

UC Berkeley

UC Berkeley Previously Published Works

Title

Autonomous Ground Vehicle Lane-Keeping LPV Model-Based Control: Dual-Rate State Estimation and Comparison of Different Real-Time Control Strategies

Permalink

<https://escholarship.org/uc/item/52r1r3sr>

Journal

Sensors, 21(4)

ISSN

1424-8220

Authors

Ducajú, Julián M Salt
Llobregat, Julián J Salt
Cuenca, Ángel
[et al.](#)

Publication Date

2021

DOI

10.3390/s21041531



Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Article

Autonomous Ground Vehicle Lane-Keeping LPV Model-Based Control: Dual-Rate State Estimation and Comparison of Different Real-Time Control Strategies

Julián M. Salt Ducajú ¹, Julián J. Salt Llobregat ^{2,*} , Ángel Cuenca ²  and Masayoshi Tomizuka ³

¹ Department of Automatic Control, LTH, Lund University, 221 00 Lund, Sweden; julian.salt_ducaju@control.lth.se

² Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, 46022 València, Spain; acuenca@isa.upv.es

³ Mechanical Engineering Department, University of California, Berkeley, CA 94720, USA; tomizuka@berkeley.edu

* Correspondence: julian@isa.upv.es

Abstract: In this contribution, we suggest two proposals to achieve fast, real-time lane-keeping control for Autonomous Ground Vehicles (AGVs). The goal of lane-keeping is to orient and keep the vehicle within a given reference path using the front wheel steering angle as the control action for a specific longitudinal velocity. While nonlinear models can describe the lateral dynamics of the vehicle in an accurate manner, they might lead to difficulties when computing some control laws such as Model Predictive Control (MPC) in real time. Therefore, our first proposal is to use a Linear Parameter Varying (LPV) model to describe the AGV's lateral dynamics, as a trade-off between computational complexity and model accuracy. Additionally, AGV sensors typically work at different measurement acquisition frequencies so that Kalman Filters (KFs) are usually needed for sensor fusion. Our second proposal is to use a Dual-Rate Extended Kalman Filter (DREFKF) to alleviate the cost of updating the internal state of the filter. To check the validity of our proposals, an LPV model-based control strategy is compared in simulations over a circuit path to another reduced computational complexity control strategy, the Inverse Kinematic Bicycle model (IKIBI), in the presence of process and measurement Gaussian noise. The LPV-MPC controller is shown to provide a more accurate lane-keeping behavior than an IKIBI control strategy. Finally, it is seen that Dual-Rate Extended Kalman Filters (DREKFs) constitute an interesting tool for providing fast vehicle state estimation in an AGV lane-keeping application.

Keywords: autonomous vehicle; dual-rate control; dual-rate EKF; MPC; LPV model



Citation: Salt Ducajú, J.M.; Salt Llobregat, J.J.; Cuenca, Á.; Tomizuka, M. Autonomous Ground Vehicle Lane-Keeping LPV Model-Based Control: Dual-Rate State Estimation and Comparison of Different Real-Time Control Strategies. *Sensors* **2021**, *21*, 1531. <https://doi.org/10.3390/s21041531>

Academic Editor: Felipe Jiménez

Received: 2 January 2021

Accepted: 15 February 2021

Published: 23 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Self-driving cars have been increasing in popularity year after year. They are the type of Autonomous Ground Vehicle (AGV) that has received the greatest share of attention, both in academia and in industry, because of the possibility that they can shift the paradigm of transportation systems. An essential concern in the development of these automated driving systems is the ability to obtain a controller that is able to make the vehicle follow a pre-established path. This problem is often considered, in a hierarchical manner, as the low level control of the AGV in opposition to a high level control, which is focused on path or trajectory generation based on the awareness of the environment that surrounds the vehicle.

The lateral vehicle control takes care of the path-tracking problem. The path is composed of a sequence of positions and orientations in the plane, and the controller has to make sure that the vehicle follows them. In order to control the vehicle, two input variables are often considered: the steering angle (which is modified by acting on the

steering wheel) and the longitudinal acceleration (modified via throttle). If the vehicle must follow a feasible collision-free pre-computed path with no time constraints, the problem is known as lane-keeping. On the contrary, if each pair of the position and orientation of the pre-defined path has a time stamp associated with them, the problem is considered as a trajectory-tracking problem. Since time constraints are not considered in this application, the steering angle can be chosen as the only control variable, disregarding the longitudinal acceleration.

To achieve successful lateral vehicle control, it is necessary that a set of sensors (GPS, IMU, and others) is embedded in the vehicle. These sensors measure some variables such as position, velocity, acceleration, orientation, and change of orientation, at different rates. The use of the celebrated Kalman filter [1–3] enables fusing all of them with the aim of being conveniently utilized by the control stage. Additionally, several authors have proposed different models to describe the lateral dynamics of the vehicle. The use of a kinematic bicycle model has been widely extended, where each axle is considered as a single wheel [4]. The model was later expanded through a dynamic expression that links, among other variables, the inertial heading time evolution with the steering wheel angle. The lateral dynamics of wheeled ground vehicles is determined by the highly nonlinear forces occurring in their tires. For this reason, most of the models that have been suggested [4–10] are nonlinear models.

Therefore, in order to use the Kalman filter in a proper way, it needs to be formulated via its extended or unscented versions (see, e.g., [11,12]). In the present approach, the Extended Kalman Filter (EKF) was chosen, not only for fusing all the data provided by the different sensing devices, but also for estimating the nonlinear behavior of the vehicle's dynamics, providing unavailable (not measurable) variables if needed and reducing the possible process and measurement noise effect. Since every sensor may work at a different rate (the GPS and velocimeter usually work at slower rates, but the IMU works at a faster rate), which may be slower than the actuation (control) rate, a multi-rate EKF may be needed. In our proposal, the different output variables are assumed to be sensed at different rates, but the internal state of the filter is only updated at a rate M -times slower than the actuation one. This leads to a Dual-Rate EKF (DREKF).

The literature on the DREKF is scarce and scattered. Some works appear in biomedicine, concretely in the field of electrocardiogram signal denoising, where the DREKF has been used in order to better estimate system states that are not updated in all time instances and avoid unwanted errors in the estimation procedure [13,14]. Unmanned Aerial Vehicles (UAVs) are another field where the DREKF has been employed with the aim of estimating state variables from few measurements, which come from a low cost, low rate GPS [15]. In robotics, although factor-graph-based methods (GT-SAM, OkVIS, Cartographer, etc.) are prevalent, the DREKF can also be utilized for ego-motion estimation so as to fuse low-rate vision and fast-rate inertial measurements in the context of the simultaneous localization and mapping problem [16,17]. To the best of the authors' knowledge, the DREKF has never been explicitly formulated in the AGV's framework.

Several authors have explored the topic of motion planning and control for AGVs (see, e.g., [18]) using different control approaches such as Linear Quadratic Regulator (LQR) control, inverse kinematics controller, Model Predictive Control (MPC), and some attempts with classical control (PID, lead-lag) [4,19]. In particular, MPC has been widely used in trajectory reference tracking for self-driving cars [8,20–25], since it enables calculating and optimizing the sequence of future control inputs by using an explicit model [26].

Depending on the control scheme selected, choosing a nonlinear model can cause a relevant increase in the calculation time, which may endanger the feasible real-time solving of the controller. On the contrary, a Linear Time-Invariant (LTI) model might be insufficient to describe the vehicle's dynamics, especially if high lateral tire forces are involved [6,9,10]. Linear Parameter Varying (LPV) models have been regarded as a trade-off between model accuracy and computational complexity [24,25,27–30].

