Fuel Efficiency Optimization in Adaptive Cruise Control: A Comparative Study of Model Predictive Control-Based Approaches

applied sciences

MDPI

Article

Fuel Efficiency Optimization in Adaptive Cruise Control: A Comparative Study of Model Predictive Control-Based Approaches

Angelo Borneo, **Federico Miretti** and **Daniela Anna Misul** *

DENERG—Department of Energy "Galileo Ferraris", CARS@Polito—Center for Automotive Research and Sustainable Mobility, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy; angelo.borneo@polito.it (A.B.); federico.miretti@polito.it (F.M.)
* Correspondence: daniela.misul@polito.it

Abstract: This work investigates the fuel efficiency potential of Adaptive Cruise Control (ACC) systems, focusing on two optimization-based control approaches for internal combustion engine (ICE) vehicles. In particular, this study compares two model predictive control (MPC) designs. In the first approach, a strictly quadratic cost is adopted, and fuel consumption is indirectly minimized by adjusting the weights assigned to state tracking and control effort. In the second approach, a fuel consumption map is explicitly included in the MPC cost function, aiming to directly minimize it. Both approaches are compared to a globally optimal benchmark obtained with dynamic programming. Although these methods have been discussed in the literature, no systematic comparison of their relative performance has been conducted, which is the primary contribution of this article. The results demonstrate that, with proper tuning, the simpler quadratic approach can achieve comparable fuel savings to the approach with explicit fuel consumption minimization, with a maximum variation of 0.5%. These results imply that the first alternative is more suitable for online implementation, due to the more favorable characteristics of the associated optimization problem.

Keywords: adaptive cruise control; model predictive control; dynamic programming; energy savings; safety

check for updates

Citation: Borneo, A.; Miretti, F.; Misul, D.A. Fuel Efficiency Optimization in Adaptive Cruise Control: A Comparative Study of Model Predictive Control-Based Approaches. *Appl. Sci.* **2024**, *14*, 9833. https://doi.org/10.3390/app14219833

Academic Editors: Redouane Khemmar, Rémi Boutteau, Karim Hammoudi and Ali Zemouche

Received: 13 September 2024
Revised: 18 October 2024
Accepted: 25 October 2024
Published: 28 October 2024

1. Introduction

The concept of mobility has undergone significant transformations in recent years, with driving automation emerging as a pivotal development. This evolution aims to improve road safety, enhance passenger comfort, and achieve energy savings [1]. Autonomous vehicles and advanced driver assistance systems (ADAS) are central to realizing these objectives [2]. Among the various ADAS, adaptive cruise control (ACC) is particularly effective in reducing energy consumption and greenhouse gas emissions by maintaining a target speed while adjusting the vehicle's velocity to follow another vehicle at a safe distance. Even straightforward ACC implementations yield modest fuel savings [3], while further reduction can be obtained through an appropriate optimization framework [4]. Furthermore, full-range ACC, which operates across the entire speed spectrum, from complete stops to highway speeds, can enhance energy savings by intelligently managing stop-and-go situations [5], which typically involve higher and more frequent acceleration and deceleration than highway cruising.

A driving factor for research in ACC developments stems from the pressing need to reduce the environmental impact of the transportation sector. Internal combustion engine (ICE) vehicles, both conventional and hybrid, which still dominate the global market [6], are significant contributors to greenhouse gas emissions. While electric vehicles (EVs) offer a promising solution, their widespread adoption is currently limited by factors such as high costs, limited driving range, and insufficient charging infrastructure. Consequently, optimizing the fuel efficiency of existing ICE vehicles remains a critical objective. Fuel consumption and emissions from the transportation sector have a considerable impact on global environmental health [7]. Even minor improvements in fuel efficiency can yield significant environmental benefits [8]. ACC systems optimized for fuel efficiency present a viable contribution to address this challenge.

This research explores the optimization of full-range ACC using dynamic programming (DP) and two model predictive control (MPC) approaches to balance fuel savings with passenger comfort while maintaining safe driving conditions.

1.1. Background

Adaptive cruise control is one of the most widely adopted ADAS technologies. It extends traditional cruise control by automatically adjusting the vehicle's speed to maintain a safe distance from the vehicle ahead [9]. This capability not only enhances driving comfort by reducing the need for manual speed adjustments and reducing frequent changes in acceleration but also has the potential to improve fuel efficiency by promoting smoother driving behavior. Moreover, full-range ACC can take care of the longitudinal control of the vehicle in urban, highway, and mixed scenarios.

Several studies have sought to improve ACC's performance by optimizing fuel consumption while ensuring safety and comfort for both conventional and hybrid vehicles [10]. In some works, pollutant emissions were also considered as control objectives [11]. Various optimization approaches, such as dynamic programming (DP) [12–14], model predictive control (MPC) [15–18], linear-quadratic regulators [19,20], and reinforcement learning solutions [21–23] have been proposed. Each approach has its merits and disadvantages.

Dynamic programming is effectively the only technique among these that can guarantee optimal solutions, albeit with high computational costs. In short, DP works by breaking down a complex decision-making process into simpler sub-problems and solving them sequentially. Bellman's principle of optimality is central to DP, enabling the formulation of an optimal policy accounting for the cumulative cost of future actions [24].

In the context of ACC, DP can be used to determine the optimal acceleration and deceleration patterns that minimize fuel consumption over a given driving cycle. However, its high computational complexity makes it impractical for real-time applications. Because of this limitation, DP is best suited for offline applications, where it remains valuable as a benchmark for evaluating the performance of other, more computationally efficient methods.

Model predictive control is advantageous for online applications due to its ability to anticipate future events and optimize control actions accordingly. MPC's capability to handle multi-variable interactions and constraints explicitly makes it a promising approach for real-time ACC optimization. Unlike DP, MPC operates in a receding horizon manner, continuously updating its predictions based on the latest state measurements.

MPC involves solving an optimization problem at each time step, where the objective is to minimize a cost function. By incorporating constraints directly into the optimization process, MPC ensures that the resulting control actions are feasible and safe. The primary challenge with MPC is maintaining real-time feasibility, especially when dealing with complex vehicle models and long prediction horizons. In a recent study [25], deep neural networks were used to generate an approximated policy that reproduces an MPC-based ACC system in order to reduce execution time and memory footprint. Nonetheless, explicit MPC implementations have been proven to be capable in real-time, such as the robust MPC algorithm developed by [26].

Linear quadratic regulators (LQRs) can be seen as a special case of model predictive control under specific conditions. Firstly, LQRs always assume that the system is linear and the cost function is quadratic [27]. In MPC, this is also assumed to be the standard problem structure, but other variations can be treated. For example, nonlinear costs can be considered, although in this case, the term nonlinear MPC is commonly used. Secondly, LQR does not handle constraints, while one of the primary strengths of MPC is its ability to explicitly incorporate constraints on inputs, states, and outputs. Lastly, LQR is inherently an infinite-horizon controller, while MPC typically operates over a finite horizon in the receding horizon technique that was previously described.

Despite all its limitations, LQR presents a notable advantage with respect to MPC in terms of computational costs. LQR results in a fixed feedback gain matrix that is computed offline, while MPC requires solving an optimization problem at each time step, which can adapt to changing conditions.

Reinforcement learning (RL) differs from the previously mentioned techniques in that it typically does not require a model of the system to be controlled. The RL controller (also called the agent) trains control policies through trial and error by interacting with the environment. It continuously improves using rewards received from the environment, without requiring a model of the system. Deep reinforcement learning, in particular, is deemed to be particularly suitable to deal with complex real-life environments [28]. Thus, RL is a powerful tool to deal with highly variable scenarios, complex environments, or where a system model is difficult to define. Its downside is that it often requires extensive training effort [29].

