Timed Elastic Band Based Trajectory Planning and MPC Based Adaptive Cruise Control for Car-Like Autonomous Vehicle[1]

Alex Kádár [2]

Abstract:

This paper presents a hierarchical control architecture for a car-like autonomous vehicle. Lateral motion is handled by a Timed Elastic Band (TEB) local planner that repeatedly optimizes a short-horizon waypoint band, while steering is generated by a Pure Pursuit controller tracking the optimized local path. Longitudinal behaviour is governed by a model predictive control (MPC) based adaptive cruise control (ACC) that regulates speed and inter-vehicle distance in free-driving and car-following modes, explicitly respecting acceleration and braking limits and enforcing a minimum safety distance. Safety is addressed through constraint-aware planning and control: the TEB optimization penalizes proximity to obstacles and violations of lane boundaries using smooth penalty terms that support stable gradient-based updates, while the MPC-ACC embeds hard bounds and distance constraints to prevent unsafe following. The integrated stack is evaluated in representative driving scenarios, including lane keeping and car-following, and is benchmarked against the MathWorks Path Following Control System (PFCS) reference model to quantify tracking behaviour and constraint compliance under identical conditions.

Keywords:

Autonomous Vehicle; Timed Elastic Band; Steering Control; Model Predictive Control; Adaptive Cruise Control.

[1] DOI: 10.12700/jceeas.2026.6.3.523

[2] PhD-Student, Budapest University of Technology and Economics, Budapest, Hungary; ORCID: 0009-0007-1767-692X; alex.kadar@edu.bme.hu.

1. Introduction

Autonomous road vehicles must generate collision-free motion while remaining stable, comfortable, and compliant with physical limits. This requirement becomes harder in everyday traffic because the environment is only partially known, other actors move unpredictably, and the feasible maneuver set is restricted by nonholonomic kinematics and actuator constraints. Practical systems therefore rely on repeated local re-planning and constraint-aware control that can react to new measurements without sacrificing dynamic feasibility.

A common and effective structure is a hierarchical separation of lateral and longitudinal tasks. Lateral motion is treated as a geometric path-following problem under steering constraints, while longitudinal motion is regulated to meet speed targets and safe headway constraints in free-driving and car-following modes. This separation reduces complexity, supports modular verification, and allows each loop to be tuned for the dominant safety risks it controls.

This paper presents a MATLAB/Simulink implementation of such a hierarchy in which a Timed Elastic Band (TEB) local planner optimizes a short-horizon band of waypoints, a Pure Pursuit controller tracks the resulting local reference, and a model predictive control based adaptive cruise control (MPC-ACC) regulates speed and distance. The safety rationale is explicit in both layers. The local planner is formulated as a constrained optimization that discourages trajectories that approach obstacles or violate lane boundaries and promotes smooth, dynamically feasible motion, reducing abrupt curvature and control spikes that can trigger loss-of-control events. The longitudinal MPC embeds hard bounds on acceleration and braking and enforces a minimum following distance, which directly targets the dominant rear-end collision mechanism in car-following scenarios. The combined effect is a system that prioritizes clearance, constraint compliance, and stable closed-loop behavior during continuous re-planning.

Performance is evaluated in representative lane-keeping and car-following scenarios and benchmarked against the MathWorks Path Following Control System (PFCS) reference model under comparable conditions to quantify tracking, constraint adherence, and safety-relevant behavior. Beyond the immediate technical contribution, the same safety-oriented local planning and predictive distance control paradigm can support advanced driver assistance and automation in regions where traffic is heterogeneous and infrastructure and enforcement are uneven, including many African road networks, with potential gains in crash reduction and transport efficiency when deployed within appropriate operational limits.

2. Literature Review on Steering and Acceleration Control

The vehicle dynamics literature commonly describes lateral motion using the bicycle model, where the vehicle is represented by a single front and a single rear wheel, and the key state variables are the lateral displacement, yaw angle, and their derivatives [1],

[2]. ISO 8855 defines the vehicle-fixed coordinate system widely used in automotive engineering, in which the x-axis denotes the longitudinal direction, the y-axis the lateral direction, and the z-axis the vertical direction. This convention forms the basis for defining path-tracking errors and for selecting the state variables used in bicycle-model formulations [1].

A large share of classical lane-keeping algorithms is geometric in nature. The Pure Pursuit method computes the required steering angle from the curvature of the circular arc defined by the current vehicle position and a look-ahead point, and it is widely used for path tracking of autonomous ground vehicles [3]. The Stanley controller combines lateral position error and heading error [4]. These methods offer simple implementation and robust operation, but they handle vehicle dynamic limits explicitly only to a limited extent.

Model-based linear controllers such as LQR, LQI, and LQG build on linearized forms of the bicycle model and compute the steering command by minimizing a quadratic cost function [2]. A key advantage of LQR and LQG-type solutions is that stability properties are well analyzable and adequate robustness can be achieved against parameter variations [8], [9]. More recent work extends the LQR structure using, for example, feedforward terms and iterative variants with soft constraints to improve tracking performance and passenger comfort [8], [9].

Nonlinear and MPC-based lateral controllers emerged to address higher-speed maneuvers and operation near tire friction limits. Nonlinear MPC schemes consider the full nonlinear bicycle model and also account for speed constraints, optimizing tracking behavior over a prediction horizon [10]. Such approaches are particularly effective for lane-change and obstacle-avoidance maneuvers, where lateral stability and accurate reference tracking are simultaneously critical [10].

Several approaches have been developed for local path planning. The Elastic Bands method treats the discrete path produced by a global planner as an energy-minimizing elastic band that continuously deforms according to nearby obstacles and the vehicle's kinematic constraints [5]. Timed Elastic Band (TEB) extends this concept by introducing an explicit time dimension and optimizing a trajectory between time-stamped path points while considering execution time, obstacle clearance, and kinematic and dynamic constraints [6]. In ROS-based environments, TEB has been extended specifically to differential-drive and car-like robots, making it suitable as a real-time local planner that can support the generation of steering commands [6].

Artificial potential field methods are also common for obstacle avoidance and local planning. Khatib's classical approach models the interaction between the robot and obstacles using virtual attractive and repulsive forces, whose resultant defines the desired motion direction [7]. Potential field methods enable reactive real-time behavior, but they suffer from local minima. MPC-based trajectory optimization provides a local planning framework that incorporates the vehicle dynamics model, path geometry, and obstacles directly into the optimization problem, and it can be used to generate a reference steering angle and or path curvature [10].

2.1 Longitudinal Acceleration Control and ACC Solutions

A classical starting point for longitudinal motion control is cruise control, which aims to maintain vehicle speed at a set reference, typically via PID control [2]. Adaptive Cruise Control (ACC) extends this concept by maintaining a safe following distance to the lead vehicle, often using a time-gap policy where the desired spacing is proportional to vehicle speed [2], [11].

Modern ACC implementations increasingly rely on model predictive control. MPC-based ACC formulations model the relative kinematics between the ego and lead vehicles in state-space form, where the state variables typically include spacing and relative velocity and may also include acceleration, while the control input is the ego vehicle acceleration [11], [12]. The objective function commonly combines spacing and speed tracking errors with penalties on acceleration and jerk, balancing comfort and safety requirements [12]. The MPC framework naturally supports explicit constraints on acceleration, speed, and minimum following distance, making it well-suited for practical implementation in embedded controllers [11], [13].

Many studies address switching between ACC operating modes (free driving, following, emergency braking) and adaptive selection of the reference speed and time gap according to traffic conditions. Alongside MPC-based approaches, alternative nonlinear control schemes are also used. Sliding-mode controllers provide strong robustness against model uncertainties and external disturbances, making them suitable for spacing-control tasks in both hybrid and conventional vehicles [14]. Fuzzy-logic ACC systems aim to capture driver experience in a rule base and achieve comfortable and safe motion by distributed regulation of speed and spacing [15]. Such systems are particularly popular on small experimental platforms and model vehicles due to their straightforward real-time implementation [15].

Neural and neuro-fuzzy approaches aim to handle the nonlinear nature of vehicle dynamics and traffic situations. For example, Takagi-Sugeno fuzzy-model-based ACC approximates longitudinal dynamics as a weighted combination of linear subsystems, enabling more effective handling of parameter variations and different speed ranges [16]. These methods often appear in combination with MPC or sliding-mode frameworks, enabling simultaneous treatment of hard constraints and robustness considerations.

