vehicles

MDPI

Article

Heading-Sensorless Autonomous Vehicles: Luenberger-Based Linear MPC for Tracking and Safe Obstacle Avoidance

Meryam El Mahri *, Tarik Jarou, Younes El Koudia, Hassan Mharzi and Salma Garmoum

Advanced Systems Engineering (ASE) Laboratory, National School of Applied Sciences (ENSAK), Ibn Tofail University, Kenitra 14000, Morocco; tarik.jarou@uit.ac.ma (T.J.); younes.elkoudia@uit.ac.ma (Y.E.K.); h.mharzi@uit.ac.ma (H.M.); salma.garmoum@uit.ac.ma (S.G.)

* Correspondence: meryam.elmahri@uit.ac.ma

Abstract

Following a trajectory and avoiding obstacles are two closely linked control tasks for an autonomous vehicle. Classical reactive controllers are tuned for one operating range, do not look ahead and have no built-in way to keep a minimum distance from an obstacle. This paper presents a Linear Time-Invariant Model Predictive Control (LTI-MPC) scheme combined with a Luenberger observer for the kinematic bicycle model. The model is linearised once around $(0, 0, 0, v_{\mathrm{nom}})$ and written in deviation coordinates so that the drift term of the predictor is exact. The observer, tuned with the discrete-time algebraic Riccati equation, rebuilds the heading from the noisy DGPS-RTK position and odometry speed, so no heading sensor is needed. Static obstacles are handled by a local re-planner with a designed clearance, and a closed-form rule links its blending length to the maximum reference heading. Tested in a closed loop with observer and measurement noise on five scenarios, the LTI-MPC reaches a tracking RMSE of 0.018 m at 4 m/s on a sinusoidal path; achieves, in simulation, a clearance of 0.98–0.99 m from the obstacle (the reference is designed for 1.0 m, the requirement is 0.5 m); and is the only controller among PID-type, Stanley and MPC whose accuracy degrades smoothly from 2 to 8 m/s with a single tuning. The observer converges within 1° in 0.24 s. A sweep of constant-radius turns shows that the linearisation stays accurate at up to 30° of heading change.

Keywords: autonomous vehicles; model predictive control; linear time-invariant systems; Luenberger observer; trajectory tracking; obstacle avoidance; kinematic bicycle model

Check for updates

Academic Editors: Mai The Vu, Anh Tuan Vo, Caoyang Yu and Gong Xiang

Received: 2 October 2026
Revised: 24 October 2026
Accepted: 30 October 2026
Published: 9 November 2026

Copyright: © 2026 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.

1. Introduction

Autonomous vehicles represent one of the most active research topics in automation, intelligent transportation and robotics. Moving from SAE level 2 to levels 4 and 5 [1] requires good perception, planning and control under many driving conditions [2]. Trajectory tracking is a central control task: the vehicle must reach and stay on a reference path by acting simultaneously on the front-wheel steering angle (lateral control) and on the longitudinal acceleration [3,4]. Many methods exist: proportional–integral–derivative (PID) control [5,6], pure-pursuit [7], Stanley control [8,9], linear-quadratic state feedback [10], and model predictive control [11–14]. Among them, model predictive control (MPC) stands out because it uses a model of the system, looks ahead over a prediction horizon, handles constraints directly and can combine several objectives in one cost function [15,16]. In a typical PID architecture, a lateral loop acts on the heading error and a longitudinal loop acts on the speed error, sometimes with a look-ahead term to reduce the lag in the curves.

These reactive schemes have well-known limits [3,4]: they do not anticipate, they are tuned for one operating range, they treat the two loops separately although they interact, they do not use the available vehicle model and, most importantly for safety, they have no built-in way to keep a minimum distance from an obstacle.

The first of these limits—the single tuning point—is what caps any fixed-gain design. In a preliminary study on the longitudinal channel of the same vehicle, a speed PID was tuned by multi-objective particle swarm optimisation over a set of grade, load, road friction and actuation delay scenarios. The optimised gains removed the overshoot and cut the energy use of a Ziegler–Nichols baseline, but the study also showed that one set of gains fixed offline cannot stay optimal once the operating conditions leave the range assumed at tuning time and that the overshoot returns as the actuation delay grows. This is the structural limit of fixed-gain PID, and it points to predictive, model-based control as the next step. The present work takes this step. Where the earlier study dealt with speed alone, we now treat lateral and longitudinal control together, swap the fixed PID gains for a model predictive controller built on the kinematic model and a prediction horizon and add two aspects that no fixed-gain law can provide: an observer that estimates a quantity that the sensors do not measure and a planning layer that converts the safety requirement into a geometric condition on the reference path. The comparison with PID-type and Stanley controllers in Section 7 reuses the protocol of the longitudinal study deliberately (each baseline is tuned once and then tested over the whole operating range), so that the robustness gained by prediction can be read directly.

Figure 1 illustrates the difference between the two families: a reactive law only reacts to the error that it sees in the present, while the MPC uses its model to start acting before the curve.

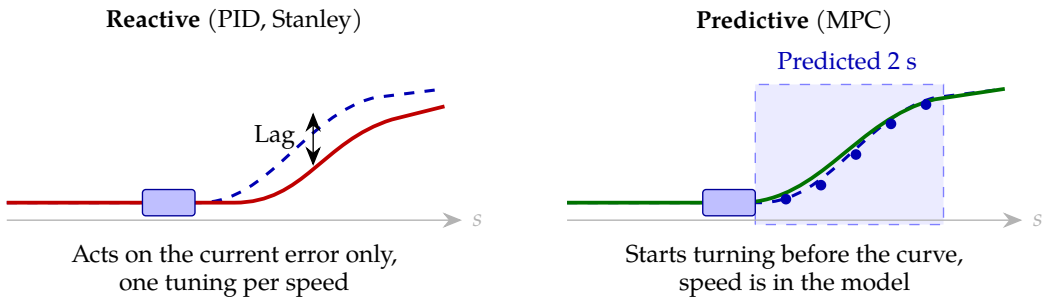


Figure 1. Reactive tracking (**left**) versus predictive tracking (**right**) in a curve. Dashed grey line: reference path; red line: trajectory of the reactive controller; green line: trajectory of the MPC; blue box: prediction horizon.

In practice, the scheme has three parts. First, a linear MPC linearises the kinematic bicycle model once around a nominal operating point $(0, 0, 0, v_{\text{nom}})$, where v_{nom} is chosen for each scenario. The resulting linear time-invariant (LTI) system gives a quadratic program that is solved in real time and whose fixed structure makes analysis and embedded implementation simpler. Second, a Luenberger observer rebuilds the heading angle ψ from noisy position and speed measurements, so no heading sensor is needed. Its gain is computed with the discrete-time algebraic Riccati equation (DARE), which gives a steady-state Kalman-type design with stable linearised error dynamics. Third, a local re-planner generates reference trajectories that keep a chosen clearance from obstacles; the MPC then tracks these references, and the safety margin is checked in simulation.

The three building blocks (linear MPC, Luenberger observer, local re-planner) are individually well known. What this paper adds is the way that they are combined and the design rules that allow the combination to work: (i) the prediction model is written in deviation coordinates so that the affine drift of the moving operating point is accounted for

exactly once, which removes a systematic longitudinal error that appears when the drift is added to an absolute-coordinate predictor; (ii) the heading is reconstructed from the position and odometry only, with the convergence/noise trade-off of the observer measured and used to choose its tuning, so that no heading sensor is required; (iii) the obstacle clearance is built into the reference by a re-planner whose blending length is tied to the linearisation range by a closed-form rule, so that the planner can never drive the controller outside the domain where its model is valid; and (iv) the evaluation uses a single-tuning, multi-speed protocol and documents the validity envelope of the linearisation, instead of reporting accuracy at one operating point. The contributions are as follows: (a) an LTI-MPC written for the four-state kinematic bicycle model with inputs (a, δ), linearised around an operating point chosen per scenario, in the convention of Polack et al. [17]; (b) a Luenberger observer with a DARE-based gain, whose convergence is studied in simulation under realistic DGPS-RTK noise; (c) a planning-and-tracking architecture for static obstacles, where the clearance is built into the reference and kept in practice because the MPC tracking error is small; (d) a MATLAB (R2022b) simulation campaign on five scenarios, all within the validity range of the kinematic model ($a_y \leq 0.5\ \mu g \approx 4.9\,\mathrm{m/s^2}$, per [17]); (e) a fair comparison with PID-type and Stanley controllers (each baseline tuned once by grid search at 4 m/s, all controllers tested from 2 to 8 m/s), which shows that the LTI-MPC gains less in raw accuracy at the tuning point (the three controllers are within a factor of two there) than in robustness over the speed range, where the PID-type law becomes unstable and Stanley degrades by a factor 1.6× at 8 m/s; and (f) a check, on the obstacle scenario, that the scheme keeps about 0.5 m of margin above the 0.5 m requirement (minimum distance ≈ 1.0 m). We also report openly where the LTI approach stops working, with scenario S3, a sweep of constant-radius turns, showing that the fixed linearisation stays accurate up to heading changes of about 30° and degrades beyond 45°.

The rest of the paper is organised as follows. Related work on vehicle MPC, state observers and obstacle avoidance is reviewed in Section 2. Section 3 introduces the vehicle model, its constraints and the reference representation; Sections 4–6 present the controller, the observer and the re-planner in turn. The simulations are reported in Section 7 and discussed in Section 8, and Section 9 closes the paper.

2. Related Work

2.1. MPC for Autonomous Vehicle Trajectory Tracking

In production vehicles, the speed and path loops are still mostly fixed-gain PID controllers, and a large part of the literature focuses on how to tune them rather than on replacing them: metaheuristics such as particle swarm, genetic or salp-swarm search take the place of hand rules for the speed loop [18,19], adaptive laws adjust the gains during operation [20], and multi-objective criteria weigh the tracking error against the overshoot, settling time and energy. All of this improves the response where the gains were tuned, but the gains stay frozen afterwards, and a reference change or a disturbance outside the tuning range is met without anticipation. Model predictive control addresses this at the level of the controller structure rather than of its gains. Recent speed controllers therefore place the powertrain lag and dead time directly into the prediction model of a receding-horizon regulator [21] or keep the PID as an inner loop under an event-triggered MPC [22]. Model predictive control removes this limit at the structural level. Its use for vehicle trajectory tracking dates back to the work of Falcone et al. [11], and it has since grown into a large body of literature. Two main families exist. In *linear time-invariant MPC* (LTI-MPC), the dynamics are linearised once around a nominal operating point: this is cheap to compute but valid over a limited range. In *linear time-varying MPC* (LTV-MPC), it is linearised at every step along the predicted trajectory: this offers better tracking at a slightly higher

cost [13,23]. Domina and Tihanyi [23] used an LTV-MPC for path following at the limit of handling and showed that successive linearisation copes well with strongly nonlinear dynamics. Chen et al. [13] proposed an LTV-MPC steering controller for emergency collision avoidance. Gain-scheduled and LPV formulations [24–26] keep the QP structure of linear MPC while updating the model along a scheduling trajectory, and robust variants based on tube [27] or polytopic vertex [28] descriptions of the parameter uncertainty have recently been proposed for path tracking; adaptive schemes that re-estimate tyre parameters online are reported in [29]. Delay-aware robust formulations are studied in [30], and the influence of the discretisation scheme (forward versus backward Euler) on closed-loop stability and warm-start efficiency is analysed in [31].

More recent works have explored hybrid formulations combining MPC with learning. Xiong et al. [32] proposed a learning-augmented MPC framework using ensemble residual dynamics learning to compensate for model mismatches under sharp curvature conditions, while Fang et al. [33] developed a distributionally robust chance-constrained linear parameter-varying MPC (DRCC-LPVMPC) accounting for model discrepancies through Wasserstein ambiguity sets. A noteworthy variant is linear parameter-varying MPC (LPV-MPC), where the system matrices depend on measurable scheduling variables; Kumar and Pachamuthu [25] presented a real-time LPV-based nonlinear MPC for robust trajectory tracking, emphasising the trade-off between fixed-speed and variable-speed control modes. Chu et al. [34] integrated MPC with PID feedback to combine anticipation with robust steady-state tracking, while Tang et al. [14] introduced an improved kinematic MPC for high-speed path tracking. In [35], Lai et al. paired an improved A* planner with kinematic and dynamic MPC on a single platform and showed where the switch from low- to medium-speed control pays off.

2.2. State Observers for Vehicle Dynamics

Whenever a quantity needed by the controller is not measured, it has to be estimated. For road vehicles, this typically concerns the lateral velocity, the sideslip angle and, when the sensor set is minimal, the heading. The Luenberger observer [36] is still a common choice, being simple and open to analysis; Brivadis et al. [37] carried its original form over to discrete-time nonlinear systems with convergence guarantees under mild observability assumptions. For vehicle sideslip and road friction, an extended Luenberger design based on local linearisation and the Routh–Hurwitz criterion is given in [5]. The extended Kalman filter remains the standard tool in practice [38,39], and recent heading pipelines that fuse dual-antenna GNSS, INS and kinematic constraints through adaptive Kalman filtering reach sub-degree accuracy [40], but they require hardware that this work deliberately omits. While an EKF fed by an IMU generally outperforms a Luenberger observer in the presence of stochastic noise, we retain the Luenberger observer for three reasons that fit a low-cost architecture: it needs no additional sensor (an IMU adds cost, calibration and gyroscope drift to compensate); its gain is computed once offline, so the run-time cost is a matrix–vector product with no covariance propagation; and its behaviour is fixed and inspectable (constant poles, see Section 5), which simplifies certification. The price is a fixed convergence/noise trade-off, quantified in Section 5; an EKF with IMU is the natural upgrade when the sensor budget allows it.

2.3. MPC-Based Obstacle Avoidance

There are three usual ways of bringing obstacles into an MPC. In the hard constraint approach, static or dynamic obstacles are encoded as inequality constraints on the optimisation problem (typically ellipsoidal exclusion regions) [41,42]. More recently, control barrier function (CBF) formulations have been embedded directly inside the MPC problem as safety

filters that guarantee the forward invariance of a safe set [43]; data-driven distributionally robust chance constraints on linearised obstacle half-spaces have also been combined with LPV-MPC and validated on a scaled vehicle [33]. In the soft constraint approach, slack variables are introduced to allow infeasibility recovery at the cost of safety guarantees that hold only with high probability [16]. Finally, in the planning and tracking decomposition, a high-level planner generates a collision-free reference trajectory and a low-level MPC tracks this reference [2,44]; this is the approach adopted in the present paper. Recent works in this context include the two-layer MPC architecture of Gratzer et al. [45], in which a real-time QP-based tracker is asynchronously informed by a mixed-integer planner, and the MPC-based planning and tracking scheme of Dong et al. [44], which combines braking (adaptive cruise) and active steering for collision avoidance with an ADMM-accelerated NMPC planner. The planning and tracking decomposition has the advantage of separating the geometric problem (avoidance) from the dynamic problem (tracking), which simplifies the QP formulation while preserving a clearance margin as a design parameter, with the bounded tracking error empirically observed. Prediction-based collision avoidance has also been demonstrated in the field on unmanned surface vehicles, with learning-based policies [46] and event-triggered nonlinear MPC [47]; the event-triggered scheme is of interest here because it reduces the number of QP solves, which is complementary to the low per-solve cost of the LTI formulation.

Figure 2 summarises the approaches reviewed in this section and shows, at each level, the option retained in this work: the simplest one that is sufficient in the low-speed domain targeted here.