1.2. Motivation

As previously mentioned, ACC systems, when intelligently controlled, can play a crucial role in this context. By optimizing speed control to minimize fuel consumption while ensuring safety and comfort, these systems can contribute to significant reductions in emissions. Optimization-based techniques like MPC and RL offer significant flexibility in defining a cost function, allowing for various formulations to achieve objectives such as fuel economy, car-following, safety, and comfort. Specifically, for MPC, two main approaches can be distinguished: one that explicitly incorporates fuel consumption in the objective functions, and another that attempts to indirectly minimize fuel consumption by other means.

For example, ref. [30] incorporates a fuel consumption map in the cost function, where the fuel flow rate is obtained via linear interpolation on experimental engine data to maximize fuel economy. Similarly, ref. [31] exploits a fuel map, but a fourth-order polynomial approximation is ultimately used in the MPC cost function rather than an interpolant function. In a different approach, ref. [32] utilizes a strictly quadratic formulation of the cost function and relies on the tracking objective to maximize fuel economy. More specifically, a reference trajectory for the acceleration is defined with an exponential attenuation of its current value; thus, a smooth response for the ACC controller is favored, which in turn should minimize fuel consumption. A similar approach was developed and tested in [33], and an impressive 12–13% improvement in fuel consumption with respect to a traditional ACC implementation based on PID (proportional–integral–derivative) control was reported.

In many ACC applications, fuel consumption is not a quadratic function of the control effort, though it is strictly monotonic. Therefore, adhering to a quadratic cost function requires significant approximations and may lead to sub-optimal performance. On the other hand, adopting a non-quadratic cost function significantly increases the complexity of the optimization problem. Thus, more sophisticated solvers and more computational power might be needed, and the optimality of the solution might not be guaranteed if the problem is non-convex.

1.3. Contributions

This work contributes to the field by conducting a comprehensive, unbiased comparison of two MPC strategies for ACC optimization:

- one with a strictly quadratic cost and indirect minimization of fuel consumption;
- a second one that explicitly includes a fuel consumption map in the MPC cost function.

To ensure a complete comparison, dynamic programming is introduced as a benchmark solution to determine the most effective approach for real-time full-range ACC optimization. The optimization problem is tailored for an internal combustion engine light-duty vehicle, which is extensively utilized in the aforementioned mixed driving scenario, encompassing both urban and highway conditions in a car-following scenario. The latter entails a single-lane and straight road. Both vehicles considered in this study share identical physical properties. The first vehicle, referred to as the lead vehicle, sets the pace, while the second, defined as the ego vehicle, follows. In this car-following scenario, the ego vehicle acts as the follower and is considered as an autonomous vehicle.

2. Methods

As mentioned in the introduction, MPC and DP are model-based optimization techniques that require a control-oriented simulation model of the system to be controlled. Also, both techniques were tested with a higher-fidelity simulation model. In this section, we first describe the equations governing the simulation model. Then, we describe the control-oriented model. Finally, we describe the implementation of the MPC algorithm as well as the DP algorithm that was used as a benchmark.

External disturbances, aside from the lead vehicle's speed, are disregarded at this stage, as the primary focus is on comparing potential energy savings.

2.1. Simulation Model

The simulation model aims to simulate an internal combustion engine vehicle in a car-following scenario. Its goal is to evaluate the potential fuel consumption benefits of full-range ACC and the effectiveness of MPC through two different formulations of the cost function. The vehicle model was developed in MATLAB, considering only longitudinal dynamics to simplify the analysis while retaining the essential dynamics for ACC.

As previously discussed, the primary vehicle model employed to describe vehicle dynamics and simulate the real vehicle, for the purpose of evaluating controller performance, incorporates a detailed representation of the longitudinal dynamics and the powertrain. The vehicle's position, speed, and acceleration were computed using a fixed reference system. The ICE provides the necessary torque to overcome road load and achieve the desired acceleration, as commanded by the controller. This setup mimics real-world driving conditions where the following vehicle adjusts its acceleration in order to proceed at the speed required to maintain a safe distance from the leading vehicle while optimizing fuel consumption.

The equation of motion [34] for the ego vehicle is given as follows:

$$m_{\mathrm{tot}}a = \eta_{\mathrm{gb}} \cdot \tau_{\mathrm{gb}} \cdot \frac{T_{\mathrm{wh}}}{r_{\mathrm{wh}}} - \left[mg\sin(\alpha) + mgf_0\cos(\alpha) + mgf_2\cos(\alpha)v^2\right] \tag{1}$$

where:

- m_{tot} is the equivalent translating mass of the vehicle, considering both translating and rotating components;
- a and v are the vehicle's acceleration and speed;
- T_{wh} is the torque at the wheels;
- f_0, f_2 are the rolling resistance coefficients;
- r_{wh} is the wheel radius;
- τ_{gb} and η_{gb} are the transmission's gear ratio and efficiency, respectively;
- α is the road slope.

The wheel speed is given as follows:

$$\omega_{\mathrm{wh}} = \frac{v}{r_{\mathrm{wh}}}. \tag{2}$$

The powertrain model includes a final drive, a six-speed gear transmission, and the engine; its goal is to link the wheel speed and torque to the engine fuel consumption. In particular, a backward vehicle model was considered, meaning that the required engine torque and speed are computed in order to match the given acceleration. The model is also influenced by the engaged gear, as this changes the transmission's gear ratio (τ_{gb}). In this work, we assume that the gear shift logic is not part of the ACC controller but rather is implemented with a traditional gear shift schedule. Therefore, the gear number was not included as a control variable for the DP and MPC algorithms.

The vehicle model also incorporates a fuel consumption map, which provides the fuel flow rate (in grams per second) as a function of the engine's torque T_{eng} and speed ω_{eng} via linear interpolation on experimental data points:

$$\dot{m}_{\text{fuel}} = \max(0, f(\omega_{\text{eng}}, T_{\text{eng}})). \tag{3}$$

2.2. Control-Oriented Model

As further described in Section 2.5, model predictive controllers require a predictive model to compute a control sequence. Since the model is used to solve an optimization problem in real-time, this control-oriented model must have a favorable structure. To do so, the best choice from a computational perspective is to use a linear model of the plant and a quadratic cost function. A simplified model was therefore derived for ACC which links the control variable u, i.e., the acceleration command, to the vehicle's motion [35]:

$$\tau \frac{d}{dt}\ddot{s}(t) + \dot{s}(t) = u(t). \tag{4}$$

Here, τ is a constant that models the inertia between the controller command and its actuation, and its value is set to 0.5; s and its derivatives $\dot{s}, \ddot{s}$ refer to the vehicle position, velocity, and acceleration; and u is the control variable.

From this, the system model was cast in state-space form as follows:

$$x_{k+1} = Ax_k + Bu_k + Gd_k \tag{5}$$

where x is the vector of state variables, u are the controls, and d are the external disturbances.

In our problem, the state variables are as follows:

$$x = \begin{bmatrix} e_{\text{d}} \\ v \\ a \end{bmatrix} \tag{6}$$

where v is the velocity of the ego vehicle and a is the acceleration of the ego vehicle. e_{d} is the distance error, i.e., the difference between the desired inter-vehicular distance and its desired value, as defined by the spacing policy (described in Section 2.3). The external disturbance is the velocity of the lead vehicle v_{lead}, which is needed to evaluate e_{d}.

Therefore, the state-space representation for our model is as follows:

$$x_{k+1} = \begin{bmatrix} 1 & -T_s & -Tt_h \\ 0 & 1 & T_s \\ T_s & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ T_s/\tau \end{bmatrix} u_k + \begin{bmatrix} T_s \\ 0 \\ 0 \end{bmatrix} d_k. \tag{7}$$

where T_s is the sampling time.

2.3. Spacing Policy

As shown in Equation (6), the distance error, e_{d}, is crucial in the problem formulation. To ensure a safe distance between vehicles, it is essential to define an appropriate spacing policy. Various strategies can be employed, ranging from the simplest approach, such as maintaining a constant safety distance, to more sophisticated methods that incorporate velocity or acceleration terms into the safety distance [36]. In this study, the inter-vehicular safety distance is determined by the following law:

$$\delta_{\text{safe}} = d_0 + t_h v_{\text{ego}} \tag{8}$$

where d_0 represents the desired distance at standstill and v_{ego} is the speed of the ego vehicle. t_h is the time headway, i.e., the elapsed time between the front of the lead vehicle and the front of the ego vehicle passing the same point on the roadway, assuming that both vehicles move at a constant relative speed.