2.2 Integrated Lateral and Longitudinal Control Architectures

Integrated control solutions aim to avoid treating lateral and longitudinal dynamics as independent subsystems and instead coordinate them within a single optimization problem. MPC-based integrated formulations define a predictive control task whose states include lateral tracking errors and longitudinal variables, while control inputs include steering angle and acceleration and, in some cases, braking force [17], [18]. This allows the controller to exploit the full maneuvering capability of the vehicle, for

example during emergency avoidance maneuvers where coordinated braking and steering is required [18].

Several works focus specifically on solving path tracking and speed control jointly within an MPC framework. Jeong and Yim present an integrated path-tracking and velocity-control algorithm that applies MPC to a vehicle with independent four-wheel steering and drive, improving tracking accuracy and stability through joint optimization of steering angle and drive torque [17]. Other studies target joint optimization of the future path and speed profile, for example via coupled lateral-longitudinal trajectory-tracking MPC schemes that treat vehicle motion as a single optimization problem [19].

In parallel, the literature contains a strong line of hierarchical structures, where a higher-level module plans a global or local trajectory and lower-level lateral and longitudinal controllers track it [18]. The advantages of the hierarchical approach include transparency, modularity, and the possibility of developing and replacing controllers with partial independence. Integrated MPC solutions, however, often deliver better performance near operating limits, where the coupling between steering and acceleration can no longer be neglected [17], [19].

The architecture applied in this paper belongs to the hierarchical category. On the lateral side, it uses a TEB-based local planner and a geometric Pure Pursuit steering controller, while on the longitudinal side, an MPC-based ACC computes the acceleration command. This structure can be positioned between established hierarchical and integrated approaches: the separation between planning and control is retained, while the ACC controller already applies model predictive principles to handle longitudinal constraints and comfort criteria, thereby partially approaching integrated MPC-based solutions.

3. Methodology and Simulation Environment

The study was conducted in MATLAB and Simulink, relying on the Automated Driving Toolbox and the Driving Scenario Designer. The Driving Scenario Designer was used to define the road environment, lane geometry, and the motion of surrounding vehicles. Within each scenario, lane markings, road curvature geometry, and the ego vehicle's reference trajectory can be specified parametrically, and the simulation model uses these signals as inputs to the TEB-based local trajectory planner.

The Simulink model follows a hierarchical architecture. At the top level, the Driving Scenario Designer generates position and velocity data that are provided to the ego-vehicle and surrounding-vehicle blocks. Building on these signals, the framework integrates the TEB trajectory planner, a Pure Pursuit based lateral controller, and an MPC-based ACC longitudinal controller. The vehicle-kinematics subsystem encapsulates both the kinematic bicycle model and the longitudinal motion model, while the feedback states are updated at every sampling instant.

The investigated scenarios can be categorized into three main groups:

- lane keeping on a curved road, where the ego vehicle autonomously tracks the lane centerline,
- straight-line car-following, where the ego vehicle regulates its speed and following distance relative to a preceding vehicle,
- a complex multi-vehicle case featuring multiple lanes, lane changes, and overtaking maneuvers, where the ego vehicle must adapt to varying traffic situations.

Identical baseline vehicle parameters and the same sampling time were used in all cases, enabling a direct comparison of the architecture's behavior across different situations.

3.1 Vehicle Model

A kinematic bicycle model is used to describe lateral motion. The vehicle is represented by a single front and a single rear wheel in a coordinate system defined along the vehicle's longitudinal axis. The state vector comprises the global position (x, y) and the vehicle orientation (θ), augmented by the longitudinal speed (v). The model inputs are the steering angle (δ) and the longitudinal acceleration (a). Together with the geometric parameters, most importantly the wheelbase (b), the kinematic bicycle model ensures that the trajectory generated by the TEB planner remains physically feasible for the ego vehicle.

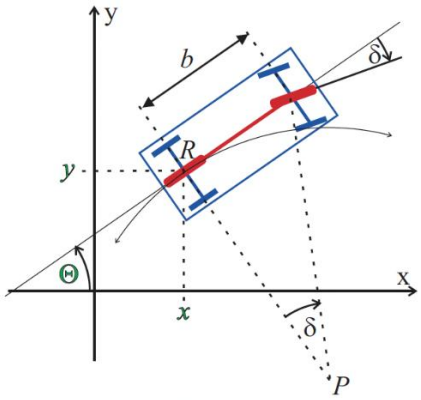


Figure 1: Car-like vehicle [20].

Kinematic model:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \cdot \cos\theta \\ v \cdot \sin\theta \\ v \cdot \dfrac{\tan\delta}{b} \\ a \end{bmatrix} \tag{1}$$

Constraints that are relevant due to vehicle geometry and the control task are enforced within the model as well. The steering angle is limited in magnitude by a maximum value consistent with the steering geometry of a car-like vehicle, set to 40°. The longitudinal acceleration is bounded by lower and upper limits representing braking and acceleration capabilities; these bounds are also chosen to remain realistic with respect

to passenger comfort and tire-road adhesion. In this work, the acceleration limits are set to -8 and 5 m/s^2.

3.2. TEB-Based Lateral Trajectory Planning

Timed Elastic Band (TEB) [21] is a modern, dynamic local planning algorithm widely used in robotics and autonomous vehicles for motion planning and navigation. Its primary objective is to enable the vehicle to progress safely and efficiently along a planned route in dynamic environments, including moving obstacles and rapidly changing conditions.

TEB represents the motion to be executed along the roadway or during a lane-change maneuver by a discrete sequence of trajectory points with an associated temporal parameterization. A fixed number of geometric path points (n) is initialized along a reference curve. This initial "band" is then iteratively deformed by an optimizer while keeping the start state fixed. Figure 2 illustrates the resulting TEB band.

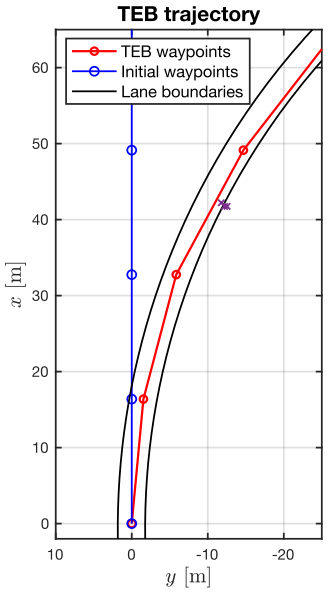


Figure 2: Formulation of the TEB points.

The band is defined by two coupled structures. First, a set of trajectory points

$$Q = \{P_i\}_{i=1\ldots n} \tag{2}$$

describes the ego vehicle position via

$$P_i = (x_i, y_i). \tag{3}$$

Second, a set of time intervals associated with consecutive points

$$\tau = \{\Delta T_i\}_{i=1\ldots n-1} \tag{4}$$

is introduced. Together, these two sets capture both the geometric shape of the path and its temporal parameterization; therefore, the TEB can be interpreted as the pair

$$B := (Q, \tau). \tag{5}$$

The planner operates in a receding-horizon manner: in each control cycle, the current ego-vehicle state becomes the new start point of the band, and the planning horizon is shifted forward along the lane.

The optimal TEB band is obtained by minimizing a cost function. The objective formulation allows the inclusion of parameters that constrain vehicle motion, such as obstacle avoidance and lane keeping. The overall cost is expressed as the weighted sum of all objective terms:

$$f(B) = \sum_k w_k \ \Gamma_k(B) \tag{6}$$

where $\Gamma_k$denotes the objective term encoding a specific motion constraint and w_k is its weight relative to the others. The optimal trajectory B^* is then obtained as

$$B^* = var \min_B f(B) \tag{7}$$

3.2.1. Objective Functions and Constraint Terms

The optimal trajectory is collision-free while exhibiting the smallest possible lateral accelerations; that is, it corresponds to the smoothest feasible collision-avoiding path available to the vehicle. Collision avoidance can be expressed through constraint terms that exclude any configuration in which the vehicle would collide with an obstacle or cross lane boundaries. For gradient-based optimization methods, a smooth, continuously differentiable penalty function is particularly advantageous because it provides well-behaved derivatives with respect to the trajectory states.

A generalized limit function can be formulated, for example, in the following form:

$$f_{limit}(z, z_m, \epsilon, S, n) = \begin{cases} 0 & for\ z \le (z_m - \epsilon) \\ \left(\frac{z-(z_m-\epsilon)}{S}\right)^n & for\ z > (z_m - \epsilon) \end{cases} \tag{8}$$

where z is a scalar "distance-like" quantity (for instance, the perpendicular distance to an obstacle or the lateral deviation from the lane centerline), z_m is a critical threshold, ϵ is a small safety margin, and S and n are parameters that control the scale and steepness of the penalty.

