassing the preceding vehicle or obstacle.

The sinusoidal and linear functions are integrated to ensure a safe double-lane change, thereby achieving an optimal trajectory to prevent collision with the preceding vehicle or obstacle. The ISO 3888 standard defines the dimensions of test tracks and procedures for evaluating the ability of passenger cars and light commercial vehicles to avoid obstacles. It consists of two parts: one for testing abrupt lane changes (ISO 3888-1) and another for obstacle avoidance maneuvers (ISO 3888-2). The test track is designed to simulate sudden lane changes, featuring a straight section followed by a curve where an obstacle is placed to assess the vehicle's response.

This standard is essential for enhancing vehicle safety by encouraging carmakers to refine their obstacle-avoidance technologies.

The double lane change path, outlined by the ISO 3888 standard, is divided into five sections [14], [15].

In the first segment, the vehicle remains in its lane for a maximum distance of 15 m for ISO 3888-1 and 12 m for ISO 3888-2 before initiating the lane change. During the second phase, the vehicle must transition to a parallel lane within 30 m for ISO 3888-1 or 13.5 m for ISO 3888-2. The third and fourth segments are employed to avoid the obstacle or preceding vehicle and return to the initial lane, respectively.

Finally, a distance of 15 m for ISO 3888-1 and 12 m for ISO 3888-2 is required after returning to the initial lane to stabilize the vehicle. To execute maneuver at the second and fourth segment of lane change maneuver as depicted in Figure 2, the lane change trajectory is computed using a sinusoidal function to ensure a smooth transition from the current lane to the adjacent lane. Where (x_{ei}, y_{ei}) is the initial location of the ego vehicle, (x_{pi}, y_{pi}) is the initial location of the preceding vehicle or obstacle, L_d represents the longitudinal distance between the ego and the preceding vehicle, and b represents a constant value of the lateral distance between the centers of two adjacent lanes.

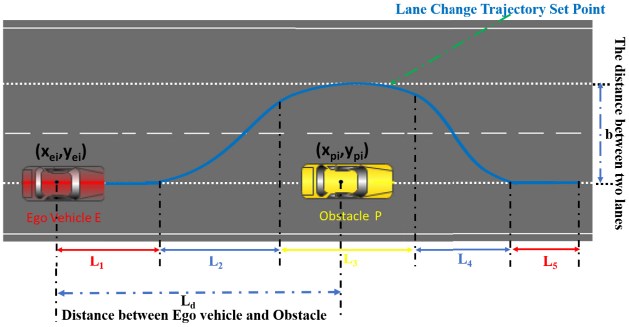
Lane Change Trajectory Set Point
The distance between two lanes
b
(x_{ei}, y_{ei})
(x_{pi}, y_{pi})
Ego Vehicle E
Obstacle P
L_1
L_2
L_3
L_4
L_5
L_d
Distance between Ego vehicle and Obstacle

FIGURE 2. Illustrating parameters for double-lane change trajectory.

In this work, the standard for the width between two adjacent lanes is considered to be 3.5 m. L_1, L_2, L_3, L_4, and L_5 represent the longitudinal distance of the five sections of the double lane change according to ISO 3888. And their lengths for ISO 3888-1 and ISO 3888-2 are described above.

As previously stated, the double-lane change trajectory is generated according to the standard using a combination of linear and sinusoidal functions. The linear function is employed to produce the path in the first, third, and last sections, maintaining the lane. Meanwhile, the sinusoidal function is utilized to create a path for changing lanes to the left and right in the second and fourth sections [16].

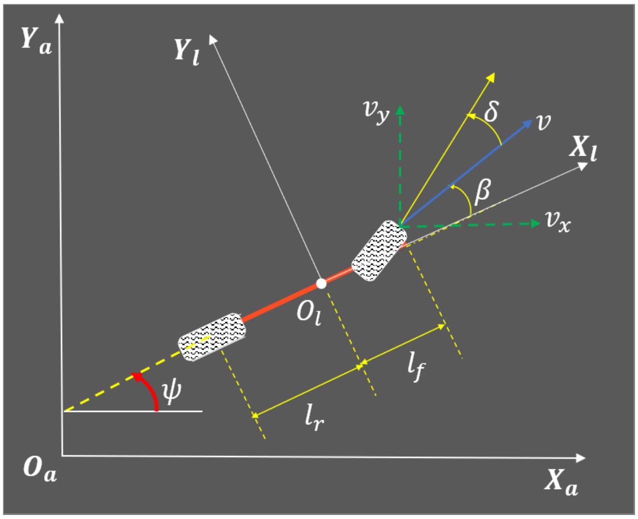

FIGURE 3. Vehicle bicycle model.

Figure 3 depicts this simplified bicycle vehicle model operating within a two-dimensional space.

Equation (1), as shown at the bottom of the next page, specifies the relationship between the lateral and longitudinal displacement trajectory functions for double-lane changes. where: y denotes the lateral displacement, and x represents the longitudinal displacement. The vehicle heading angle is equivalent to the inverse tangent of the slope of the trajectory. However, for small slope (angle) variations, the heading angle of the vehicle (ψ) is approximated to the slope of the trajectory. Hence, the vehicle's heading angle (ψ) is determined by differentiating the lateral displacement (y) with respect to the longitudinal displacement (x), as depicted in equation (2), as shown at the bottom of the next page.

III. VEHICLE MODELING

The accuracy and complexity of the vehicle model greatly affect the lane change process. A more complex model can better represent motion, but it increases controller design and computational costs. A simpler model reduces complexity, but it may not withstand non-linear effects and introduce uncertainty [11], [17], [18]. The model under consideration in this work is a basic bicycle car-like kinematic model to characterize the movement of an autonomous vehicle. This model simplifies by neglecting direct factors such as forces, torque, and inertia that influence motion. By representing a four-wheeled vehicle with only two wheels, one at the front and one at the rear, the model reduces control to managing two wheels and one steering angle.

The state of the vehicle, which describes the condition of the vehicle at time step k during its motion, is represented by the following set of state variables:

$$x(k) = (x(k), y(k), \psi(k), V_x(k), V_y(k), r(k)) \tag{3}$$

At each time step k, $x(k)$, and $y(k)$ respectively represent the abscissa and ordinate of the position of the vehicle's center of gravity. $\psi(k)$ denotes the vehicle's heading angle in the global frame at time step k, while $V_x(k)$ and $V_y(k)$ represent the longitudinal and lateral velocities at the same time step, expressed in the same frame. $r(k)$ is the yaw rate at time step k. The resultant velocity V is the norm of the longitudinal velocity V_x and lateral velocity V_y.

This paper introduces an advanced control strategy aimed at achieving precise lane changes. MPC law is developed to regulate both the longitudinal acceleration (a_x) and steering angle (δ). This ensures stable and precise movements in the longitudinal and lateral directions, respectively. This approach ensures effective lane change management by keeping longitudinal velocity variations under control.

The angle $\beta(k)$ denotes the side slip angle of the vehicle at time step k, calculated using equation (4) where v_x and v_y represent the longitudinal and lateral velocities of the vehicle in a body frame. These velocities in a body frame are then transformed into the global frame using equation (6).

$$\beta(k) = \tan^{-1} \frac{v_y(k)}{v_x(k)} \tag{4}$$

Considering the current state, the motion of the ego vehicle can be expressed using the following discrete state equations:

$$x(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \psi(k+1) \\ Vx(k+1) \\ Vy(k+1) \\ r(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + Vx(k)\Delta t \\ y(k) + Vy(k)\Delta t \\ \psi(k) + r(k)\Delta t \\ V(k)\cos(\psi(k) + \beta(k)) + a_x\Delta t \\ V(k)\sin(\psi(k) + \beta(k)) \\ \frac{V(k)\delta(k)}{lr+lf} \end{bmatrix} \tag{5}$$

where, Δt is the sampling time step duration, l_f is the length between the front wheel and the center of the vehicle, and l_r is the length between the rear wheel and the vehicle's center.

$$\begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} = \begin{bmatrix} cos(\psi(k)) & -sin(\psi(k)) \\ sin(\psi(k)) & cos(\psi(k)) \end{bmatrix} \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix} \tag{6}$$

IV. STATE ESTIMATION

The state estimation unit is responsible for accurately determining the vehicle's state, considering factors such as the mathematical model, sensor data, previous state, and control input. However, the presence of sensor noise and model imperfections can lead to inaccuracies in estimation, thereby compromising lane change safety. To address this issue, estimators, including Kalman filters KF, EKF, UKF, and AUKF, are designed to optimize the estimation of autonomous vehicle states.

These filters recursively process noisy sensor data and potentially inaccurate models to enhance estimation precision. A comparative analysis of these estimators will be presented in the results and discussion section.

A. STANDARD KALMAN FILTER

The Kalman filter, a recursive mathematical algorithm, is commonly utilized for estimating the state of a linear system based on noisy measurement inputs, the system model, its previous state, and control input. Illustrated in Figure 4, the algorithm comprises two phases: prediction and update. In the prediction step, the filter forecasts the system's state at the next time step using the current state and system model. Subsequently, in the update step, it refines its state estimate based on the current measurement input [19].