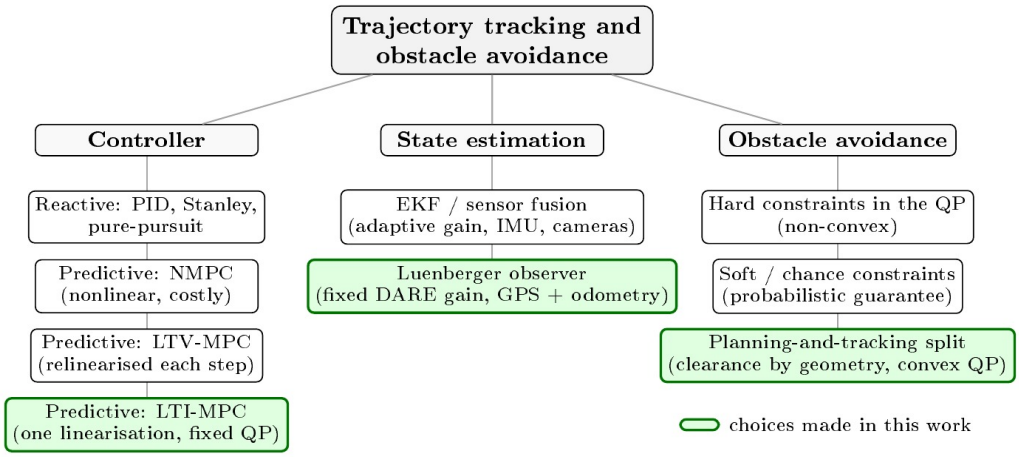


Figure 2. Overview of the reviewed approaches.

3. Vehicle Modelling and Reference Trajectories

3.1. Kinematic Bicycle Model

Figure 3 summarises the problem addressed in this paper. A vehicle must follow a reference path and pass a static obstacle with a clearance of at least d_{safe}. At every sampling period T_s, the controller decides the two inputs, acceleration a and steering angle δ, from the measured position (X, Y) and speed v_x; the heading ψ is not measured and must be estimated. This section defines the vehicle model, the constraints and the reference trajectories used throughout.

The vehicle is modelled by the kinematic bicycle model, in which the wheels of the same axle are merged into a single virtual wheel located at the centre of the axle. This model relies on the no-slip rolling assumption and neglects tyre forces. According to Polack et al. [17], the kinematic model remains consistent as long as the lateral acceleration $a_y = v^2/R$ stays

below the threshold $a_y \leq 0.5\ \mu g \approx 4.9\,\mathrm{m/s^2}$ (with $\mu = 1$, dry road), which restricts the use of the model to low-speed urban driving applications, automatic parking, mobile robots and autonomous shuttles. All scenarios in this paper respect this validity criterion.

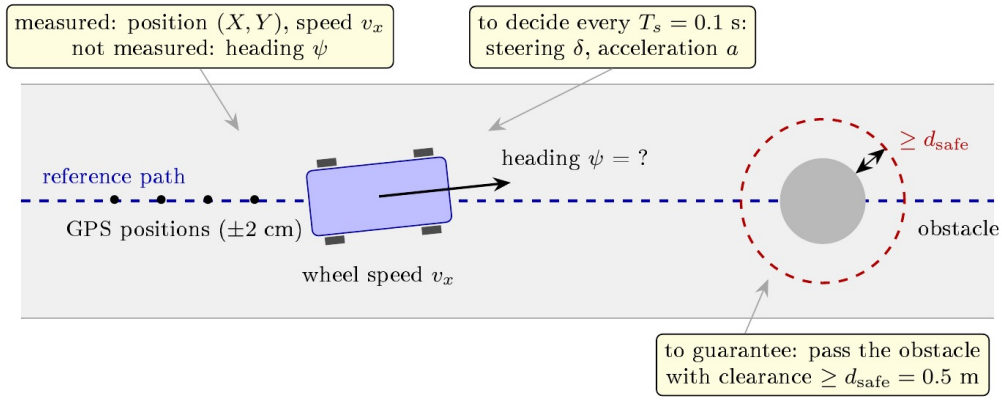


Figure 3. Problem statement: path to follow, obstacle to pass, measured and unmeasured variables. Blue dashed line: reference path; black dots: GPS position fixes; grey disk: obstacle; red dashed circle: safety zone of radius $r_o + d_{\mathrm{safe}}$.

Let (X, Y) denote the position of the rear-axle centre in the global frame, ψ the heading angle, v_x the longitudinal speed, δ the front-wheel steering angle, and $L = l_f + l_r$ the wheelbase. The kinematic equations of motion read

$$\dot{X} = v_x \cos\psi, \quad \dot{Y} = v_x \sin\psi, \quad \dot{\psi} = \frac{v_x}{L}\tan\delta, \quad \dot{v}_x = a, \tag{1}$$

where a is the longitudinal acceleration command. The system has four states and two control inputs, $x = (X, Y, \psi, v_x)^\top \in \mathbb{R}^4$ and $u = (a, \delta)^\top \in \mathbb{R}^2$, and can be written compactly as $\dot{x}(t) = f(x(t), u(t))$. This formulation is consistent with the standard convention in autonomous driving applications [17,48]. The geometric layout of the model is illustrated in Figure 4.

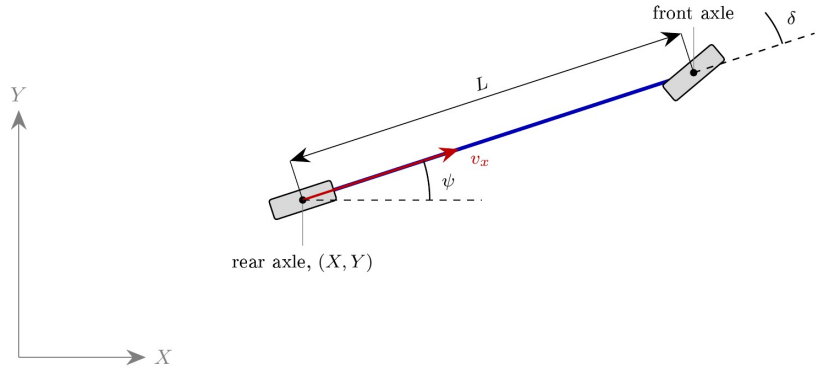


Figure 4. Kinematic bicycle model and notation.

3.2. Constraints

The vehicle is subject to physical constraints on the steering angle and its rate, $|\delta| \leq \delta_{\max} = 35°$ and $|\dot{\delta}| \leq \dot{\delta}_{\max} = 120°/\mathrm{s}$, and on the longitudinal acceleration, $a_{\min} = -5 \leq a \leq a_{\max} = +3\,\mathrm{m/s^2}$, together with a jerk limitation $|\dot{a}| \leq 3\,\mathrm{m/s^3}$. The asymmetric bounds on a reflect the difference between accelerating and braking capabilities in a typical passenger vehicle.

3.3. Reference Trajectory Representation

A reference trajectory is described by a parameterised curve $x_r(s) = (X_r(s), Y_r(s), \psi_r(s), v_{x,r}(s))^\top$, where $s \in [0, s_f]$ is an arc length parameter. The reference heading ψ_r is consistent with the curve tangent, $\psi_r(s) = \arctan_2(dY_r/ds, dX_r/ds)$. For real-time MPC, the reference is sampled along the prediction horizon at intervals $\Delta s = v_{x,r} T_s$, where T_s is the control sampling period.

4. Linear MPC Formulation

This section builds the LTI-MPC that tracks the reference while respecting the constraints above. Unlike LTV-MPC, which re-linearises at every step, the scheme linearises the kinematic model once around a fixed operating point, which gives a linear time-invariant (LTI) prediction model. There are four reasons for this choice.

(i) *The linearisation is valid over the target range.* In low-speed urban driving ($|\psi| \lesssim 30°$, $|\delta| \lesssim 30°$), the model error stays small around the linearisation point. Outside this range (highway speeds, emergency manoeuvres), LTV-MPC or NMPC would be better.

(ii) *Low computational cost.* The matrices A, B and d are computed once and stored; LTV-MPC has to re-linearise along the predicted trajectory at every step, which typically adds 5–15 ms per step; a systematic comparison of linear and nonlinear MPC solve times across solvers and CPUs is given in [49].

(iii) *Simple stability analysis.* With constant matrices, the classical tools of linear-quadratic control (Lyapunov stability, robustness margins, pole placement) apply directly and give clear guarantees on the linearised closed loop. In the LTV case, these tools become much heavier (parametric LMIs, time-varying ISS).

(iv) *Easy embedded deployment.* The QP has a fixed structure and can be solved by a general solver such as MATLAB's `quadprog` or, when speed matters, by an embedded solver such as OSQP [50] or the code-generating ACADO toolkit [51], which make full use of warm starting. The fixed matrices A and B can also be stored in ROM, which simplifies code generation for automotive ECUs.

The price to pay is a loss in accuracy when the state moves away from the linearisation point. We reduce it by adapting the operating-point speed v_{nom} to each scenario, which is a simple form of gain scheduling (Section 4.1).

4.1. Linearisation Around an Adaptive Operating Point

The vehicle dynamics are linearised around the operating point

$$\bar{x} = (0, 0, 0, v_{\text{nom}})^\top, \qquad \bar{u} = (0, 0)^\top, \tag{2}$$

where v_{nom} is the nominal cruising speed of the considered scenario. In our experiments, $v_{\text{nom}} = 5\,\text{m/s}$ for scenarios S1 and S3, $v_{\text{nom}} = 4\,\text{m/s}$ for S4 and v_{nom} is chosen to match the test speed for the multi-speed scenarios S2 (2, 4, 6 m/s) and S5 (3, 5, 7 m/s). Adapting v_{nom} to each scenario keeps the linearisation accurate over the relevant operating range, while preserving the LTI structure during operation. Two points deserve to be made explicit. First, changing v_{nom} is a model adaptation, not controller retuning: the weights Q, R, $R_{\Delta u}$, P, the horizon N and the constraints are the same in every scenario in this paper, and only the entries $A_{23} = v_{\text{nom}} T_s$ and $B_{32} = v_{\text{nom}} T_s / L$ of the prediction model change. Second, in operation, v_{nom} is taken from the mission: a low-speed vehicle follows a planned route with a known cruise speed, so the planner provides v_{nom} together with the reference trajectory; it is held constant along a route segment and may be switched between segments, since the matrices for a new value are obtained in closed form from (3). The comparison in Section 7.6, where the MPC is run from 2 to 8 m/s, shows the behaviour when the actual speed departs

from the design speed; the sensitivity test reported there (Section 7.6) quantifies the loss in accuracy for a ± 1 m/s mismatch.

The first-order Taylor expansion of f around $(\bar{x}, \bar{u})$ gives $f(x,u) \approx A^c(x-\bar{x}) + B^c(u-\bar{u})$, with the continuous-time Jacobians evaluated at $\bar{x}$ and $\bar{u}$. For the kinematic model, since $\sin(0) = 0$, $\cos(0) = 1$, $\tan(0) = 0$ and $1/\cos^2(0) = 1$, the Jacobians take the simple closed form

$$A^c = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & v_{\text{nom}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad B^c = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & v_{\text{nom}}/L \\ 1 & 0 \end{pmatrix}. \tag{3}$$

This compact form reveals the key coupling of the kinematic model: the lateral velocity $\dot{Y}$ is proportional to $v_{\text{nom}}\psi$, while the heading rate $\dot{\psi}$ is proportional to $(v_{\text{nom}}/L)\delta$.

4.2. Discretisation in Deviation Coordinates

We discretise the system using the explicit Euler method with sampling period $T_s = 0.1$ s, yielding $A = I + A^c T_s$ and $B = B^c T_s$. The operating point $\bar{x}$ is not an equilibrium, since $f(\bar{x}, \bar{u}) = (v_{\text{nom}}, 0, 0, 0)^\top \neq 0$: the vehicle keeps advancing along X at the nominal speed. The prediction model is therefore written in *deviation coordinates* $\tilde{x} = x - \bar{x}$ (and $\tilde{u} = u - \bar{u} = u$), for which the Euler-discretised first-order expansion reads

$$\tilde{x}_{j+1} = A\tilde{x}_j + Bu_j + d, \qquad d = T_s f(\bar{x}, \bar{u}) = (v_{\text{nom}} T_s, 0, 0, 0)^\top, \tag{4}$$

Here, d is the constant term that appears because the operating point is not an equilibrium: the vehicle keeps moving along X at the nominal speed. Since $A^c\bar{x} = f(\bar{x}, \bar{u})$ for the kinematic model, (4) holds exactly for the linearised dynamics and is the same as $x_{j+1} = Ax_j + Bu_j$ written in absolute coordinates. Two aspects follow for the implementation. A predictor working on deviations $\tilde{x}$ must contain d: without it, the predicted vehicle falls behind by a constant amount per step, the error grows with the horizon and the tracking of a straight line fails. A predictor written in absolute coordinates, on the other hand, must not be given d, because the term $A_{14}v_{x,j}$ already produces the advance $v_{\text{nom}}T_s$, and the drift would then be counted twice. The three cases are illustrated in Figure 5.

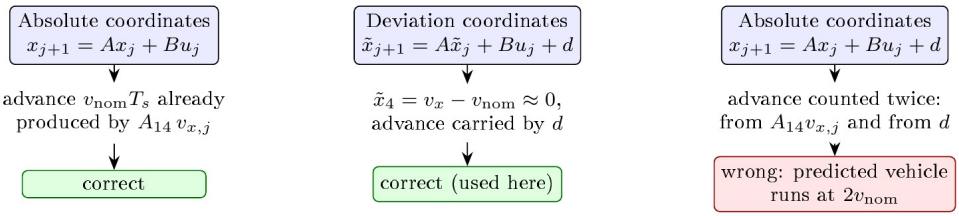


Figure 5. Correct and incorrect means of including the drift term d in the predictor.

The sampling period $T_s = 0.1$ s is a standard choice for automotive MPC [11,12], balancing responsiveness (typical actuator bandwidth ≈ 5 Hz) with the computational load (10 Hz control rate).

4.3. Cost Function

Over the prediction horizon of N steps, the MPC cost penalises three aspects: the tracking error on the four state components, the size of the two inputs and the change in the inputs from one step to the next (which keeps the commands smooth and helps to respect the actuator rate limits). The predicted states x_j and the reference samples $x_{r,j}$ are both expressed in deviation coordinates, $\tilde{x}_j = x_j - \bar{x}$ and $\tilde{x}_{r,j} = x_{r,j} - \bar{x}$; since the offset $\bar{x}$ cancels in

the difference, $\tilde{x}_j - \tilde{x}_{r,j} = x_j - x_{r,j}$, and the cost can be written with the physical components of the state $x_j = (X_j, Y_j, \psi_j, v_{x,j})^\top$ and of the input $u_j = (a_j, \delta_j)^\top$:

$$\begin{aligned} J = &\sum_{j=1}^{N-1} \left[q_X (X_j - X_{r,j})^2 + q_Y (Y_j - Y_{r,j})^2 + q_\psi (\psi_j - \psi_{r,j})^2 + q_{v_x} (v_{x,j} - v_{x,r,j})^2 \right] \\ &+ \sum_{j=0}^{N-1} \left[r_a a_j^2 + r_\delta \delta_j^2 + r_{\Delta a} (a_j - a_{j-1})^2 + r_{\Delta\delta} (\delta_j - \delta_{j-1})^2 \right] \\ &+ p_X (X_N - X_{r,N})^2 + p_Y (Y_N - Y_{r,N})^2 + p_\psi (\psi_N - \psi_{r,N})^2 + p_{v_x} (v_{x,N} - v_{x,r,N})^2. \end{aligned} \tag{5}$$

The state terms start at $j = 1$: x_0 is the current estimate, which no input can change, so its term would be a constant. The final state x_N has its own weight. In compact form,

$$J = \sum_{j=1}^{N-1} \|x_j - x_{r,j}\|_Q^2 + \sum_{j=0}^{N-1} \left[\|u_j\|_R^2 + \|u_j - u_{j-1}\|_{R_{\Delta u}}^2 \right] + \|x_N - x_{r,N}\|_P^2, \tag{6}$$

where $\|v\|_M^2 = v^\top M v$ and the weighting matrices are diagonal:

$$\begin{aligned} Q &= \mathrm{diag}(q_X, q_Y, q_\psi, q_{v_x}), \qquad R = \mathrm{diag}(r_a, r_\delta), \\ R_{\Delta u} &= \mathrm{diag}(r_{\Delta a}, r_{\Delta\delta}), \qquad P = \mathrm{diag}(p_X, p_Y, p_\psi, p_{v_x}). \end{aligned} \tag{7}$$

Each term is a square in its own unit (m^2, rad^2, $(\mathrm{m/s})^2$, $(\mathrm{m/s}^2)^2$), so the weights of Table 1 also perform the unit conversion between them.