Using the generalized limit function above, the obstacle-avoidance objective term is defined as

$$\Gamma_{o,\text{limit}} = f_{\text{limit}}(-d_0, -d_{\text{limit}}, \epsilon_o, S_o, n_o) \tag{9}$$

where d_0 denotes the perpendicular distance between two TEB trajectory points and an obstacle, and d_{limit} specifies the minimum allowable clearance from the obstacle.

Lane keeping can be enforced in an analogous manner using limit functions. Let y be the lateral deviation of a TEB waypoint from the lane centerline and let y_{min} and y_{max} denote the lateral coordinates of the right and left lane boundaries, respectively. The corresponding lane-related objective terms are:

$$\Gamma_{Ll,limit} = f_{limit}(y, y_{max}, \epsilon_L, S_L, n_L) \tag{10}$$

$$\Gamma_{Lr,limit} = f_{limit}(-y, -y_{min}, \epsilon_L, S_L, n_L) \tag{11}$$

Together, $\Gamma_{o,\text{limit}}$, $\Gamma_{Ll,\text{limit}}$, and $\Gamma_{Lr,\text{limit}}$ provide a compact representation of obstacle-clearance and lane-boundary requirements within the TEB optimization. The aggregated constraint objective can therefore be written as:

$$f(B) = w_o \cdot \Gamma_{o,limit}(B) + w_L \cdot [(\Gamma_{Ll,limit}(B) + \Gamma_{Lr,limit}(B)] \quad (12)$$

where w_o and w_L are weighting factors.

The hyperparameters must be selected such that the vehicle follows the lane centerline with minimal tracking error while maintaining collision-free motion, and such that excessive steering intervention is avoided for passenger comfort. Based on testing, the following parameter values were found to provide satisfactory performance:

$$\epsilon_o = 0.154, \quad S_o = 0.96, \quad n_o = 2.8, \quad w_o = 1.5$$
$$\epsilon_L = 0.088, \quad S_L = 1.20, \quad n_L = 1.2, \quad w_L = 5$$

3.3. Pure Pursuit Based Angle Generation

In the implemented TEB algorithm, the local path at each time instant is represented by a fixed number of five waypoints expressed in the ego-vehicle coordinate frame. The first point always coincides with the current vehicle position, which corresponds to [0,0] in the ego frame, while the remaining four points are placed ahead along the vehicle's longitudinal axis. Since the TEB is regenerated at every sampling step around the current pose, the vehicle effectively needs to align with the next point in the sequence, that is, the second TEB waypoint. Therefore, the reference heading is determined from the direction between the first and the second points. The corresponding direction angle can be computed using the $atan2$ function:

$$\theta = atan2(y_2 - y_1,\ x_2 - x_1) \quad (13)$$

where $(x_1, y_1) = (0,0)$ is the current ego position, and (x_2, y_2) are the coordinates of the second TEB point. This provides a reference heading angle θ for the vehicle. However, tracking this heading angle alone is not sufficient. The steering command must be computed such that, taking the kinematic bicycle model geometry and the wheelbase into account, the vehicle reaches the desired waypoint within a specified look-ahead distance. In practice, this task is handled by the Pure Pursuit algorithm, which is a geometric path-tracking method.

The core idea of Pure Pursuit [3] is that the vehicle tracks a predefined target point (look-ahead point) located on the path at a fixed distance ahead. From the current position, the vehicle attempts to reach this target as if it were traveling along a circular arc. Accordingly, the steering angle is chosen such that the vehicle would follow a circle whose arc intersects the target point. In the vehicle coordinate frame, the target point is given by (x, y), and the look-ahead distance is $L_d = \sqrt{x^2 + y^2}$. Using the angular deviation θ relative to the vehicle's heading, the desired curvature can be computed as:

$$\kappa = \frac{2 \cdot sin(\theta)}{L_d} \quad (14)$$

The steering angle δ is then obtained as:

$$\delta = arctan(b \cdot \kappa) \quad (15)$$

where b denotes the wheelbase. The computed steering command, optionally scaled by a gain factor, is subsequently saturated according to the allowable steering limits. Figure 3 illustrates the geometric construction underlying the Pure Pursuit method.

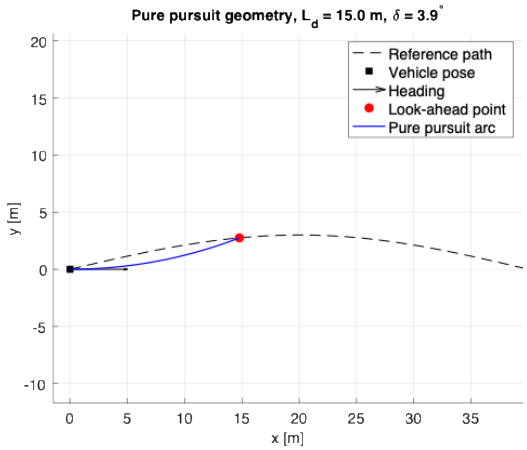


Figure 3: Pure Pursuit geometry.

3.4 MPC-Based Adaptive Cruise Control

Model Predictive Control (MPC) [12] is an advanced control method that computes the control input in a predictive manner using a dynamic model, such that future outputs track reference values while respecting constraints. The core principle of MPC is that, at every sampling instant, it solves a finite-horizon optimization problem based on the system model. Over a predefined prediction horizon, an optimal sequence of control inputs is computed by minimizing a cost function subject to operational constraints. Within this receding-horizon strategy, only the first control input of the optimized sequence is applied to the system; at the next sampling instant, the entire procedure is repeated using the updated current state. As a result, MPC can anticipate the system's behavior and respond to predicted disturbances or reference changes before they occur, unlike conventional PID controllers.

The MPC-based adaptive speed and distance control algorithm aims to regulate both the ego vehicle speed and its headway to a preceding vehicle:

Following (ACC) mode: If a preceding vehicle (lead vehicle) is present, the MPC controls the ego-vehicle acceleration to maintain a safe following distance. The desired spacing is defined using a time-gap policy with headway time T_h and a minimum standstill distance d_0. The reference distance is $d_{\text{ref}} = d_0 + T_h v_{\text{ego}}$, where v_{ego} denotes the current speed of the ego vehicle.

Free (Cruise) mode: If there is no slower vehicle ahead of the ego vehicle, the controller maintains a driver-defined target speed (v_{set}). In this mode, the MPC is responsible for reaching and sustaining the desired speed while respecting the vehicle's dynamic constraints.

In the implementation, the MPC optimization problem is formulated and solved at each time step using YALMIP [22]. YALMIP is a MATLAB-based modeling environment that enables a declarative specification of the decision variables, objective function, and constraints of an optimization problem, and then automatically converts the formulation into a solver-compatible numerical form. The outcome of the optimization

is the acceleration command (a_{cmd}) to be applied immediately, which determines the throttle and braking actuation of the vehicle.

Accurate MPC computation requires several components to be defined precisely. First, a discrete-time state-space model is needed, built around a state vector that captures the following distance and the vehicle speed. This is paired with a quadratic cost function that penalizes following-distance error, speed deviation, and control effort. In addition, constraints must be specified to represent the physical acceleration limits of the vehicle and the minimum permissible following distance. Finally, selecting the prediction horizon N and the sampling time T_s completes the control problem specification. The following subsections detail these elements: the state-space model formulation, the structure of the cost function and the constraints, and how these components are integrated into the YALMIP-based MPC formulation.

3.4.1 Dynamic Model and State Variable Definition

For the car-following mode, three states are defined:

1. gap: the distance between the ego vehicle and the preceding (lead) vehicle, that is, the longitudinal spacing. This is computed from the difference between the vehicle positions while accounting for vehicle geometry, corresponding to the actual distance between the front of the ego vehicle and the rear of the lead vehicle.
2. v: the longitudinal speed of the ego vehicle.
3. z: an integrator state used to compensate for offset (steady-state error). This allows the system to maintain the target following distance or speed reference accurately over time by eliminating residual error.