The general state space model for the Kalman filter involves two equations: the state estimate equation and the observation equation. The discrete-time state estimate equation in a linear system, the Kalman filter expresses it as follows:

$$x(k+1) = Ax(k) + Bu(k) + W(k) \tag{7}$$

where x is the state estimate vector, u is the control input vector, A is the state matrix, B is the control input matrix and W represents the system or process white Gaussian noises with corresponding system noise covariance matrix Q. Matrices A and B are determined from the vehicle model and their matrices are given by:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \Delta t \\ 0 & 0 \\ \frac{V}{lf+lr} & 0 \end{bmatrix},$$

where the current vehicle state ($x(k)$) and the control input ($u(k)$) matrices at time step k are given by:

$$x(k) = \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \\ Vx(k) \\ Vy(k) \\ r(k) \end{bmatrix} \quad \text{and} \quad u(k) = \begin{bmatrix} \delta(k) \\ a_x(k) \end{bmatrix}$$

The observation equation is also expressed with the following linear equation:

$$y(k) = Cx(k) + V(k) \tag{8}$$

where y is the measurement vector of the current state, V is the measurement of white Gaussian noises with corresponding measurement noise covariance matrix R, and C is the output matrix represented as n by n identity matrix, where n is the number of state parameters.

As shown in Figure 4, at the prediction stage of the Kalman filter, the state ($\hat{x}$) and the covariance error ($\hat{P}$) are estimated using the previous state, previous covariance error, and control input.

During the updated phase, the KF determines the Kalman gain and updates the predicted state $x(k+1)$ and covariance error $P(k+1)$ of the system using the current measurement input $u(k)$. The variable K which is indicated in Figure 4 represents the Kalman gain. It determines how much the measurement is included in the new state estimate. Finally, the updated estimated state and covariance error become the current state and covariance error, which become input for the next state estimation.

$$y = \begin{cases} y_{ei}, & \text{for } x_{ei} \le x < x_{pi} - L_2 - \frac{L_3}{2} \\ y_{ei} + \frac{b}{2\pi}\left(\frac{2\pi}{L_2}\cdot(x-(x_{pi}-L_2-\frac{L_3}{2})) - \sin\left(\frac{2\pi}{L_2}\cdot(x-(x_{pi}-L_2-\frac{L_3}{2}))\right)\right), & \text{for } x_{pi} - L_2 - \frac{L_3}{2} \le x < x_{pi} - \frac{L_3}{2} \\ y_{ei} + b, & \text{for } x_{pi} - \frac{L_3}{2} \le x < x_{pi} + \frac{L_3}{2} \\ y_{ei} + b - \frac{b}{2\pi}\left(\frac{2\pi}{L_4}\cdot(x-(x_{pi}+\frac{L_3}{2})) - \sin\left(\frac{2\pi}{L_4}\cdot(x-(x_{pi}+\frac{L_3}{2}))\right)\right), & \text{for } x_{pi} + \frac{L_3}{2} \le x < x_{pi} + \frac{L_3}{2} + L_4 \\ y_{ei}, & \text{for } x_{pi} + \frac{L_3}{2} + L_4 \le x \le x_{pi} + \frac{L_3}{2} + L_4 + L_5 \end{cases} \tag{1}$$

$$\psi = \begin{cases} 0, & \text{for } x_{ei} \le x < x_{pi} - L_2 - \frac{L_3}{2} \\ \frac{b}{L_2}\left(1 - \cos\left(\frac{2\pi}{L_2}\cdot(x-(x_{pi}-L_2-\frac{L_3}{2}))\right)\right), & \text{for } x_{pi} - L_2 - \frac{L_3}{2} \le x < x_{pi} - \frac{L_3}{2} \\ 0, & \text{for } x_{pi} - \frac{L_3}{2} \le x < x_{pi} + \frac{L_3}{2} \\ -\frac{b}{L_4}\left(1 - \cos\left(\frac{2\pi}{L_4}\cdot(x-(x_{pi}+\frac{L_3}{2}))\right)\right), & \text{for } x_{pi} + \frac{L_3}{2} \le x < x_{pi} + \frac{L_3}{2} + L_4 \\ 0, & \text{for } x_{pi} + \frac{L_3}{2} + L_4 \le x \le x_{pi} + \frac{L_3}{2} + L_4 + L_5 \end{cases} \tag{2}$$

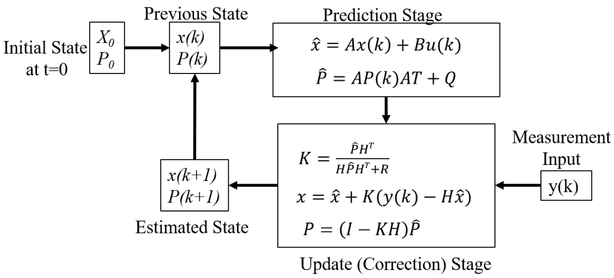

FIGURE 4. Standard Kalman filter algorithm.

B. EXTENDED KALMAN FILTER

The standard Kalman filter operates effectively with linear systems. However, when dealing with autonomous vehicles, which often exhibit nonlinear behaviors, relying solely on the standard Kalman filter becomes impractical for state estimation.

Consider a scenario where a vehicle initially moves straight and then transitions into a turn, perhaps for a lane change. Using a standard Kalman filter to predict the vehicle's next state may lead to position estimations that deviate from the actual path taken. Consequently, due to the inherent nonlinearity of vehicle dynamics, it becomes necessary to employ a nonlinear variant of the Kalman filter, such as the Extended Kalman Filter (EKF), for accurate state estimation.

The EKF addresses this challenge by approximating the nonlinear state equations through linearization via a first-order Taylor expansion. Although this simplification disregards higher-order terms in the state function, it enables the estimation of vehicle states more effectively than a standard Kalman filter in nonlinear scenarios [20], [21], [22], [23].

The state and observation equations of the model are characterized by nonlinear functions denoted as $f(x, u)$ and $h(x)$, respectively.

$$x = f(x, u) + w \tag{9}$$

$$y = h(x) + v \tag{10}$$

where $f(x, u)$ is a nonlinear function representation of the system process, which takes the state vector x and the control input u as the parameter, $h(x)$ is a nonlinear observation function, w is the process noise of the system and v represents the measurement noise.

The nonlinear vehicle model state function for $x = [x, y, \psi, V_x, V_y, r]$ and $u = [\delta, a_x]$ is defined as:

$$f(x, u) = \begin{bmatrix} f1(x,u) \\ f2(x,u) \\ f3(x,u) \\ f4(x,u) \\ f5(x,u) \\ f6(x,u) \end{bmatrix} = \begin{bmatrix} x(k) + Vx(k)\Delta t \\ y(k) + Vy(k)\Delta t \\ \psi(k) + r(k)\Delta t \\ V(k)\cos(\psi(k) + \beta(k)) + a_x\Delta t \\ V(k)\sin(\psi(k) + \beta(k)) \\ \frac{V(k)\delta(k)}{lr+lf} \end{bmatrix} \tag{11}$$

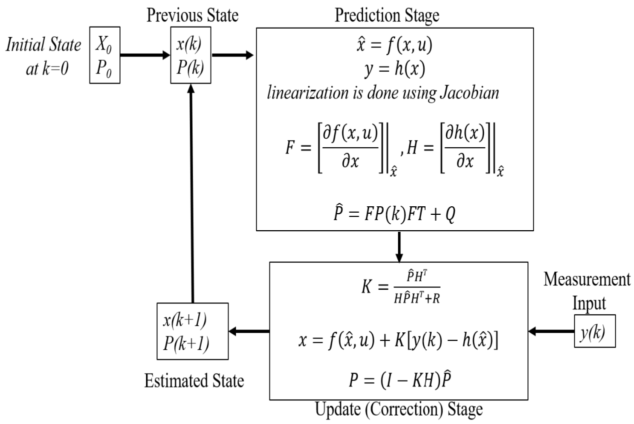
Initial State
at k=0
X_0
P_0
Previous State
x(k)
P(k)
Prediction Stage
$\hat{x} = f(x,u)$
$y = h(x)$
linearization is done using Jacobian
$F = \left[\frac{\partial f(x,u)}{\partial x}\right]\Big|_{\hat{x}}, H = \left[\frac{\partial h(x)}{\partial x}\right]\Big|_{\hat{x}}$
$\hat{P} = FP(k)FT + Q$
$K = \frac{\hat{P}H^T}{H\hat{P}H^T+R}$
$x = f(\hat{x},u) + K[y(k) - h(\hat{x})]$
$P = (I - KH)\hat{P}$
Measurement
Input
y(k)
x(k+1)
P(k+1)
Estimated State
Update (Correction) Stage

FIGURE 5. Extended Kalman filter algorithm.

$$h(x, u) = \begin{bmatrix} h1(x) \\ h2(x) \\ h3(x) \\ h4(x) \\ h5(x) \\ h6(x) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \\ Vx(k) \\ Vy(k) \\ r(k) \end{bmatrix} \tag{12}$$

Figure 5 below depicts the step to estimate the state and the covariance matrix in the Extended Kalman filter.