Table 1. LTI-MPC parameters.

Parameter	Symbol	Value
Horizon and timing		
Prediction horizon	N	20
Sampling period	T_s	0.1 s
State tracking weights (stage)		
Longitudinal position weight	q_X	1
Lateral position weight	q_Y	100
Heading weight	q_ψ	50
Longitudinal speed weight	q_{v_x}	10
Control magnitude weights		
Acceleration weight	r_a	0.1
Steering weight	r_δ	5.0
Control rate weights		
Acceleration rate weight	$r_{\Delta a}$	1.0
Steering rate weight	$r_{\Delta\delta}$	30.0
Terminal weights ($P = 2Q$)		
Terminal X weight	p_X	2
Terminal Y weight	p_Y	200
Terminal ψ weight	p_ψ	100
Terminal v_x weight	p_{v_x}	20
Operating point		
Nominal cruising speed	v_{nom}	2–7 m/s (scenario-dependent)

The weights indicate what the controller should prioritise. Staying in the lane and pointing along it are the safety-critical quantities, so q_Y and q_ψ are set much higher than q_X and q_{v_x}; being slightly ahead of or behind the reference along X has no negative impact and is corrected through the speed. Steering is weighted more than acceleration ($r_\delta > r_a$)

because a smooth steering command matters for comfort and tyre wear, while moderate changes in acceleration are acceptable at low speed. Smoothness comes mainly from the rate weights $R_{\Delta u}$: $r_{\Delta\delta}$ is the largest weight in the problem, and it is what prevents the steering rate limit of Section 3 from being reached. For $j = 0$, the rate term uses u_{-1}, the input applied at the previous sampling instant t_{k-1}, so that the command is continuous from one MPC iteration to the next.

The terminal weight P stands for the cost beyond the horizon. The theoretically exact choice is the solution of the discrete-time Riccati equation of the unconstrained LQR problem; we use the simpler choice $P = 2Q$, which gives a stable closed loop in all scenarios of Section 7 at no computational cost.

The complete LTI-MPC problem at time t_k reads

$$\begin{aligned} &\min_{u_0,\dots,u_{N-1}} J \\ &\quad \text{s.t. } \tilde{x}_0 = \hat{x}(t_k) - \bar{x}, \quad u_{-1} = u_0^\star(t_{k-1}), \\ &\qquad \tilde{x}_{j+1} = A\tilde{x}_j + Bu_j + d, \\ &\qquad u_{\min} \le u_j \le u_{\max}, \quad |u_j - u_{j-1}| \le \Delta u_{\max}, \quad j = 0, \dots, N-1, \end{aligned} \tag{8}$$

where $\hat{x}(t_k)$ is the current state estimate provided by the Luenberger observer (Section 5). The bounds are applied component-wise, with $u_{\min} = (a_{\min}, -\delta_{\max})^\top$, $u_{\max} = (a_{\max}, \delta_{\max})^\top$ and $\Delta u_{\max} = (\dot{a}_{\max}, \dot{\delta}_{\max})^\top T_s$. The receding-horizon implementation applies only the first control $u_0^\star$ from the optimal sequence and then re-solves (8) at the next sampling instant. Problem (8) is a convex quadratic program with $N \cdot n_u = 20 \times 2 = 40$ decision variables (the $n_u = 2$ inputs at each of the N prediction steps; the predicted states are eliminated through the model); it is solved using MATLAB's `quadprog` from the Optimization Toolbox [52], using its interior-point convex algorithm, which is well suited to medium-scale convex QPs. Alternative solvers, such as the dedicated MPC solvers in MATLAB's Model Predictive Control Toolbox [53], can be used interchangeably without algorithmic modification.

4.4. Algorithm and Tuning

Algorithm 1 summarises the LTI-MPC procedure. Note that, in contrast to LTV-MPC schemes, the matrices A, B and d are computed only once during initialisation, after which the QP is solved with constant prediction matrices.

Algorithm 1 LTI-MPC for trajectory tracking

Require: Reference $x_r(\cdot)$, current estimate $\hat{x}(t_k)$, previous input u_{-1}, operating point v_{nom}.
Ensure: Control $u_0^\star$ to apply at t_k.
1: **Initialisation (once):** compute A^c, B^c at $\bar{x} = (0, 0, 0, v_{\text{nom}})^\top$; discretise to obtain A, B, d.
2: Sample reference: $x_{r,j} \leftarrow x_r(s(t_k) + jv_{\text{nom}}T_s)$, $j = 0, \dots, N$.
3: Solve QP (8) (with constant A, B, d) to obtain $u^\star_{0:N-1}$.
4: **return** $u_0^\star$.

Figure 6 shows how Algorithm 1 works over time. At each sampling instant t_k, the controller solves the QP (8) over a prediction horizon of N steps, obtains the optimal control sequence $u^\star_{0:N-1}$, applies only the first element $u_0^\star$ to the vehicle, and, at the next instant t_{k+1}, re-solves the QP using fresh state measurements and a window shifted one step forward. In the figure, the predicted states $\{x_j\}_{j=0}^N$ appear in blue against the reference (dashed grey), the applied input $u_0^\star$ in red in the lower panel, the discarded inputs $u^\star_{1:N-1}$ in light blue and the shifted window in orange.

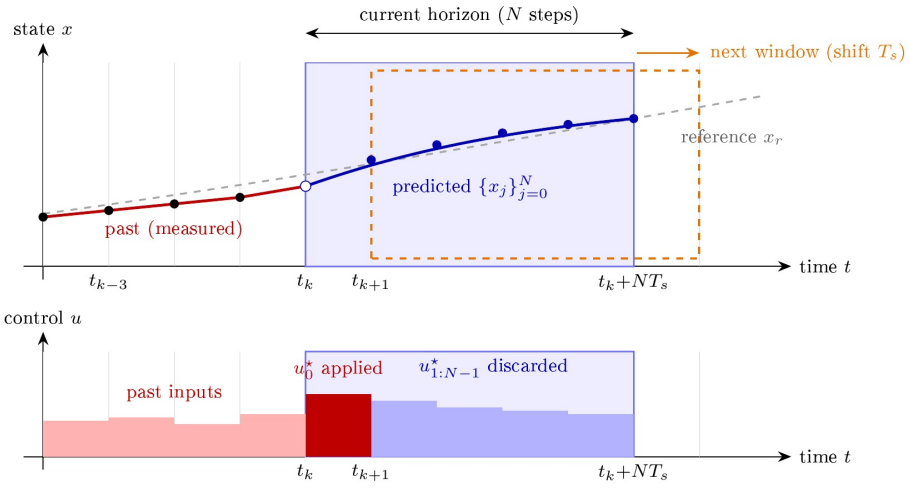


Figure 6. Receding horizon principle: predicted states (**top**) and input sequence (**bottom**).

The MPC weighting matrices are tuned through a hybrid approach combining engineering intuition and parametric optimisation on the sinusoidal scenario. The retained values are reported in Table 1, with the weights expressed component-wise as in (5).

5. Luenberger State Observer

In a realistic deployment, the heading angle ψ is not directly available from low-cost sensors (low-end IMUs are noisy and prone to drift, while higher-end inertial units significantly increase the sensor cost). We therefore design a Luenberger observer that reconstructs ψ from position and longitudinal speed measurements only.

5.1. Measurement Model and Observability

We assume that the measurable quantities are $y(t) = (X_m, Y_m, v_{x,m})^\top = Cx(t) + \nu(t)$, where

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 4} \tag{9}$$

extracts the first, second and fourth components of the state and ν is the measurement noise. In a typical configuration, X_m and Y_m come from differential GPS-RTK, and $v_{x,m}$ comes from wheel encoder odometry; the heading ψ is not directly measured.

Observability Analysis

The observability of the linearised model around a nominal operating point is studied via the Kalman observability matrix $O = \left(C^\top, (CA)^\top, (CA^2)^\top, (CA^3)^\top\right)^\top \in \mathbb{R}^{12\times 4}$. A direct calculation for the linearised system (4) at $\bar{x} = (0, 0, 0, v_{\text{nom}})^\top$ yields

$$\operatorname{rank}(O) = \begin{cases} 4 & \text{if } v_{\text{nom}} > 0 \\ 3 & \text{if } v_{\text{nom}} = 0 \end{cases}. \tag{10}$$

The system is therefore fully observable as long as the vehicle is in motion. Intuitively, ψ can be inferred from the temporal evolution of (X, Y), since $\dot{Y}/\dot{X} \approx \tan\psi$ for small ψ; this inference becomes impossible when the vehicle is stationary.

The rank condition holds for every $v_{\text{nom}} > 0$, but the matrix O becomes poorly conditioned as the speed drops. Table 2 lists its smallest singular value $\sigma_{\min}(O)$ for the discrete-time system at $T_s^{\text{obs}} = 10\,\text{ms}$: below $v_{\text{nom}} = 0.5\,\text{m/s}$, the condition number $\kappa(O) = \sigma_{\max}/\sigma_{\min}$

exceeds 200, so the heading is observable in theory but poorly in practice. This is why the convergence result below assumes $v_x > 0.5\,\mathrm{m/s}$. Figure 7 shows the geometry behind this.

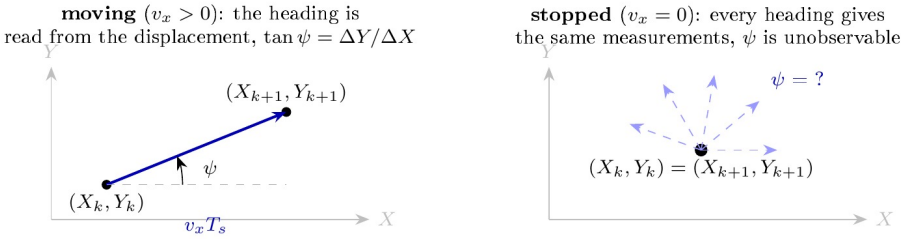


Figure 7. Observability of the heading from GPS fixes: moving (**left**) and stationary (**right**).

Table 2. Conditioning of the observability matrix for several values of v_{nom}.

v_{nom} (m/s)	rank ($\mathcal{O}$)	$\sigma_{\min}$ ($\mathcal{O}$)	κ ($\mathcal{O}$)
0.0	3	0	∞
0.1	4	2.2×10^{-3}	901
0.3	4	6.7×10^{-3}	300
0.5	4	1.1×10^{-2}	180
1.0	4	2.2×10^{-2}	90
5.0	4	1.1×10^{-1}	18

5.2. Observer Design

The observer is built on the same linearised model as the MPC, in the deviation coordinates of Section 4.2. Its continuous-time form is

$$\dot{\hat{\tilde{x}}}(t) = A_c\hat{\tilde{x}}(t) + B_c u(t) + f(\bar{x}, \bar{u}) + L\big(y(t) - C\hat{\tilde{x}}(t) - C\bar{x}\big), \tag{11}$$

where $L \in \mathbb{R}^{4\times 3}$ is the observer gain and $C\bar{x} = (0, 0, v_{\mathrm{nom}})^\top$ accounts for the offset between the deviation and measured coordinates. The estimation error $e = \tilde{x} - \hat{\tilde{x}}$ satisfies $\dot{e} = (A_c - LC)e - L\nu + \varepsilon(x, u)$, where ε collects the linearisation residual of the true nonlinear dynamics. We design L via a Kalman-like steady-state approach: the structure is that of a Luenberger observer (constant gain, no online covariance propagation), and the gain is the steady-state Kalman gain, so the two names describe the same estimator; we keep the name Luenberger to stress that no covariance is updated at runtime. Specifically, the DARE is solved for the discretised system with process noise covariance Q_{KF} and measurement noise covariance $R_{\mathrm{KF}} = \mathrm{diag}(0.01, 0.01, 0.01)$, yielding

$$L_d = P_\infty C^\top (C P_\infty C^\top + R_{\mathrm{KF}})^{-1}, \tag{12}$$

where P_∞ is the steady-state solution of the DARE (MATLAB `idare`). The observer is implemented in discrete time with $T_s^{\mathrm{obs}} = 0.01\,\mathrm{s}$ (ten times faster than the MPC) using the prediction–correction recursion

$$\hat{\tilde{x}}_{k+1}^{-} = A\hat{\tilde{x}}_k + Bu_k + d, \qquad \hat{\tilde{x}}_{k+1} = \hat{\tilde{x}}_{k+1}^{-} + L_d\big(y_{k+1} - C\hat{\tilde{x}}_{k+1}^{-} - C\bar{x}\big), \tag{13}$$

where A, B and d are the deviation-coordinate matrices of (4) discretised at T_s^{obs}. The data flow is illustrated in Figure 8: the predictor propagates the previous estimate with the applied control and the drift d, the innovation $y_{k+1} - C\hat{x}_{k+1}^{-}$ is weighted by the constant gain L_d (computed once offline via the DARE) and the corrected estimate is both fed back for the next step and delivered to the MPC.

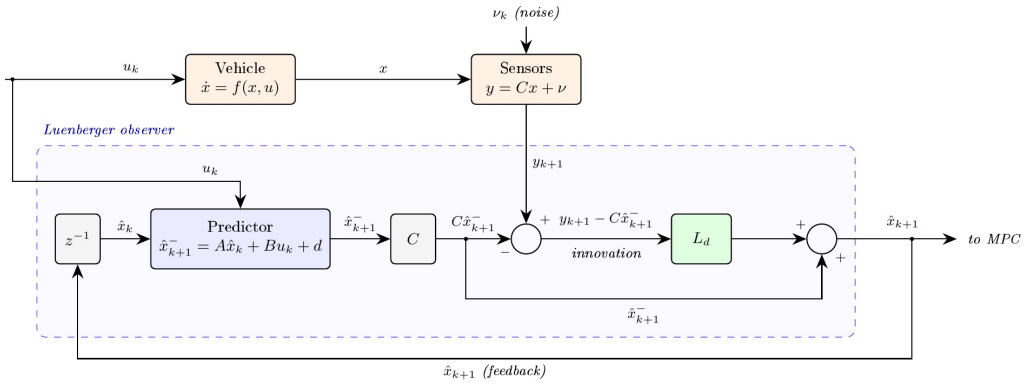


Figure 8. Block diagram of the discrete-time Luenberger observer.