The longitudinal motion of the ego vehicle and the spacing dynamics are approximated by an affine discrete-time state-space model with sampling time Δt. In following (ACC) mode, the model can be written in the form:

$$x_{k+1} = Ax_k + Ba_k + c \tag{16}$$

where the state vector is $x := [gap \quad v \quad z]^T$, the control input a_k is the ego-vehicle longitudinal acceleration (positive for acceleration and negative for braking), and c is an exogenous term capturing the influence of the lead vehicle speed. The corresponding matrices are:

$$A = \begin{pmatrix} 1 & -\Delta t & 0 \\ 0 & 1 & 0 \\ \Delta t & -T_h\,\Delta t & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ \Delta t \\ 0 \end{pmatrix}, \qquad c = \begin{pmatrix} \Delta t\, v_{\text{lead}} \\ 0 \\ 0 \end{pmatrix} \tag{17}$$

Component-wise interpretation:

Spacing dynamics (gap): The first row of the state update, together with the term c, updates the spacing state:

$$gap_{k+1} = gap_k + \Delta t(v_{\text{lead}} - v_k) \tag{18}$$

Here, v_{lead} is the speed of the lead vehicle, and v_k is the current ego speed. The equation expresses that the gap increases when the lead vehicle moves faster than the ego vehicle and decreases when the ego vehicle moves faster. Accordingly, the $-\Delta t$ coefficient in

the first-row models the effect of ego speed, while the $\Delta t v_{\text{lead}}$ term in c adds the contribution of the lead vehicle speed.

Speed dynamics (v): The second row of the model captures the ego-vehicle speed evolution:

$$v_{k+1} = v_k + \Delta t\, a_k \tag{19}$$

where a_k is the acceleration commanded by the controller. This indicates that the new speed equals the previous speed plus the change induced by acceleration integrated over the sampling interval.

Integrator dynamics (z): The third row updates the integrator state as:

$$z_{k+1} = z_k + \Delta t\,(gap_k - T_h\, v_k) \tag{20}$$

This shows that z integrates over time the difference between the actual spacing and the speed-proportional spacing term ($T_h\, v_k$). The purpose is offset-free tracking: z is effectively the integral of the dynamic spacing error. For example, if the ego vehicle is too close (the actual gap is smaller than $T_h v$), the integrator evolves accordingly, encouraging the controller to reduce acceleration or apply braking. It is important to note that the constant standstill distance term d_0 is not embedded directly in the integrator dynamics. Instead, d_0 appears as part of the reference distance. This is necessary because directly integrating a constant (nonzero) error contribution would introduce an undesirable drift in the integrator state. As formulated, z integrates only the speed-dependent component of the spacing error, ensuring that no persistent error remains in following distance under dynamic conditions.

In summary, the matrices A, B, c define the predictive model used to forecast the evolution of spacing and speed over the next N steps for a given sequence of acceleration inputs. Model parameters include the time headway T_h (default value 3 s) and the discrete step Δt (100 ms), both of which directly affect the state-transition matrix A.

Free (No-Lead) Mode Model: When no lead vehicle is present, the spacing state and its dynamics are no longer meaningful, because the ego vehicle can accelerate freely up to the set speed. In this case, the model reduces to a two-state system:

$$x_f = [v \quad z_v]^T \tag{21}$$

that is, only the ego speed and an integrator state for the speed error are retained. The discrete-time state-space model for free-driving mode is:

$$A_f = \begin{pmatrix} 1 & 0 \\ \Delta t & 1 \end{pmatrix}, \qquad B_f = \begin{pmatrix} \Delta t \\ 0 \end{pmatrix}, \qquad c_f = \begin{pmatrix} 0 \\ -\Delta t\, v_{ref} \end{pmatrix} \tag{22}$$

where c_f injects the reference speed into the integrator dynamics. Since v_{ref} is a constant desired value, its discrete integral appears as $-\Delta t\, v_{ref}$). In component form, the free-mode update equations are:

$$v_{k+1} = v_k + \Delta t\, a_k \tag{23}$$

$$z_{v,k+1} = z_{v,k} + \Delta t\,(v_k - v_{ref}) \tag{24}$$

Here v_{ref} is the desired speed (the driver-defined v_{set}). The integrator state z_v accumulates the speed error (current speed minus reference), ensuring that the cruise-

control behavior maintains the set speed accurately in steady state, without residual speed error.

3.4.2 MPC Cost Function

At each time step, the MPC controller solves an optimization problem that minimizes a predefined cost function. This cost function numerically encodes the desired system objectives and penalizes undesired behavior. In the present case, the cost consists of multiple terms, each weighting a specific aspect: following-distance error, speed error, the integrator state, the magnitude of the control input, and the variability of the control input.

The stage cost summed over the horizon steps $k = 1 \dots N$ includes the following components:

Following-distance error (weight: Q_{gap}):

$$Q_{gap}\left(gap_k - (d_0 + T_h\, v_k)\right)^2 \tag{25}$$

This term penalizes deviations of the instantaneous following distance gap_k from the reference distance $d_0 + T_h v_k$. A larger Q_{gap} forces the optimizer to prioritize maintaining the desired spacing.

Speed error (weight: Q_v):

$$Q_v\left(v_k - v_{\text{ref},k}\right)^2 \tag{26}$$

This term measures the deviation of vehicle speed from a reference speed v_{ref}. The interpretation of the reference depends on whether a lead vehicle is present:

Following mode: $v_{ref,k} = v_{lead}$, the speed of the lead vehicle. In this case, the cost penalizes relative speed, that is, situations where the ego vehicle is moving too fast or too slow compared to the lead vehicle. In practice, the reference speed is also bounded by the set cruise speed: if $v_{lead} > v_{set}$, the vehicle does not attempt to match the lead vehicle speed but instead tracks the user-defined set speed.

Free mode: $v_{ref} = v_{set}$, the predefined desired speed. The cost then represents speed-tracking error, and the controller minimizes it to drive v_k toward v_{set}.

Integrator state penalty (weight: Q_I):

$$Q_I\, z_k^2 \tag{27}$$

In free mode, the corresponding term is $Q_I\, z_{v,k}^2$. This term penalizes the squared integrator state. Here, z_k is the integrated (accumulated) dynamic spacing error, while $z_{v,k}$ is the integral of the speed error. The role of this penalty is to encourage the integrator state to converge toward zero, thereby removing steady-state error. In this work, the same Q_I weight is applied to the integrator state in both following and free modes, ensuring comparable steady-state error suppression across operating conditions.

Acceleration command magnitude (weight: R):

$$R\, a_k^2 \tag{28}$$

This regularization discourages unnecessarily large acceleration commands. Physically, it leads to more comfortable actuation by avoiding overly aggressive throttle or braking

when it is not required. If R is too small, the MPC may issue very large accelerations for short intervals to reduce other errors. If R is too large, the controller can become overly conservative and respond insufficiently to spacing or speed errors.

Jerk penalty, i.e., change in acceleration (weight: S_{jerk}):

$$S_{\text{jerk}}\,(a_k - a_{k-1})^2 \tag{29}$$

applied for each $k > 1$. This term penalizes variability in the acceleration command and promotes smooth actuation. By penalizing jerk, the MPC avoids abrupt switching between acceleration and braking, which improves both ride comfort and closed-loop stability. If S_{jerk} is large, the controller tends to select a more gradual acceleration profile, even if that implies slower correction of other errors.

Terminal integrator penalty (weight: Q_{IN}):

$$Q_{IN}\,z_N^2 \tag{30}$$

In free mode, the corresponding term is $Q_{IN}\,z_{v,N}^2$. This penalizes the integrator state at the final horizon step. Its purpose is to encourage the MPC to plan toward zero integrated error by the end of the horizon. The terminal weight enforces the zero steady-state error requirement without forcing an excessively large integrator penalty at earlier steps. The same terminal weight Q_{IN} is used in both following and free modes, so the end-of-horizon zero-error requirement is enforced with comparable strength in both cases.

The full-horizon MPC cost function can therefore be written as:

$$J_{MPC} = \sum_{k=1}^{N} \Big[Q_{\text{gap}} \left(gap_k - (d_0 + T_h v_k) \right)^2 + Q_v \left(v_k - v_{ref,k} \right)^2 + \\ + Q_I\, z_k^2 + R\, a_k^2 + S_{\text{jerk}}\,(a_k - a_{k-1})^2 \Big] + Q_{IN}\, z_N^2 \tag{31}$$

The effect of the MPC weighting factors was analyzed through a targeted parameter study. Starting from a baseline configuration, the weights Q_{gap}, Q_v, R, and S_{jerk} were varied systematically. The integrator-related weights were kept constant in all cases, specifically $Q_I = 0.1$and $Q_{IN} = 10$.