To linearize the nonlinear process and observation functions, EKF uses Jacobian or first-order terms of the Taylor series expansion of state and observation functions.

$$F = \left[\frac{\partial f(x,u)}{\partial x}\right]\Big|_{\hat{x},u}, \qquad H = \left[\frac{\partial h(x)}{\partial x}\right]\Big|_{\hat{x}} \tag{13}$$

$$F = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x} \\ \frac{\partial f_2(x,u)}{\partial x} \\ \frac{\partial f_3(x,u)}{\partial x} \\ \frac{\partial f_4(x,u)}{\partial x} \\ \frac{\partial f_5(x,u)}{\partial x} \\ \frac{\partial f_6(x,u)}{\partial x} \end{bmatrix} = \frac{\partial}{\partial x} \begin{bmatrix} x(k) + Vx(k)\Delta t \\ y(k) + Vy(k)\Delta t \\ \psi(k) + r(k)\Delta t \\ V(k)\cos(\psi(k) + \beta(k)) + a_x\Delta t \\ V(k)\sin(\psi(k) + \beta(k)) \\ \frac{V(k)\delta(k)}{lr+lf} \end{bmatrix} \tag{14}$$

$$H = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x} \\ \frac{\partial h_2(x)}{\partial x} \\ \frac{\partial h_3(x)}{\partial x} \\ \frac{\partial h_4(x)}{\partial x} \\ \frac{\partial h_5(x)}{\partial x} \\ \frac{\partial h_6(x)}{\partial x} \end{bmatrix} = \frac{\partial}{\partial x} \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \\ Vx(k) \\ Vy(k) \\ r(k) \end{bmatrix} \tag{15}$$

where F is the state matrix and H is the output matrix. After the linearization process, the current covariance matrix is calculated from the state transition matrix F and previously estimated covariance error at the prediction stage as follows.

$$\hat{P} = FP(k)F^T + Q \tag{16}$$

where Q is the system process noise covariance. At the update stage, the state and covariance estimates are corrected using

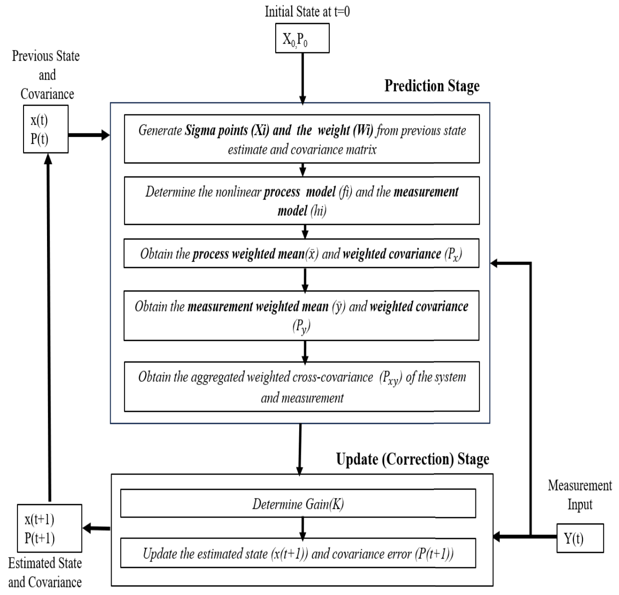
Initial State at t=0
$X_{0,}P_0$
Previous State and Covariance
x(t)
P(t)
Prediction Stage
Generate Sigma points (Xi) and the weight (Wi) from previous state estimate and covariance matrix
Determine the nonlinear process model (fi) and the measurement model (hi)
Obtain the process weighted mean($\bar{x}$) and weighted covariance (P_x)
Obtain the measurement weighted mean ($\bar{y}$) and weighted covariance (P_y)
Obtain the aggregated weighted cross-covariance (P_{xy}) of the system and measurement
Update (Correction) Stage
Determine Gain(K)
Update the estimated state (x(t+1)) and covariance error (P(t+1))
Measurement Input
Y(t)
x(t+1)
P(t+1)
Estimated State and Covariance

FIGURE 6. Overall flowchart of unscented kalman filter algorithm.

the measurement input (y) and Kalman gain (K).

$$K = \frac{\hat{P}H^T}{H\hat{P}H^T + R} \tag{17}$$

$$x(k+1) = f(\hat{x}, u) + K[y(k) - h(\hat{x})] \tag{18}$$

$$P(k+1) = (I - KH)\hat{P} \tag{19}$$

EKF has a limitation when the function is highly non-linear since it uses only first-order derivatives to linearize the system. Thus the EKF is not optimal when the system is highly nonlinear. The other issue, it is difficult to calculate the Jacobians and needs a high computational cost.

C. UNSCENTED KALMAN FILTER

The EKF simplifies the complexity of the nonlinear system by using first-order Taylor expansion, which ignores higher-order terms in the state function [20], [24].

The Unscented Kalman filter(UKF) ignores the Jacobian operations in the EKF by propagating the distribution through nonlinear transformation on a set of so-called sigma points [24], [25], [26]. That is, the probability distribution of sample points and the corresponding weights of sampling points are computed. These sigma points are selected and used as values in the nonlinear system model. It is an unbiased, minimum-mean squared error estimator of a dynamic system with the state vector and covariance matrix. Using an unscented transformation method, the weighted set of sigma points propagates the means and covariance matrix of nonlinear transformations of random variables.

The unscented transform propagates the mean and covariance estimates into the nonlinear transform by assigning the weights to the corresponding sigma points [27].

Similar to other Kalman filter methods, the UKF operates in two phases: the prediction phase and the correction phase, as illustrated in Figure 6.

In the prediction phase, the UKF performs the following steps:

- Generate sigma points X_i and the corresponding scalar weight W_i. The sigma points are calculated from the previous state estimate and the covariance matrix P_x of the random variable as inputs and distributed around the previous estimated state based on the covariance matrix. The number of sigma points and the corresponding weights generated are $2n+1$, where n is the number of parameters that represent the vehicle's state. In this work, the vehicle state is represented by six variables, so $n = 6$.

$$X_i = \begin{cases} X_0 = \hat{x} & i = 0 \\ X_i = \hat{x} + (\sqrt{(n+\lambda)}P_x) & i = 1, .., n \\ X_i = \hat{x} - (\sqrt{(n+\lambda)}P_x) & i = n+1, .., 2n \end{cases}. \tag{20}$$

$$W_i = \begin{cases} W_0 = \dfrac{\lambda}{n+\lambda} & i = 0 \\ W_i = \dfrac{1}{2(n+\lambda)} & i = 1, .., 2n \end{cases}. \tag{21}$$

where λ is a scaling parameter and n is the number of parameters in the given state.

- Determine the nonlinear process model function $f_i = f(X_i, u)$ by passing the generated sigma points, where i represents the iteration from 0 to $2 \times n$.

$$f_i = f(X_i, u) = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \\ f_3(x,u) \\ f_4(x,u) \\ f_5(x,u) \\ f_6(x,u) \end{bmatrix} = \begin{bmatrix} x(k) + Vx(k)\Delta t \\ y(k) + Vy(k)\Delta t \\ \psi(k) + r(k)\Delta t \\ V(k)\cos(\psi(k) + \beta(k)) + a_x\Delta t \\ V(k)\sin(\psi(k) + \beta(k)) \\ \frac{V(k)\delta(k)}{lr+lf} \end{bmatrix} \tag{22}$$

- Determine the measurement model function $h_i = h(f_i)$ by passing predicted measurement input y_i, where i represents the iteration from 0 to $2 \times n$.

$$h_i = h(f_i) = \begin{bmatrix} h_1(f_{1i}) \\ h_2(f_{2i}) \\ h_3(f_{3i}) \\ h_4(f_{4i}) \\ h_5(f_{5i}) \\ h_6(f_{6i}) \end{bmatrix} = \begin{bmatrix} f_{1i} \\ f_{2i} \\ f_{3i} \\ f_{4i} \\ f_{5i} \\ f_{6i} \end{bmatrix} \tag{23}$$

- Compute the weighted process mean and covariance by transforming the sigma points through f_i.

$$\hat{x} = \sum_{i=0}^{2n} W_i \times f_i \tag{24}$$

$$P_x = \sum_{i=0}^{2n} W_i(f_i - \hat{x})(f_i - \hat{x})^T + Q \tag{25}$$

  In this equation, Q represents the state noise covariance matrix, a diagonal matrix that describes the behavior of the noise in the system or process.
- Compute the weighted measurement mean and covariance by transforming the sigma points through h_i.

$$\hat{y} = \sum_{i=0}^{2n} W_i \times h_i \tag{26}$$

$$P_y = \sum_{i=0}^{2n} W_i(h_i - \hat{y})(h_i - \hat{y})^T + R \tag{27}$$

  Here, R denotes the measurement noise covariance matrix which is a diagonal matrix that describes the behavior of the noise that exists in the measurement.
- Obtain the weighted cross-covariance matrix P_{xy} which measures the correlation between the predicted state and the predicted measurement.
  The cross-covariance matrix, P_{xy}, illustrates how the predicted state influences the predicted measurement.

$$P_{xy} = \sum_{i=0}^{2n} W_i(f_i - \hat{x})(h_i - \hat{y})^T \tag{28}$$

- During the update phase, the UKF performs the following steps:
  - Calculate the Kalman gain (K)

$$K = \frac{P_{xy}}{P_y} \tag{29}$$

  - Update the predicted state and covariance matrix.

$$x(t+1) = \hat{x} + K[y - \hat{y}] \tag{30}$$

$$P(t+1) = P_x - KP_yK^T \tag{31}$$

D. ADAPTIVE UNSCENTED KALMAN FILTER

The UKF is extensively employed to tackle the non-linear issues in vehicle state estimation. However, when the system noise covariance is unknown, the standard UKF algorithm is not able to manage the change in system noises and withstand the effect of gross errors [28], [29], [30].