Previous contributions on LPV-MPC for reference tracking in ground vehicles have shown promising results [24,25]. Here, the nonlinearities of the vehicle's dynamics are embedded into the model's varying parameters, which may cause prediction errors for long time horizons if the variation from the operating point is meaningful throughout this time interval. In a recent contribution [31], a learning algorithm for vehicular dynamics [32] was applied to an LPV car dynamics' model [27], to optimize the prediction results over a long prediction horizon. It would be interesting to analyze its behavior in a realistic scenario when used in model-based control.

Moreover, as previously mentioned, in this control problem, some sensors work at a slower rate. In order to reach a good control performance, this rate may not be appropriate to update the controller output. Then, instead of using a DREKF to provide a single, fast-rate controller with faster estimates, a dual-rate controller may be considered to generate faster control actions from slower measurements.

The main contributions of this article are three. First, we present an LPV-MPC design that considers a model identified specifically for longer time scale predictions such as the ones handled by MPC. Second, we introduce a Dual-Rate EKF (DREKF) that allows a fast state update using, but not limited to, slow and noisy measurements in an autonomous vehicle control context. Third, we compare and analyze two different low computational complexity, dual-rate approaches for lateral vehicle controlling in the presence of process and measurement Gaussian noise. The first of the two approaches considered in the third contribution uses a single, fast-rate feedforward controller, which is designed from an Inverse Kinematics Bicycle (IKIBI) model. The second considers an MPC controller, which can be designed from a new LPV optimized model and with a prediction horizon that allows generating a fast-rate control signal from the slow-rate measurements.

The paper is organized as follows. Section 2 details the design aspects for each control approach (IKIBI and MPC). Then, the DREKF is introduced in Section 3. Simulated experiments are introduced and justified in Section 4, and their results are presented and discussed in Section 5. Finally, some conclusions summarize the present work in Section 6.

2. Control Strategies

There are diverse control laws devoted to vehicle lane-keeping, commonly called steering controllers. In this section, two widely used methods with some variations are considered: the Inverse Kinematic Bicycle model (IKIBI) and the Linear Parameter Varying-Model Predictive Control (LPV-MPC).

In both cases, the purpose is to use the steering front wheels' angle δ as the control action in order to follow the desired path. The complete path, $[X \ Y \ \Psi]_{traj}$, is planned offline, and depending on the controller election, the next yaw rate, r_{ref} , or yaw position goal, Ψ_{ref} , is delivered by a pure pursuit procedure with a coherent look-ahead distance L [7,19,33,34]. Figure 1 shows a schematic view of this process. The Dual-Rate Extended Kalman Filter (DREKF) proposed for state estimation can also be seen in Figure 1 and is further explained in Section 3.

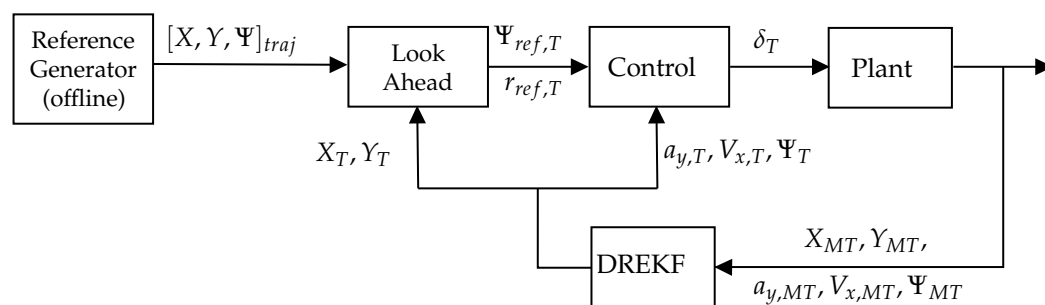


Figure 1. Proposed closed-loop control. DREKF, Dual-Rate Extended Kalman Filter.

2.1. Inverse Kinematic Bicycle Model-Based Controller

In this work, the IKIBI is used by adding a proportional feedforward controller in order to consider the yaw rate measurement r . The control law yields:

$$\delta(k+1) = \left[\text{atan}_2 \left(\frac{r_{ref} L_v}{v_x(k)} + K_p (r_{ref}(k+1) - r(k)) \right) \right] \gamma \quad (1)$$

where K_p is the feedforward controller's proportional gain, r_{ref} is the yaw rate goal established by the pure pursuit, L_v is the vehicle's longitudinal length, and γ is a vehicle coefficient that translates the tire angle into the steering angle. Since the input signal is considered directly as the tire angle, $\gamma = 1$. Furthermore, the function atan_2 represents the fourth-quadrant inverse tangent.

2.2. Linear Parameter Varying-Model Predictive Control

Model predictive control can be used for lateral vehicle control [8,20,21]. A linear model of the system should be considered to implement an MPC controller in real time in order to avoid computational delays [21,35,36]. The lateral dynamics' model that is used for this controller was presented in [27]:

$$\dot{\psi}(k) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \delta(k) \quad (2)$$

where $\dot{\psi}$ is the yaw rate in the body frame coordinates of the vehicle and δ the front steering angle. The presented model is Linear Parameter Varying (LPV), and its coefficients (b_{0-2} and $a_{1,2}$) depend on the lateral acceleration and longitudinal velocity in the local body frame of the vehicle. Previous research has suggested an optimized method for identifying the system's parameters [31] by minimizing the model prediction errors over a long time horizon.

Since the goal of this controller is to follow a trajectory reference described in terms of position (X and Y) and orientation (Ψ) in absolute coordinates, it is interesting to set the orientation of the vehicle (ψ) as the output of the system rather than its rate of change ($\dot{\psi}$). A forward Euler method is used where:

$$\psi(k) = \frac{T}{z-1} \dot{\psi}(k) = \frac{Tz^{-1}}{1-z^{-1}} \dot{\psi}(k) \quad (3)$$

where T is the sampling period of the system. Then:

$$\psi(k) = \frac{Tb_0 z^{-1} + Tb_1 z^{-2} + Tb_2 z^{-3}}{1 + (a_1 - 1)z^{-1} + (a_2 - a_1)z^{-2} - a_2 z^{-3}} \delta(k) \quad (4)$$

Therefore, the model used in MPC is:

$$x(k+1) = Ax(k) + Bu(k) \quad (5)$$

$$y(k) = Cx(k) + Du(k) \quad (6)$$

where u and y are the discrete-time input ($\delta(k)$) and output ($\psi(k)$) variables, respectively, and:

$$A = \begin{bmatrix} 1 - a_1 & a_1 - a_2 & a_2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (7)$$

$$B = [1 \ 0 \ 0]^T \quad (8)$$

$$C = [Tb_0 \ Tb_1 \ Tb_2] \quad (9)$$

$$D = 0 \quad (10)$$

The quadratic cost function chosen for solving this problem at each time step k is:

$$J_U(k) = \sum_{i=k+1}^{k+M} (y(k) - y_{ref}(k))^T Q (y(k) - y_{ref}(k)) + \sum_{i=k}^{k+M-1} u(k)^T R u(k) \quad (11)$$

where U is the input signal sequence of the control horizon that minimizes the cost function over the MPC prediction horizon for every metaperiod and Q and R are positive semi-definite weight matrices that penalize the controlled variables and inputs, respectively. Furthermore, $y_{ref}(k)$ is determined by the look-ahead algorithm.