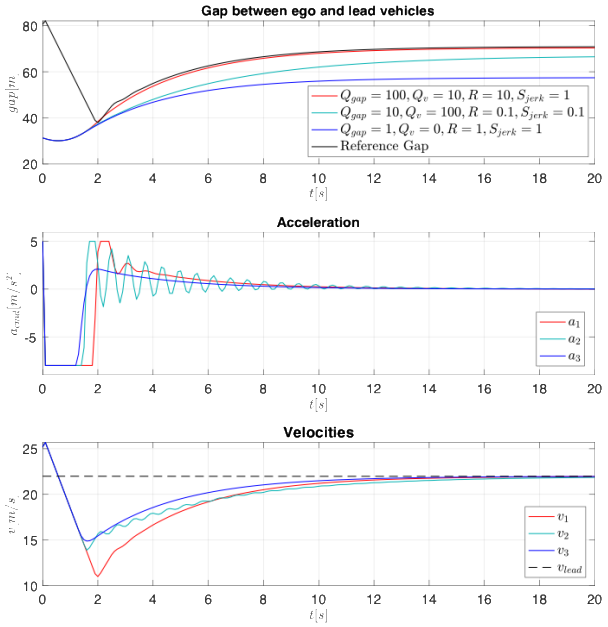


Figure 4: MPC testing under different parameter settings.

Figure 4 illustrates MPC behavior under different parameter settings. The figure was generated in a simulation in which the ego vehicle had to follow a slower vehicle on a straight road at a safe distance, with overtaking prohibited. In the upper plot, the ideal following distance as a function of ego speed is shown by the black curve. The simulation with tuned parameters corresponds to the red curves, where the vehicle reaches the desired following distance after approximately 15 seconds. The dark-blue case represents a configuration with very small weights, where the controller becomes overly sluggish and does not optimize sufficiently to achieve the required following distance; in contrast, it still tracks the lead vehicle speed (black dashed line). The green-shaded case shows a configuration in which jerk is hardly penalized, leading to oscillations in both acceleration and vehicle speed.

For the final ACC settings used in this paper, the weight vector corresponding to the red curves in Figure 4 was selected:

$$Q_{\text{gap}} = 100, \quad Q_v = 10, \quad R = 10, \quad S_{\text{jerk}} = 1, \quad Q_I = 0{,}1, \quad Q_{IN} = 10.$$

This combination provided a practical trade-off: the following distance approaches the reference curve quickly without overshoot, the speed profile matches the lead vehicle well, and the acceleration signal does not exhibit persistent large-amplitude oscillations, making the maneuver acceptable from a passenger-comfort perspective.

3.4.3 Constraints and Safety Conditions in the MPC

In real-world operation, vehicle dynamics and safety requirements impose constraints on the controller that must be respected within the MPC optimization. The following linear constraints are included.

Acceleration bounds: For each time step, lower and upper limits are imposed on the acceleration command:

$$a_{min} \leq a_k \leq a_{max}, \quad \forall k = 0 \dots N-1 \tag{32}$$

The default values are $a_{min} = -8\ {}^{m}/_{s^2}$ and $a_{max} = 5\ {}^{m}/_{s^2}$. The key point is that the controller must remain within this range and cannot request stronger braking or acceleration than permitted by these bounds. This constraint models the system's physical limitations (maximum engine force, maximum braking force, tire-road adhesion, etc.). Also, it supports safety and comfort, since excessive deceleration or acceleration can be unsafe or uncomfortable for passengers.

Minimum following distance (collision avoidance): For every k, the spacing is constrained such that

$$\text{gap}_k \geq d_0 - \delta \tag{33}$$

where d_0 is the minimum safety distance (default value 5 m), and δ is an optional slack allowance. For example, if $\delta = 0.5$ the condition becomes $\text{gap}_k \geq 4.5$ m. By default, $\delta = 0$ meaning a hard constraint is enforced. This constraint ensures that the MPC cannot plan a trajectory that would lead to a collision; that is, the ego vehicle cannot approach the lead vehicle closer than the minimum safety distance.

At each sampling instant, the controller checks whether a valid lead vehicle exists ahead of the ego vehicle. If an object is detected in front of the ego vehicle within the lane and within the relevant range, the MPC switches to following mode. If no such lead vehicle exists (no detection, or the previously tracked lead vehicle leaves the defined range), the controller switches to free mode.

Based on the model and cost function above, the MPC optimization problem can be formulated as a quadratic program: the objective is quadratic, while the dynamics and constraints are linear. Using the YALMIP framework, the required control input can be computed at each time step.

4. Simulation Results

The objective of this chapter is to experimentally validate the behavior of the integrated system in controlled, reproducible simulations. The simulation environment is introduced, alongside the investigated case types: pure lane keeping, multi-lane overtaking, and longitudinal car-following under varying traffic conditions.

In the simulations, the proposed TEB + Pure Pursuit + MPC-ACC architecture and the Path Following Control System (PFCS) reference model were executed using identical road geometry and the same lead-vehicle trajectory. This setup enables a direct comparison of lateral path-following performance and longitudinal following behavior. The simulations were run on an Apple MacBook Pro M3 Pro laptop, and the runtime was also evaluated. Here, runtime refers to the total wall-clock time of a simulation run, including scenario initialization and execution of the control algorithms, but excluding model compilation time.

4.1 Lane Keeping

The purpose of this experiment is to evaluate lane-keeping performance on a curved road segment. The track is a closed loop with varying curvature. The vehicle operates in its own coordinate frame, and all sensor and planner-module signals are interpreted in ego coordinates. During the simulation, the ego vehicle traveled at a constant speed of 100 km/h (27.78 m/s).

Trajectory planning was performed using the Timed Elastic Band approach. As an initial guess, a small set of seed points is placed along a straight line; the TEB cost function then deforms these points to match the actual lane geometry while accounting for lane-boundary constraints and smoothness requirements.

Lane-keeping quality is characterized by the lateral deviation and its root-mean-square (RMS) error. The lane width is 3.6 m; subtracting the ego vehicle footprint yields the admissible lateral bounds within which the vehicle must remain when treated as a point for this evaluation. In this case, the lateral deviation was required to stay within $[-0.9, 0.9]$ m. The safe lane boundaries are therefore indicated by the ± 0.9 m limits, as shown in the "Lateral Deviation" plot.

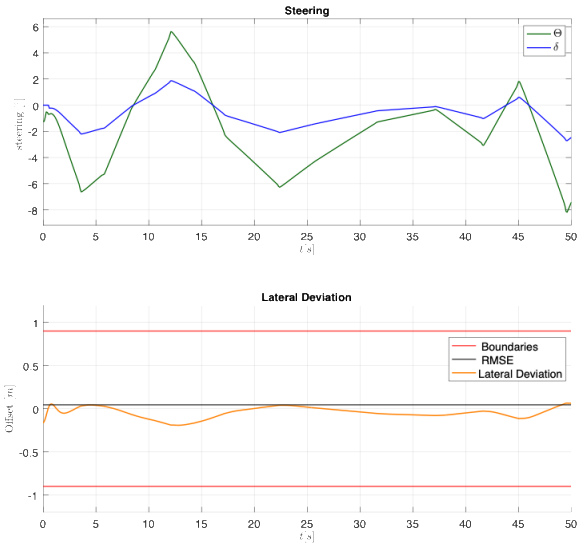


Figure 5: Steering angle and lateral deviation response.

Figure 5 presents the simulation results. The upper panel shows the evolution of the steering angle over time. The Pure Pursuit controller effectively aligns the steering command with the TEB waypoints, and the vehicle remains within the lane boundaries throughout the simulation. The lower panel shows the lateral deviation, which remains within geometrically admissible bounds, with a slight maximum deviation from the lane centerline (19 cm).

4.2 Testing the Adaptive Cruise Control

To evaluate the performance of the adaptive cruise control (ACC), a simulation environment was constructed in which the ego vehicle travels along a long, straight road

segment with strictly in-lane motion. The test road features continuous solid lines on both sides; therefore, lane changes are prohibited throughout the simulation. This setup eliminates lateral maneuvers, allowing the assessment to focus exclusively on longitudinal control quality. The MPC-based ACC controls the ego vehicle described earlier, incorporating a 3-second headway policy and the vehicle's dynamic acceleration and braking limits.

The traffic scenario includes a single preceding vehicle (lead vehicle) following a predefined speed profile. Initially, the lead vehicle travels more slowly than the ego vehicle. It then decelerates and accelerates in multiple phases, finally settling at a moderate speed. This creates realistic, continuously changing conditions for evaluating the spacing response. During the simulation, the ACC must maintain the desired time-gap distance by adapting to the lead vehicle's instantaneous speed and acceleration, avoiding excessive acceleration and braking, and ensuring collision-free motion.