Thus, adaptive UKF (AUKF) based on the Sage-Husa filter will be the best solution to compensate for the system noise. This algorithm directly adopts the Sage-Husa noise estimation technique so that the system noise matrices are kept as positive or semi-positive [29]. It is a maximum likelihood estimation algorithm that adjusts the process and measurement noise covariance matrices based on the observed data [31].

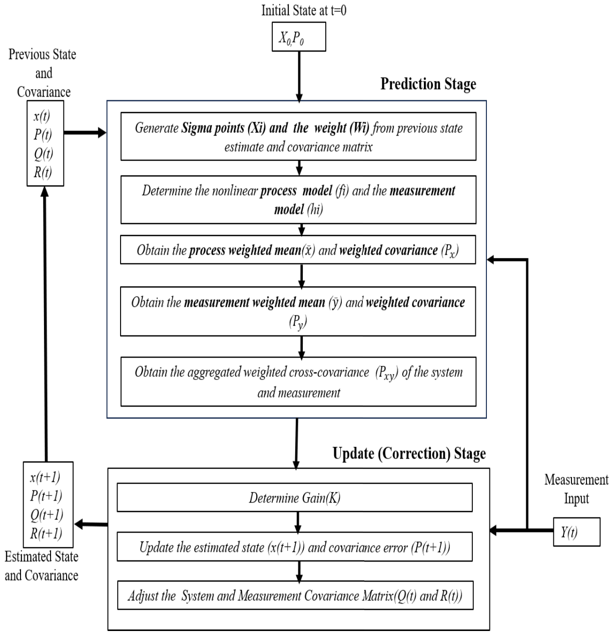
Initial State at t=0
X_0, P_0
Previous State and Covariance
x(t)
P(t)
Q(t)
R(t)
Prediction Stage
Generate Sigma points (Xi) and the weight (Wi) from previous state estimate and covariance matrix
Determine the nonlinear process model (fi) and the measurement model (hi)
Obtain the process weighted mean($\bar{x}$) and weighted covariance (P_x)
Obtain the measurement weighted mean ($\bar{y}$) and weighted covariance (P_y)
Obtain the aggregated weighted cross-covariance (P_{xy}) of the system and measurement
Update (Correction) Stage
Determine Gain(K)
Update the estimated state (x(t+1)) and covariance error (P(t+1))
Adjust the System and Measurement Covariance Matrix(Q(t) and R(t))
x(t+1)
P(t+1)
Q(t+1)
R(t+1)
Estimated State and Covariance
Measurement Input
Y(t)

FIGURE 7. Adaptive unscented kalman filter algorithm.

The estimation process for the AUKF closely mirrors that of the UKF. However, in the case of AUKF, the adaptive process extends beyond updating the estimated state and error. It iteratively refines the covariance of system and measurement noise (Q and R) to better adapt to varying noise conditions.

The entire process of the AUKF is detailed in Figure 7, illustrating its iterative nature and its capability to adjust system noise for robust state estimation in vehicle systems.

The adaptive process to determine Q and R is shown below in the equations:

$$d_k = \frac{1-\alpha}{1-\alpha^n} \tag{32}$$

$$q_k = (1-d_k)q_{k-1} + d_k(x - \hat{x}) \tag{33}$$

$$r_k = (1-d_k)r_{k-1} + d_k(y - \hat{y}) \tag{34}$$

$$v_k = y - \hat{y} - r_k \tag{35}$$

$$Q_k = (1-d_k)Q_{k-1} + d_k(K(v_k v_k^T)K^T + P - P_x) \tag{36}$$

$$R_k = (1-d_k)R_{k-1} + d_k(v_k v_k^T - P_y) \tag{37}$$

q_k denotes the mean of the system noise vector, while Q_k represents the system noise variance matrix; whereas R_k represents the measurement noise variance matrix; r_k is the mean of the measurement noise vector; v_k is a residual factor representing the discrepancy between the actual and expected observations; n is the tuning parameter and α is a correction factor, with the range of values is $0.95 < \alpha < 0.99$ [28], [29].

To ensure the system noise covariance matrix Q_k and the measurement noise covariance matrix R_k stay positive or semi-positive definite values. If Q_k and R_k are a negative definite value, the Q_k and R_k are calculated by the following equations:

$$Q_k = (1 - d_k)Q_{k-1} + d_k(K(v_k v_k^T)K^T + P) \tag{38}$$

$$R_k = (1 - d_k)R_{k-1} + d_k(v_k v_k^T) \tag{39}$$

V. MODEL PREDICTIVE CONTROL DESIGN

To ensure optimal control inputs for a smooth and safe lane change, it's essential to manage both the lateral and longitudinal motions of the vehicle effectively.

This study focuses on the precise control of these motions: the steering angle actuator handles lateral movement, while the throttle/brake actuator manages longitudinal motion. Adjusting the throttle regulates fuel consumption, affecting acceleration and deceleration, while the steering angle dictates the vehicle's direction by turning the front wheels. Since lateral and longitudinal maneuvers are interconnected, simultaneous control of both inputs is crucial for executing lane changes that are both safe and comfortable [32].

Designing multiple input and output systems using traditional controllers like PID, Stanly, and Pure Pursuit can pose significant challenges [17], [33], [34], [35], [36]. This difficulty arises from the interdependence between longitudinal and lateral motion. In contrast, the MPC controller excels in managing both lateral and longitudinal control inputs simultaneously, while considering their interactions [32]. MPC functions as a feedback controller, leveraging the system's model to predict its future behavior. It then addresses an online optimization problem to determine the optimal controller action, driving the predicted output towards the reference. Furthermore, MPC accommodates constraints on control inputs, ensuring that the constraints are not violated to avoid undesirable outcomes.

The general MPC formulation for autonomous vehicle lateral and longitudinal control can be expressed as the following discrete-time optimal control problem:

$$min_{\{uk\}_{i=1}^{N}} J(x_{k,i}, u_{k,i}) \quad \forall i = 1, 2, \ldots, N \tag{40a}$$

Subject to:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k && (40b) \\ u_{min} &< u_{k,i} < u_{max} && (40c) \\ \Delta u_{min} &< \Delta u_{k,i} < \Delta u_{max} \\ &\forall i = 1, 2, \ldots, N && (40d) \end{aligned}$$

where x_k represents the state vector at time step k, which includes the vehicle's lateral position, longitudinal position, yaw angle, lateral velocity, longitudinal velocity, and yaw rate, $x_k = [x, y, \psi, V_x, V_y, r]$ and u_k denotes the control input vector at time step k, which is composed of the steering angle (δ) and longitudinal acceleration (a_x) of the vehicle, $u_k = [\delta_k, a_{xk}]$.

Equation (40b) represents the vehicle model used to predict the vehicle's future state with respect to the predicted control inputs of the vehicle (u_k).

This work focuses solely on controlling input constraints, aiming to ensure a safe and comfortable lane change. The steering angle and longitudinal acceleration are restricted within specific intervals: $a_x \in [-1\,\text{m/s}^2, 1\,\text{m/s}^2]$ for longitudinal acceleration and $\delta \in [-28.6°, 28.6°]$ for steering angle, allowing for smooth lateral movement while maintaining constant longitudinal velocity. To enhance tracking of the generated lane change trajectory and stabilize longitudinal motion changes, the cost function incorporates errors in lateral distance, heading angle, and longitudinal velocity between reference and predicted target states.