In the spacing policy, δ_{safe} is the distance computed using the desired time headway, which in this work is set to be $t_h = 1.4$. Consistently, with Equation (8), a constant time headway implies a safety distance growing linearly with respect to the velocity. Reference [37] suggests time headway in the interval [1 s, 2 s] based on safety and road capacity concerns.

Finally, the distance error, first introduced in Section 2.2, is defined as the difference between the current inter-vehicular distance δ and δ_{safe}:

$$e_{\text{d}} = \delta - \delta_{\text{safe}} = d_{lead} - d_{\text{ego}} - (d_0 + t_h v_{\text{ego}}). \tag{9}$$

2.4. Dynamic Programming

Dynamic programming is a mathematical optimization method used to solve multi-stage decision problems by breaking them down into simpler sub-problems. DP is particularly useful in problems where decisions need to be made sequentially and the outcome depends on previous decisions. At each stage, the system progresses according to its inherent dynamics and is affected by the decisions made. If the system's evolution is entirely predictable, the problem is classified as deterministic, as for the problem under investigation. Conversely, if random phenomena impact the system's evolution, the problem is considered stochastic.

In ACC optimization, DP is used to compute the optimal speed and acceleration profiles that minimize fuel consumption while maintaining a safe distance from the lead vehicle and satisfactory driver comfort. The DP algorithm considers all possible acceleration and deceleration scenarios at each time step and evaluates their impact on the total cost over a finite horizon. The Bellman equation guides the recursive computation of the optimal policy, which minimizes the cumulative cost [38]:

$$J^*(x_k) = \min_{u_k}\{g(x_k, u_k) + J^*(x_{t+1})\}, \tag{10}$$

where:

- $J^*(x_k)$ is the optimal cost-to-go from state x_k;
- $g(x_k, u_k)$ is the immediate cost incurred by applying control u_k at state x_k;
- x_{t+1} is the state at the next time step resulting from control u_k.

In our framework, the total cost J includes two terms. The first is the fuel flow rate to minimize the overall fuel consumption. The second is a penalty term to enforce acceleration constraints. The cost was formulated as follows:

$$J = \sum_{t=0}^{T} \Big(\dot{m}_{\text{fuel}}(t) \cdot T_s + w_1 a^2(t)\Big), \tag{11}$$

where w_1 is a weighting factor. The penalty term ensures that the vehicle avoids abrupt accelerations or decelerations that could compromise passenger comfort or safety.

The optimization problem was implemented in a MATLAB environment, and a MATLAB tool for general-purpose DP optimization, called DynaProg [39], was used to solve the problem. For the current application, the optimization tool allows us to set unfeasibility, thus constraining the problem easily. For this reason, to ensure that the vehicle maintains a safe distance from the lead vehicle, constraints on the distance error are set. The minimum distance constraint is set to avoid collision between the two vehicles and is formulated as follows:

$$e_{\mathrm{d,min}} = max(-0.9\, t_{h,\mathrm{des}} v_{\mathrm{ego}}, -20) \quad (12)$$

By imposing this constraint, we ensure that the ego vehicle does not fall below the standstill distance. However, we allow the relative distance to be less than the predefined safe distance without risking a collision. The maximum distance error constraint is instead a constant value:

$$e_{\mathrm{d,max}} = 30 \quad (13)$$

Setting this threshold is crucial to prevent the inter-vehicle distance from becoming too large, which increases the likelihood of other vehicles cutting in. It is indeed important to take into account how other traffic participants react to ecological driving strategies [40,41].

2.5. Model Predictive Control

2.5.1. MPC Algorithm and Implementation

Model predictive control is suitable for online applications, predicting future states to optimize control actions. MPC uses a predictive model to compute a control sequence, minimizing a cost function over a finite horizon. The process involves iterating on the control sequence, applying only the first control input, and updating the model with new measurements at each time step. To facilitate the online application of the controller, a linear MPC approach was chosen, necessitating a linear plant model for predictions and a quadratic cost function. The linear model allows the MPC to predict the system states over the prediction horizon, consisting of H_p steps, each with a duration of T_s.

The MPC algorithm follows these steps:

1. Prediction model: Use the vehicle model to predict future states over the prediction horizon based on the current state and control inputs.
2. Optimization: Solve an optimization problem to find the control inputs that minimize the cost function, subject to constraints.
3. Implementation: Apply the first control input from the optimized sequence.
4. Update: Update the model with new state measurements and repeat the process.

The prediction model accounts for vehicle dynamics and the behavior of the lead vehicle. The optimization problem is solved using numerical methods, typically with a dedicated solver.

The control inputs include throttle and brake commands, which are adjusted to achieve the desired speed and distance. The implementation of the first control input uses the detailed vehicle model to end up in the update phase with the new state measurements.

The MPC controller was implemented in a MATLAB environment using the YALMIP toolbox [42], which allows for the formulation of optimization problems using algebraic equations and acts as an interface to an array of solvers. The optimization problem was transcribed in an implicit, and hence sparse, formulation [43] and solved with a semidefinite programming solver from the MOSEK software, 10.2.1 version number package [44]. As mentioned above, the problem is in a quadratic form; thus, the chosen solver is computationally efficient for this class of optimization problems. To ensure feasibility and stability, practical strategies were employed, including the introduction of soft constraints, as detailed in the following section, and the use of a sufficiently long prediction horizon.

To meet the performance requirements, two formulations for the cost function are proposed. The first one includes fuel consumption, distance error, velocity, acceleration, and a penalty on control effort. The second one does not include fuel consumption. The reason behind this is that the control problem of a fuel consumption-oriented solution is typically not as simple as a linear problem with a quadratic cost function. In [15], the authors state that under certain conditions, the fuel consumption can be considered proportional to the acceleration. Our goal is to demonstrate that by appropriately tuning weights, the two controllers can achieve comparable results, and we validate their goodness by comparing them with DP.

2.5.2. Cost Function with Explicit Fuel Economy Objective

As mentioned in the introduction, two different cost functions were formulated for the MPC implementation. Both attempt to realize the car-following objective while minimizing fuel consumption. Terms related to comfort and safety are also included to ensure a realistic implementation.

The first cost function does not conform to a quadratic form and was formulated as follows:

$$J_{\mathrm{FC}} = \sum_{k=0}^{H_p-1} \left(w_1 \dot{m}_{\mathrm{fuel},k+1} + w_2 e_{\mathrm{d},k+1}^2 + w_3 a_{k+1}^2 + w_4 u_k + w_5 \zeta_{e,k}^2 + w_6 \zeta_{u,k}^2 \right) \tag{14}$$

where H_p is the prediction horizon, $w_1, \ldots, w_6$ are the weighting factors of each term of the cost function, and ζ_e, ζ_u are slack variables, which are variables used to soften the constraints. Constraint softening is useful to improve the computational efficiency of the optimization problem and to guarantee that a solution is found even if no physical solution can be computed with the corresponding hard constraints. In such cases, the solver is allowed to search for a feasible solution outside the boundaries but penalizes this action using slack variables in the cost function [45].

These constraints were formulated to achieve the car-following objective without violating the vehicle's acceleration limits:

$$e_{d\min} + \zeta_e \leq e_{\mathrm{d}} \leq e_{d\max} + \zeta_e \tag{15}$$

$$v_{\min} \leq v \tag{16}$$

$$u_{\min} + \zeta_u \leq u \leq u_{\max} + \zeta_u \tag{17}$$

where $e_{d\min} = 0$, $e_{d\max} = 25$, $v_{\min} = 0$, $u_{\min} = -1$ and $u_{\max} = 1$.