The results can be summarized through three key characteristics. Figure 6 shows the evolution of the actual inter-vehicle distance together with the reference following distance targeted by the MPC. After the transient phase, the controller maintains the gap close to the reference. During acceleration and deceleration intervals, the spacing does not enter a critical range, and no sustained under-following occurs. The middle plot shows the speed profiles of the ego and lead vehicles and illustrates that the ACC gradually aligns the ego speed with the lead vehicle's speed while avoiding abrupt speed changes. The bottom panel shows the acceleration command generated by the MPC; after the initial settling phase, the actuation remains predominantly continuous with a small amplitude, supporting comfortable, physically realistic behavior.

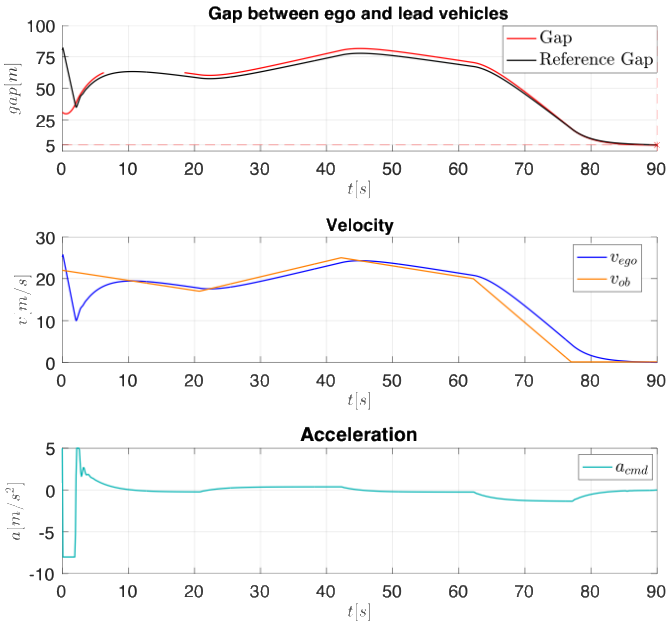


Figure 6: ACC simulation results.

The most critical segment of the simulation is the final emergency-braking maneuver, in which the lead vehicle decelerates to zero in a short time. In this situation, the MPC-based ACC applies progressively stronger braking as required, and the ego vehicle comes to a stop while maintaining an inter-vehicle distance of approximately 5 m. This indicates that, even under the imposed dynamic constraints, the controller can ensure a safe stop without collision while maintaining spacing within the defined safety envelope throughout the maneuver.

4.3 Lane Following in the Presence of Obstacles

This experiment considers a two-lane curved road segment. The ego vehicle travels in the outer lane while multiple vehicles appear ahead with different speed profiles. In certain sections, the lane markings are dashed; therefore, overtaking is permitted in principle. The ego vehicle follows the waypoints generated by the TEB-based planner, while longitudinal motion is governed by the MPC controller, which applies the 3-second time-headway policy.

4.3.1 Simulation Procedure

1. A slower vehicle moves into the ego vehicle's path in the outer lane. In the overtaking lane, a car approaches from behind at high speed and passes the ego vehicle, therefore a lane change cannot be initiated yet.
2. Once the higher-speed vehicle in the inner lane has passed the ego vehicle and reached a safe distance ahead, the lane change is initiated.
3. Traffic in the left lane is locally faster, therefore the MPC transitions to the set-speed target, since the lead vehicle is significantly quicker and outside the relevant following-distance range.
4. The speed of the lead vehicle in the inner lane decreases, so the ego vehicle merges back into the original outer lane.
5. Near the end of the segment, the lead vehicle performs an emergency braking maneuver and comes to a complete stop. The ACC applies the kinematic braking limit, and the ego vehicle stops safely within the targeted standstill spacing. During the emergency stop at the end of the segment, the TEB-based planner does not generate an evasive maneuver because the scenario provides no adjacent lane into which the vehicle could legally and safely move; therefore, the only admissible solution is a safe stop. If a situation arose in which the ego vehicle could not stop in time even when respecting the braking limit, the TEB optimization would, through the cost function, search for an alternative laterally shifted path and attempt to find a collision-minimizing trajectory, provided that lane geometry and traffic rules allow such a maneuver.

4.3.2 Simulation Evaluation

Gap between ego and lead vehicles: The upper part of Figure 7 shows the time evolution of the measured gap and the reference gap. In the final segment, when the

lead vehicle comes to a stop, the gap decreases monotonically while tracking the reference curve.

Steering: The steering-related plots indicate that, during fast lane-change transitions, the TEB-generated path induces larger and more abrupt lateral commands (θ), which are then smoothed by the Pure Pursuit controller (δ). After the maneuver is completed, the controller returns to a small-amplitude, damped behavior.

Acceleration: The lower plot shows the acceleration response. The signal reaches saturation at multiple points where the MPC enforces the braking constraint, which would be perceived as a strong jerk by passengers in a real vehicle. These large amplitudes and high jerk values result from a weight selection in the cost function that prioritizes safety and spacing accuracy over comfort. In the final segment, the acceleration remains negative, resulting in braking to a complete stop.

Lateral deviation: The lateral deviation remains within the lane boundaries throughout the scenario, showing brief peaks during lane-change transitions before settling back close to the lane centerline.

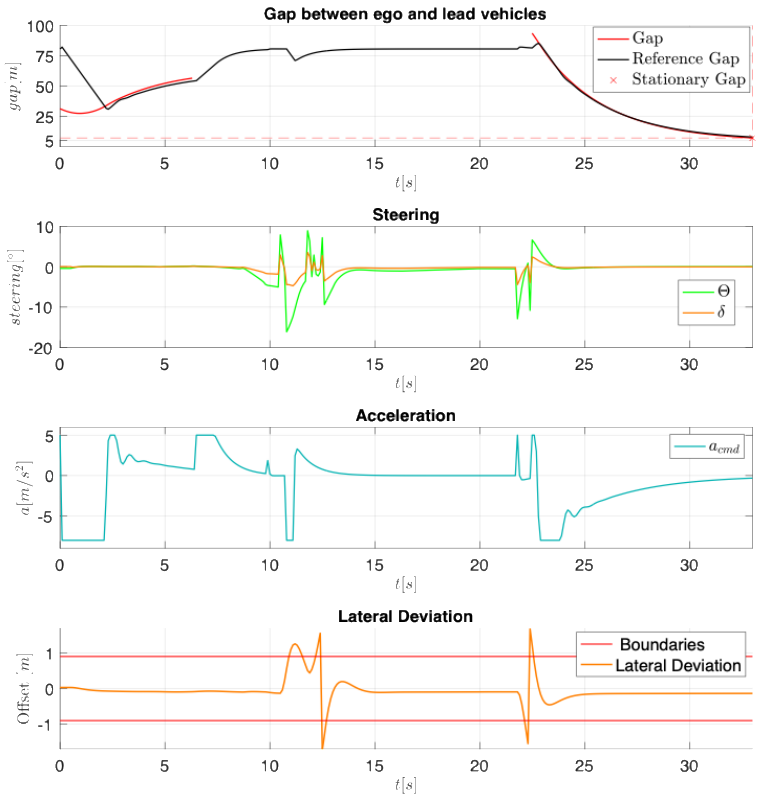


Figure 7: Simulation results in a complex environment.

4.3.3 Summary of Results

At all times, the ego vehicle selected a trajectory that remained compliant with lane-admissibility constraints. The ACC maintained the 3-second time-headway policy throughout the scenario. In the final segment, when the lead vehicle came to a complete

stop, the ego vehicle safely decelerated to a standstill, reaching a final standstill spacing of approximately 5 m.

4.3.4 Interpretation

The presented case illustrates the interaction between the TEB-based planner and the ACC. The TEB searches for lane-change opportunities with comparatively aggressive lateral intent, while the longitudinal dynamics ultimately remain constrained by the ACC limits. Together, this yields a conservative maneuvering behavior that primarily preserves safe following.

4.4 Comparison of TEB+MPC and the Path Following Control System

The Path Following Control System (PFCS) is a MATLAB/Simulink reference model that provides integrated lateral and longitudinal control for a car-like vehicle. The block follows a predefined reference path, typically the lane centerline, and uses nonlinear MPC to compute steering-angle and longitudinal-acceleration commands while ensuring the desired speed tracking and maintaining the headway to a lead vehicle. However, PFCS does not perform independent obstacle-avoidance planning or plan lane-change maneuvers; it focuses exclusively on motion along the prescribed path and on car-following behavior.