In addition, the rate of change of control inputs (Δu) is included in the cost function to mitigate discomfort caused by abrupt control input changes. This involves considering the difference between consecutive control inputs ($\Delta u = u_k - u_{k-1}$) for both steering angle and longitudinal acceleration. Thus, the following quadratic cost function in equation (40a) is implemented for accurate lane change maneuvers.

$$\begin{aligned} J(x_k, u_k) = \sum_{i=1}^{N} & W_{de}||d_{e(k+i)}||^2 + W_{\psi}||\psi_{e(k+i)}||^2 \\ & + W_v||v_{e(k+i)}||^2 \\ & + \sum_{i=2}^{N} W_{\delta}||\delta_{(k+i)} - \delta_{(k+i-1)}||^2 \\ & + W_{a_x}||a_{x(k+i)} - a_{x(k+i-1)}||^2 \\ & \forall i = 1, 2, \ldots, N \end{aligned} \tag{41}$$

where d_e is the distance error or cross-tracking error between the predictive position $(x_k|_{pred}, y_k|_{pred})$ and the reference position $(x_k|_{ref}, y_k|_{ref})$, which is calculated as follows:

$$de_{k+i} = \sqrt{(x_{(k+i)}|_{ref} - x_{(k+i)}|_{pred})^2 + (y_{(k+i)}|_{ref} - y_{(k+i)}|_{pred})^2} \tag{42}$$

ψ_e represents the heading error of the vehicle, which is the difference between the predicted and reference yaw angles.

$$\psi_{e(k+i)} = \psi_{(k+i)}|_{ref} - \psi_{(k+i)}|_{pred} \tag{43}$$

In which v_e denotes the velocity error, which minimizes the velocity error and helps to find the corresponding optimal controller inputs. W_{de}, W_{ψ}, W_v, W_{δ}, and W_{ax} are weighting coefficients that are tuned according to the control objectives to find optimal steering and longitudinal acceleration control inputs.

VI. RESULT AND DISCUSSION

To ensure the accuracy of the proposed system in lane change maneuvers, rigorous tests were conducted to meet ISO 3888-1 and ISO 3888-2 standards. These standards provide autonomous vehicle researchers with a comprehensive evaluation of the system's capabilities and ensure compliance with safety criteria.

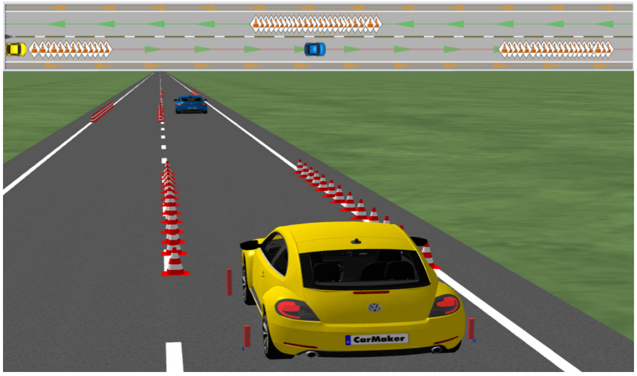

FIGURE 8. Setting a restricted area on a simple two-lane straight road with pylons on IPG Carmaker Road Scenario.

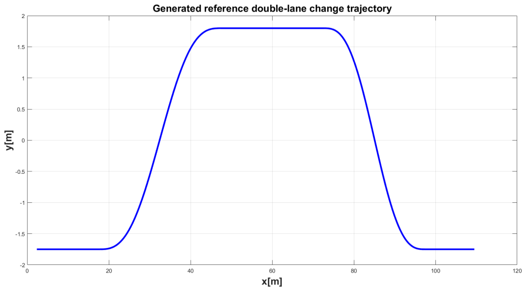

FIGURE 9. Generated reference double-lane change trajectory according to the ISO 3888-1 standards.

All system components, including the trajectory generator, model predictive control, and state estimators, were carefully implemented using MATLAB.

Furthermore, to accurately recreate real-world traffic situations and road conditions, IPG Carmaker was used to create a highly realistic simulation environment. This precise simulation allowed for a thorough evaluation of the vehicle's behavior during lane change maneuvers.

Special attention was given to the lane change process, which was subjected to testing according to ISO 3888 standards, and the outcomes were analyzed. The test is run for 6.2 *seconds* over a total distance of 130 m with a 72 km/h forward speed of the ego vehicle. The distance between the ego and preceding vehicles is around 58 m. The vehicle selected on IPG Carmaker has a 1.5 m width and a 1.67 m long wheelbase.

Additionally, the accuracy and computational complexity of the KF, EKF, UKF, and AUKF state estimator algorithms are compared for the six vehicle states in both tests. The state estimators were combined with a double-lane change trajectory generator and MPC to complete the lane change trajectory process.

The white noise is added to the inertial measurement sensor input of the state estimator to investigate which state estimator filter is best for reducing measurement and process noise.

Following the configuration of the IPG Carmaker, two simulations were performed. One involved executing an ISO 3888-1 double-lane change to overtake a slow-moving vehicle, while the other involved utilizing ISO 3888-2 protocols to navigate around obstacles. The subsequent section delves into a discussion of the obtained results.

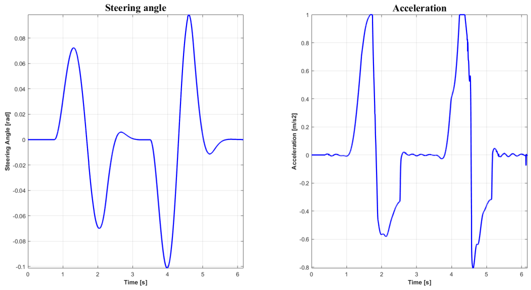

FIGURE 10. Generated steering angle and longitudinal acceleration control inputs for double lane change according to ISO 3888-1 standard.

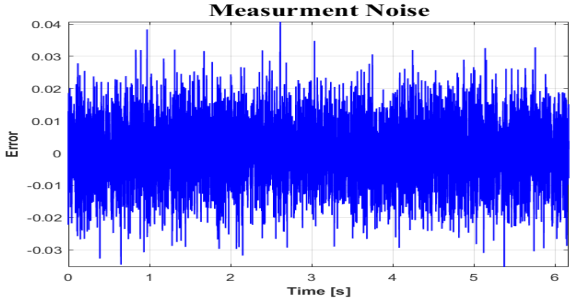

FIGURE 11. White noise to be added to the measurement inertial sensor input.

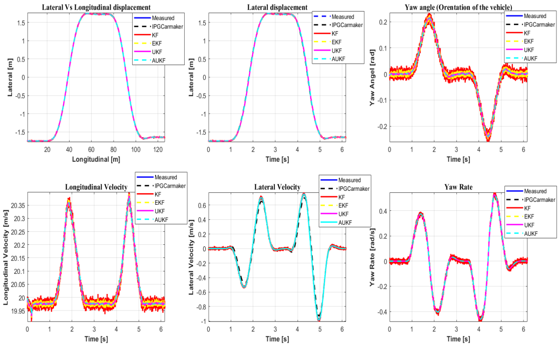

FIGURE 12. Estimated states (the longitudinal displacement, lateral displacement, yaw angle, longitudinal velocity, lateral velocity, and yaw rate).

A. DOUBLE LANE CHANGE TEST WITH ISO 3888-1

In this section, our focus shifts to assessing the autonomous vehicle's proficiency in executing a double-lane change seamlessly while ensuring control, stability, and obstacle avoidance. For this evaluation, the vehicle undergoes a test where its lateral movement is constrained by a sequence of pylons arranged in an alley, adhering to the specifications outlined in the ISO 3888-1 standard.

Utilizing IPG Carmaker, we create road scenarios incorporating different traffic signs. The pylons are strategically positioned according to the standard's guidelines, specifically in the first, third, and fifth sections, as illustrated in Figure 8.

A double-lane change trajectory, which is used to take over the preceding vehicle, is generated according to the standard

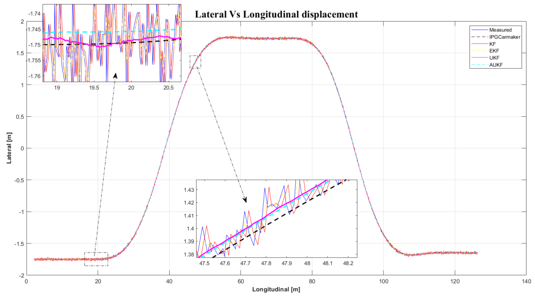

FIGURE 13. Estimated longitudinal versus lateral displacement.

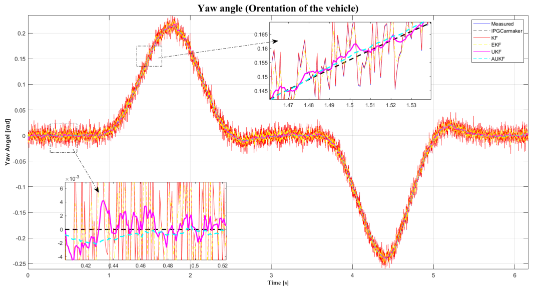

FIGURE 14. Estimated yaw angle.

ISO 3888-1 using equation (1). The generated reference double-lane change trajectory ($y[m]$ with respect to $x[m]$) is shown in Figure 9.

From the generated lane change trajectory, the steering angle (δ) and longitudinal acceleration control inputs are derived, which become the inputs to the state estimator after the MPC adjusts them to evaluate the next state of the vehicle, $[x, y, \psi, V_x, V_y, r]$. The steering angle control input (δ) is determined from the orientation (ψ) of the vehicle, which is a slope of the trajectory calculated by derivation of the generated reference lane change trajectory, which is shown in Figure 9. Then, the steering angle is calculated from the change in the orientation angle.

Figure 10 below shows the generated steering angle and the corresponding longitudinal acceleration for perfect double-lane change according to the standard ISO 3888-1 after being controlled by the MPC.