This optimization problem is subject to the discrete-time model of the system (5), (6) and a set of linear constraints on the control and output that preserve the physical feasibility of the solution:

$$FU(k) \leq f \quad (12)$$

$$GY(k) \leq g \quad (13)$$

where Y is the sequence of discrete-time output variables in the prediction horizon of the MPC problem. Furthermore, the rate of change of the output variable is limited by the slew rate, S , to avoid abrupt vehicle turns that would have a detrimental effect on the passengers' comfort:

$$l_\infty(y(k+1) - y(k)) \leq S \quad (14)$$

The choice of the cost function as convex, as well as a linear model, and convex constraint sets makes the whole problem convex, which is beneficial for the computation of the problem since if a solution exists, it is the globally optimal one [37].

Finally, one of the aspects of MPC is the generation of the sequence of the M future discrete-time control actions to achieve the goal reference. However, it is typical that only the first control action is injected. Therefore, MPC is a natural dual-rate control in the sense that it calculates M future control actions with each of the measurement data.

3. Dual-Rate Extended Kalman Filter

In previous work [31], the set of hardware available (an inertial measurement unit, a differential GPS, and a computer) for data acquisition in the car was able to measure X , Y , Ψ , and V_x , but V_y and r were difficult to access.

Moreover, the measurements are available at different frequencies, the GPS's at a slow frequency (about 10 Hz) and the same for velocity acquisition. The orientation ψ is acquired by the IMU with a frequency of 100 Hz. For this reason, dual-rate control is a natural proposal to deal with this problem, assuming slow-rate measurements, but a fast (M -times faster) steering control action. The acceleration could be varied at a slow frequency. In the case of the dual-rate EKF, it may be needed when some of the measurements are not available due to its slow-rate acquisition (for instance, X and Y from the GPS). Then, state estimation is carried out at a faster rate from (V_x, ψ) .

Analogously, the DREKF includes a linearization procedure, which is based on the use of the Jacobian matrix (a matrix of partial derivatives). At each time step, this matrix is evaluated with the current predicted states. The DREKF, different from a standard EKF, carries out some slow-rate computations (such as the correction stage) only when output variables are available, that is when they are sensed. Otherwise, predictions are shifted to the next iteration.

The DREKF presented in this section takes a nonlinear model based on Newton's second law that uses the bicycle model and assumes a constant tire load [5,24,29] for state estimation. Expressing the model in discrete time for period T yields:

$$F_{yf}(k) = -C_{af} \arctan\left(\frac{V_y(k-1) + r(k-1)a}{\max(V_x(k-1), V_{\min})} - \delta(k-1)\right) \quad (15)$$

$$F_{yr}(k) = -C_{ar} \arctan\left(\frac{V_y(k-1) - r(k-1)b}{\max(V_x(k-1), V_{\min})}\right) \quad (16)$$

$$a_x(k) = a_x(k-1) \quad (17)$$

$$a_y(k) = -V_x(k-1)r(k-1) + \frac{F_{yf}(k-1) + F_{yr}(k-1)}{m} \quad (18)$$

$$\dot{r}(k) = \frac{aF_{yf}(k-1) \cos(\delta(k-1)) - bF_{yr}}{I_{zz}} \quad (19)$$

$$V_x(k) = V_x(k-1) + T \cdot a_x(k-1) \quad (20)$$

$$V_y(k) = V_y(k-1) + T \cdot a_y(k-1) \quad (21)$$

$$r(k) = r(k-1) + T \cdot \dot{r}(k-1) \quad (22)$$

$$X(k) = X(k-1) + T[V_x(k-1) \cos(\psi(k-1)) - V_y(k-1) \sin(\psi(k-1))] \quad (23)$$

$$Y(k) = Y(k-1) + T[V_x(k-1) \sin(\psi(k-1)) + V_y(k-1) \cos(\psi(k-1))] \quad (24)$$

$$\psi(k) = \psi(k-1) + T \cdot r(k-1) \quad (25)$$

where, as mentioned earlier, X and Y are the position coordinates in the absolute inertial frame, Ψ is the orientation coordinate also in absolute coordinates, ψ is the orientation coordinate in the body-frame coordinates, r its rate of change, and V_x , V_y , a_x , and a_y are the velocities and accelerations, respectively, in the body-frame coordinates. Moreover, the different constants that appear in these equations are defined in Appendix A, where their numerical values are also given to allow the reproducibility of the results. Additionally, let us denote this global nonlinear dynamic model as the next state-space representation

$$\begin{cases} \xi(k) = f(\xi(k-1), n_1(k-1), u(k-1)) \\ z(k) = h(\xi(k), n_2(k)) \end{cases} \quad (26)$$

where the AGV state $\xi(k)$ is composed of $(V_x(k), V_y(k), X(k), Y(k), \psi(k), r(k))^T$, the control signal is $u(k-1) = (a_x(k-1), \delta(k-1))^T$, the output consists of $z(k) = (V_x(k), X(k), Y(k), \psi(k))^T$, and $n_1(k-1)$ and $n_2(k)$ are process and measurement noises, respectively, which are both assumed to be zero mean multivariate Gaussian noises with variance $\bar{Q}(k) = 0.01$ and $\bar{R}(k) = 0.01$, respectively.

Assuming that the notation $\hat{\xi}(j|i)$ means the state estimated for the instant jT at the instant iT , the prediction and correction steps of the DREKF are defined as follows:

- Fast-rate calculations:
 - Prediction of the next state $\hat{\xi}(k|k-1)$ and propagation of the covariance $P(k|k-1)$. These computations are calculated $\forall k$:

$$\begin{aligned} \hat{\xi}(k|k-1) &= f(\hat{\xi}(k-1|k-1), n_1(k-1), u(k-1)) \\ P(k|k-1) &= A(k)P(k-1|k-1)A(k)^T + L(k)\bar{Q}(k-1)L(k)^T \end{aligned} \quad (27)$$

where $\hat{\xi}(0) = E[\xi(0)]$ $P(0) = E[(\xi(0) - E[\xi(0)])(\xi(0) - E[\xi(0)])^T]$, $E[\cdot]$ being the expectation, and where $A(k)$ and $L(k)$ are Jacobian matrices computed in

order to respectively linearize the process model about the current state and about the process noise:

$$\begin{aligned}\bar{A}(k) &= \left. \frac{\partial f}{\partial \xi} \right|_{\hat{\xi}(k-1|k-1), n_1(k-1), u(k-1)} \\ L(k) &= \left. \frac{\partial f}{\partial n_1} \right|_{\hat{\xi}(k-1|k-1), n_1(k-1), u(k-1)}\end{aligned}\quad (28)$$

- State and covariance shifts, for $\hat{\xi}(k|k)$ and $P(k|k)$, respectively. These computations are calculated when measurements are not provided, that is for $k \neq MT$:

$$\begin{aligned}\hat{\xi}(k|k) &= \hat{\xi}(k|k-1) \\ P(k|k) &= P(k|k-1)\end{aligned}\quad (29)$$

- Slow-rate calculations, which are computed when measurements are provided, that is for $k = MT$:

- Prediction of the future output $\hat{z}(k)$:

$$\hat{z}(k) = h(\hat{\xi}(k|k-1), n_2(k))\quad (30)$$

- Computation of the Kalman filter gain $K(k)$:

$$K(k) = P(k|k-1)H(k)^T \left(H(k)P(k|k-1)H(k)^T + M(k)\bar{R}(k)M(k)^T \right)^{-1}\quad (31)$$

where $H(k)$ and $M(k)$ are Jacobian matrices calculated in order to respectively linearize the output model about the predicted next state and about the measurement noise:

$$\begin{aligned}H(k) &= \left. \frac{\partial h}{\partial \xi} \right|_{\hat{\xi}(k|k-1), n_2(k)} \\ M(k) &= \left. \frac{\partial h}{\partial n_2} \right|_{\hat{\xi}(k|k-1), n_2(k)}\end{aligned}\quad (32)$$

- Correction of the state $\hat{\xi}(k|k)$ and correction of the covariance $P(k|k)$:

$$\begin{aligned}\hat{\xi}(k|k) &= \hat{\xi}(k|k-1) + K(k)(z(k) - \hat{z}(k)) \\ P(k|k) &= K(k)\bar{R}(k)K(k)^T + (I - K(k)H(k))P(k|k-1)(I - K(k)H(k))^T\end{aligned}\quad (33)$$

Finally, it should be mentioned that if $M = 1$ is assumed, the state and covariance shifts in (29) are replaced by the corrections in (33), resulting in a (single-rate) EKF.