The two control architectures were compared on the same scenario using five runs, based on average runtime and lane-keeping accuracy. The proposed TEB+MPC system achieved an average runtime of 10.193 s, whereas PFCS required 13.355 s. Thus, the TEB+MPC approach reduced simulation runtime by approximately 24%. This difference indicates that combining TEB-based local planning with a separate MPC-ACC longitudinal controller imposes a lower computational burden than the fully integrated PFCS solution in this setup.

Lateral tracking accuracy was compared using RMSE relative to the lane centerline. For TEB+MPC, the lateral RMSE was 0.026 m, while for PFCS it was 0.021 m. This indicates that PFCS kept the lane centerline slightly more accurately; however, the difference is on the order of millimeters and is negligible in practical terms. A key distinction is that PFCS is limited to lane-center tracking and car-following and does not perform lane changes for obstacle avoidance. In contrast, the TEB+MPC architecture can execute lane-change maneuvers, although with a marginally higher lateral error. Overall, the TEB+MPC solution provides near-identical lane-keeping accuracy at lower runtime while enabling more functionally complex maneuvers.

5. Safety Rationale and Potential Benefits for African Road Contexts

The proposed TEB + Pure Pursuit + MPC-ACC stack improves safety primarily by making constraints explicit and by separating safety-critical functions into modules that can be validated and tuned independently.

Constraint-aware motion generation reduces collision risk. The TEB planner optimizes a short-horizon trajectory while directly penalizing proximity to obstacles and violations of lane boundaries. This produces trajectories that maintain clearance and remain within the drivable corridor, rather than relying solely on geometric tracking of a fixed reference line. In safety terms, the planner encodes keep-out regions and lane admissibility into the optimization objective, which reduces the likelihood of generating an infeasible or unsafe lateral plan when the environment changes.

Predictive longitudinal control maintains a safety distance under disturbances. The MPC-based ACC uses a predictive model to regulate speed and headway. Minimum following distance and acceleration limits are handled as explicit constraints, ensuring that the controller does not request physically impossible or unsafe braking or acceleration. The time-headway policy provides a speed-scaled safety buffer, which is robust to typical lead-vehicle speed fluctuations.

Conservative fallback behavior in constrained situations. When lateral options are not available (for example, no legal adjacent lane), the architecture naturally defaults to longitudinal safety: the ACC can decelerate to maintain a minimum gap and, if necessary, stop safely. This is safety-relevant because it prevents "planning into a dead end" where the vehicle commits to a lateral maneuver that cannot be completed without violating lane or clearance constraints.

Reduced computational load supports timely reactions. The measured runtime advantage versus PFCS indicates that, in this implementation, the modular planner-controller stack can be computationally lighter than an integrated nonlinear MPC reference model while maintaining comparable lateral accuracy. Lower compute load helps preserve real-time feasibility, a safety prerequisite for timely responses to traffic changes.

5.1 Potential Benefits for African Road Context

Road traffic injury remains a major public-health and development burden in Africa, with the region carrying a disproportionate share (19%) of global road deaths and persistent structural gaps in road-safety management and enforcement capacity [23-25]. Systematic reviews also highlight the scale and under-measurement of crash outcomes in many settings, reinforcing the need for interventions that are robust to imperfect infrastructure and heterogeneous road users [26-28], [29].

The proposed TEB + Pure Pursuit + MPC-ACC methodology aligns with these needs because it encodes safety requirements directly into the motion-generation and control loop rather than assuming ideal lane structure or entirely predictable traffic. The TEB layer can explicitly penalize lane-boundary violation and low-clearance proximity, which is relevant when hazards are localized and appear with limited warning. The MPC-ACC layer enforces a time-headway policy and actuator bounds as constraints, which support rear-end crash prevention by maintaining a speed-scaled safety buffer and ensuring braking commands remain physically feasible under emergency

deceleration scenarios. These design choices target common, high-severity crash modes emphasized in global and Africa-focused road-safety reporting, particularly loss-of-control on curves and collision risk in car-following and stop scenarios [23], [25].

From a deployment perspective, African AV discussions emphasize readiness constraints such as infrastructure variability, governance, and public acceptance, alongside the potential for safety gains if the technology is implemented responsibly [33]. Empirical work from Ghana shows that user perception and intention to adopt AV-related technologies are nontrivial and context-dependent, suggesting that incremental, safety-centered driver-assistance functions may be a practical pathway before full autonomy [30], [31]. In parallel, legal and governance analyses argue that regulation must account for local societal considerations and responsible adoption frameworks, which favor architectures with transparent safety rules and auditable constraints rather than opaque behavior [32].

Practically, the methodology supports a staged introduction: first as advanced driver-assistance functions (lane keeping, headway control, emergency braking behavior under constraints), then as higher automation where the environment permits. This incremental strategy is consistent with the need to strengthen institutional road-safety capacity and governance mechanisms highlighted in Africa-wide assessments [25].

6. Conclusions and Summary

Based on simulation studies, it can be concluded that the TEB-based local planner, combined with the Pure Pursuit geometric steering controller, is well-suited for lateral control of a car-like vehicle. In the lane-keeping tests on the curved track, the ego vehicle remained within the lane boundaries throughout the simulation, with a maximum lateral deviation of approximately 0.19 m, which is negligible relative to the lane width. The steering signal follows the TEB-generated waypoints, while Pure Pursuit, accounting for the kinematic bicycle model, produces steering commands that correspond to maneuvers feasible for real vehicles.

For longitudinal motion, the MPC-based ACC controller enforced the 3-second time-headway policy while respecting acceleration limits and comfort-related criteria. On a straight road, with a lead vehicle that alternated between deceleration and acceleration, the following distance stabilized around the reference after the transient response, the speed profile matched the lead vehicle's dynamics, and saturation occurred only where required by safety constraints. In the emergency-braking scenario, the ego vehicle stopped without collision, maintaining a final standstill spacing of approximately 5 m. This confirms that, under the selected parameterization, the MPC can generate a safe braking command that satisfies the physical constraints.

In the complex multi-lane scenario, the integrated architecture exhibited conservative but consistent behavior. Laterally, the TEB sought overtaking opportunities with relatively aggressive intent; however, the longitudinal MPC-ACC limited ego speed in multiple instances due to spacing and acceleration constraints. Overall, the TEB + Pure

Pursuit + MPC-ACC combination provided stable, collision-free behavior consistent with the defined safety requirements across the investigated scenarios.

6.1 Contribution to the Literature

The presented architecture aligns with hierarchical solutions commonly reported in the literature, while implementing a less frequently discussed combination in its specific form: TEB-based local trajectory planning is coupled with a geometric Pure Pursuit steering controller and an MPC-based ACC longitudinal controller within a unified Simulink environment. One contribution of the paper is that it details how TEB can be integrated as a local, lane- and obstacle-aware planner specifically for a car-like vehicle, while keeping lateral control simple and interpretable through a well-established geometric tracking method.

A second contribution is the ACC formulation, which explicitly incorporates relative distance, relative speed, and an integrator state, and augments the objective with a jerk penalty. The results demonstrate how the selection of weighting factors shapes the trade-off between spacing accuracy, speed tracking, and passenger comfort. The final weight vector identified through the parameter study provides a configuration that balances tracking performance and comfort while maintaining the imposed safety constraints.

By evaluating the integrated simulation of the lateral and longitudinal modules across multiple scenarios of increasing complexity, the study shows that a hierarchical architecture, while only partially optimization-based, can approach behaviors typically associated with fully integrated MPC solutions while retaining the transparency of a modular structure. The obtained results, therefore, provide a bridge between purely geometric control approaches and fully unified, joint-optimization control schemes.

Compared to the PFCS reference model, the TEB+MPC architecture achieved roughly a 25% reduction in simulation runtime, while the lane-keeping RMSE was only a few millimeters higher than the PFCS value. Given that PFCS is limited to lane-center tracking, whereas the TEB-based approach also supports lane-change maneuvers and explicit obstacle avoidance, the results indicate that the proposed structure offers a favorable trade-off between computational demand, tracking accuracy, and functional complexity.

6.2 Future Research Directions

Several directions for further development emerge from this work. One key extension is the explicit temporal enrichment of the TEB formulation, in which trajectory points are associated with time parameters, and the objective includes penalties on longitudinal acceleration and jerk. This would bring lateral planning and longitudinal motion closer together, allowing the TEB optimization to address lane boundaries, obstacles, and dynamic constraints in a more unified manner.