Balancing different control objectives, constraint softening, and achieving satisfying performances requires accurate tuning of the controller parameters and hence the weighting factors. In this work these are determined through trial and error.

As explained in Section 2.1, the fuel flow rate map (Equation (3)) is a linear interpolant; thus, it is not a smooth function, and its derivatives present discontinuities. While this is not a challenge for dynamic programming, it makes the MPC optimization problem significantly harder and might seriously hamper its real-time capabilities. To overcome this issue, the fuel flow rate map was approximated with a polynomial function. In particular, we found that a first-order polynomial was sufficient to achieve a reasonably accurate fit for our engine map:

$$\hat{\dot{m}}_{\mathrm{fuel}} = p_{00} + p_{10}\omega_{\mathrm{eng}} + p_{01}T_{\mathrm{eng}} \tag{18}$$

where p_{00}, p_{01} and p_{10} represent the fit coefficients.

Figure 1 shows the fuel flow rate obtained by simulating the vehicle on the UDDS driving cycle using the fuel consumption map and its polynomial approximation. As the figure shows, the first-order polynomial overestimates fuel consumption by approximately 6%.

Equation (18) was then rewritten as a function of the state variables only, hence giving:

$$\hat{\dot{m}}_{\mathrm{fuel}} = p_{00} + p_{10}\frac{x(2)}{r_{\mathrm{wh}}}\tau_{\mathrm{gb}}\tau_{\mathrm{fd}} + p_{01}\frac{(f_0 + f_2 x(2)^2 + m x(3))r_{\mathrm{wh}}}{\tau_{\mathrm{gb}}\tau_{\mathrm{fd}}}, \tag{19}$$

so that the cost function can be written explicitly. We will refer to this controller as $\mathrm{MPC}_{\mathrm{FC}}$ in the following sections.

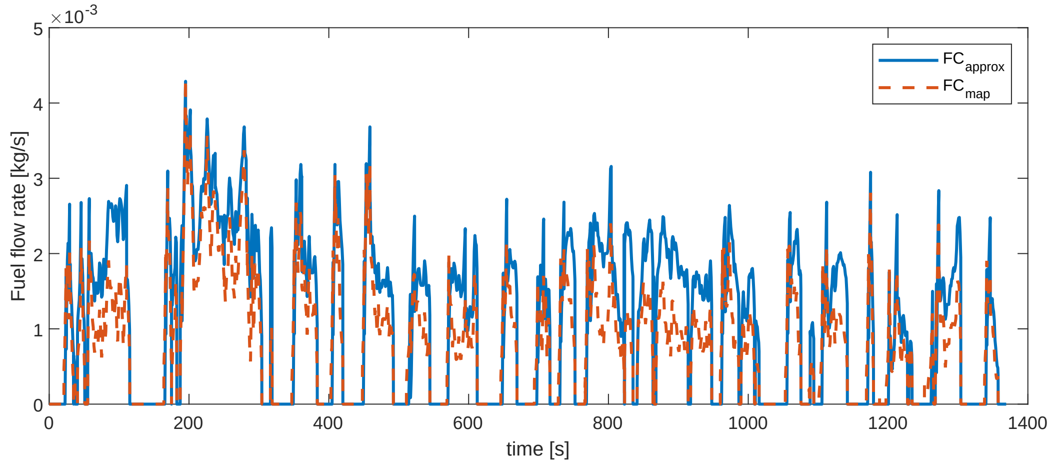

Figure 1. Fuel flow rate results along UDDS driving cycle with map and polynomial function.

2.5.3. Cost Function with Implicit Fuel Economy Objective

Since the implementation of the fuel consumption term in the cost function not only requires algebraic manipulation of its formulation but also several parameters to tune, and given the consideration made in [15], which states that fuel consumption is mainly dominated by vehicle acceleration and that in a car-following scenario, the latter increases as the absolute value of the acceleration increases, the following cost function is proposed, aiming to achieve similar results to the previous section:

$$J_{\text{noFC}} = \sum_{k=0}^{H_p-1} \left(x_k^T Q x_k + r u_k^2 + w_1 \zeta_{e,k}^2 + w_2 \zeta_{u,k}^2 \right) \tag{20}$$

where Q is a diagonal matrix of weights, and r, w_1 and w_2 are the weighting factors of the penalization terms of the cost function. The latter is indeed structured to minimize the variations in the states from an instant to the following one, with a higher weight on acceleration, and to penalize large values for the control variable. The ζ terms, as for the previous formulation (Equation (14)), are slack variables for constraint relaxation.

The problem is constrained as in Equations (15) and (17). These constraints are incorporated into the optimization problem, ensuring that the control actions are both optimal and feasible. MPC's ability to handle constraints is one of its key advantages.

3. Simulation Setup

To evaluate the performance of the two MPCs in optimizing ACC systems, we tested both implementations with the simulation model introduced in Section 2.1. The same model was also used to derive the DP benchmark. The simulations included various driving scenarios to assess the effectiveness of each control strategy in different conditions.

3.1. Driving Scenarios

The driving scenarios include standard driving cycles and real ones to mimic real-world conditions, including:

- Urban driving: Frequent stops and starts, lower speeds, and varying traffic conditions.
- Highway driving: Higher speeds, fewer stops, and steady traffic flow.
- Mixed driving: A combination of urban and highway conditions.

Each scenario included a lead vehicle following a predefined speed profile, and the ego vehicle equipped with ACC adjusted its speed accordingly. The lead vehicle's speed profiles include different patterns, such as constant speed, gradual acceleration, sudden deceleration, and stop-and-go behavior.

3.2. Performance Metrics

The performance of the DP and MPC-based ACC systems was evaluated using the following metrics:

- Fuel consumption: Total fuel consumption over the driving cycle.
- Fuel consumption savings: Fuel percentage savings compared to the fuel consumption arising from the optimal acceleration profile of the driving cycle.
- Tracking: Deviation from the desired following distance and reference speed.
- Acceleration smoothness: Variability in acceleration, reflecting passenger comfort.

These metrics provided a comprehensive assessment of each control strategy's ability to optimize fuel consumption while ensuring safety and comfort.

3.3. Test Cycles

The controllers were tested on five driving cycles: three standard cycles, denoted as UDDS, ARDC, and AUDC, and two real cycles denoted as RD1 and RD2. Table 1 summarizes the main characteristics of these cycles.

- UDDS: represents city driving conditions and is used for light-duty vehicles, also including stop-and-go simulations;
- ARDC: represents rural driving conditions, reaching higher speeds than 100 km/h;
- AUDC: represents urban conditions;
- RD1: represents urban conditions;
- RD2: represents urban conditions.

Table 1. Mean velocity and RMS of acceleration for each driving cycle.

	UDDS	**ARDC**	**AUDC**	**RD1**	**RD2**
Duration (s)	1369	1082	993	381	435
Mean velocity (m/s)	8.7520	15.9487	4.8992	5.5080	4.9888
Max velocity (m/s)	25.347	30.972	16.028	14.427	18.073
RMS acceleration (m/s^2)	0.6091	0.6289	0.7785	0.7370	0.7013

4. Results and Discussion

To ensure a fair comparison between the different proposed methods, each was tested across the five driving cycles previously described. All controllers were implemented on the same vehicle model developed in MATLAB for consistency.

4.1. Fuel Consumption

As expected, all algorithms showed significant reductions in fuel consumption compared to a baseline lead vehicle. The greatest fuel savings occurred at lower speeds and in scenarios with frequent speed changes. Using SP as the benchmark, fuel savings ranged from a minimum of 6.9% in rural conditions to a maximum of 22% in urban driving conditions. These results reflect the theoretical best performance that an ACC control could achieve if the entire lead vehicle's speed profile for the whole drive cycle was known in advance. Thus, it is no surprise that the MPC controller achieves on average about 70% of the fuel benefit. In urban driving scenarios, MPC achieved fuel savings of up to 17.3%.