4. Implementation

In this section, we present the experiments that were performed in order to compare the two proposed controllers for lane-keeping and justify the appeal of using a DREKF in this application.

The design choices for the controllers and some simulation details are presented first, followed by a discussion of the tests' selection. Afterwards, we introduce the cost indexes that quantify each controllers' performance, and we present the results obtained.

4.1. Simulation Details and Design Choices for the Controllers

Simulations were carried out using the vehicle parameters of a 2017 Lincoln MKZ on a circuit path. The sampling period of the simulated discrete-time plant was assumed to be $T = 0.01$ s, which is the same as the fastest acquisition frequency of sensors installed in the test-bed vehicle.

The IKIBI-based controller design resulted in $K_p = 0.55$, and as mentioned earlier, $\gamma = 1$. On the other hand, the LPV model parameters used for the MPC strategy were obtained from previous research results [31]. Moreover, the convex optimization problem of the MPC was solved using CVXGEN [38].

Moreover, the prediction and the control horizons in the MPC problem were chosen to be equal to 10 steps ($N = 10$), to ensure a small computation time, and the weighting matrices that penalize the output deviation from its reference and the input were, $Q = 1$ and $R = 0.001$, respectively. Since the main goal of this implementation was to obtain an accurate control, the vehicle's orientation error was penalized much more heavily compared to the input signal.

Furthermore, to verify that the low computational complexity of the two controllers allows a real-time implementation, the calculation time for each controller was computed throughout the entire trajectory. The results are shown in Figure 2. It can be seen that all calculations were below the vehicle's sampling period, which was equal to 10 milliseconds. The average and standard deviation for the IKIBI and the MPC controllers were equal to 0.003 ± 0.001 milliseconds and 0.637 ± 0.276 milliseconds, respectively. Furthermore, the slowest execution time recorded was 0.0040 milliseconds for the IKIBI controller and 2.086 milliseconds for the MPC controller.

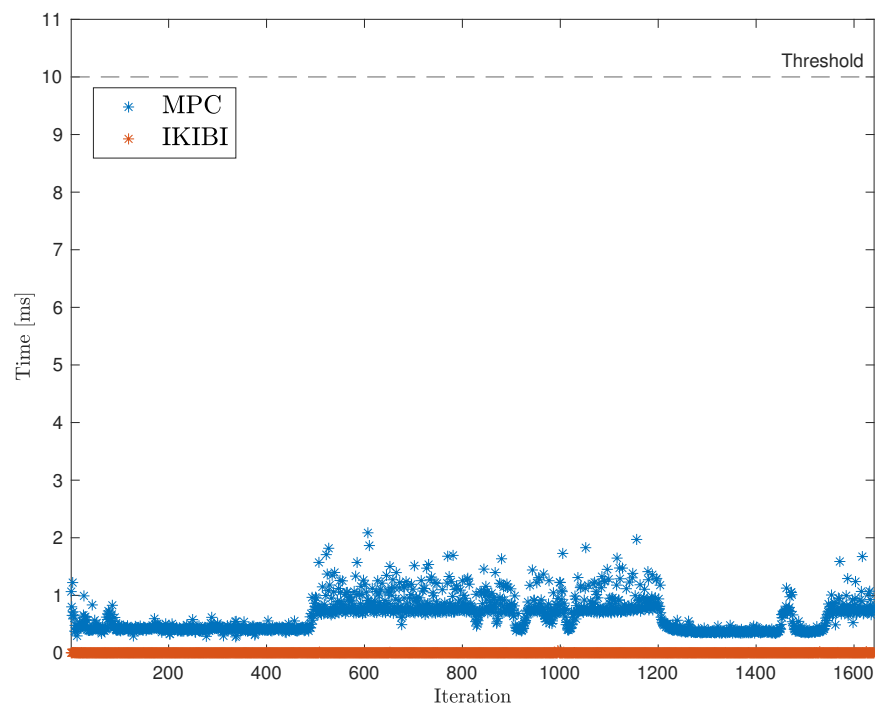


Figure 2. Calculation duration throughout the trajectory.

4.2. Performed Tests' Selection

The performed tests compared the behavior of the two proposed controllers, the IKIBI-based controller and MPC-based controller. The tests were performed focusing on the lateral dynamics of the vehicle, and therefore, it was assumed that a longitudinal controller was able to maintain a constant longitudinal speed throughout the entire trajectory ($a_x = 0$). The tests were performed using two different longitudinal velocities: 8 and 12 m/s.

The circuit that was used to generate the path references includes abrupt lateral movements such as a fast double-lane turn and a 180 degree turn. These maneuvers are so aggressive that when driving the real car through this path, the longitudinal velocity was as low as 2.5 m/s in the most critical segments. Therefore, using constant longitudinal velocities of 8 and 12 m/s would allow us to drive close to the vehicle's dynamic limits.

Moreover, the most realistic tests would assume that new sensor data would be obtained every 0.1 seconds, so $M = 10$. Thus, only in the presence of the DREKF, both controllers would be able to receive new data every $T = 0.01$ s. However, if a single rate EKF (SREKF) were implemented instead of the DREKF, the controllers would have to be calculated every $MT = 0.1$ s.

In this last situation, the MPC-based controller was still able to provide a different control signal every T since $M \leq N$, which means that the first M discrete-time control signals ($M = 10$) of the control horizon ($N = 10$) would be used at every controller calculation. On the contrary, the IKIBI-based controller had to calculate one control signal every MT in the absence of the DREKF. Finally, both controllers were tested considering process and measurement noises and also without these noises.

4.3. Cost Indexes Used to Measure Performance

Two different cost indexes were used in order to better quantify and compare each control solution in each of the tests:

- J_1 , which is based on the ℓ_2 -norm, and its goal is to provide a measure of how accurately the path is followed:

$$J_1 = \sum_{k=1}^l \min_{1 \leq k' \leq l} \sqrt{(X_k - X_{ref,k'})^2 + (Y_k - Y_{ref,k'})^2} \quad (34)$$

where l is the number of iterations required by the AGV to reach the final point of the path, $(X, Y)_k$ is the current AGV position, and $(X_{ref}, Y_{ref})_{k'}$ is the nearest kinematic position reference to the current AGV position.

- J_2 , which is based on the ℓ_∞ -norm and is defined to obtain the maximum difference between the desired path and the current AGV position:

$$J_2 = \max_{1 \leq k \leq l} \left\{ \min_{1 \leq k' \leq l} \sqrt{(X_k - X_{ref,k'})^2 + (Y_k - Y_{ref,k'})^2} \right\} \quad (35)$$

5. Results and Discussion

This section shows and discusses the results that were obtained from the different tests.