5.2.1. Choosing the Heading Process Noise Weight

The entry of Q_{KF} associated with ψ decides the extent to which the observer believes the model rather than the measurements for the unmeasured state and therefore how quickly $\hat{\psi}$ converges versus how much measurement noise it passes on. This point matters here because $\hat{\psi}$ feeds an MPC with a large heading weight q_ψ: any noise that is in $\hat{\psi}$ ends up in the steering command. Table 3 quantifies this trade-off in the sinusoidal scenario at 4 m/s, with an initial heading error of $-3.2°$ and the noise levels of Section 7; it reports the dominant time constant τ_{dom}, the time $t_{1°}$ to reach $|\tilde{\psi}| < 1°$, the RMS estimation error after 1 s, the closed-loop tracking RMSE and the mean steering variation per MPC step $\overline{|\Delta\delta|}$. The fastest setting (10^{-2}) converges the fastest but leaves the largest RMS heading error and a mean step-to-step steering variation above $4°$, while 10^{-3} more than halves both figures at the cost of slightly slower convergence, and smaller values bring diminishing returns while slowing convergence to about 1 s. We retain $Q_{\mathrm{KF}} = \mathrm{diag}(10^{-3}, 10^{-3}, 10^{-3}, 10^{-3})$ for all simulations.

Table 3. Observer trade-off versus the heading process noise weight $Q_{\mathrm{KF},\psi}$.

$Q_{\mathrm{KF},\psi}$	τ_{dom} (ms)	$t_{1°}$ (s)	$\tilde{\psi}$ RMS (°)	Tracking RMSE (m)	$\overline{\lvert\Delta\delta\rvert}$ (°)
10^{-2}	53	0.16	1.23	0.023	4.24
10^{-3}	236	0.24	0.47	0.017	1.88
3×10^{-4}	444	0.55	0.31	0.016	1.35
10^{-4}	779	0.99	0.25	0.015	1.08

5.2.2. Closed-Loop Poles

For $v_{\mathrm{nom}} = 4\,\mathrm{m/s}$ and the retained parameters, the DARE yields the closed-loop poles of $A - L_d C$ given in Table 4. All poles lie strictly inside the unit disc ($|z|_{\max} = 0.959$). The dominant time constant (in continuous time) is $\tau_{\mathrm{dom}} = 236\,\mathrm{ms}$, consistent with the empirical time of 0.24 s needed to bring an initial $3.2°$ heading error below $1°$ ($3.2\,e^{-t/\tau_{\mathrm{dom}}} < 1 \Rightarrow t \approx 1.16\,\tau_{\mathrm{dom}}$); see Section 7.8.

Table 4. Closed-loop observer poles at $v_{\mathrm{nom}} = 4\,\mathrm{m/s}$.

Pole (Discrete z)	Continuous s (rad/s)	τ (ms)
$0.730 \pm 0.005j$	$-31.5 \pm 0.6j$	32
0.744	−29.5	34
0.959	−4.2	236

5.2.3. Convergence of the Observer

Assume $|\psi - \psi_0| < \pi/4$, $v_x > 0.5\,\mathrm{m/s}$ and $|\delta| < \delta_{\max}$. Then, for a small enough initial estimation error $e(0)$ and bounded measurement noise $\|\nu\|_\infty \le \bar{\nu}$, the observer satisfies the input-to-state stability bound

$$\|e(t)\| \le c_1 e^{-\lambda t}\|e(0)\| + c_2\bar{\nu}, \qquad \forall t \ge 0, \tag{14}$$

with decay rate $\lambda \ge |\mathrm{Re}(s_{\mathrm{dom}})| = 4.2\,\mathrm{rad/s}$ (i.e., $\tau_{\mathrm{dom}} = 236\,\mathrm{ms}$) and constants $c_1, c_2 > 0$ determined by the spectrum of $A_c - LC$.

Why this holds. The dominant terms in the linearised error dynamics are governed by $A_c - LC$, whose eigenvalues lie strictly in the open left half-plane by DARE construction. The exponential decay rate is bounded below by the slowest pole. The higher-order terms $\varepsilon(x, u)$ are bounded under the stated assumptions on ψ, v_x and δ, which guarantee that the linearisation remains accurate. A complete proof follows standard converse Lyapunov arguments for locally Lipschitz perturbations of an exponentially stable linear system; see, e.g., [37] for the discrete-time nonlinear setting.

Note that the observer validity region ($|\psi - \psi_0| < \pi/4$) is wider than the accurate tracking envelope of the LTI-MPC ($|\psi| \lesssim 30°$, Section 7.3): the estimate converges even in manoeuvres where the controller's tracking accuracy degrades.

5.2.4. Parametric Robustness

The observer is structurally robust to uncertainties in the wheelbase L: since neither A_c nor C depends on L (only B_c does), the spectrum of $A_c - LC$ is independent of L. A wheelbase mismatch only enters the error dynamics through the input term $(B_c - \hat{B}_c)u$, a bounded disturbance proportional to $|\Delta L|\,|\delta|$ that shifts the steady-state estimation error without affecting convergence. If the gain L_d is computed for v_{nom} while the vehicle actually drives at v_x, the dominant time constant changes: with L_d designed at 4 m/s and $v_x \in \{2, 6\}\,\mathrm{m/s}$, the eigenvalues of $A(v_x) - L_d(v_{\mathrm{nom}})C$ give τ_{dom} between 138 and 520 ms, with the loop staying stable in all cases. Updating v_{nom} per scenario, as the MPC does, brings the observer back to its nominal time constant.

6. Local Trajectory Re-Planning for Obstacle Avoidance

The LTI-MPC of Section 4 excels in trajectory tracking, but it does not natively reason about obstacles. To enable obstacle avoidance, we introduce a local re-planner that modifies the reference trajectory to ensure a designed clearance to all known static obstacles. A static obstacle is modelled as a disk $O_i = \{(x, y) : (x - x_{o,i})^2 + (y - y_{o,i})^2 \le r_{o,i}^2\}$, where $(x_{o,i}, y_{o,i})$ is the centre and $r_{o,i}$ the physical radius. The vehicle is treated as a point (its footprint is folded into d_{safe}) and has to stay at least d_{safe} away from every obstacle—that is, $\sqrt{(X - x_{o,i})^2 + (Y - y_{o,i})^2} \ge r_{o,i} + d_{\mathrm{safe}} \triangleq R_i$.

Starting from a baseline reference $(X_r(s), Y_r(s))$ and an obstacle O_i, the re-planner builds a new path $(\tilde{X}_r(s), \tilde{Y}_r(s))$ that follows the baseline away from the obstacle, moves sideways near it so that the clearance condition holds and stays continuously differentiable so that the MPC can track it as well as the original. We adopt a Gaussian bell weighted lateral displacement,

$$\tilde{Y}_r(s) = Y_r(s) + (Y_{\mathrm{tgt}} - Y_r(s))\exp\left(-\frac{(X_r(s) - x_{o,i})^2}{2\sigma_o^2}\right), \tag{15}$$

Here, $Y_{\mathrm{tgt}} = y_{o,i} + \varepsilon(R_i + \delta_{\mathrm{margin}})$ is the lateral position to reach while passing, $\varepsilon \in \{+1, -1\}$ selects the side, and $\sigma_o = L_{\mathrm{blend}}/2$ sets how far along the path the deviation extends; L_{blend} is a design parameter, usually 15–25 m so that the reference heading stays within the linear range and 20 m in Section 7. Figure 9 shows the construction: the baseline $Y_r(s)$ (dashed

blue) becomes $\tilde{Y}_r(s)$ (solid red), which moves around the obstacle (grey disk) outside its safety zone of radius R_i (dashed orange) at the height Y_{tgt}; the blending length $L_{\text{blend}} = 2\sigma_o$ is the distance between the two inflection points, where the reference heading is the largest.

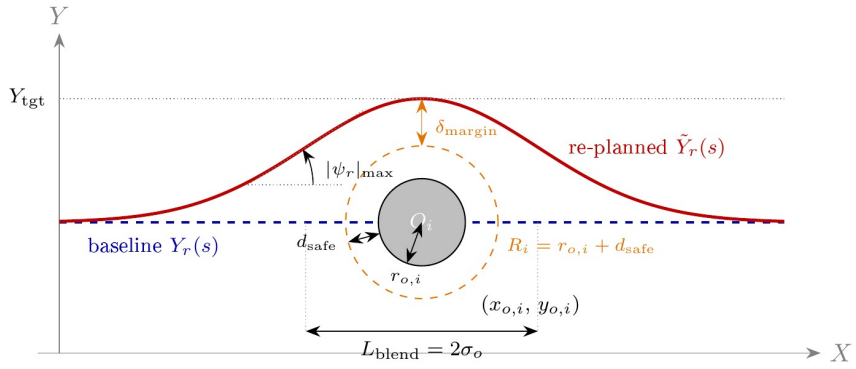


Figure 9. Construction of the re-planned reference around an obstacle.

L_{blend} cannot be chosen freely: a shorter bell gives a sharper manoeuvre, a longer one keeps the heading small and the choice has to respect the range where the linearisation is valid. The maximum reference heading angle produced by the Gaussian bell re-planner follows the closed-form relation

$$|\psi_r|_{\max} = \arctan\left(\frac{2A_Y}{L_{\text{blend}}\sqrt{e}}\right), \tag{16}$$

where A_Y is the target lateral displacement (e.g., $A_Y = 2.5\,\text{m}$ for the obstacle avoidance scenario) and $e \approx 2.718$ is Euler's number. With $L_{\text{blend}} = 10\,\text{m}$, this yields $|\psi_r|_{\max} \approx 16.9°$, which consumes more than half of the accurate tracking envelope ($|\psi| \lesssim 30°$, Section 7.3) and leaves little margin for the tracking error and for the additional heading excursions caused by measurement noise. With $L_{\text{blend}} = 20\,\text{m}$, the formula gives $|\psi_r|_{\max} \approx 8.6°$, which is well within the linear regime, and the closed-loop system becomes robust and stable. This scenario uses $L_{\text{blend}} = 20\,\text{m}$.

Clearance of the Re-Planned Reference

If $|Y_{\text{tgt}} - y_{o,i}| \geq R_i + \delta_{\text{margin}}$, then the re-planned reference (15) satisfies

$$\sqrt{(\tilde{X}_r - x_{o,i})^2 + (\tilde{Y}_r - y_{o,i})^2} \geq R_i + \delta_{\text{margin}} - E(\sigma_o),$$

where $E(\sigma_o) \to 0$ as $\sigma_o \to 0^+$.

This property concerns only the reference trajectory generated by the re-planner. The minimum distance actually achieved by the vehicle is $d_{\min}^{\text{veh}} \geq d_{\min}^{\text{ref}} - \|e_{\text{track}}\|_\infty$. A formal safety guarantee would require an Input-to-State Stability (ISS) analysis providing an explicit bound on $\|e_{\text{track}}\|_\infty$. We report empirical tracking error bounds in Section 7; an analytical ISS bound is left for future work.

While the re-planner ensures that the reference satisfies the safety constraint, the MPC tracking error introduces a small additional perturbation. Empirically (see Section 7), the achieved clearance remains well above d_{safe} provided that the lateral displacement parameter is chosen with a reasonable margin (we use 2.5 m for an obstacle radius of 1.5 m). For several obstacles, (15) can be applied sequentially; when two obstacles are closer than about L_{blend}, their bells overlap and the resulting reference must be checked against both safety zones. Section 7.5 reports a two-obstacle case, and Section 8.3 states the limits of this extension.

7. Simulation Results

This section reports the simulation results obtained with the proposed LTI-MPC + Luenberger framework in five representative scenarios. The simulation environment is implemented in MATLAB and consists of (i) a continuous-time vehicle simulator using the nonlinear kinematic dynamics (1) integrated by a fourth-order Runge–Kutta scheme with step $T_s^{\text{sim}} = 10\,\text{ms}$; (ii) the LTI-MPC, with the QP (8) solved at each control step ($T_s = 100\,\text{ms}$) by `quadprog` (interior-point convex); (iii) the Luenberger observer, run at the simulation rate with the gain L_d pre-computed once via `idare`. Vehicle parameters are reported in Table 5 (the same vehicle as in the preliminary longitudinal PID study) and MPC parameters in Table 1.

Figure 10 shows the closed loop that is simulated. The reference trajectory $x_r(s)$ first proceeds through the local re-planner of Section 6, which returns a collision-free reference $\tilde{x}_r$ around the known static obstacles; the LTI-MPC of Section 4 tracks $\tilde{x}_r$ using the state estimate $\hat{x}$; the vehicle is measured through $y = Cx + \nu$ (noisy DGPS position and odometry speed), and the Luenberger observer of Section 5 rebuilds the unmeasured heading ψ from y and the applied control u, which closes the loop.

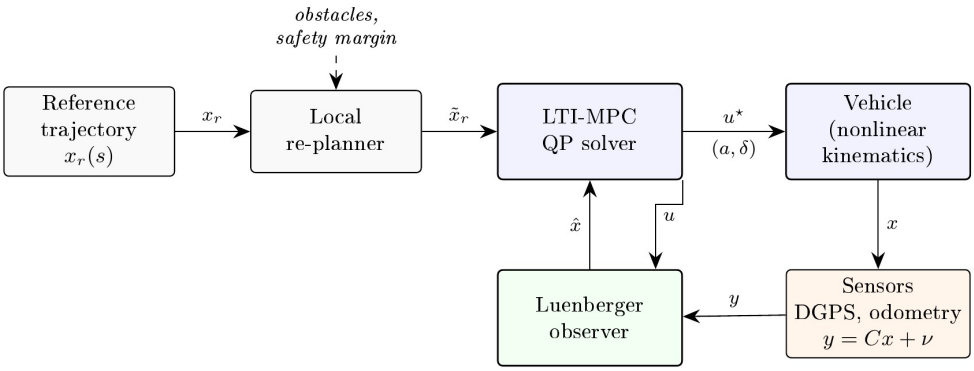


Figure 10. Overall architecture: re-planner, LTI-MPC, vehicle, sensors and Luenberger observer.

All scenarios are run in a closed loop with the Luenberger observer in the feedback path and with additive measurement noise $\sigma_X = \sigma_Y = 0.02\,\text{m}$ (typical of differential GPS-RTK in open-sky conditions) and $\sigma_{v_x} = 0.05\,\text{m/s}$ (typical of wheel encoder odometry). The observer is initialised with $\hat{\tilde{x}}(0) = (0, 0, \psi(0), v_x(0) - v_{\text{nom}})^\top$, except in Section 7.8, where an initial heading error is introduced deliberately. Tracking errors are measured on the true state. Steady-state metrics are computed for $t > 2\,\text{s}$, excluding the last second before the end of the reference path, and the simulation is stopped when the prediction horizon reaches the end of the path, so that no end-of-path transient contaminates the metrics. Where useful, results with perfect state feedback (no noise, no observer) are also reported to separate the control-limited error from the noise-limited error.

Table 5. Vehicle parameters.