As shown in Table 2, the two MPC controllers exhibit similar behavior, with the fuel consumption-based one performing marginally better, achieving improvements of approximately 0.2% to 0.5%.

To thoroughly assess the controllers' performances, the percentage of fuel consumption reduction is considered. As shown in Table 3, the percentage savings for each proposed controller along the five driving cycles exhibit two trends:

- The performance difference between the two MPC controllers is minimal, with a maximum variation of 0.5%, indicating highly comparable results;

- Both MPC controllers and DP show higher beneficial impacts in urban areas and driving cycles characterized by lower average speeds and higher-acceleration RMS, reflecting the efficiency of the control strategies in more dynamic driving conditions.

Table 2. Fuel consumption in Kg for each tested driving cycle and controller.

	UDDS	ARDC	AUDC	RD1	RD2
Leader (kg)	1.1102	1.5676	0.5518	0.2288	0.2479
MPC_{noFC} (kg)	1.0727	1.5127	0.4576	0.2017	0.2147
MPC_{FC} (kg)	1.0692	1.5085	0.4563	0.2006	0.2142
DP (kg)	1.0141	1.4590	0.4293	0.1900	0.1953

Table 3. Percentage fuel savings for each tested driving cycle and controller.

	UDDS	ARDC	AUDC	RD1	RD2
MPC_{noFC}	3.4%	3.5%	17.1%	11.8%	13.4%
MPC_{FC}	3.7%	3.8%	17.3%	12.3%	13.6%
DP	8.6%	6.9%	22.2%	16.9%	21.2%

The difference in fuel savings between DP and MPC can be attributed to the nature of each method. As an offline optimization technique, DP is capable of exploring a broader solution space to find the global optimum. In contrast, MPC, being an online optimization method, focuses on local optima within the prediction horizon. However, MPC's ability to continuously update its predictions based on real-time data allows it to adapt to changing conditions effectively.

4.2. Tracking and Passenger Comfort

Passenger comfort, evaluated based on acceleration smoothness, and performance tracking of MPC-based controllers were initially examined using the UDDS driving cycle, as illustrated in Figure 2.

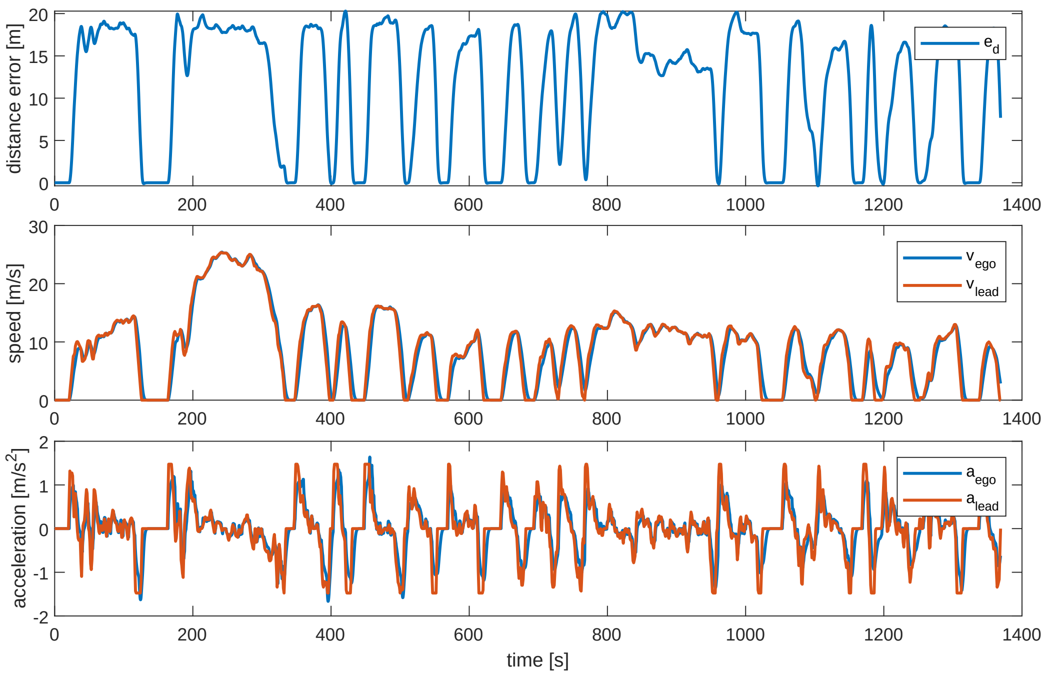

Figure 2. Distance error, MPC−equipped ego, and lead velocities and accelerations along UDDS driving cycle.

For improved clarity, Figure 3 provides a zoomed-in view of the velocity and acceleration profiles, highlighting the controller's ability to produce smoother profiles compared to the lead vehicle. A comprehensive comparison, incorporating the lead vehicle, MPC, and baseline DP velocity and acceleration profiles, is presented in Figure 4 within the same zoomed window. Additionally, Figure 5 extends this comparison to the AUDC driving cycle.

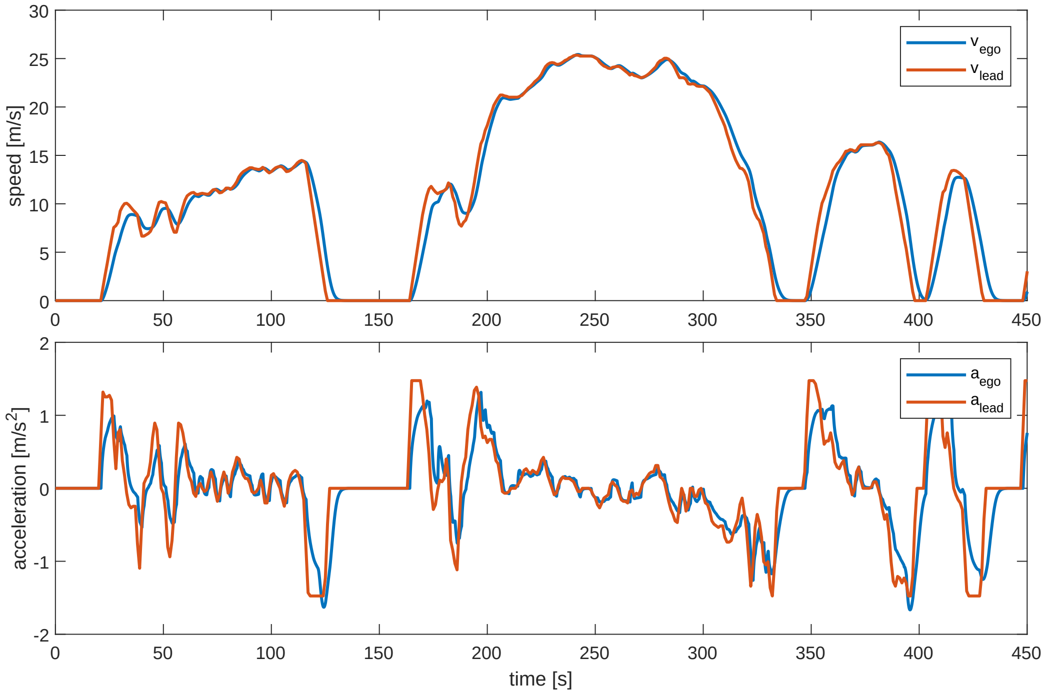

Figure 3. MPC−equipped ego and lead velocities and accelerations along UDDS driving cycle, [0, 450 s] zoom.