5.1. Noiseless, Fast Sensor Feedback Test

The first experiment considered the situation where sensor feedback is received every T (fast sampling rate). Therefore, the controllers can also directly calculate the input signal (steering angle) every T . Moreover, since it was assumed in this test that there is no measurement or process noises, there was no need for a filter.

This test was used to compare each of the two controllers that we proposed for this application. Figures 3–5 show the results. Figure 3 plots the X and Y coordinates of each simulation, and Figures 4 and 5 show the temporal evolution of the steering angle and the yaw rate, respectively.

Because of the abruptness of the maneuvers, a degradation of the behavior when the longitudinal velocity of the vehicle is higher is clearly observed in Figure 3. Such an aggressive maneuver is handled by each of the controllers in two different ways.

On the one hand, the IKIBI controller simply increases the steering angle in order to achieve a higher yaw rate. While this may suffice for more moderate maneuvers, in a real scenario, the front wheel angular position is physically bounded, and therefore, the control signal calculated with this controller would not be feasible.

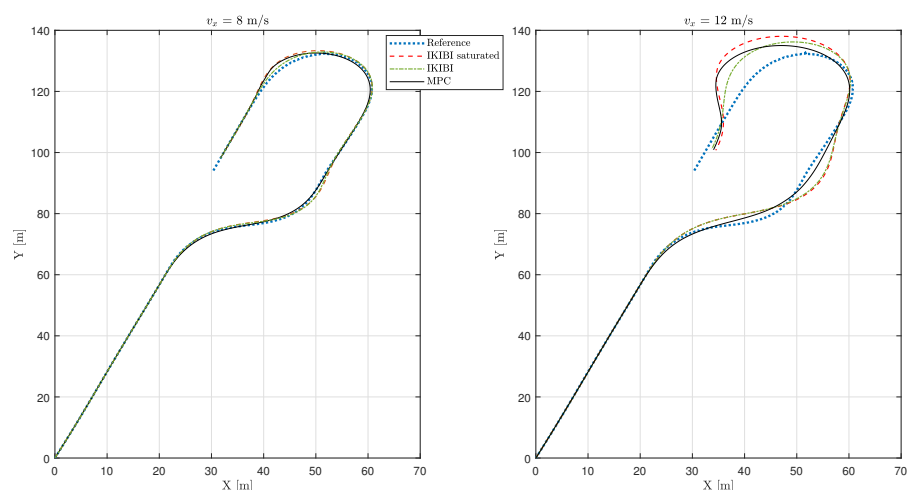


Figure 3. Vehicle path: noiseless, fast sensor feedback test. (left) $v_x = 8$ m/s and (right) $v_x = 12$ m/s.

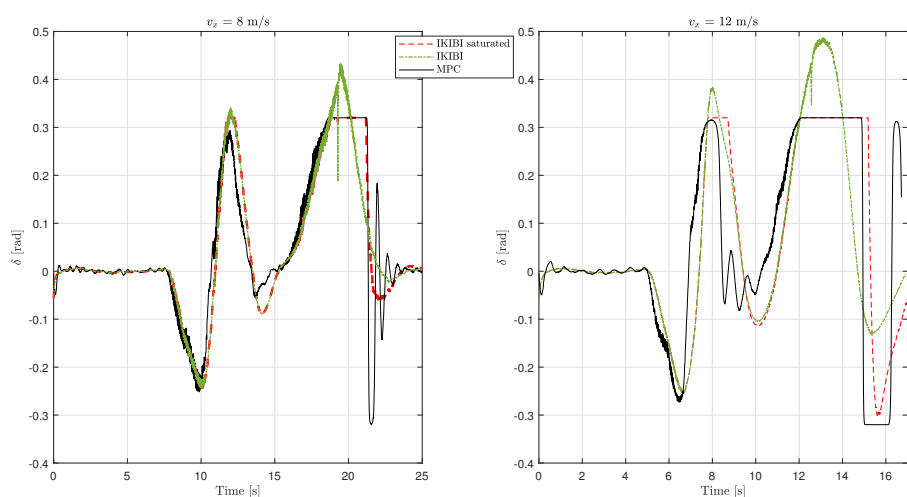


Figure 4. Front wheel steering temporal evolution: noiseless, fast sensor feedback test. (left) $v_x = 8$ m/s and (right) $v_x = 12$ m/s.

The degradation in the lane-following accuracy when the control signal is saturated for the IKIBI controller with the maximum steering angle of the car can be seen in Figure 3. Here, the front wheels cannot physically turn more than 0.32 radians. This degradation becomes more noticeable the more aggressive the maneuver is; here, the higher the longitudinal velocity is.

On the other hand, MPC can explicitly consider in its calculations that the front wheel steering angle has to be bounded to never violate the physical limitations of the real vehicle. Moreover, because of the prediction horizon, when the car has to perform an abrupt maneuver, the MPC anticipates it and starts steering the wheel before the time that the IKIBI controller does.

As mentioned, Figure 4 shows the front-wheel steering temporal evolution. It can be seen here how the MPC controller is able to keep the steering angle inside the desired boundaries, whereas the IKIBI controller will saturate.

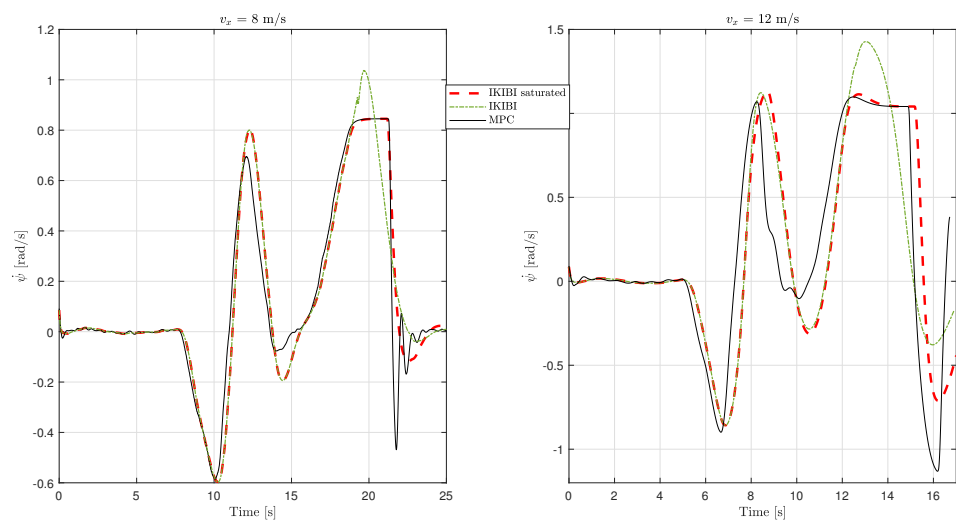


Figure 5. Yaw rate temporal evolution: noiseless, fast sensor feedback test. (**left**) $v_x = 8$ m/s and (**right**) $v_x = 12$ m/s.

Moreover, Figure 5 plots the temporal evolution of the yaw rate throughout the trajectory. As mentioned earlier, because of the predictive nature of the MPC controller in a longer term horizon than the IKIBI controller, it is able to anticipate when a big turn is required and starts steering the vehicle earlier than the other controller analyzed.

As a consequence, the trajectory whose input references were generated by MPC will be smoother. Moreover, MPC can explicitly control the feeling of comfort experienced by the vehicle passengers using the expression (14). Since this expression acts by limiting the yaw rate, the driving experience will be more satisfying when using MPC rather than the IKIBI controller.