To assess the robustness of the proposed filters, the white Gaussian noise shown in Figure 11 is generated and added to the measurements of the inertial sensors. This process helps to determine which state estimator can maintain performance despite this measurement noise.

Figure 12 illustrates the estimated state variables of the vehicle (including longitudinal displacement, lateral displacement, yaw angle, longitudinal velocity, lateral velocity, and yaw rate) for a vehicle moving at a speed of 72 km/h for 6.2 seconds. It compares the outcomes derived from the four-state estimator, the measured inputs, and the output from IPG Carmaker, presenting them together for comparison.

The results of each estimation method were compared with IPG Carmaker estimation results and noisy measured data to assess accuracy. By looking closely at specific

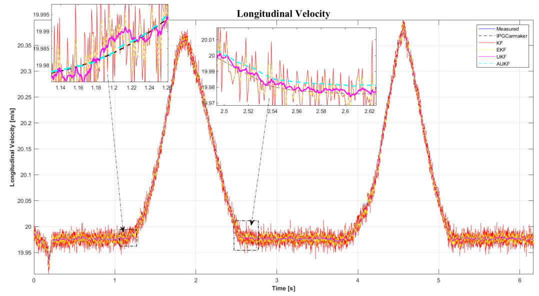

FIGURE 15. Estimated longitudinal velocity.

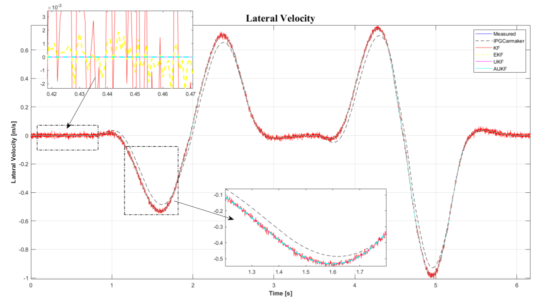

FIGURE 16. Estimated lateral velocity.

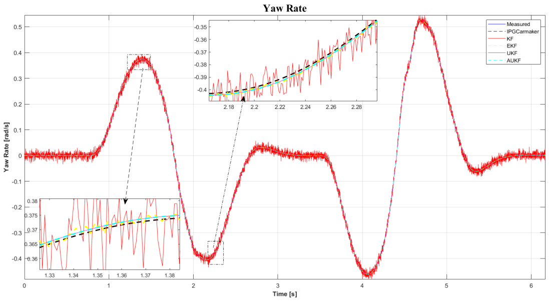

FIGURE 17. Estimated yaw rate.

sections of the estimated graph, we identified the state estimation algorithm that produces the most accurate results. The estimated lateral and longitudinal displacements are shown in Figure 13. When zoomed in, we can observe that UKF and AUKF accurately estimate both longitudinal and lateral displacements. However, AUKF demonstrated slightly superior accuracy and convergence compared to UKF, unlike classical KF and EKF, which yielded the least accurate estimates.

Figure 14 depicts the estimated yaw angle. Upon closer examination in the zoomed section, it becomes evident that UKF and AUKF provide satisfactory estimations of the yaw angle. In contrast, the yaw angle estimation results obtained with standard KF and EKF appear noisy, and the graphs are not smooth, which indicates their estimation is not accurate.

Figure 15 shows the estimated longitudinal velocity. Upon closer examination in the zoomed section, it becomes apparent that AUKF provides more accurate estimates of the longitudinal velocity compared to the other filters. Additionally, the results from AUKF closely align with the

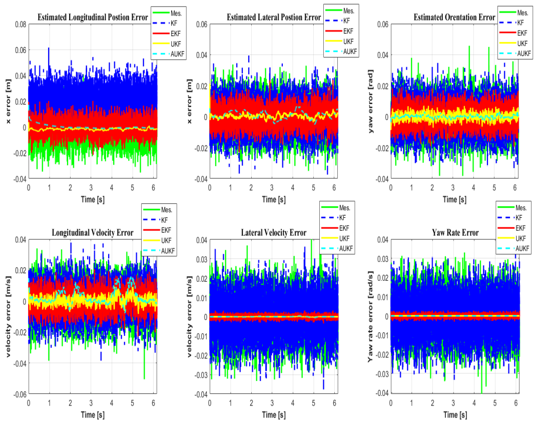

FIGURE 18. Error on the estimated states.

TABLE 1. RMS state estimation and measurement errors.

RMS Error	Measured	KF	EKF	UKF	AUKF
Longitudinal Displacement	10×10^{-3}	10×10^{-3}	5.75×10^{-3}	1.86×10^{-3}	1.70×10^{-3}
Lateral Displacement	10×10^{-3}	9.996×10^{-3}	5.75×10^{-3}	1.43×10^{-3}	2.69×10^{-4}
Yaw Angle	9.97×10^{-3}	9.993×10^{-3}	5.79×10^{-3}	2.17×10^{-4}	1.00×10^{-4}
Longitudinal Velocity	9.998×10^{-3}	9.996×10^{-3}	5.78×10^{-3}	2.53×10^{-3}	5.08×10^{-4}
Lateral Velocity	10×10^{-3}	9.99×10^{-3}	9.18×10^{-4}	6.37×10^{-5}	1.75×10^{-5}
Yaw Rate	10×10^{-3}	9.99×10^{-3}	9.08×10^{-4}	4.37×10^{-17}	4.37×10^{-17}

desired longitudinal velocity, whereas the estimations from the other filters, including UKF, diverge from the desired longitudinal velocity.

Figure 16 illustrates the estimated lateral velocity. Upon closer inspection in the zoomed section, it is apparent that both UKF and AUKF provide satisfactory estimations of the lateral velocity.

Figure 17 illustrates the estimated yaw rate. A closer look at the zoomed section reveals that AUKF accurately estimates the yaw rate and maintains a more stable trajectory. Additionally, AUKF demonstrates faster convergence to the desired yaw rate compared to the classic KF and EKF estimations, which exhibit instability and fail to converge within the specified time frame.

Comparing estimation errors helps to determine which estimators are performing well. Figure 18 illustrates the estimation error for each filter. The AUKF stands out with low estimation errors for all six state variables, represented in cyan. The UKF performs well in estimating the yaw angle and yaw rate. However, the classic KF and EKF filters show high estimation errors and struggle to effectively reduce measurement noise.

The aforementioned Figure 18 compares the estimators based on the estimated error value in the time domain. However, the most effective method for comparing the accuracy of state estimation filters is through root mean square error (RMS error), which provides a quantitative measurement of error. The estimated state RMS errors are calculated using equation (44), and the RMS errors for the six state variables of each filter are presented in Table 1.

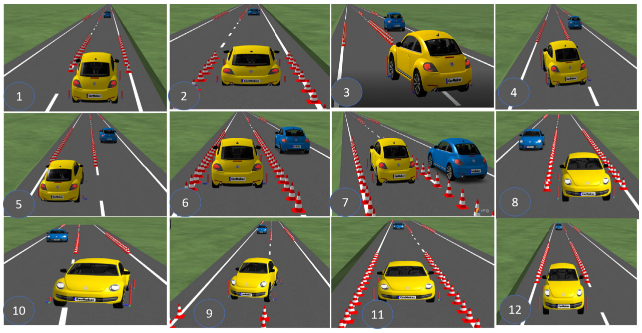

FIGURE 19. Some snapshot images taken from IPGMovie of IPG CarMaker for ISO 3888-1 double lane change visualization.

$$RMSerror = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{44}$$

where x_i is the actual state value, $\hat{x}_i$ is the estimated state value, and n is the number of estimated state values throughout the process.

Regarding the computational complexity of the state estimators, KF, EKF, UKF and AUKF algorithms are compared for double lane change with MATLAB simulation. The simulation is carried out in MATLAB R2024a running on a laptop computer with 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz and 16 GB RAM. With this performance of a laptop, the computational complexity of the state estimator indicates that the AUKF is the slowest estimator and the classic KF is the fastest estimator followed by UKF. The EKF and AUKF have the highest computational complexity. The reason for the difference in computational complexity is the way that each filter updates its state estimate.

The steering angle generated was also fed in real-time to the IPG CarMake vehicle. The double-lane change maneuver was visualized using IPGMovie in IPG CarMaker's simulation environment. The simulation results from IPG CarMaker demonstrate that the lane change algorithm can execute lane changes successfully without colliding with the preceding vehicle or other road users, adhering to the restricted area defined by pylons.

Throughout the lane change maneuver, the trajectory of the ego vehicle remains smooth and stable. Snapshot images of the vehicle maneuver during the lane change, presented in Figure 19, indicate that the vehicle completed the double lane change without deviating from the lateral constraints set by the pylons, as per ISO 3888-1 standard. This confirms the effectiveness of the proposed system in performing safe double-lane change maneuvers.

B. OBSTACLE AVOIDANCE TEST WITH ISO 3888-2

The ISO 3888-2 standard test is very crucial for evaluating the ability of the autonomous vehicle to avoid obstacles safely while maintaining stability and control of the vehicle.