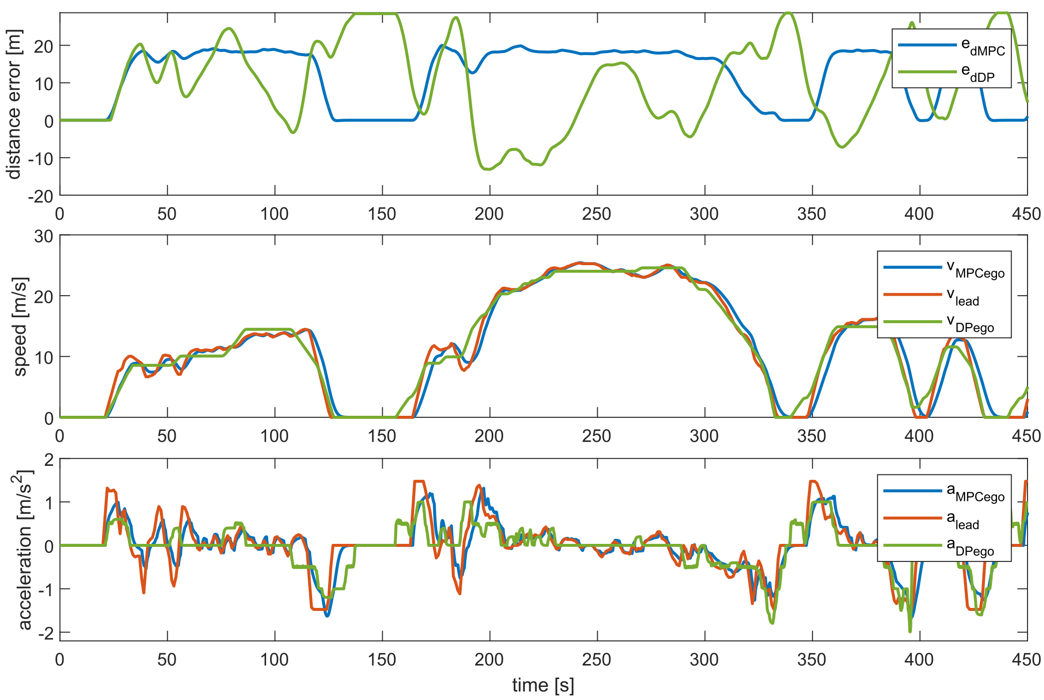

Figure 4. Comparison of velocity and acceleration profiles of ego equipped with fuel consumption MPC and DP controllers with lead along UDDS driving cycle, [0, 450 s] zoom.

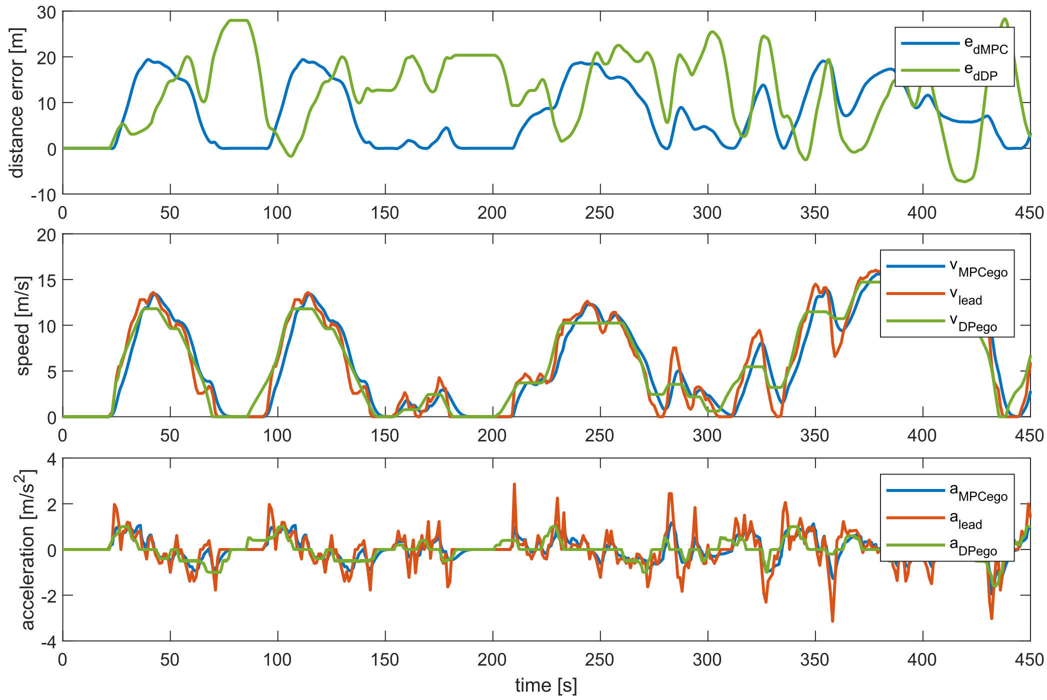

Figure 5. Comparison of velocity and acceleration profiles of ego equipped with fuel consumption MPC and DP controllers with lead along AUDC driving cycle, [0, 450 s] zoom.

Based on the preliminary findings discussed in Section 4.1, only the MPC_{FC} results are displayed, as both MPC controllers produced nearly identical outcomes. Furthermore, Figures 4 and 5 showcase that while DP achieved greater smoothness in velocity due to its access to the entire drive cycle, its acceleration profile occasionally exhibited sharper transitions than MPC in dynamic scenarios. Table 4 further corroborates these findings, emphasizing the minimal differences between the results of the two MPC controllers.

Table 4. Acceleration RMS comparison between the lead vehicle and the ego vehicle equipped with MPC and DP.

	Leader	**MPC_{FC}**	**MPC_{noFC}**	**DP**
RMS acc UDDS (m/s^2)	0.6091	0.4924	0.4963	0.5218
RMS acc AUDC (m/s^2)	0.7785	0.4728	0.4764	0.4412

4.3. Distance Error

The magnitude of the distance error plays an important role in optimizing the controllers' performance. Larger errors provide more leeway for the controller to smooth out the vehicle's speed and acceleration profiles instead of rigidly tracking the lead vehicle's behavior. Figures 4 and 5 demonstrate that the DP algorithm extensively utilizes its available margin owing to its prior knowledge of the driving cycle. This allows DP to dynamically adjust the distance error, even occasionally resulting in negative values, although it never leads to zero or negative inter-vehicle distances. The MPC controller, operating with less foresight, exhibits a more conservative use of the distance error. Notably, in scenarios with greater dynamic changes, both DP and MPC exploit this parameter more actively, as illustrated in Figures 6 and 7, which also depict the velocity and acceleration profiles for completeness.

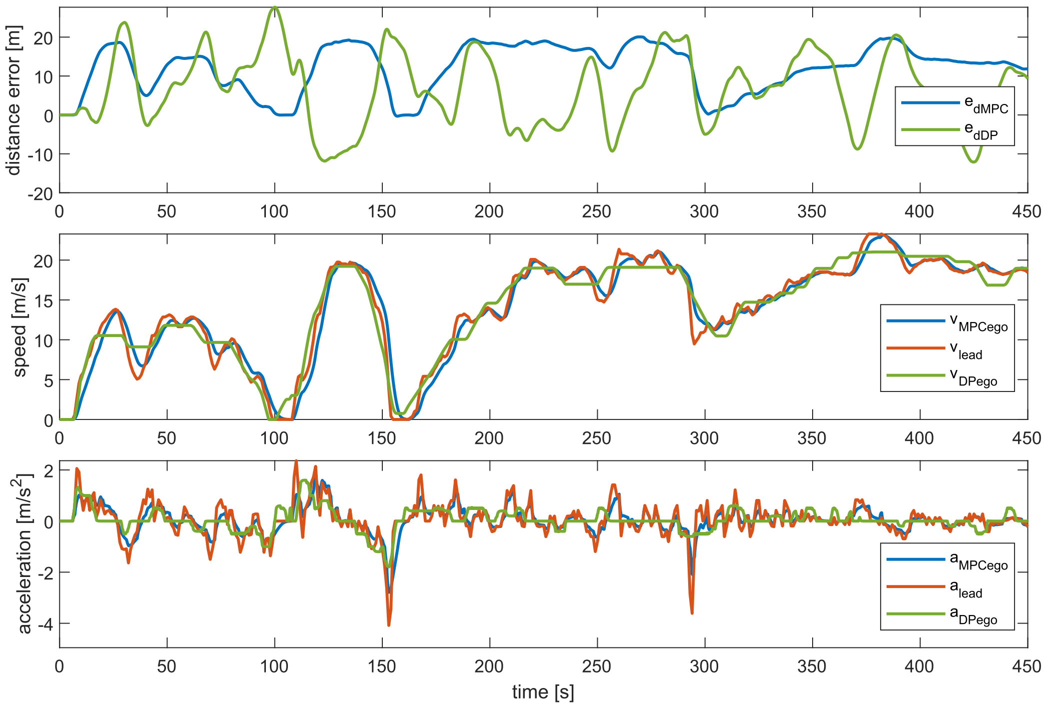

Figure 6. Comparison of distance error, velocity, and acceleration profiles of ego equipped with fuel consumption MPC and DP controllers with lead along ARDC driving cycle, [0, 450 s] zoom.