Finally, Table 1 shows the performance cost indexes for each of the controllers in this fast, noiseless test. It can be seen how, by explicitly considering the physical limitations of the vehicle such as the maximum front-wheel steering angle over a prediction horizon, the MPC is able to follow the reference path more accurately than its IKIBI counterpart.

Table 1. Cost indexes: noiseless, fast sensor feedback test.

Controller	$v_x = 8$ m/s		$v_x = 12$ m/s	
	J_1 (m)	J_2 (m)	J_1 (m)	J_2 (m)
IKIBI	492.13	0.9	2090.2	5.15
IKIBI saturated	667.3	1.88	3036.1	8.39
MPC	561	1.67	1817.2	6.5

5.2. Fast Sensor Feedback Test with Noise Using EKF

Process and measurement noises are present in a real scenario for this lane-keeping application. Unfortunately, the previous test was observed to turn unstably if these noises were present. Thus, the use of EKF is justified.

Figure 6 plots the planar coordinates of the trajectories in the case where both of these noises are present and an EKF is implemented. As mentioned, since using an EKF is essential to have a stable trajectory, we will not show the unstable results for the tests that did not consider using the EKF.

The behavior seen in Figure 6 is analogous to the former experiment that did not consider noises: the behavior degrades when increasing the longitudinal velocity of the vehicle, and the IKIBI controller is saturated. On the other hand, MPC is still able to control the system from this velocity.

Table 2 shows a quantitative version of what is graphically presented in Figure 6. The MPC controller allows a more accurate lane-keeping behavior when compared to the

proposed IKIBI controller, and this is accentuated the more extreme the situation is: in the presence of measurement and process noises and with high longitudinal velocities.

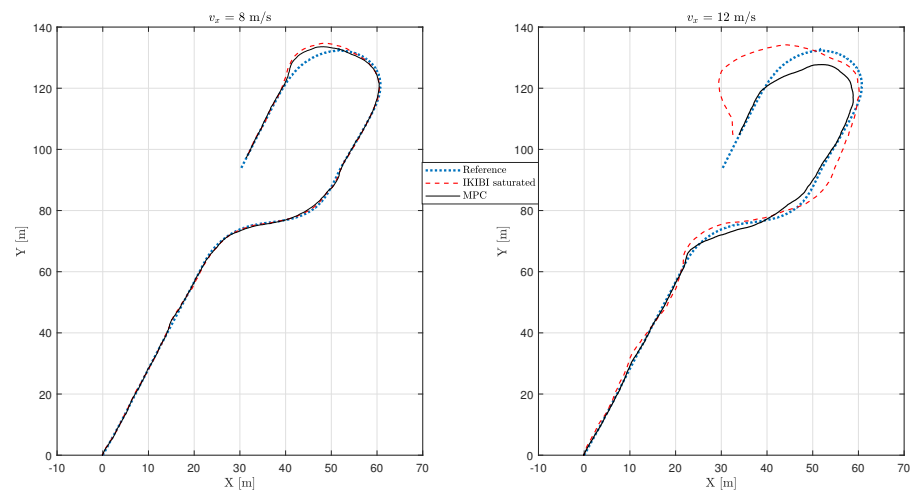


Figure 6. Vehicle path: fast sensor feedback test with noise using EKF. (**left**) $v_x = 8$ m/s and (**right**) $v_x = 12$ m/s.

Table 2. Cost indexes: fast sensor feedback test with noise using EKF.

Controller	$v_x = 8$ m/s		$v_x = 12$ m/s	
	J_1 (m)	J_2 (m)	J_1 (m)	J_2 (m)
IKIBI saturated	999.4	3.42	3660.9	10.77
MPC	834.3	2.63	1269.5	4.54

5.3. Noiseless, Slow Sensor Feedback Test

Nonetheless, the most relevant situation occurs when sensor measurements are not updated every T , but they are updated every MT (here, $M = 10$). In this situation, the controllers have to be calculated M -times slower than in the previous situations. This test explores the situation where no EKF is used and there is no measurement or process noise.

For the IKIBI controller, this situation will necessarily involve keeping the control action constant throughout MT . However, MPC is capable of acting differently. Even though usually, MPC calculates a control sequence over a whole prediction horizon, but only the first control action of this sequence is applied, it is also possible to apply the different control actions of the control horizon if the update rate of the MPC calculations is not fast enough.

Figure 7 shows the comparison between an IKIBI controller calculated every MT and an MPC controller that is calculated every MT , but updates its control signal every T because it uses its entire control horizon.

The disadvantage of this implementation strategy for the MPC controller is that the anticipation ability of MPC is lost, especially in this application where the control horizon is equal to the prediction horizon. As a consequence, the lane-keeping behavior degrades, as seen in Table 3.

However, the MPC strategy is still a more accurate option than the IKIBI controller because of its ability to explicitly constrain physical variables such as the steering angle of the front wheels.

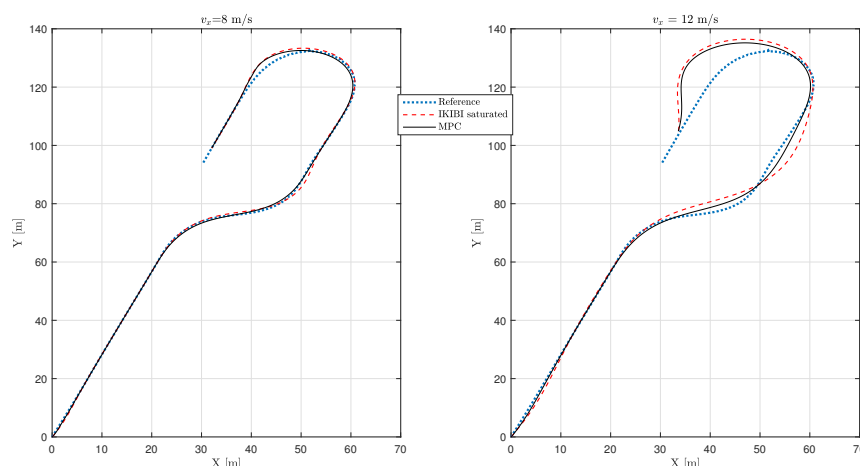


Figure 7. Vehicle path test: noiseless, slow sensor feedback test: (left) $v_x = 8$ m/s and (right) $v_x = 12$ m/s.

Table 3. Cost indexes: noiseless, slow sensor feedback test.

Controller	$v_x = 8$ m/s		$v_x = 12$ m/s	
	J_1 (m)	J_2 (m)	J_1 (m)	J_2 (m)
IKIBI saturated	800.9	1.91	3039.8	7.49
MPC	613.8	1.69	2026.6	6.86

5.4. Slow Sensor Feedback Test with Noise Using the DREKF

Finally, we also considered the situation where the sensor feedback was obtained at a slow rate (every MT) and there were process and measurement noises. The initial test performed in these conditions was to analyze the behavior of each of the two controllers when a Single-Rate EKF (SREKF) was used with a slow sampling frequency. The controllers were also meant to be calculated every MT . However, neither of the two controller strategies (MPC and IKIBI) were able to produce a stable lane-keeping behavior in this situation.

Thus, the use of a Dual-Rate EKF (DREKF) is necessary. The DREKF has the ability to provide new measurements every T while only updating its internal matrices and acquires measured variables every MT . Figure 8 shows the results for implementing the DREKF to calculate both controllers every T while only receiving new sensor data every MT . Moreover, Table 4 shows the cost indexes for this experiment. It can be seen how the DREKF allows an accurate lane-keeping behavior in situations where only slow and noisy sensor feedback is available.