Parameter	Symbol	Value
Wheelbase	$L = l_f + l_r$	2.60 m ($l_f = 1.20$, $l_r = 1.40$)
Mass	m	1500 kg
Steering saturation	$\delta_{\max}$	$\pm 35^\circ$
Max. steering rate	$\dot{\delta}_{\max}$	$120^\circ/\text{s}$
Max. acceleration/deceleration	$a_{\max}, a_{\min}$	$+3, -5\,\text{m/s}^2$
Max. jerk	$\dot{a}_{\max}$	$3\,\text{m/s}^3$

7.1. Scenario S1: Straight Line with Lateral Disturbance

The vehicle travels along a straight line at $v_{x,r} = 5\,\mathrm{m/s}$. A lateral velocity disturbance

$$w(t) = 0.4\,h(t)\,\sin\big(2\pi(t-8)/3\big)\ \mathrm{m/s}, \qquad t \in [8, 11]\ \mathrm{s},$$

is added to $\dot{Y}$, where h is a Hann window ensuring smooth onset and offset. Figure 11 presents the trajectory, the lateral and heading errors and the steering command; the orange band marks the disturbance interval. Outside the disturbance interval, the tracking error stays at the noise floor of the observer; during the disturbance, the LTI-MPC limits the lateral deviation to $|e_y|_{\max} = 0.109\,\mathrm{m}$ and returns to within 1 cm of the reference 0.54 s after the disturbance ends. Over the whole run, the RMSE is 0.024 m, and the steering peak of 9.88° stays well away from saturation. Because the MPC looks $NT_s = 2\,\mathrm{s}$ ahead, the correction starts gently and the vehicle returns to the line without overshoot.

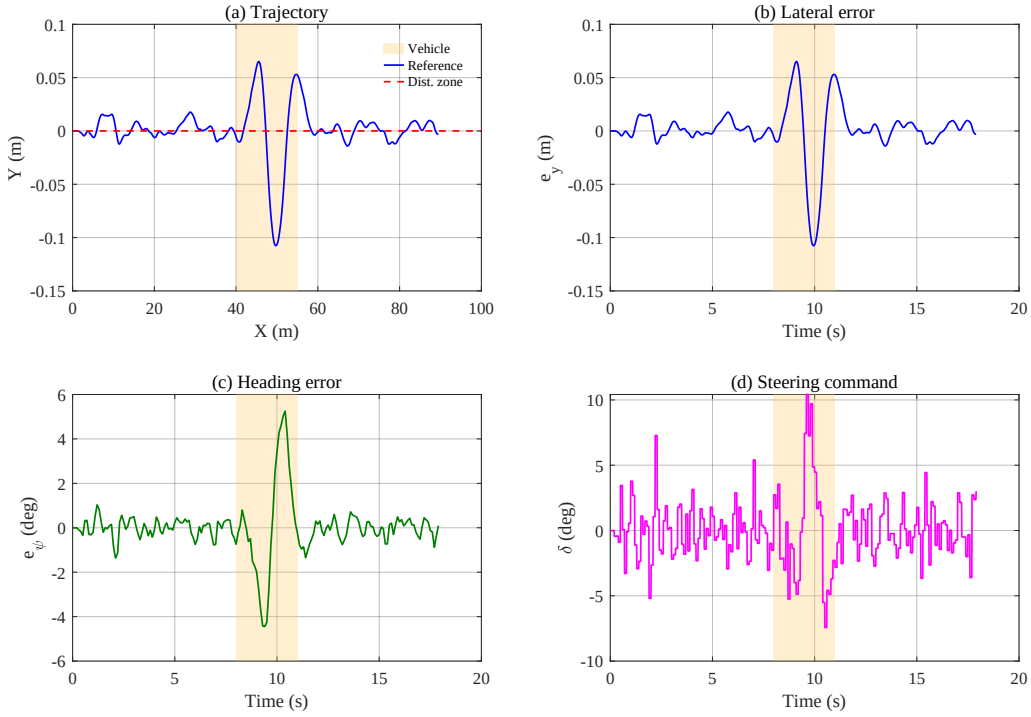


Figure 11. Scenario S1, straight line with a lateral disturbance: (**a**) trajectory, (**b**) lateral error, (**c**) heading error, (**d**) steering command. In (**a**), the blue line is the vehicle trajectory and the red dashed line the reference; the shaded band marks the disturbance interval.

7.2. Scenario S2: Sinusoidal Trajectory at Multiple Speeds

Here, the reference is a sinusoid, $Y_r(X) = 1.5\sin(2\pi X/30)\,\mathrm{m}$, followed over 120 m. Its tightest radius of curvature is $R_{\min} = 1/(1.5(2\pi/30)^2) \approx 15.2\,\mathrm{m}$, so the Polack criterion allows speeds of up to $v_{\max} = \sqrt{0.5\mu g R_{\min}} \approx 8.6\,\mathrm{m/s}$. The maximum reference heading is $\psi_{\max} = \arctan(1.5 \cdot 2\pi/30) \approx 17.4°$, within the LTI envelope. We test $v_x = 2, 4, 6\,\mathrm{m/s}$ with v_{nom} adapted to each speed. Figure 12 reports the trajectories, lateral errors, steering commands and headings; Table 6 reports the steady-state metrics ($t > 2\,\mathrm{s}$) with the observer in the loop under 2 cm/0.05 m/s measurement noise ("noisy") and with exact state feedback ("perfect").

The steady-state RMSE grows with the speed, reflecting the increased difficulty of tracking under tighter time constants. With perfect state feedback, the steering peaks stay at the geometric value $\arctan(L\kappa_{\max}) \approx 9.7°$ and the RMSE is control-limited; with the observer in the loop, the RMSE increases by a few millimetres and the steering peaks rise to 13.7–15.3°, reflecting the heading estimate noise discussed in Section 5. The noise

matters most at 2 m/s, where the observer receives the least kinematic information per second (Table 2). Even so, the worst lateral error, 0.052 m at 6 m/s, is about ten times smaller than the tolerances usually accepted for lane keeping in a town.

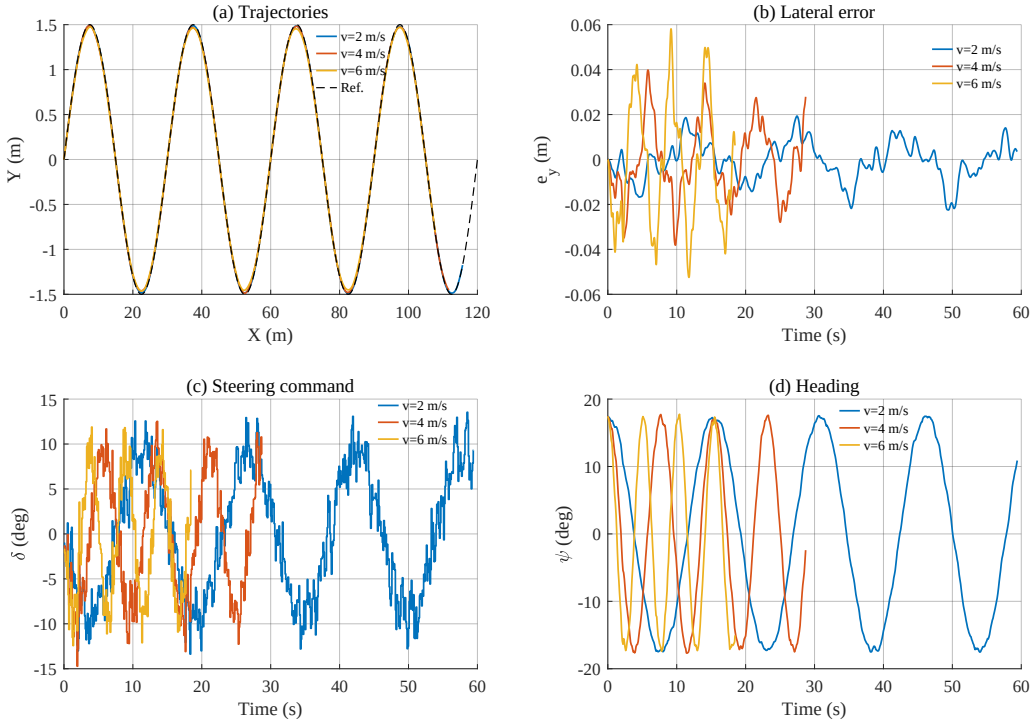


Figure 12. Scenario S2, sinusoidal reference at 2, 4 and 6 m/s: (**a**) trajectories, (**b**) lateral error, (**c**) steering, (**d**) heading.

Table 6. Scenario S2: steady-state metrics with and without measurement noise.

v_x (m/s)	Noisy (Observer in the Loop)			Perfect State Feedback	
	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	$\lvert \delta \rvert_{\max}$ (°)	RMSE (m)	$\lvert \delta \rvert_{\max}$ (°)
2	0.008	0.025	15.3	0.008	9.6
4	0.018	0.040	13.7	0.016	9.5
6	0.026	0.052	12.4	0.027	9.4

7.3. Scenario S3: Constant-Radius Turn Sweep (Envelope of the LTI Approximation)

The path has three parts: a 10 m straight line, a circular arc of radius $R = 50$ m that turns the heading by $\theta \in \{15°, 30°, 45°, 60°\}$ and a 60 m straight exit, all driven at $v_x = 5$ m/s. The lateral acceleration, $a_y = 25/50 = 0.5\,\text{m/s}^2$, is well inside the Polack limit, so any error that appears comes from the fixed linearisation and not from the kinematic model itself. In the arc, the exact steering is $\delta_{\text{geom}} = \arctan(L/R) = 2.98°$. Figure 13 shows the trajectories (references dashed), the speed, the lateral error and the steering (1 s moving average) around δ_{geom}; on the exit straight, the error settles at a value that grows with the exit heading. The metrics are given in Table 7: the exit RMSE is taken over the last 5 s of the valid window, and $|\delta|_{\max}$ is the raw peak of the applied steering, noise included, whereas the figure shows the 1 s average.

The scenario was built to identify where a linearisation around $\psi = 0$ stops working, and two findings emerge. On the exit straight, the vehicle keeps a constant lateral offset that increases with the exit heading: under 5 cm up to 30°, 0.149 m at 45°, 0.400 m at 60°. The predictor uses $\dot{Y} = v\psi$ and $\dot{X} = v$ in place of $v \sin\psi$ and $v\cos\psi$; the error is about 5 % on the sine and 13 % on the cosine at 30°, and it is 17% and 50% at 60°. The speed

also drops during the turn (Figure 13b): the predictor believes that the vehicle advances along X more rapidly than in reality, by the factor $1/\cos\psi$, so the controller slows down to keep the predicted X error small. This is entirely an effect of the model and not of the vehicle. Neither effect leads to instability: the raw steering peaks stay within $\pm 6.9^\circ$ (Table 7), and the smoothed command of Figure 13d holds the geometric value δ_{geom} during the arc, for a duration that grows with the arc angle, before returning to zero on the exit straight. The accuracy simply degrades gracefully with $|\psi|$. Practically, reference trajectories for the proposed LTI-MPC should be designed with $|\psi_r| \lesssim 30^\circ$ (exit error at the centimetre level), which is naturally satisfied by typical urban paths and by the re-planner of Section 6; for high-heading manoeuvres, LTV-MPC with online re-linearisation or nonlinear MPC would be required.

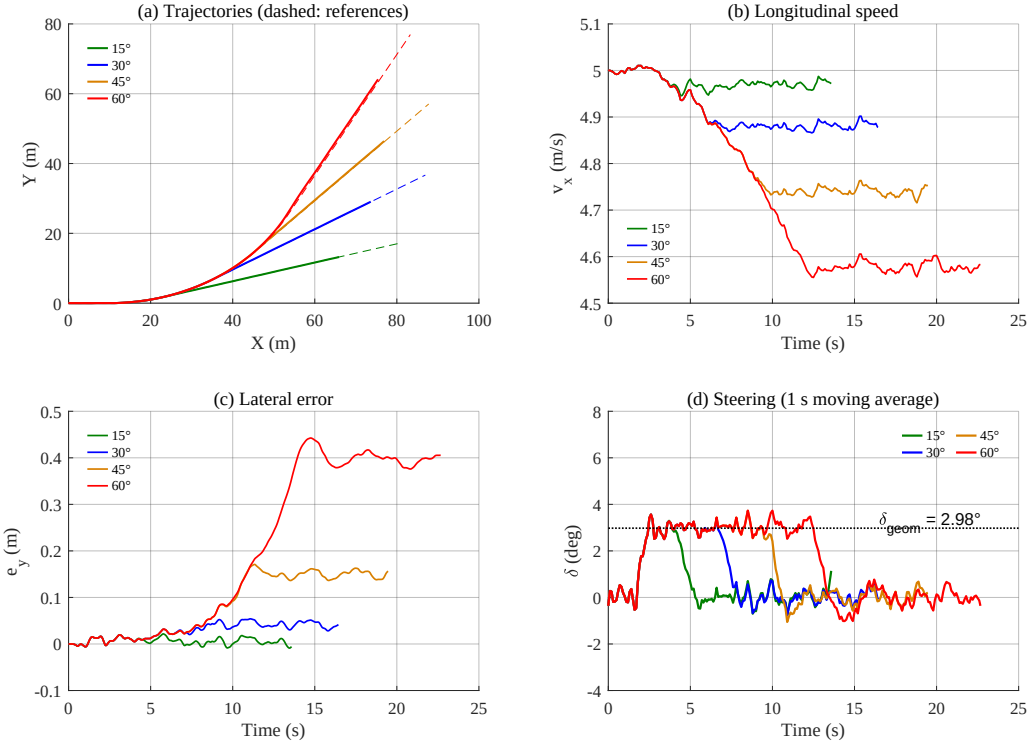


Figure 13. Scenario S3, constant-radius turns from 15° to 60°: (**a**) trajectories, (**b**) speed, (**c**) lateral error, (**d**) steering.

Table 7. Scenario S3: metrics for each arc angle.

Arc Angle (°)	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	Exit-Straight RMSE (m)	$\lvert \delta \rvert_{\max}$ (°)
15	0.011	0.028	0.007	6.8
30	0.036	0.053	0.042	6.8
45	0.112	0.169	0.149	6.8
60	0.283	0.436	0.400	6.9

7.4. Scenario S4: Extended Serpentine with Curvature Inversions

The reference combines two harmonics, $Y_r(X) = 1.0\sin(2\pi X/40) + 0.3\sin(2\pi X/20)$, over 160 m at $v_x = 4$ m/s. It shows multiple curvature inversions and amplitude modulations while keeping the maximum heading within the LTI envelope ($\psi_{\max} \approx 14.1^\circ$). Figure 14 reports the trajectory, lateral error, steering (with the $\pm\delta_{\max}$ limits) and heading. The LTI-MPC achieves a steady-state RMSE of 0.012 m and a peak lateral error $|e_y|_{\max} = 0.028$ m, tracking the highly varying reference with centimetre accuracy throughout. The peak steering effort reaches $|\delta|_{\max} = 10.1^\circ$, well below saturation, with no sustained oscillation.