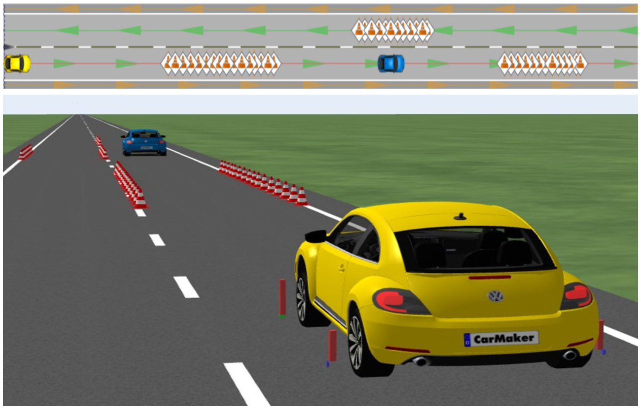

FIGURE 20. Setting simple two-lane straight road with pylons on IPG Carmaker Road Scenario.

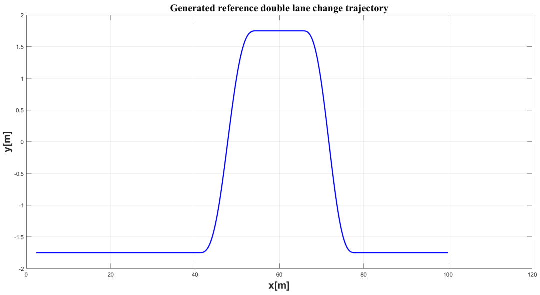

FIGURE 21. Generated double-lane change reference trajectory according to ISO 3888-2 to avoid obstacle.

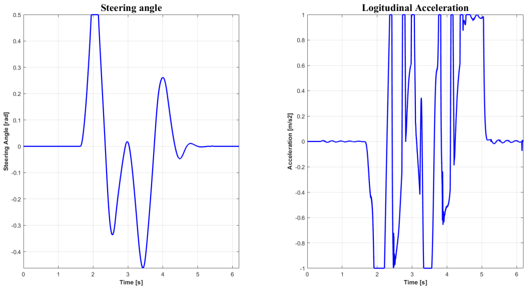

FIGURE 22. Generated steering angle control input for obstacle avoidance according to ISO 3888-2.

It can assess the autonomous vehicle's ability to react quickly and effectively to unexpected obstacles or hazards. As done above for the ISO 3888-1 test, the vehicle's lateral position is restricted by a pylon alley.

Figure 20 shows the lateral movement restriction made with pylons, which are placed according to ISO 3888-2 standards in the first, third, and fifth sections.

A double-lane change trajectory, which is used to avoid obstacles, is generated according to the standard ISO 3888-2 using equation (1). The generated reference double-lane change trajectory is shown in Figure 21 below. Relative to ISO 3888-1, the ISO 3888-2 lane change trajectory has a very steep slope, which needs a higher steering rate to make the lane change.

The steering angle has been generated from the change in the orientation angle of the vehicle in real time. Figure 22 below shows the generated steering angle for perfect obstacle avoidance according to the standard ISO 3888-2 after the MPC controls it.

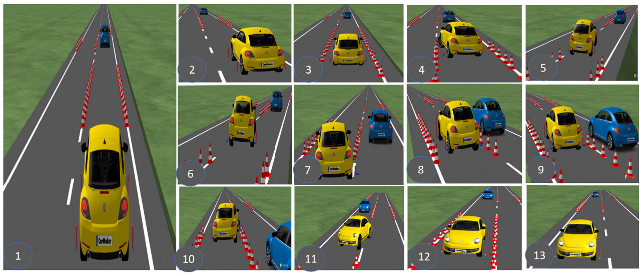

FIGURE 23. Some snapshot images taken from the IPGMovie of IPG CarMaker for ISO 3888-2 double lane change visualization. The images are numbered according to the sequence of navigation.

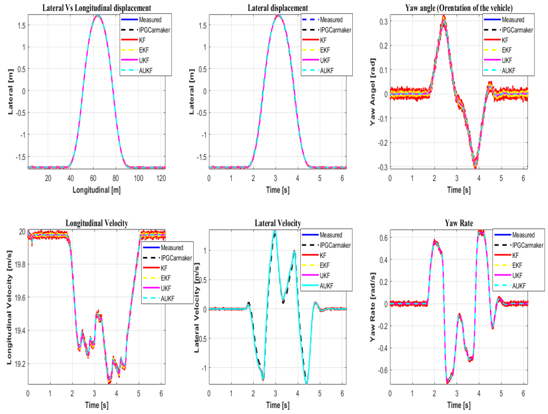

FIGURE 24. Estimated states (the longitudinal displacement, lateral displacement, yaw angle, longitudinal velocity, lateral velocity, and yaw rate).

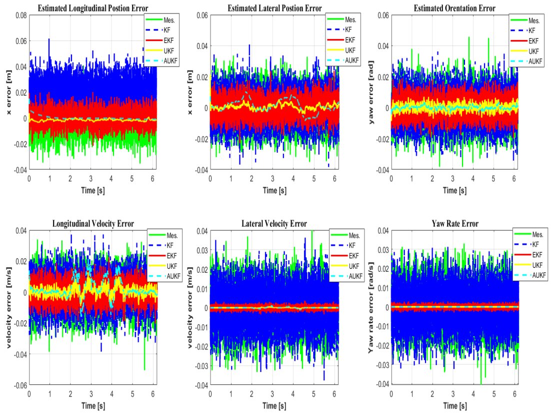

FIGURE 25. Error on the estimated states.

The double-lane change to avoid obstacles has been visualized on IPG CarMaker using IPGMovie. IPG CarMaker's simulation result shows that the lane change algorithm can successfully move through the restricted area by pylons to avoid obstacles on the road. The ego vehicle's trajectory is smooth and stable throughout the lane change maneuver. Some snapshot images of the vehicle maneuver during the lane change in Figure 23 below show that the vehicle has avoided the obstacle by completing the double lane change without crossing the restricted lateral constraint set with pylons according to standard ISO 3888-2.

The estimated states and errors of the vehicle states for avoiding obstacles according to ISO 3888-2 standards are shown in the figures below. The results show the AUKF with the trajectory generator and MPC created a smooth lane change trajectory with the smallest estimation error relative to the other estimator.

VII. CONCLUSION

This paper introduces a novel methodology aimed at achieving precise double-lane changes for autonomous vehicles, adhering closely to the stringent requirements set forth by ISO 3888-1 and ISO 3888-2 standards. The proposed approach integrates a suite of components, including a lane change trajectory generator, Kalman filters, and a model predictive controller.

The trajectory generator functions to develop safe and efficient lane change paths, while the state estimator precisely assesses the vehicle's position, velocity, and orientation. Subsequently, control actions are carried out by the MPC controller to ensure the vehicle accurately tracks the designated trajectory. Employing both linear and sinusoidal function-based trajectories, managed by the MPC, yields stable lane change paths that meet ISO standards. The MPC offers enhanced performance and robustness by simultaneously controlling longitudinal and lateral motion.

Integrating real-time interaction with surrounding vehicles enhances safety and reliability. Among state estimation methods, the AUKF emerges as the most accurate, particularly in handling nonlinearities. Comparatively, the Classic Kalman Filter (KF) performs poorly due to its linear nature, while the EKF struggles with higher-order nonlinear systems. Despite its higher computational complexity, the AUKF is considered the optimal choice for state estimation in lane changes.

Experimental validation, conducted using IPG Carmaker and MATLAB/Simulink, confirms the effectiveness of integrating AUKF with the model predictive controller. This integration ensures stable and smooth lane-change maneuvers while meeting the safety standards delineated in ISO 3888-1 and ISO 3888-2.

REFERENCES

[1] M. Nikowitz, *Fully Autonomous Vehicles: Visions of the Future or Still Reality*. Berlin, Germany: Epubli, 2015.

[2] J. Becker, M.-B. A. Colas, S. Nordbruch, and M. Fausten, "Bosch's vision and roadmap toward fully autonomous driving," *Road vehicle Autom.*, vol. 1, pp. 49–59, Aug. 2014.

[3] M. A. Khan, H. E. Sayed, S. Malik, T. Zia, J. Khan, N. Alkaabi, and H. Ignatious, "Level-5 autonomous driving—Are we there yet? A review of research literature," *ACM Comput. Surveys*, vol. 55, no. 2, pp. 1–38, Feb. 2023.

[4] S. Abdallaoui, E.-H. Aglzim, A. Chaibet, and A. Kribèche, "Thorough review analysis of safe control of autonomous vehicles: Path planning and navigation techniques," *Energies*, vol. 15, no. 4, p. 1358, Feb. 2022.

[5] Y. Liu, B. Zhou, X. Wang, L. Li, S. Cheng, Z. Chen, G. Li, and L. Zhang, "Dynamic lane-changing trajectory planning for autonomous vehicles based on discrete global trajectory," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8513–8527, Jul. 2022.

[6] X. Jin, G. Yin, and N. Chen, "Advanced estimation techniques for vehicle system dynamic state: A survey," *Sensors*, vol. 19, no. 19, p. 4289, Oct. 2019.