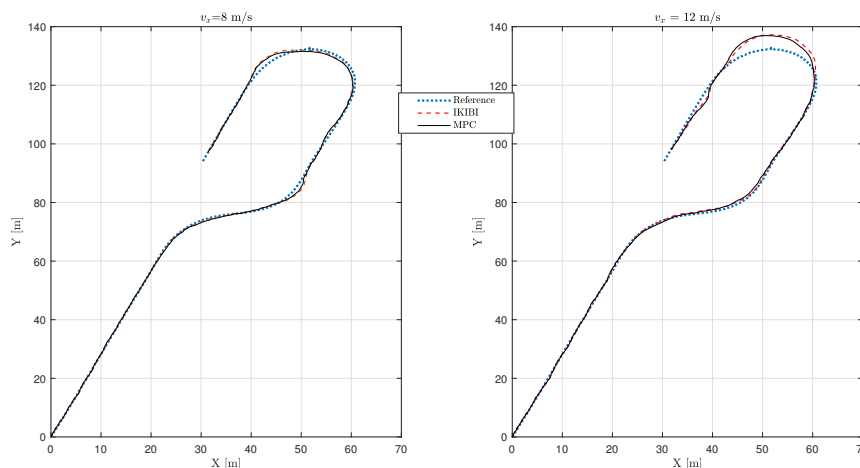


Figure 8. Vehicle path: slow sensor feedback test with noise using the DREKF. (left) $v_x = 8$ m/s (right) $v_x = 12$ m/s.

Table 4. Cost indexes: slow sensor feedback test with noise using the DREKF.

Controller	$v_x = 8 \text{ m/s}$		$v_x = 12 \text{ m/s}$	
	$J_1 \text{ (m)}$	$J_2 \text{ (m)}$	$J_1 \text{ (m)}$	$J_2 \text{ (m)}$
IKIBI saturated	764.76	1.58	1057.3	4.67
MPC	738	1.3	1040.2	4.75

6. Conclusions

The formulation of the model predictive control problem is especially well suited for controlling self-driving cars since it is able to take into consideration long prediction horizons that would be especially important in the event of abrupt maneuvers and in the presence of measurement and process noise.

Additionally, the physical limitations of the vehicle can be explicitly considered, and the comfort of the passengers can be directly taken into consideration by using this control scheme.

For these reasons, MPC provides a more accurate lane-keeping behavior than an IKIBI control strategy. The difference in the accuracy of each of the two controllers can be quantified by the cost indexes introduced in Section 4.

The use of EKF has been essential to obtain a stable behavior of the system in this application when measurement and process noises are present. If the update rate of the sensor data is fast enough, it will suffice to use a standard EKF, called the SREKF in this work.

However, if the update rate of the filter's internal state is too slow, a DREKF should be used, since it will allow providing new variable estimations every T to the controllers so that they can be calculated at a fast rate while updating the internal state every MT .

One alternative to the use of a DREKF would be to use all the input sequence of the control horizon when calculating the MPC controller every MT . Nonetheless, this is a suboptimal solution since there is a loss of the anticipation ability, which is characteristic of MPC. Furthermore, this alternative is only feasible when noise is not present, which happens scarcely in a real application.

Finally, we observe that including a DREKF allows obtaining a degree of lane-keeping accuracy with a slow and noisy sensor feedback similar to the one obtained for the test where there was no noise and sensor data were acquired at a fast rate for both proposed controllers.

Author Contributions: Conceptualization, J.M.S.D., M.T., and J.J.S.L.; methodology, J.M.S.D. and Á.C.; software, J.M.S.D. and J.J.S.L.; validation, J.J.S.L. and Á.C.; formal analysis, J.M.S. and Á.C.; resources, M.T.; writing, original draft preparation, J.M.S.D.; writing, review and editing, J.M.S.D. and Á.C.; visualization, J.J.S.L.; supervision, J.M.S.D.; funding acquisition, J.J.S.L. and Á.C. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the Spanish government Grant Number RTI2018-096590-B-I00.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Acknowledgments: The first author is a member of the ELLIIT Strategic Research Area at Lund University. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
IKIBI	Inverse Kinematics Bicycle
EKF	Extended Kalman Filter
AGV	Autonomous Ground Vehicle
LPV	Linear Parameter Varying
IMU	Inertial Measurement Unit
DOF	Degrees of Freedom

Appendix A. Simulation Model

Simulations were performed using MATLAB Simulink's Vehicle Dynamics Blockset. This toolbox is equipped with a Vehicle Body 3DOF block that implements a rigid two-axle vehicle body model to calculate longitudinal, lateral, and yaw motion. The dynamic equations of the internal model used in this block are [39]:

$$\ddot{y} = -\dot{x}r + \frac{F_{yf} + F_{yr} + F_{y,ext}}{m} \quad (A1)$$

$$\dot{r} = \frac{aF_{yf} - bF_{yr} + M_{z,ext}}{I_{zz}} \quad (A2)$$

$$\dot{\Psi} = r \quad (A3)$$

where F_{yf} and F_{yr} are the lateral forces applied to the front and rear wheels, respectively, along the vehicle-fixed y-axis. Additionally, the external forces that act on the vehicle center of gravity are:

$$F_{x/y/z,ext} = F_{d,x/y/z} + F_{x/y/z,input} \quad (A4)$$

$$M_{x/y/z,ext} = M_{d,x/y/z} + M_{x/y/z,input} \quad (A5)$$

where the subindex d indicates that they are drag forces/moments. Then:

$$F_{xft} = 0 \quad (A6)$$

$$F_{xft} = -C_{\alpha f} \alpha_f \mu_f \frac{F_{zf}}{F_{z,nom}} \quad (A7)$$

$$F_{xrt} = 0 \quad (A8)$$

$$F_{xrt} = -C_{\alpha r} \alpha_r \mu_r \frac{F_{zr}}{F_{z,nom}} \quad (A9)$$

where the subindex t indicates that the forces are acting on the tires. Here,

$$F_{zf} = \frac{bmg - (\ddot{x} - \dot{y}r)mh + hF_{x,ext} + bF_{z,ext} - M_{y,ext}}{a + b} \quad (A10)$$

$$F_{zr} = \frac{amg + (\ddot{x} - \dot{y}r)mh - hF_{x,ext} + aF_{z,ext} + M_{y,ext}}{a + b} \quad (A11)$$

Moreover, the tire forces can be calculated with the slip angles (α):

$$\alpha_f = \arctan\left(\frac{\dot{y} + ar}{\dot{x}}\right) - \delta_f \quad (\text{A12})$$

$$\alpha_r = \arctan\left(\frac{\dot{y} - br}{\dot{x}}\right) - \delta_r \quad (\text{A13})$$

$$F_{xf} = F_{xft} \cos(\delta_f) - F_{yft} \sin(\delta_f) \quad (\text{A14})$$

$$F_{yf} = -F_{xft} \sin(\delta_f) + F_{yft} \cos(\delta_f) \quad (\text{A15})$$

$$F_{xr} = F_{xrt} \cos(\delta_r) - F_{yrt} \sin(\delta_r) \quad (\text{A16})$$

$$F_{yr} = -F_{xrt} \sin(\delta_r) + F_{yrt} \cos(\delta_r) \quad (\text{A17})$$

The physical variables needed to calculate these equations are:

- m , the vehicle body mass.
- a and b , the distance of the front and rear wheels, respectively, from the normal projection point of vehicle's CG onto the common axle plane.
- I_{zz} , the vehicle body moment of inertia about the vehicle-fixed z-axis.
- C_{α} , cornering stiffness. This constant represents a linear approximation for the relationship between the slip angle, α , and the lateral force, F_y .
- μ , the wheel friction coefficient.
- h , the height of the vehicle's center of gravity above the axle plane.