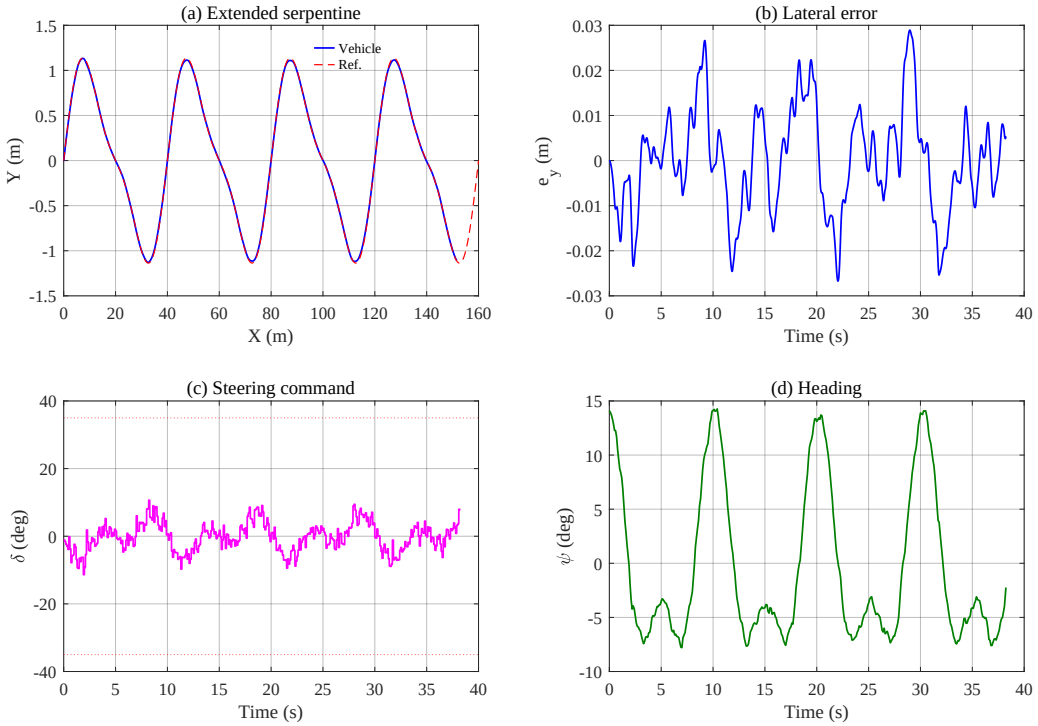


Figure 14. Scenario S4, extended serpentine at 4 m/s: (**a**) trajectory, (**b**) lateral error, (**c**) steering, (**d**) heading.

7.5. Scenario S5: Static Obstacle Avoidance

The vehicle approaches a static obstacle of radius $r_o = 1.5\,\text{m}$ centred at $(x_o, y_o) = (40, 0)\,\text{m}$ while travelling along an initially straight reference. The required clearance to the obstacle surface is $d_{\text{safe}} = 0.5\,\text{m}$, giving an avoidance radius $R_o = r_o + d_{\text{safe}} = 2.0\,\text{m}$. The re-planner generates the avoidance trajectory (15) with $Y_{\text{tgt}} = +2.5\,\text{m}$ (passage on the left, $\delta_{\text{margin}} = 0.5\,\text{m}$) and $L_{\text{blend}} = 20\,\text{m}$, so that $|\psi_r|_{\max} = 8.6°$ by (16). We test $v_x = 3, 5, 7\,\text{m/s}$ with v_{nom} set to each test speed. Figure 15 reports the trajectories, the distance from the vehicle to the obstacle surface versus the safety threshold, the lateral error with respect to the re-planned reference and the steering command (1 s moving average); Table 8 reports the metrics, where $d_{\min}$ is the minimum distance from the rear-axle centre to the obstacle surface and $|\delta|_{\max}$ is the raw peak of the applied command.

The LTI-MPC + re-planner architecture achieves a minimum distance to the obstacle surface of $d_{\min} \in [0.98, 0.99]\,\text{m}$ at all tested speeds, i.e., twice the required clearance $d_{\text{safe}} = 0.5\,\text{m}$. The result is consistent with the bound $d_{\min}^{\text{veh}} \geq d_{\min}^{\text{ref}} - \|e_{\text{track}}\|_\infty$ of Section 6: the re-planned reference passes at $Y_{\text{tgt}} - r_o = 1.0\,\text{m}$ from the surface, and the MPC tracks it with $|e_y|_{\max} \leq 0.030\,\text{m}$, so the achieved clearance is within a few centimetres of the designed one. This scenario carries the main practical contribution of this paper. A re-planner that writes the clearance into the reference, followed by an MPC that tracks the reference with a very small error (RMSE at most 0.013 m at every speed), gives a system that meets the safety requirement by construction at the reference level and, in simulation, at the vehicle level. The steering needed for this is moderate: the raw peak is at most 7.0° (Table 8) and the smoothed command of Figure 15d stays within roughly ±5°.

Table 8. Scenario S5: obstacle avoidance metrics.

v_x (m/s)	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	$\lvert \delta \rvert_{\max}$ (°)	$d_{\min}$ (m)	$d_{\min} \geq d_{\text{safe}}$
3	0.008	0.016	7.0	0.98	✓
5	0.010	0.021	5.9	0.99	✓
7	0.013	0.030	7.0	0.99	✓

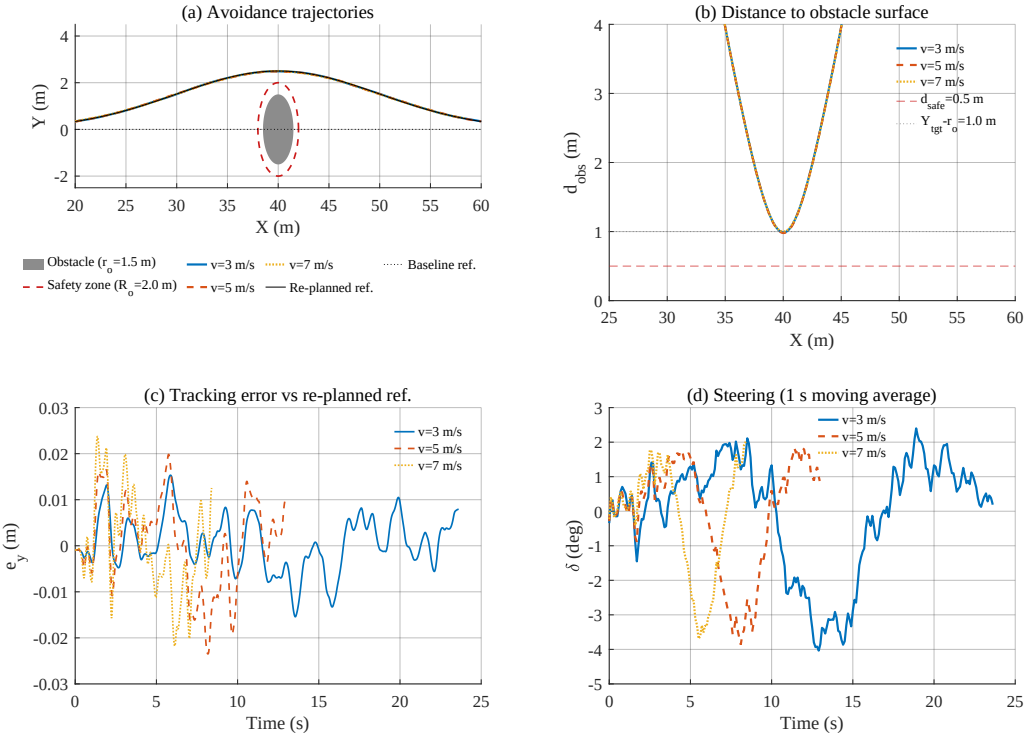


Figure 15. Scenario S5, static obstacle at 3, 5 and 7 m/s: (**a**) trajectories, (**b**) distance to the obstacle, (**c**) lateral error, (**d**) steering.

Two Spaced Obstacles

To test the sequential application of (15), a second obstacle of the same radius is placed at $(x_o, y_o) = (55, 0)$ m, i.e., 15 m behind the first one, which is less than $L_{\text{blend}} = 20$ m, so that the two bells overlap; both are passed on the same side with $Y_{\text{tgt}} = 2.5$ m. Figure 16 shows the resulting reference and the vehicle trajectory at 5 m/s. The two overlapping bells merge into a single smooth excursion whose maximum reference heading is 8.9°, barely above the 8.6° of a single bell. The minimum distance to the obstacle surfaces is 0.99 m for the first obstacle and 1.00 m for the second, the tracking RMSE is 0.009 m and the raw steering peak is 8.0°, so the clearance of the single-obstacle case is preserved. When the obstacles are on opposite sides, or closer than about $L_{\text{blend}}/2$, the superposition can reduce the clearance or produce a heading above $|\psi_r|_{\max}$; the planner then has to be run with the combined reference checked against every safety zone, and, if the check fails, the vehicle should slow down rather than rely on (15) alone. We therefore limit the claim of this paper to isolated or well-separated static obstacles and to pairs passed on the same side.

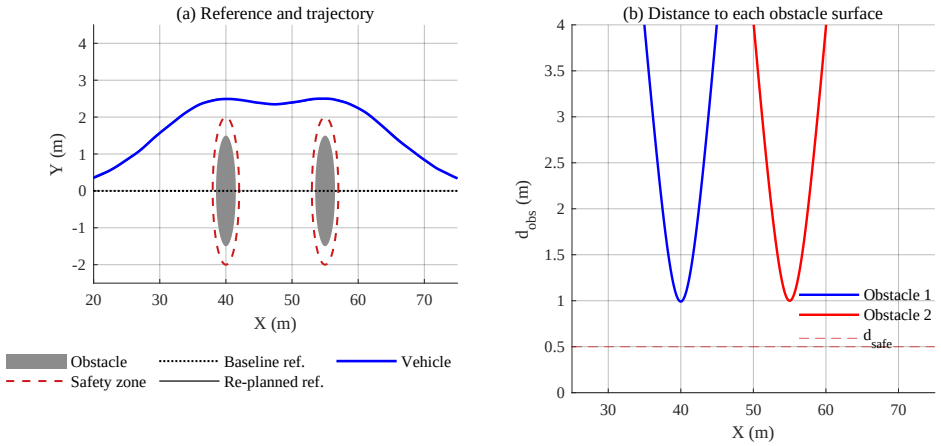


Figure 16. Two obstacles 15 m apart (S5, $v_x = 5\,\mathrm{m/s}$): (**a**) re-planned reference and trajectory, (**b**) clearance from each obstacle. In (**a**), the grey disks are the obstacles, the red dashed circles their safety zones, the dotted line the baseline reference, the thin solid line the re-planned reference and the blue line the vehicle trajectory; in (**b**), the two solid lines are the distances to each obstacle surface and the red dashed line is d_{safe}.

7.6. Comparison with Baseline Controllers (PID-Type and Stanley)

To quantify the benefit of the predictive formulation over classical reactive controllers, we conduct a head-to-head comparison on the sinusoidal benchmark of Section 7.2; the two baselines are the reactive laws that are most often retained in recent comparative and hybrid studies [54]. The PID-type baseline uses a proportional lateral law with heading feedback, $\delta = -(K_y e_y + K_\psi e_\psi)$ (a PD-like structure in the path error sense), together with a PI longitudinal speed loop; the Stanley controller implements the front-axle cross-track law $\delta = -e_\psi - \arctan\left(k\, e_y^{\mathrm{front}} / (k_{\mathrm{soft}} + v_x)\right)$ [9]. All three controllers receive the same observer estimate and the same measurement noise and are subject to the same steering, steering rate, acceleration and jerk limits.

Sensitivity to the Design Speed

To separate the effect of a speed mismatch from the comparison with the baselines, the MPC designed at $v_{\mathrm{nom}} = 4\,\mathrm{m/s}$ was also run, without any change, at 3 and 5 m/s on the sinusoidal reference; Table 9 gives the resulting steady-state RMSE next to the value obtained when v_{nom} matches the test speed. A mismatch of $\pm 1\,\mathrm{m/s}$ raises the RMSE from 0.013 to 0.080 m at 3 m/s and from 0.025 to 0.123 m at 5 m/s, i.e., by a factor of five to six, while the steering peaks are essentially unchanged and the closed loop remains stable; the error is a slow lateral offset, not an oscillation. This is the price of the fixed linearisation: the lateral gain of the prediction model, $v_{\mathrm{nom}} T_s$, no longer matches that of the vehicle, and the controller under- or over-estimates the effect of its own steering. In practice, v_{nom} must therefore follow the cruise speed to well within 1 m/s, which is what the segment-wise selection of Section 4.1 provides and which the comparison below assumes.

Table 9. Sensitivity of the LTI-MPC to a mismatch between the design speed v_{nom} and the actual speed (S2, observer and noise in the loop).

Actual Speed (m/s)	v_{nom} (m/s)	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	$\lvert \delta \rvert_{\max}$ (°)
3	3 (matched)	0.013	0.036	13.3
3	4 (mismatch −1)	0.080	0.124	14.1
5	5 (matched)	0.025	0.050	13.5
5	4 (mismatch +1)	0.123	0.192	13.1

The comparison protocol is designed to be fair and to test the "single tuning per operating range" limitation of reactive controllers, in the same spirit as the scenario ensemble evaluation used for the longitudinal PID in the preliminary study (there, the swept parameters were the grade, load, friction and latency; here, the swept parameter is the cruise speed, which is the one that changes the lateral plant gain v_{nom}^2/L most directly). The baseline gains are tuned by a parametric sweep on S2 at 4 m/s to minimise the steady-state RMSE (retained values $K_y = 4.0$, $K_\psi = 6.0$, $k = 5.0$, $k_{\mathrm{soft}} = 2.0$; PI speed gains 1.0 and 0.2), exactly as the MPC weights were tuned, and all controllers are then run without retuning at $v_x = 2, 4, 6, 8$ m/s. Figure 17 compares the trajectories, lateral errors and steering commands (0.2 s moving average) of the three controllers at 6 m/s and reports the steady-state RMSE across speeds (bars clipped at 0.3 m for readability); Table 10 gives the steady-state metrics ($t > 2$ s) with observer and measurement noise, together with the ratio of each controller's RMSE to the LTI-MPC RMSE at the same speed.

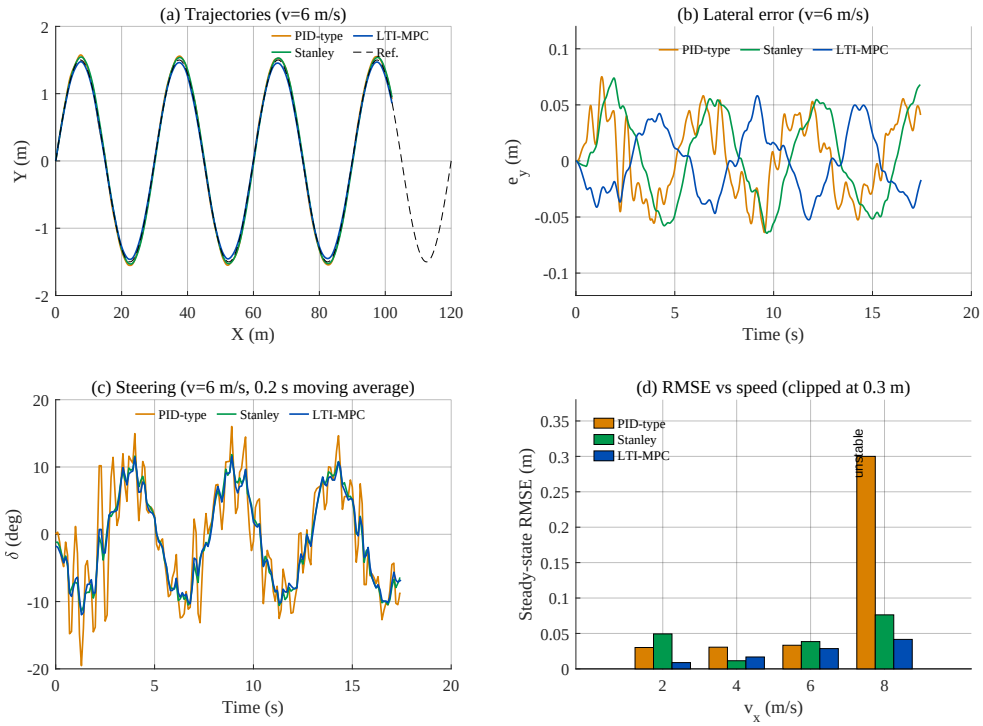


Figure 17. PID-type, Stanley and LTI-MPC on the sinusoidal reference: (**a**) trajectories, (**b**) lateral error, (**c**) steering at 6 m/s; (**d**) RMSE versus speed.