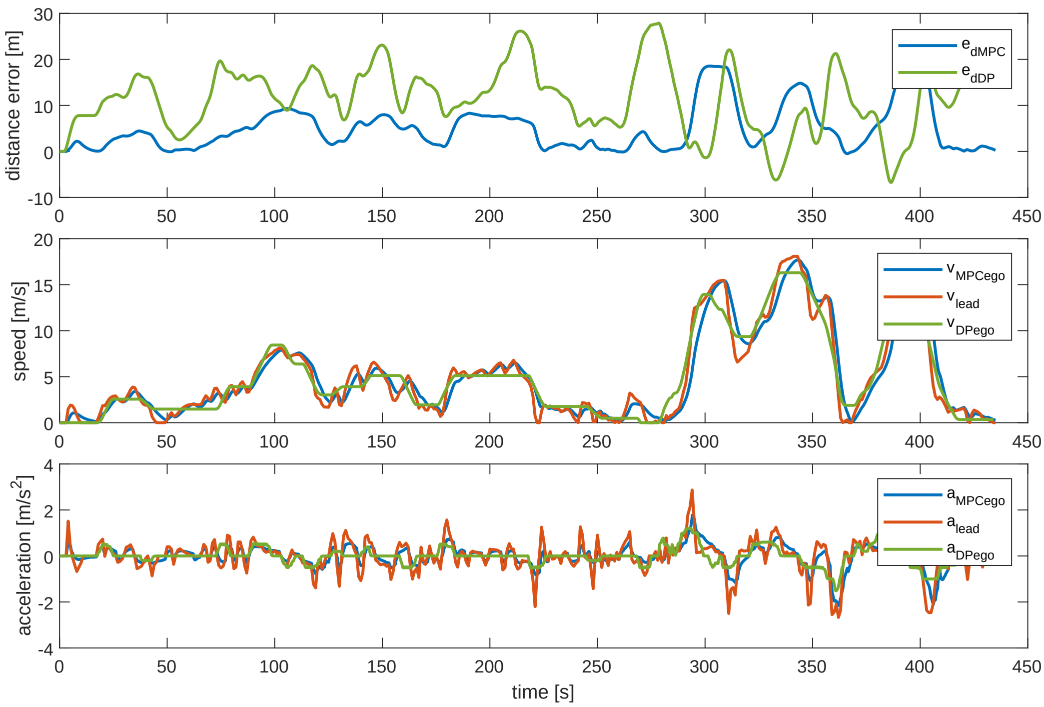

Figure 7. Comparison of velocity and acceleration profiles of ego equipped with fuel consumption MPC and DP controllers with lead along RD2 driving cycle, [0, 450 s] zoom.

5. Conclusions

This paper presents a comparative analysis of two model predictive control strategies aimed at optimizing adaptive cruise control systems for enhanced fuel efficiency and passenger comfort while ensuring safe driving behavior, with their key difference being the formulation of their cost functions:

- One strategy explicitly incorporates a fuel consumption term;
- The other indirectly addresses fuel efficiency by focusing on acceleration smoothing.

The two MPC strategies are also benchmarked against a dynamic programming controller. The results show that both MPC controllers yield comparable outcomes in terms of computational efficiency, making them suitable for real-time implementation. In terms of performance, the MPC_{FC}, which includes the explicit fuel consumption term, marginally outperforms the alternative strategy, though it operates with slightly slower computational speed. However, the performance improvements are minimal, calling into question the added complexity and effort required for implementing MPC_{FC}. Furthermore, despite both controllers having the same number of tunable parameters, the tuning process for MPC_{FC} was more time-consuming. Additionally, the development of MPC_{FC} involved several assumptions and approximations, such as maintaining a constant gear ratio over the prediction horizon and applying polynomial fitting to the fuel map, which may not always be readily available.

In conclusion, while MPC_{FC} achieved slight performance advantages, the demonstrated potential for both MPC strategies to reduce fuel consumption and improve passenger comfort makes them both promising candidates for future ACC systems. As the automotive industry progresses towards more sustainable and efficient mobility solutions, advanced control strategies like MPC will be crucial in achieving these objectives. Given the marginal differences in performance, a controller that is easier to tune and implement may prove more viable for commercial deployment. Nonetheless, further validation in more realistic simulation environments is essential to fully explore and refine this comparison. In future work, we plan to test the developed algorithms in a more complex simulation environment and with a higher modeling fidelity in order to properly capture the vehicle and powertrain dynamics as well as the interference due to traffic and unexpected maneuvers.

Author Contributions: Conceptualization, F.M. and A.B.; methodology, F.M. and A.B.; software, A.B.; formal analysis, F.M. and A.B.; investigation, A.B.; writing—original draft preparation, F.M. and A.B.; writing—review and editing, F.M. and D.A.M.; supervision, D.A.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 original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Brown, A.; Gonder, J.; Repac, B. An Analysis of Possible Energy Impacts of Automated Vehicles. In *Road Vehicle Automation*; Lecture Notes in Mobility; Springer International Publishing: Cham, Switzerland, 2014; pp. 137–153. [CrossRef]
2. Vahidi, A.; Sciarretta, A. Energy saving potentials of connected and automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 822–843. [CrossRef]
3. Benmimoun, M.; Pütz, A.; Zlocki, A.; Eckstein, L. euroFOT: Field Operational Test and Impact Assessment of Advanced Driver Assistance Systems: Final Results. In *Proceedings of the FISITA 2012 World Automotive Congress*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 537–547. [CrossRef]
4. Zlocki, A.; Themann, P. Methodology for quantification of fuel reduction potential for adaptive cruise control relevant driving strategies. *IET Intell. Transp. Syst.* **2014**, *8*, 68–75. [CrossRef]
5. Stern, R.E.; Cui, S.; Delle Monache, M.L.; Bhadani, R.; Bunting, M.; Churchill, M.; Hamilton, N.; Haulcy, R.; Pohlmann, H.; Wu, F.; et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 205–221. [CrossRef]
6. IEA (Ed.) *Global EV Outlook 2024: Moving Towards Increased Affordability*; IEA: Paris, France, 2024.
7. IEA (Ed.) *Global CO_2 Emissions by Sector, 2019–2022*; IEA: Paris, France, 2023.
8. Kopelias, P.; Demiridi, E.; Vogiatzis, K.; Skabardonis, A.; Zafiropoulou, V. Connected & autonomous vehicles—Environmental impacts—A review. *Sci. Total Environ.* **2020**, *712*, 135237. [CrossRef] [PubMed]