Furthermore, subscripts f and r refer to the front and rear axles, respectively.

Finally, Table A1 shows the numerical values used for the constants presented earlier.

Table A1. Simulation model constants.

Constant	Value	Unit
m	1800	kg
a	1.6	m
b	1.65	m
μ_f	0.6	-
μ_r	0.6	-
$C_{\alpha,f}$	120	kN/rad
$C_{\alpha,r}$	110	kN/rad
h	0.35	m
I_{zz}	3270	kg · m ²

References

1. Haykin, S. *Kalman Filtering and Neural Networks*; Wiley Online Library: Hoboken, NJ, USA, 2001.
2. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 2006; Volume 378.
3. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
4. Rajamani, R. *Vehicle Dynamics and Control*; Springer: New York, NY, USA, 2006. [[CrossRef](#)]
5. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The Stanford Entry in the Urban Challenge. *J. Field Robot.* **2008**, *25*, 569–597. [[CrossRef](#)]
6. Domina, A.; Tihanyi, V. Modelling the dynamic behavior of the steering system for low speed autonomous path tracking. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 535–540. [[CrossRef](#)]
7. Kuwata, Y.; Teo, J.; Fiore, G.; Karaman, S.; Frazzoli, E.; How, J.P. Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control. Syst. Technol.* **2009**, *17*, 1105–1118. [[CrossRef](#)]
8. Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 1094–1099. [[CrossRef](#)]
9. Bakker, E.; Nyborg, L.; Pacejka, H. *Tyre Modeling for Use in Vehicle Dynamics Studies*; SAE Technical Paper 870421; SAE: Warrendale, PA, USA, 1987.
10. Doumiati, M.; Victorino, A.; Charara, A.; Lechner, D. A method to estimate the lateral tire force and the sideslip angle of a vehicle: Experimental validation. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 6936–6942. [[CrossRef](#)]

11. Wang, Y.; Chen, W.; Tomizuka, M. Extended kalman filtering for robot joint angle estimation using mems inertial sensors. *IFAC Proc. Vol.* **2013**, *46*, 406–413. [[CrossRef](#)]
12. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
13. Hesar, H.D.; Mohebbi, M. A Multi Rate Marginalized Particle Extended Kalman Filter for P and T Wave Segmentation in ECG Signals. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 112–122. [[CrossRef](#)] [[PubMed](#)]
14. Akhbari, M.; Shamsollahi, M.B.; Jutten, C. Twave alternans detection in ecg using Extended Kalman Filter and dualrate EKF. In Proceedings of the 2014 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 1–5 September 2014; pp. 2500–2504.
15. Grillo, C.; Vitrano, F. State estimation of a nonlinear unmanned aerial vehicle model using an Extended Kalman Filter. In Proceedings of the 15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, Dayton, OH, USA, 28 April–1 May 2008; p. 2529.
16. Armesto, L.; Tornero, J.; Vincze, M. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *Int. J. Robot. Res.* **2007**, *26*, 577–589. [[CrossRef](#)]
17. Armesto, L.; Tornero, J. SLAM based on Kalman filter for multi-rate fusion of laser and encoder measurements. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 2, pp. 1860–1865.
18. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [[CrossRef](#)]
19. Ljungqvist, O. Motion Planning and Stabilization for a Reversing Truck and Trailer System. Master's Thesis, Linköping University, Linköping, Sweden, 2015.
20. Borrelli, F.; Falcone, P.; Keviczky, T.; Asgari, J.; Hrovat, D. MPC-Based Approach to Active Steering for Autonomous Vehicle Systems. *Int. J. Veh. Auton. Syst.* **2005**, *3*, 265–291. [[CrossRef](#)]
21. Carvalho, A.; Gao, Y.; Gray, A.; Tseng, E.; Borrelli, F. Predictive control of an autonomous ground vehicle using an iterative linearization approach. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, The Hague, The Netherlands, 6–9 October 2013; pp. 2335–2340. [[CrossRef](#)]
22. García, C.; Prett, D.; Morari, M. Model Predictive Control: Theory and Practice—a Survey. *Automatica* **1989**, *25*, 335–348. [[CrossRef](#)]
23. Mayne, D.; Rawlings, J.; Rao, C.; Scokaert, P. Constrained Model Predictive Control: Stability and Optimality. *Automatica* **2000**, *36*, 789–814. [[CrossRef](#)]
24. Besselmann, T.; Morari, M. Autonomous vehicle steering using explicit LPV-MPC. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 2628–2633.
25. Alcalá, E.; Puig, V.; Quevedo, J. LPV-MPC control for autonomous vehicles. *IFAC-PapersOnLine* **2019**, *52*, 106–113. [[CrossRef](#)]
26. Johansson, R. *Model-Based Predictive and Adaptive Control*; KFS Studentbokhandel: Lund, Sweden, 2020.
27. Cerone, V.; Piga, D.; Regruto, D. Set-membership LPV model identification of vehicle lateral dynamics. *Automatica* **2011**, *47*, 1794–1799. [[CrossRef](#)]
28. Jain, V.; Kolbe, U.; Breuel, G.; Stiller, C. Reacting to Multi-Obstacle Emergency Scenarios Using Linear Time Varying Model Predictive Control. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1822–1829. [[CrossRef](#)]
29. Keviczky, T.; Balas, G. Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance. *J. Guid. Control. Dyn.* **2006**, *29*, 680–694. [[CrossRef](#)]
30. Farag Saeed Ramadan, A.; Sayed, A.; Shehata, O.; Garcia, F.; Tadjine, H.; Matthes, E. Dynamics Platooning Model and Protocols for Self-Driving Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1974–1980. [[CrossRef](#)]
31. Salt Ducaju, J.M.; Tang, C.; Chan, C.Y.; Tomizuka, M. Application Specific System Identification for Model-Based Control in Self-Driving Cars. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020.
32. Abbeel, P.; Ganapathi, V.; Ng, A. Learning vehicular dynamics, with application to modeling helicopters. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 1–8.
33. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Technical Report; Carnegie-Mellon UNIV Pittsburgh PA Robotics INST: Pittsburgh, PA, USA, 1992.
34. Buehler, M.; Iagnemma, K.; Singh, S. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*; Springer: New York, NY, USA, 2009; Volume 56.
35. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, E.; Hrovat, D. Predictive Active Steering Control for Autonomous Vehicle Systems. *IEEE Trans. Control. Syst. Technol.* **2007**, *15*, 566–580. [[CrossRef](#)]
36. Yan, Y.; Wang, J.; Zhang, K.; Mingcong, C.; Chen, J. Path Planning using a Kinematic Driver-Vehicle-Road Model with Consideration of Driver's Characteristics. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2259–2264. [[CrossRef](#)]
37. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
38. Mattingley, J.; Boyd, S. CVXGEN: A Code Generator for Embedded Convex Optimization. *Optim. Eng.* **2012**, *12*, 1–27. [[CrossRef](#)]
39. Gillespie, T.D. *Fundamentals of Vehicle Dynamics*; Society of Automotive Engineers (SAE): Warrendale, PA, USA, 1992; Volume 400.