Table 10. PID-type, Stanley and LTI-MPC on scenario S2 across speeds.

v_x (m/s)	Controller	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	$\lvert \delta \rvert_{\max}$ (°)	RMSE Ratio vs. MPC
2	PID-type	0.031	0.049	24.6	3.7×
	Stanley	0.047	0.077	16.7	5.6×
	LTI-MPC	0.008	0.025	15.3	1.0×
4	PID-type	0.031	0.056	20.6	1.7×
	Stanley	0.013	0.028	13.2	0.7×
	LTI-MPC	0.018	0.040	13.7	1.0×
6	PID-type	0.038	0.108	27.0	1.5×
	Stanley	0.043	0.076	12.1	1.7×
	LTI-MPC	0.026	0.052	12.4	1.0×
8	PID-type	unstable	–	35.0	–
	Stanley	0.073	0.130	13.7	1.6×
	LTI-MPC	0.045	0.082	13.6	1.0×

Three observations follow. At the tuning speed of 4 m/s, a well-tuned Stanley controller performs as well as the LTI-MPC on this easy, unconstrained reference (RMSE ratio 0.7×), and the PID-type law is 1.7× worse; with exact state feedback, the three RMSE values are 0.016 m (MPC), 0.010 m (Stanley) and 0.030 m (PID-type). Order-of-magnitude gains over reactive controllers on such a benchmark can only be obtained against poorly tuned baselines, and we make no such claim. Away from 4 m/s, the situation changes. The loop gain of a reactive controller grows with the speed, so both baselines lose accuracy at the two ends of the range (Stanley is 5.6× worse than the MPC at 2 m/s and 1.7× worse at 6 m/s), and, at 8 m/s, the PID-type controller becomes unstable under the actuator rate limits. The LTI-MPC, which carries the speed in its prediction model and enforces the rate limits in the QP, moves from 0.008 m at 2 m/s to 0.045 m at 8 m/s without any discontinuity. Third, the MPC produces the smoothest steering at every speed: the raw peaks of Table 10 are 12.4° for the MPC against 27.0° for the PID-type law at 6 m/s, and the residual high-frequency activity visible on the PID-type trace of Figure 17c is absent from the MPC trace, reflecting the explicit penalisation of the control rate $r_{\Delta\delta}$. The advantage of the predictive formulation therefore lies less in the raw accuracy at a single operating point than in its robustness across the operating range, its explicit constraint handling and, through the re-planner, its ability to reason about obstacles, none of which the reactive laws provide.

7.7. Computational Performance

Table 11 reports the QP solve time per MPC iteration measured over 287 consecutive iterations in the sinusoidal scenario at 4 m/s, and Figure 18 shows its distribution. The times are the wall-clock times of MATLAB's `quadprog` (interior-point convex algorithm, default tolerances) measured with `tic/toc` on a laptop with an Intel Core i7-1165G7 (2.80 GHz) and 16 GB of RAM under Windows; the first call, which includes just-in-time compilation, is excluded. The values in the table are the mean, median, minimum and maximum over the run. In the reported run, the solver took 1.96 ms on average (median 1.91 ms, between 1.64 and 2.83 ms) over 287 iterations; a second, independent run averaged 2.20 ms. In both cases, more than 97 % of the 100 ms period is left unused. An embedded operator-splitting solver such as OSQP [50] would cut the solve time to a fraction of a millisecond, and an event-triggered scheme in the spirit of [47] would cut the number of solves. What makes this cheap is the LTI structure: the linearisation, the discretisation and the stacking of the QP matrices are applied once at start-up, and only the linear term g and the rate bounds change from one step to the next.

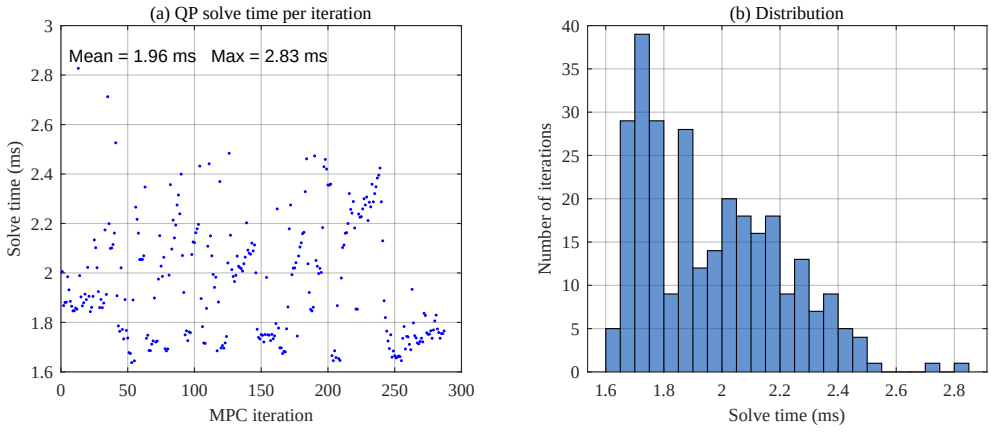


Figure 18. QP solve time per MPC iteration: (**a**) time series, (**b**) histogram.

Table 11. QP solve time per MPC iteration.

Operation	Time (ms)
Linearisation and discretisation (cached, LTI)	~0.01
QP solve—mean	1.96
QP solve—median	1.91
QP solve—max.	2.83
QP solve—min.	1.64

7.8. Observer Convergence and Robustness to Noise

Figure 19 illustrates the convergence of the Luenberger observer on the sinusoidal trajectory at 4 m/s with the measurement noise of Section 7: the true versus estimated heading, the estimation error with the $\pm 1^\circ$ band and the lateral position (true, noisy measurement and estimate). Starting with a heading error of -3.2°, the observer brings it below 1° in 0.24 s, in line with the dominant time constant of Table 4. Once converged ($t > 1$ s), the RMS error of $\hat{\psi}$ settles at 0.47°, with peaks of 1.7° where the curvature changes the fastest.

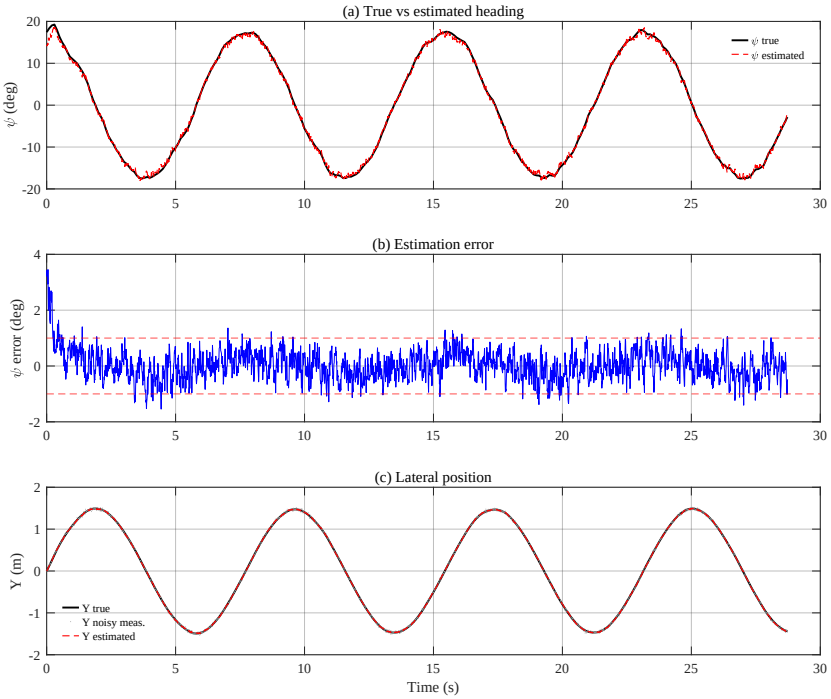


Figure 19. Convergence of the observer at 4 m/s: (**a**) true and estimated heading, (**b**) estimation error, (**c**) lateral position. In (**b**), the blue line is the heading estimation error and the red dashed lines mark the $\pm 1^\circ$ band.

8. Discussion

8.1. Strengths of the Proposed Scheme

Within the range where the linearisation holds, the LTI-MPC tracks structured paths with a lateral RMSE of between 0.008 and 0.026 m at 2–6 m/s, observer and realistic noise included, which is comfortably within the requirements of urban driving. With a constant A and B, the standard stability tools of linear-quadratic control (terminal cost, terminal set) apply to the linearised loop directly; determining such a certificate for the present tuning is left for later work. The safety margin comes from the pair re-planner plus MPC and is verified in simulation: on S5, the tracking error never exceeds 0.030 m, so the achieved clearance is within a few centimetres of the designed one, and relation (16) informs the designer of the required blending length in order for the re-planned path to stay in the

linear range. In a fair comparison, the predictive controller is the only one of the three whose accuracy degrades smoothly from 2 to 8 m/s with a single tuning. The QP solve time leaves a large real-time margin, and the Luenberger observer removes the need for a heading sensor.

8.2. From Fixed-Gain PID to Predictive Control

The results of Section 7.6 close the loop opened by the longitudinal study. There, the best fixed-gain PID that offline optimisation could find still lost its margin as the delay grew. Here, the PID-type lateral law, tuned by the same type of offline search, becomes unstable when the cruise speed moves away from its tuning point, while the LTI-MPC, which has the speed in its prediction model, degrades smoothly. The two experiments show the same aspect from two sides: a fixed gain cannot follow a plant whose gain changes, but a model can. The three elements added in this work resolve the three gaps found in the fixed-gain study: the prediction horizon gives the anticipation that a PID lacks, the observer gives the state that a PID has to perform without and the re-planner gives the safety mechanism that no choice of gains can provide.

8.3. Limitations

Figure 20 places the five scenarios in the (speed, heading) plane and shows where the LTI approximation is accurate, degraded or no longer applicable.

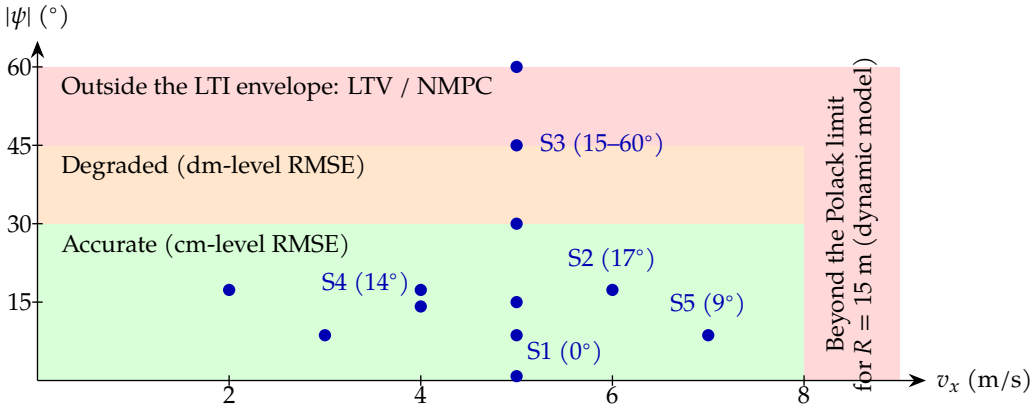


Figure 20. Operating envelope of the LTI-MPC in the (speed, heading) plane, with the test points of the five scenarios. Blue dots: operating points (speed, maximum heading) of scenarios S1 to S5; green, orange and red areas: accurate, degraded and out-of-range regions of the linearisation.

Several limits should be stated. The first concerns the simulation itself. The vehicle is simulated with the same kinematic bicycle equations that the controller is designed from; tyre forces, steering backlash and actuator delay are absent, so the simulation cannot reveal the errors that these effects would introduce on hardware. To give at least an indication, scenario S2 at 4 m/s was repeated with a simulation model that was deliberately different from the design model: the wheelbase increased by 15 %, a first-order steering actuator with a 100 ms time constant was included and GPS noise was raised to 10 cm RMS with a slowly varying bias, representative of a degraded urban canyon fix. Table 12 gives the result. A 15 % wheelbase error and a 100 ms steering lag are absorbed almost without effect (RMSE 0.015–0.019 m against 0.016 m for the nominal run of this test, which differs from the 0.018 m of Table 6 only by the noise realisation; the steering peaks are slightly larger), because the MPC corrects the resulting model error at every step. The degraded GPS is a different matter: 10 cm noise with a slowly varying bias of about 15 cm raises the RMSE to 0.205 m and drives the steering to its 35° limit, although the loop does not diverge. A slowly drifting

position bias is indistinguishable, for the observer, from a real lateral offset, so it enters directly into the tracking error, while the heading estimate amplifies the measurement noise into steering activity. For urban use, this is the limitation that matters the most, and it is the main reason to move to an EKF with an IMU (Section 2.2). None of these tests substitutes experiments on a vehicle, which remain the real validation and the next step.

Other limits follow from the modelling choices. The kinematic bicycle model holds only up to $a_y \leq 4.9\,\mathrm{m/s^2}$ [17]; faster driving or tighter curves call for a dynamic model with tyre forces. The re-planner knows only static obstacles and a point vehicle: moving obstacles would need a motion model, and a real footprint requires a larger R_i. The observer converges only locally and its heading estimate is noisy enough to appear in the steering (Table 3); an EKF with an IMU would help in this regard. The Gaussian bell is a convenient heuristic, and a more principled planner could produce smoother manoeuvres. Three further caveats should be stated plainly. The validation is numerical, and the vehicle is simulated with the same kinematic model used to design the controller, which makes the test easier than it would be on hardware; a high-fidelity simulator or a scaled vehicle is needed before any claim about deployment. The safety margin is observed, not proven; a proof would require an ISS bound on the tracking error. Moreover, the behaviour under DGPS noise that is far above the design level, as in urban canyons, has only been mentioned briefly and would likely require retuning R_{KF} or an EKF.

Table 12. Robustness on S2 at 4 m/s when the simulated vehicle differs from the design model (observer and noise in the loop).

Simulation Model	RMSE (m)	$\lvert e_y \rvert_{\max}$ (m)	$\lvert \delta \rvert_{\max}$ (°)
Nominal (this run)	0.016	0.033	13.2
Wheelbase +15 %	0.015	0.031	14.7
Steering actuator lag 100 ms	0.019	0.042	14.3
GPS noise 10 cm RMS + bias	0.205	0.496	35.0
All three combined	0.211	0.492	35.0

8.4. Comparison with Hard Constraint Formulations

Another option is to place the obstacle directly as a hard constraint in the MPC QP. This idea is appealing but brings three practical problems: staying outside a disk is a non-convex constraint, and its linearisation introduces errors that can leave the QP without a feasible solution; the MPC has to decide on which side to pass, which calls for ad hoc rules or for a planner above it; and such formulations react strongly to the solver settings. Separating planning from tracking overcomes all three: the geometry is handled by the planner and the dynamics by the MPC, and the QP remains an ordinary convex tracking problem with box constraints on the inputs.