9. Xiao, L.; Gao, F. A comprehensive review of the development of adaptive cruise control systems. *Veh. Syst. Dyn.* **2010**, *48*, 1167–1192. [CrossRef]
10. Anselma, P.G. Optimization-Driven Powertrain-Oriented Adaptive Cruise Control to Improve Energy Saving and Passenger Comfort. *Energies* **2021**, *14*, 2897. [CrossRef]
11. Schmied, R.; Waschl, H.; Quirynen, R.; Diehl, M.; del Re, L. Nonlinear MPC for Emission Efficient Cooperative Adaptive Cruise Control. *IFAC-PapersOnLine* **2015**, *48*, 160–165. [CrossRef]
12. Ma, F.; Yang, Y.; Wang, J.; Li, X.; Wu, G.; Zhao, Y.; Wu, L.; Aksun-Guvenc, B.; Guvenc, L. Eco-driving-based cooperative adaptive cruise control of connected vehicles platoon at signalized intersections. *Transp. Res. Part D Transp. Environ.* **2021**, *92*, 102746. [CrossRef]
13. Weißmann, A.; Görges, D.; Lin, X. Energy-optimal adaptive cruise control combining model predictive control and dynamic programming. *Control Eng. Pract.* **2018**, *72*, 125–137. [CrossRef]
14. Luo, Y.; Chen, T.; Zhang, S.; Li, K. Intelligent Hybrid Electric Vehicle ACC with Coordinated Control of Tracking Ability, Fuel Economy, and Ride Comfort. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2303–2308. [CrossRef]
15. Li, S.; Li, K.; Rajamani, R.; Wang, J. Model predictive multi-objective vehicular adaptive cruise control. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 556–566. [CrossRef]
16. Vajedi, M.; Azad, N. Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 113–122. [CrossRef]
17. Musa, A.; Miretti, F.; Misul, D. MPC-Based Cooperative Longitudinal Control for Vehicle Strings in a Realistic Driving Environment. In *Proceedings of the SAE Technical Paper Series*; SAE International: Warrendale, PA, USA, 2023. [CrossRef]
18. Fiorillo, C.; Mauro, M.; Biswas, A.; Bonfitto, A.; Emadi, A. Designing a Real-Time Implementable Optimal Adaptive Cruise Control for Improving Battery Health and Energy Consumption in EVs through V2V Communication. *Energies* **2024**, *17*, 1986. [CrossRef]
19. Pan, C.; Huang, A.; Wang, J.; Chen, L.; Liang, J.; Zhou, W.; Wang, L.; Yang, J. Energy-optimal adaptive cruise control strategy for electric vehicles based on model predictive control. *Energy* **2022**, *241*, 122793. [CrossRef]
20. Borneo, A.; Zerbato, L.; Miretti, F.; Tota, A.; Galvagno, E.; Misul, D.A. Platooning Cooperative Adaptive Cruise Control for Dynamic Performance and Energy Saving: A Comparative Study of Linear Quadratic and Reinforcement Learning-Based Controllers. *Appl. Sci.* **2023**, *13*, 10459. [CrossRef]
21. Zhu, M.; Wang, Y.; Pu, Z.; Hu, J.; Wang, X.; Ke, R. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transp. Res. Part C Emerg. Technol.* **2020**, *117*, 102662. [CrossRef]
22. Acquarone, M.; Borneo, A.; Misul, D.A. Acceleration control strategy for Battery Electric Vehicle based on Deep Reinforcement Learning in V2V driving. In Proceedings of the 2022 IEEE Transportation Electrification Conference & Expo (ITEC), Anaheim, CA, USA, 15–17 June 2022. [CrossRef]
23. Borneo, A.; Miretti, F.; Acquarone, M.; Misul, D. *Battery Electric Vehicle Control Strategy for String Stability Based on Deep Reinforcement Learning in V2V Driving*; SAE Technical Paper Series; SAE International: Warrendale, PA, USA, 2023. [CrossRef]
24. Bertsekas, D.P. *Dynamic Programming and Optimal Control*; Athena Scientific: Nashua, NH, USA, 2012.
25. Nguyen, D.G.; Park, S.; Park, J.; Kim, D.; Eo, J.S.; Han, K. An MPC Approximation Approach for Adaptive Cruise Control With Reduced Computational Complexity and Low Memory Footprint. *IEEE Trans. Intell. Veh.* **2024**, *9*, 3154–3167. [CrossRef]
26. Lin, X.; Görges, D. Robust Model Predictive Control of Linear Systems With Predictable Disturbance With Application to Multiobjective Adaptive Cruise Control. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 1460–1475. [CrossRef]
27. Ostertag, E. *Mono- and Multivariable Control and Estimation: Linear, Quadratic and LMI Methods*; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
28. Selvaraj, D.; Hegde, S.; Amati, N.; Deflorio, F.; Chiasserini, C. An ML-Aided Reinforcement Learning Approach for Challenging Vehicle Maneuvers. *IEEE Trans. Intell. Veh.* **2023**, *8*, 1686–1698. [CrossRef]
29. Lin, Y.; McPhee, J.; Azad, N.L. Comparison of Deep Reinforcement Learning and Model Predictive Control for Adaptive Cruise Control. *IEEE Trans. Intell. Veh.* **2021**, *6*, 221–231. [CrossRef]
30. Li, L.; Wang, X.; Song, J. Fuel consumption optimization for smart hybrid electric vehicle during a car-following process. *Mech. Syst. Signal Process.* **2017**, *87*, 17–29. [CrossRef]
31. Li, S.E.; Guo, Q.; Xu, S.; Duan, J.; Li, S.; Li, C.; Su, K. Performance Enhanced Predictive Control for Adaptive Cruise Control System Considering Road Elevation Information. *IEEE Trans. Intell. Veh.* **2017**, *2*, 150–160. [CrossRef]
32. Luo, L.h.; Liu, H.; Li, P.; Wang, H. Model predictive control for adaptive cruise control with multi-objectives: Comfort, fuel-economy, safety and car-following. *J. Zhejiang Univ. Sci. A* **2010**, *11*, 191–201. [CrossRef]
33. Nie, Z.; Farzaneh, H. Adaptive Cruise Control for Eco-Driving Based on Model Predictive Control Algorithm. *Appl. Sci.* **2020**, *10*, 5271. [CrossRef]
34. Guzzella, L.; Sciarretta, A. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
35. Bageshwar, V.; Garrard, W.; Rajamani, R. Model Predictive Control of Transitional Maneuvers for Adaptive Cruise Control Vehicles. *IEEE Trans. Veh. Technol.* **2004**, *53*, 1573–1585. [CrossRef]
36. Wu, C.; Xu, Z.; Liu, Y.; Fu, C.; Li, K.; Hu, M. Spacing Policies for Adaptive Cruise Control: A Survey. *IEEE Access* **2020**, *8*, 50149–50162. [CrossRef]

37. Vogel, K. A comparison of headway and time to collision as safety indicators. *Accid. Anal. Prev.* **2003**, *35*, 427–433. [CrossRef]
38. Bertsekas, D.P. *Bertsekas*, 4th ed.; Athena Scientific: Belmont, MA, USA, 2017; Volume 1.
39. Miretti, F.; Misul, D.; Spessa, E. DynaProg: Deterministic Dynamic Programming solver for finite horizon multi-stage decision problems. *SoftwareX* **2021**, *14*, 100690. [CrossRef]
40. Hülsebusch, D.; Salfeld, M.; Xia, Y.; Gauterin, F. Adaptive Cruise Control: A Behavioral Assessment of Following Traffic Participants Due to Energy Efficient Driving Strategies. In *Proceedings of the FISITA 2012 World Automotive Congress*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 209–220. [CrossRef]
41. Hulsebusch, D.; Salfeld, M.; Ponomarev, I.; Gauterin, F. The impact of energy efficient driving strategies on rear-end safety. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013. [CrossRef]
42. Lofberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2–4 September 2004; pp. 284–289. [CrossRef]
43. Jerez, J.L.; Kerrigan, E.C.; Constantinides, G.A. A condensed and sparse QP formulation for predictive control. In Proceedings of the IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011. [CrossRef]
44. MOSEK ApS. *The MOSEK Optimization Suite. Version 10.2.1*; MOSEK ApS: København, Denmark, 2024.
45. Zeilinger, M.N.; Jones, C.N.; Morari, M. Robust stability properties of soft constrained MPC. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010. [CrossRef]