A second direction is a fully integrated lateral-longitudinal MPC formulation, where steering angle and acceleration are computed within a single optimization problem. This could be particularly beneficial for emergency avoidance maneuvers and high-utilization situations near adhesion limits, where the conservative behavior of the current hierarchical structure may excessively restrict maneuverability.

Finally, the next step beyond simulation is experimental validation on real vehicles or small-scale test platforms in a hardware-in-the-loop setup. Such experiments would enable the assessment of sensor acquisition, delays, and model mismatch, and would support automatic tuning of TEB and MPC parameters, for instance, using Bayesian optimization. In the long term, this could enable robust deployment of the proposed architecture in real road environments.

Conflict of Interest

The authors hereby declare that they have no financial interest in this manuscript.

Notes on Contributor

Alex Kádár is a PhD student in the Doctoral School of Electrical Engineering at the Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, Department of Control Engineering and Information Technology. His research focuses on local trajectory planning and predictive control for car-like autonomous robots, including Timed Elastic Band-based path planning and model predictive control.

References

Ackaah, W., Leslie, V. L. D., & Osei, K. K. (2021). Perception of autonomous vehicles – A Ghanaian perspective. *Transportation Research Interdisciplinary Perspectives*, *11*, 100437. https://doi.org/https://doi.org/10.1016/j.trip.2021.100437

Ackaah, W., Leslie, V. L. D., & Osei, K. K. (2022). The adoption of self-driving vehicles in Africa: insight from Ghana. *Urban, Planning and Transport Research*, *10*(1), 333-357. https://doi.org/10.1080/21650020.2022.2092548

Adeloye, D., Thompson, J. Y., Akanbi, M. A., Azuh, D., Samuel, V., Omoregbe, N., & Ayo, C. K. (2016). The burden of road traffic crashes, injuries and deaths in Africa: a systematic review and meta-analysis. *Bull World Health Organ*, *94*(7), 510-521a. https://doi.org/10.2471/blt.15.163121

Africa Transport Policy Program (SSATP). (2025). Africa status report on road safety 2025.

Al-Gabalawy, M., Hosny, N., & Aborisha, A.-h. (2021). Model predictive control for a basic adaptive cruise control. *International Journal of Dynamics and Control*, *9*, 1-12. https://doi.org/10.1007/s40435-020-00732-w

Alomari, K., Mendoza, R., Sundermann, S., Goehring, D., & Rojas, R. (2020). *Fuzzy Logic-based Adaptive Cruise Control for Autonomous Model Car*. https://doi.org/10.5220/0010175101210130

Artuñedo, A., Moreno-Gonzalez, M., & Villagra, J. (2024). Lateral control for autonomous vehicles: A comparative evaluation. *Annual Reviews in Control*, *57*, 100910. https://doi.org/https://doi.org/10.1016/j.arcontrol.2023.100910

Bhalla, K., Harrison, J. E., & Shahraz, S. (2014). *Burden of road injuries in Sub-Saharan Africa*. World Bank Global Road Safety Facility.

Coulter, R. C. (1992). *Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01*. Carnegie Mellon University, Robotics Institute.

Effoduh, J. O. (2023). Regulating self-driving cars: An African perspective [Preprint]. https://doi.org/10.32920/24242872.v1

Falcone, P., Borrelli, F., Asgari, J., Tseng, E., & Hrovat, D. (2007). *A model predictive control approach for combined braking and steering in autonomous vehicles*. https://doi.org/10.1109/MED.2007.4433694

Ganji, B., Kouzani, A., Khoo, S., & Shams-Zahraei, M. (2014). Adaptive cruise control of a HEV using sliding mode control. *Expert Systems with Applications: An International Journal*, *41*, 607-615. https://doi.org/10.1016/j.eswa.2013.07.085

Gincsainé Dr. Szádeczky-Kardoss, E. (n.d.). Robotkarok és mobilis robotok előadás jegyzet. In *Mobilis robotok – Bevezetés*. Budapesti Műszaki Egyetem.

International Organization for Standardization (ISO). (2011). *Road vehicles: Vehicle dynamics and road-holding ability: Vocabulary*. (ISO 8855:2011). Geneva, Switzerland

Jeong, Y., & Yim, S. (2021). Model Predictive Control-Based Integrated Path Tracking and Velocity Control for Autonomous Vehicle with Four-Wheel Independent Steering and Driving. *Electronics*, *10*(22), 2812.

Keller, M., Hoffmann, F., Hass, C., Bertram, T., & Seewald, A. (2014). Planning of Optimal Collision Avoidance Trajectories with Timed Elastic Bands. *IFAC Proceedings Volumes*, *47*(3), 9822-9827. https://doi.org/https://doi.org/10.3182/20140824-6-ZA-1003.01143

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, *5*(1), 90-98. https://doi.org/10.1177/027836498600500106

Kurse, T. K., Gebresenbet, G., & Daba, G. F. (2024). Prospects for Implementation of Autonomous Vehicles and Associated Infrastructure in Developing Countries. *Infrastructures*, *9*(12), 237.

Lagarde, E. (2007). Road traffic injury is an escalating burden in Africa and deserves proportionate research efforts. *PLoS Med*, *4*(6), e170. https://doi.org/10.1371/journal.pmed.0040170

Li, S., Sun, Z., Liu, Y., Ji, A., Zhang, S., & Chen, L. (2023). Coupled lateral and longitudinal trajectory tracking control with a modified vehicle kinematics model

for autonomous vehicles. *Intelligent Transportation Infrastructure*, *2*, liad009. https://doi.org/10.1093/iti/liad009

Löfberg, J. (2004). A toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation* (pp. 284-289). IEEE Cat. No.04CH37508. https://doi.org/10.1109/CACSD.2004.1393890

Naus, G. J. L., Vugts, R. P. A., Ploeg, J., van de Molengraft, M. J. G., & Steinbuch, M. (2010). String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach. *IEEE Transactions on Vehicular Technology*, *59*, 4268-4279.

Nie, Z., & Farzaneh, H. (2020). Adaptive Cruise Control for Eco-Driving Based on Model Predictive Control Algorithm. *Applied Sciences*, *10*(15), 5271.

Quinlan, S., & Khatib, O. (1993). Elastic bands: connecting path planning and control. *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 802-807 vol.802.

Rajamani, R. (2012). Vehicle Dynamics and control. In *Mechanical Engineering Series*. https://doi.org/10.1007/978-1-4614-1433-9

Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., & Bertram, T. (2013). Efficient trajectory optimization using a sparse model. *2013 European Conference on Mobile Robots*, 138-143. https://doi.org/10.1109/ECMR.2013.6698833

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C.,…Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, *23*(9), 661-692. https://doi.org/https://doi.org/10.1002/rob.20147

Vissoci, J. R. N., Shogilev, D. J., Krebs, E., Andrade, L., Vieira, I. F., Toomey, N., Portero Batilana, A., Haglund, M., & Staton, C. A. (2017). Road traffic injury in sub-Saharan African countries: A systematic review and summary of observational studies. *Traffic Inj Prev*, *18*(7), 767-773. https://doi.org/10.1080/15389588.2017.1314470

Wang, Z., Sun, K., Ma, S., Sun, L., Gao, W., & Dong, Z. (2022). Improved Linear Quadratic Regulator Lateral Path Tracking Approach Based on a Real-Time Updated Algorithm with Fuzzy Control and Cosine Similarity for Autonomous Vehicles. *Electronics*, *11*(22), 3703.

World Health Organization. (2023). Global status report on road safety 2023.

World Health Organization Regional Office for Africa. (July 16, 2024). Road traffic deaths rise in the African region, but down globally, WHO report. WHO AFRO.

Zhang, C., Wei, X., Wang, Z., Zhang, H., & Guo, X. (2022). T-S fuzzy-model-based adaptive cruise control for longitudinal car-following considering vehicle lateral stability. *Intelligence & Robotics*, *2*(4), 371-390. https://doi.org/10.20517/ir.2022.26

Zhu, G., Jie, H., & Hong, W. (2023). Nonlinear Model Predictive Path Tracking Control for Autonomous Vehicles Based on Orthogonal Collocation Method.

International Journal of Control, Automation and Systems, *21*(1), 257-270. https://doi.org/10.1007/s12555-021-0812-7

Figure captions

Figure 1: Car-like vehicle [20] 51
Figure 2: Formulation of the TEB points 52
Figure 3: Pure Pursuit geometry 55
Figure 4: MPC testing under different parameter settings 60
Figure 5: Steering angle and lateral deviation response 62
Figure 6: ACC simulation results 63
Figure 7: Simulation results in a complex environment 65