8.5. Practical Deployment

Before applying the scheme to a vehicle, several additions would be sensible: an EKF that fuses GPS, IMU and odometry; a dynamic bicycle model with Pacejka tyres for higher speeds; a probabilistic obstacle description with chance-constrained MPC [33]; tests on a scaled vehicle; and a three-level architecture in which a mission planner selects waypoints, a local planner shapes obstacle-aware references and the LTI-MPC tracks them.

8.6. Sensitivity to Design Parameters

The horizon N sets the balance between accuracy and computation: with $N \leq 10$, the controller sees too little of the road ahead; with $N \geq 30$, it gains almost nothing in accuracy while the QP grows (its size is linear in N and its solve time is more than linear); and

$N = 20$ lies between these cases. The lateral weight $q_Y = 100$ needed no adjustment across scenarios. The rate weights decide the smoothness: if too small, the steering chatters; if too large, the response lags. Here, $r_{\Delta\delta} = 30$ gives smooth commands that still react. On the observer side, $Q_{\mathrm{KF},\psi}$ fixes the trade-off of Table 3. On the planner side, (16) gives the blending length: for a 2.5 m sideways move, $L_{\mathrm{blend}} \geq 20$ m keeps $|\psi_r|_{\max}$ under 9°.

9. Conclusions and Future Work

We have described a linear time-invariant MPC, coupled with a Luenberger observer, for following a path and avoiding obstacles with a vehicle represented by the kinematic bicycle model. It continues a preliminary study of fixed-gain longitudinal PID control, whose limited robustness had pointed towards model-based anticipation. Three needs of low-speed urban driving are met: seeing ahead through a prediction horizon, respecting the physical limits of the actuators directly in the optimisation and keeping a minimum distance from obstacles by separating planning from tracking.

The main contributions are as follows: an LTI-MPC for the four-state kinematic model with inputs (a, δ), linearised once around $(0, 0, 0, v_{\mathrm{nom}})$ and written in deviation coordinates with the exact drift $d = (v_{\mathrm{nom}} T_s, 0, 0, 0)^\top$; a Luenberger observer designed with the DARE, which rebuilds the heading from the DGPS-RTK position and odometry with 0.47° RMS accuracy and converges in 0.24 s, with a measured convergence/noise trade-off used to choose its tuning; a local re-planner with a closed-form rule linking the blending length to the maximum reference heading, which builds a clearance of 1.0 m into the reference, with the vehicle reaching 0.98–0.99 m in simulation at 3–7 m/s against a 0.5 m requirement, where the difference is the small tracking error; a fair comparison with PID-type and Stanley controllers (tuned once at 4 m/s, tested from 2 to 8 m/s) showing that the predictive controller is the only one whose accuracy degrades smoothly over the range, with RMSE ratios of 3.7×/5.6× at 2 m/s and 1.5×/1.7× at 6 m/s, and an unstable PID-type controller at 8 m/s; and a simulation campaign on five scenarios, all with the observer and realistic noise in the loop, giving a tracking RMSE of 0.008–0.026 m at 2–6 m/s inside the LTI validity range. The sweep of constant-radius turns in scenario S3 shows where the fixed linearisation stops: the exit straight error stays at the centimetre level for up to 30° of heading change and reaches 0.400 m at 60°, so LTV-MPC or NMPC would be needed for large-heading manoeuvres.

The QP solve time of 1.96 ms leaves a real-time margin above 98.0% at the 100 ms sampling period on a laptop-class CPU. This margin suggests that the scheme is a candidate for embedded automotive hardware in low-speed uses such as automatic parking, shuttles, delivery robots and automated guided vehicles, but the claim has to be confirmed by timing the QP on an embedded platform, which has not been performed here.

Future work includes (i) moving obstacles, with spatio-temporal planning and chance-constrained MPC; (ii) higher speeds, with a dynamic vehicle model and Pacejka tyre forces; (iii) hardware-in-the-loop tests on a scaled vehicle; (iv) an ISS analysis of the closed-loop tracking error, to convert the empirical clearance margin into a proven safety guarantee; (v) coupling with perception, to handle uncertain obstacle detections with robust or distributionally robust MPC; and (vi) closing the gap left by the longitudinal study, namely powertrain lag and perception-to-actuation delay, by adding a delay model to the MPC predictor and by allowing the observer to estimate the grade and load disturbances that a fixed-gain law can only react to.

Author Contributions: Conceptualisation, M.E.M. and T.J.; methodology, M.E.M.; software, M.E.M.; validation, M.E.M., Y.E.K. and S.G.; formal analysis, M.E.M.; investigation, M.E.M.; visualisation, M.E.M.; writing—original draft preparation, M.E.M.; writing—review and editing, T.J., Y.E.K., H.M. and S.G.; supervision, T.J. and H.M.; project administration, H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The MATLAB simulation code that reproduces every figure and table in this article is available from the corresponding author upon request.

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

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
NMPC	Nonlinear Model Predictive Control
QP	Quadratic Program
DARE	Discrete-Time Algebraic Riccati Equation
EKF	Extended Kalman Filter
ISS	Input-to-State Stability
DGPS-RTK	Differential GPS with Real-Time Kinematic correction
PID	Proportional–Integral–Derivative
RMSE	Root Mean Square Error
ECU	Electronic Control Unit

References

1. *SAE Standard J3016*; Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE International: Warrendale, PA, USA, 2021.
2. Schwarting, W.; Alonso-Mora, J.; Rus, D. Planning and decision-making for autonomous vehicles. *Annu. Rev. Control Robot. Auton. Syst.* **2018**, *1*, 187–210. [CrossRef]
3. Amer, N.H.; Zamzuri, H.; Hudha, K.; Kadir, Z.A. Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges. *J. Intell. Robot. Syst.* **2017**, *86*, 225–254. [CrossRef]
4. 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]
5. Stephant, J.; Charara, A.; Meizel, D. Virtual sensor: Application to vehicle sideslip angle and transversal forces. *IEEE Trans. Ind. Electron.* **2004**, *51*, 278–287. [CrossRef]
6. Marino, R.; Scalzi, S.; Netto, M. Nested PID steering control for lane keeping in autonomous vehicles. *Control Eng. Pract.* **2011**, *19*, 1459–1467. [CrossRef]
7. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Technical Report CMU-RI-TR-92-01; Carnegie Mellon University, The Robotics Institute: Pittsburgh, PA, USA, 1992.
8. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. Stanley: The robot that won the DARPA Grand Challenge. *J. Field Robot.* **2006**, *23*, 661–692. [CrossRef]
9. Hoffmann, G.M.; Tomlin, C.J.; Montemerlo, M.; Thrun, S. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 2296–2301.
10. Samson, C. Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots. *IEEE Trans. Autom. Control* **1995**, *40*, 64–77. [CrossRef]
11. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [CrossRef]
12. Borrelli, F.; Bemporad, A.; Morari, M. *Predictive Control for Linear and Hybrid Systems*; Cambridge University Press: Cambridge, UK, 2017.
13. Chen, T.; Chen, X.; Chen, X.; Xu, B. Linear time-varying MPC-based autonomous emergency steering control for collision avoidance. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9360–9371. [CrossRef]
14. Tang, L.; Yan, F.; Zou, B.; Wang, K.; Lv, C. An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles. *IEEE Access* **2020**, *8*, 51400–51413. [CrossRef]
15. Camacho, E.F.; Bordons, C. *Model Predictive Control*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2013.

16. Rawlings, J.B.; Mayne, D.Q.; Diehl, M. *Model Predictive Control: Theory, Computation, and Design*, 2nd ed.; Nob Hill Publishing: El Cerrito, CA, USA, 2017.
17. Polack, P.; Altché, F.; D'Andréa-Novel, B.; de La Fortelle, A. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 11–14 June 2017; pp. 812–818.
18. Kebbati, Y.; Ait-Oufroukh, N.; Vigneron, V.; Ichalal, D.; Gruyer, D. Optimized self-adaptive PID speed control for autonomous vehicles. In Proceedings of the 2021 26th International Conference on Automation and Computing (ICAC), Portsmouth, UK, 2–4 September 2021; pp. 1–6.
19. Elias, G.H.S.; Al-Moadhen, A.; Kamil, H. Optimizing the PID controller to control the longitudinal motion of autonomous vehicles. *AIP Conf. Proc.* **2023**, *2591*, 040045. [CrossRef]
20. Simorgh, A.; Marashian, A.; Razminia, A. Adaptive PID control design for longitudinal velocity control of autonomous vehicles. In Proceedings of the 2019 6th International Conference on Control, Instrumentation and Automation (ICCIA), Sanandaj, Iran, 30–31 October 2019; pp. 1–6.
21. Lee, J.; Jo, K. Model predictive control with powertrain delay consideration for longitudinal speed tracking of autonomous electric vehicles. *World Electr. Veh. J.* **2024**, *15*, 433. [CrossRef]
22. Liao, J.; Li, H.; Yang, Q.; Zhou, Y. PID-based hierarchical event-triggered MPC for USV trajectory tracking. In Proceedings of the IECON 2025—51st Annual Conference of the IEEE Industrial Electronics Society, Madrid, Spain, 14–17 October 2025.
23. Domina, Á.; Tihanyi, V. Linear MPC approach for automated vehicle path following at the limit of handling. *Sensors* **2022**, *22*, 5807. [CrossRef]
24. Tian, Y.; Yao, Q.; Hang, P.; Wang, S. A gain-scheduled robust controller for autonomous vehicles path tracking based on LPV system with MPC and H_∞. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9350–9362. [CrossRef]
25. Kumar, N.; Pachamuthu, R. Real-time LPV-based non-linear model predictive control for robust trajectory tracking in autonomous vehicles. *arXiv* **2025**, arXiv:2506.04684.
26. Nezami, M.; Karachalios, D.S.; Schildbach, G.; Abbas, H.S. Obstacle avoidance of autonomous vehicles: An LPVMPC with scheduling trust region. *arXiv* **2024**, arXiv:2405.02030.
27. Chen, Y.; Sun, Y.; Li, J.; He, J.; He, C. Tube-based robust model predictive control for autonomous vehicle with complex road scenarios. *Appl. Sci.* **2025**, *15*, 6471. [CrossRef]
28. Wang, X.; Dou, G.; Chen, T.; Lu, J. Improved robust model predictive trajectory tracking control for intelligent vehicles based on multi-cell hyperbody vertex modeling and double-layer optimization. *Sensors* **2025**, *25*, 6537. [CrossRef]
29. Tan, W.; Wang, M.; Ma, K. Research on intelligent vehicle trajectory tracking control based on improved adaptive MPC. *Sensors* **2024**, *24*, 2316. [CrossRef]
30. Kalaria, D.; Lin, Q.; Dolan, J.M. Delay-aware robust control for safe autonomous driving and racing. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 7140–7150. [CrossRef]
31. Huang, Z.; Li, H.; Li, W.; Liu, J.; Huang, C.; Yang, Z.; Fang, W. A new trajectory tracking algorithm for autonomous vehicles based on model predictive control. *Sensors* **2021**, *21*, 7165. [CrossRef]
32. Xiong, L.; Liu, M.; Xie, Z.; Leng, B.; Zhang, Y. Learning-augmented MPC for autonomous vehicle path tracking via ensemble residual dynamics learning. *Sensors* **2026**, *26*, 340. [CrossRef]
33. Fang, S.; Li, X.; Wu, C.; Yu, K. DRCC-LPVMPC: Robust data-driven control for autonomous driving and obstacle avoidance. *arXiv* **2026**, arXiv:2603.14408.
34. Chu, D.; Li, H.; Zhao, C.; Zhou, T. Trajectory tracking of autonomous vehicle based on model predictive control with PID feedback. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2239–2250. [CrossRef]
35. Lai, F.; Huang, C.; Sun, J. Intelligent vehicle path planning and tracking control using improved A* and MPC algorithms. *Proc. Inst. Mech. Eng. D J. Automob. Eng.* **2025**, *240*, 7323–7338. [CrossRef]
36. Luenberger, D.G. An introduction to observers. *IEEE Trans. Autom. Control* **1971**, *16*, 596–602. [CrossRef]
37. Brivadis, L.; Andrieu, V.; Serres, U. Luenberger observers for discrete-time nonlinear systems. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 3435–3440. [CrossRef]
38. Best, M.C.; Gordon, T.J.; Dixon, P.J. An extended adaptive Kalman filter for real-time state estimation of vehicle handling dynamics. *Veh. Syst. Dyn.* **2000**, *34*, 57–75. [CrossRef]
39. Ryu, J.; Gerdes, J.C. Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control. *J. Dyn. Syst. Meas. Control* **2004**, *126*, 243–254. [CrossRef]
40. Jiao, H.; Xu, X.; Chen, S.; Guo, N.; Yu, Z. Improving vehicle heading angle accuracy based on dual-antenna GNSS/INS/barometer integration using adaptive Kalman filter. *Sensors* **2024**, *24*, 1034. [CrossRef]
41. Gao, Y.; Gray, A.; Tseng, H.E.; Borrelli, F. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Veh. Syst. Dyn.* **2014**, *52*, 802–823. [CrossRef]

42. Liniger, A.; Domahidi, A.; Morari, M. Optimization-based autonomous racing of 1:43 scale RC cars. *Optim. Control Appl. Methods* **2015**, *36*, 628–647. [CrossRef]
43. Zeng, J.; Zhang, B.; Sreenath, K. Safety-critical model predictive control with discrete-time control barrier function. In Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021; pp. 3882–3889. [CrossRef]
44. Dong, D.; Ye, H.; Luo, W.; Wen, J.; Huang, D. Collision avoidance path planning and tracking control for autonomous vehicles based on model predictive control. *Sensors* **2024**, *24*, 5211. [CrossRef]
45. Gratzer, A.L.; Broger, M.M.; Schirrer, A.; Jakubek, S. Two-layer MPC architecture for efficient mixed-integer-informed obstacle avoidance in real-time. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 13767–13784. [CrossRef]
46. Lou, M.; Yang, X.; Hu, J.; Shen, H.; Xu, B.; Xiang, Z.; Zhang, B. Design and field test of collision avoidance method with prediction for USVs: A deep deterministic policy gradient approach. *IEEE Internet Things J.* **2025**, *12*, 3363–3372. [CrossRef]
47. Hu, J.; Yang, X.; Lou, M.; Ye, H.; Shen, H.; Xiang, Z. Design, simulation, and field testing of an intelligent control algorithm based on event-triggered and nonlinear MPC for USVs. *IEEE Internet Things J.* **2025**, *12*, 22155–22167. [CrossRef]
48. Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 1094–1099.
49. Li, W.; Zhang, X.; Wang, Y.; Xie, S. Comparison of linear and nonlinear model predictive control in path following of underactuated unmanned surface vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 575. [CrossRef]
50. Stellato, B.; Banjac, G.; Goulart, P.; Bemporad, A.; Boyd, S. OSQP: An operator splitting solver for quadratic programs. *Math. Program. Comput.* **2020**, *12*, 637–672. [CrossRef]
51. Houska, B.; Ferreau, H.J.; Diehl, M. ACADO toolkit—An open-source framework for automatic control and dynamic optimization. *Optim. Control Appl. Methods* **2011**, *32*, 298–312. [CrossRef]
52. MathWorks. *Optimization Toolbox User's Guide (R2022b)*; The MathWorks, Inc.: Natick, MA, USA, 2022.
53. MathWorks. *Model Predictive Control Toolbox User's Guide (R2022b)*; The MathWorks, Inc.: Natick, MA, USA, 2022.
54. Al-Jumaili, M.H.; Özok, Y.E. New control model for autonomous vehicles using integration of model predictive and Stanley based controllers. *Sci. Rep.* **2024**, *14*, 19872. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.