[7] H. Sazgar and A. K. Khalaji, "Nonlinear integrated control with friction estimation for automatic lane change on the highways," *Proc. Inst. Mech. Engineers, Part K, J. Multi-Body Dyn.*, vol. 236, no. 3, pp. 453–469, Sep. 2022.

[8] A. Ghaffari, B. Gharehpapagh, A. Khodayari, and S. Salehinia, "Longitudinal and lateral movement control of car following maneuver using fuzzy sliding mode control," in *Proc. IEEE 23rd Int. Symp. Ind. Electron.*, Jun. 2014, pp. 150–155.

[9] S. Zhang, X. Liu, G. Deng, J. Ou, E. Yang, S. Yang, and T. Li, "Longitudinal and lateral control strategies for automatic lane change to avoid collision in vehicle high-speed driving," *Sensors*, vol. 23, no. 11, p. 5301, Jun. 2023.

[10] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 1–14.

[11] V. Fors, B. Olofsson, and E. Frisk, "Resilient branching MPC for multi-vehicle traffic scenarios using adversarial disturbance sequences," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 4, pp. 838–848, Dec. 2022.

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

[13] D. Iberraken, "Safe trajectories and sequential Bayesian decision-making architecture for reliable autonomous vehicle navigation," M.S. thesis, Autom., Université Clermont Auvergne, Oct. 2020. [Online]. Available: https://theses.hal.science/tel-03158717 and https://theses.hal.science/tel-03158717/file/2020CLFAC043_IBERRAKEN.pdf

[14] S. Angelis, M. Tidlund, A. Leledakis, M. Lidberg, M. Nybacka, and D. Katzourakis, "Optimal steering for double-lane change entry speed maximization," in *Proc. ACEV'14 Int. Symp. Adv. Vehicle Control*, 2014, pp. 1–9.

[15] H. Jamaluddin, E. P. Ping, and K. Hudha, "Hardware-in-the-loop simulation of automatic steering control for double lane change and sine steer manoeuvres," *Int. J. Vehicle Auto. Syst.*, vol. 10, no. 1/2, p. 67, 2012.

[16] R. Zhang, Z. Zhang, Z. Guan, Y. Li, and Z. Li, "Autonomous lane changing control for intelligent vehicles," *Cluster Comput.*, vol. 22, no. S4, pp. 8657–8667, Jul. 2019.

[17] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Review and performance evaluation of path tracking controllers of autonomous vehicles," *IET Intell. Transp. Syst.*, vol. 15, no. 5, pp. 646–670, May 2021.

[18] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–20.

[19] N. H. Ali and G. M. Hassan, "Kalman filter tracking," *Int. J. Comput. Appl.*, vol. 89, no. 9, pp. 1–23, 2014.

[20] D. Qi, J. Feng, X. Ni, and L. Wang, "Maximum correntropy extended Kalman filter for vehicle state observation," *Int. J. Automot. Technol.*, vol. 24, no. 2, pp. 377–388, Apr. 2023.

[21] H. Ahmadi Jeyed and A. Ghaffari, "Nonlinear estimator design based on extended Kalman filter approach for state estimation of articulated heavy vehicle," *Proc. Inst. Mech. Engineers, Part K, J. Multi-Body Dyn.*, vol. 233, no. 2, pp. 254–265, Jun. 2019.

[22] D. Qi, J. Feng, W. Wan, and B. Song, "A novel maximum correntropy adaptive extended Kalman filter for vehicle state estimation under non-Gaussian noise," *Meas. Sci. Technol.*, vol. 34, no. 2, Feb. 2023, Art. no. 025114.

[23] T. Kim and T.-H. Park, "Extended Kalman filter (EKF) design for vehicle position tracking using reliability function of radar and LiDAR," *Sensors*, vol. 20, no. 15, p. 4126, Jul. 2020.

[24] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Jun. 2000, pp. 1–24.

[25] B. Zheng, P. Fu, B. Li, and X. Yuan, "A robust adaptive unscented Kalman filter for nonlinear estimation with uncertain noise covariance," *Sensors*, vol. 18, no. 3, p. 808, Mar. 2018.

[26] M. Impraimakis and A. W. Smyth, "An unscented Kalman filter method for real time input-parameter-state estimation," *Mech. Syst. Signal Process.*, vol. 162, Jan. 2022, Art. no. 108026.

[27] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[28] B. Ge, H. Zhang, L. Jiang, Z. Li, and M. M. Butt, "Adaptive unscented Kalman filter for target tracking with unknown time-varying noise covariance," *Sensors*, vol. 19, pp. 1–19, Oct. 2019.

[29] J. Wang, T. Xu, and Z. Wang, "Adaptive robust unscented Kalman filter for AUV acoustic navigation," *Sensors*, vol. 20, no. 1, p. 60, Dec. 2019.

[30] Y. Zhang, M. Li, Y. Zhang, Z. Hu, Q. Sun, and B. Lu, "An enhanced adaptive unscented Kalman filter for vehicle state estimation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[31] X. Wang, A. Wang, D. Wang, Y. Xiong, B. Liang, and Y. Qi, "A modified sage-husa adaptive Kalman filter for state estimation of electric vehicle servo control system," *Energy Rep.*, vol. 8, pp. 20–27, Aug. 2022.

[32] D. Stenger, R. Ritschel, F. Krabbes, R. Voßwinkel, and H. Richter, "What is the best way to optimally parameterize the MPC cost function for vehicle guidance?" *Mathematics*, vol. 11, no. 2, p. 465, Jan. 2023.

[33] S. Singh, "Longitudinal velocity control of autonomous ground vehicle using PID and PI controller," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 1, pp. 504–510, Jan. 2021.

[34] T. Hossain, H. Habibullah, and R. Islam, "Steering and speed control system design for autonomous vehicles by developing an optimal hybrid controller to track reference trajectory," *Machines*, vol. 10, no. 6, p. 420, May 2022.

[35] G. H. S. Elias, A. Al-Moadhen, and H. Kamil, "Optimizing the pid controller to control the longitudinal motion of autonomous vehicles," in *AIP Conf. Proc.*, vol. 2591, 2023, pp. 1–29.

[36] L. Song, J. Li, Z. Wei, K. Yang, E. Hashemi, and H. Wang, "Longitudinal and lateral control methods from single vehicle to autonomous platoon," *Green Energy Intell. Transp.*, vol. 2, no. 2, Apr. 2023, Art. no. 100066.

HEYI MULUNEH HAILU received the B.Sc. degree in electrical engineering from Hawassa University, Hawassa, Ethiopia, and the M.Sc. degree in electronics engineering from the Politecnico di Torino, Torino, Italy. He is currently pursuing the Ph.D. degree in automatic control and robotics with the University of Burgundy, Nevers, France.

His research interests include autonomous vehicles, self-driving cars, automatic control systems, robotics, nonlinear systems, estimation theory, and mathematical modeling.

SALIM HIMA received the B.S. and M.Sc. degrees in control systems from the University of Ferhat-Abbas, Sétif, Algeria, in 1997 and 1998, respectively, the M.Sc. degree in robotics from the University Pierre et Marie Curie, Paris 6, in 2000, and the Ph.D. degree in robotics from the University of Evry Val d'Essonne, in 2005.

From 2006 to 2008, he was a Postdoctoral Fellow with the IBISC Laboratory, University of Evry Val d'Essonne, where he worked on two projects, SIMACOM and VIGISIM, funded by the French national research agency and related to ground transportation security. From 2009 to 2011, he was a Postdoctoral Fellow with the LIVIC-IFSTTAR Laboratory involved in the HaveIt European project for the development of highly automated vehicles for intelligent transport. He has been an Associate Professor with the Energy and Systems Department, ESME Engineering School, Paris, since 2012, and has been an Associate Member of the IBISC Laboratory of Evry, Université-Paris Saclay, since 2019. His research interests include dynamical system analysis and control, robust control, optimal control, observers, motion planning, autonomous ground vehicles, and autonomous aerial vehicles.

AHMED CHAIBET received the Engineering degree from the National Institute of Hydrocarbons and Chemistry, Boumerdès, Algeria, in 1999, the master's degree in virtual reality and control of complex systems from Versailles Saint-Quentin-en-Yvelines University, Versailles, France, in 2002, the Ph.D. degree in automatic control from the University of Evry Val d'Essonne, Evry, France, in 2006, and the Habilitation à Diriger des Recherches (Accreditation to Supervise Research) degree from the University of Paris XI, Orsay, France, in July 2019.

He currently holds the position of an Associate Professor in automatic control with the University of Bourgogne, France, where he is also affiliated with the DRIVE Department. Prior to joining the University of Bourgogne, he was an Associate Professor with the ESTACA College, specializing in the education of automotive, aeronautical, space, railway, and naval engineers with the Paris Saclay Campus. His ongoing research at the DRIVE Department is centered on developing robust linear and nonlinear control laws for the automated guidance of on-ground vehicles. Additionally, he focuses on energy management systems and powertrains of electric vehicles, with a specific emphasis on the automotive context. He is also keenly interested in control law reconfiguration and fault-tolerant control (FTC).