Slope Considered Online Nonlinear Trajectory Planning with Differential Energy Model for Autonomous Driving

Zhaofeng Tian[1], Lichen Xia[1], and Weisong Shi[1]

Abstract—**Achieving energy-efficient trajectory planning for autonomous driving remains a challenge due to the limitations of model-agnostic approaches. This study addresses this gap by introducing an online nonlinear programming trajectory optimization framework that integrates a differentiable energy model into autonomous systems. By leveraging traffic and slope profile predictions within a safety-critical framework, the proposed method enhances fuel efficiency for both sedans and diesel trucks by 3.71% and 7.15%, respectively, when compared to traditional model-agnostic quadratic programming techniques. These improvements translate to a potential $6.14 billion economic benefit for the U.S. trucking industry. This work bridges the gap between model-agnostic autonomous driving and model-aware ECO-driving, highlighting a practical pathway for integrating energy efficiency into real-time trajectory planning.**

Index Terms—**Trajectory Optimization, Energy Efficiency**

I. INTRODUCTION

In 2021, truck transportation in the US has covered a remarkable distance of 327.48 billion miles [1], with fuel costs representing about 30% of the Total Cost of Ownership (TCO) [2], [3]. Despite achieving basic safe operations, current studies and practices in Autonomous Driving (AD) trajectory planning have yet to provide substantial evidence of energy savings under an energy model agnostic framework. This study proposes a novel **energy-model-aware** planning paradigm that achieves online nonlinear trajectory optimization on a differentiable energy model.

In the commonly used AD trajectory generation method, cubic or polynomial curve generation, collision avoidance to both static and dynamic obstacles would be the top priority [4], [5], [6], [7], the energy-saving performance is usually considered a byproduct of trajectory smoothing on acceleration or jerk profile (general energy), instead of an explicit objective under an energy model agnostic frame. Although applicable to various vehicle types, using a general energy representation often lacks robust proof for energy optimality. In contrast, Ecological Driving (ECO-driving) specifically assists drivers in operating vehicles within a high-efficiency zone, providing promising insights for energy-efficient autonomous driving that leverages precise energy models (model-based) for trajectory optimization. To explore the potential of incorporating ECO-driving strategies, related works are investigated below.

Pressing and releasing the gas pedal, vehicles can Pulse and Glide (PnG), allowing the engine to operate at high-efficiency points [8]. In converting the PnG problem into a discrete Optimal Control Problem (OCP), variables such as engine rotation speed, torque, and gear ratio are considered. Study [9] solves the discrete OCP using an enumeration strategy and then addresses nonlinear problems. Beyond Internal Combustion Engines (ICE) vehicles, a Genetic Algorithm (GA) is applied for offline optimization of PnG in Electric Vehicles (EV), which has proven energy-efficient compared to constant speed driving [10]. However, real-time application and safety concerns remain challenges for PnG strategies.

Optimizing a speed profile given a global driving cycle and terrain prior, Eco-driving can achieve better energy efficiency. Some studies consider the improvement produced by taking the terrain info i.e. slope profile into vehicle control [11], [12], [13], [14]. Study [11] and [12] use the Dynamic Programming (DP) method to iteratively search an energy-saving speed profiling and gear selection over a lookahead elevation horizon for a heavy diesel truck, and could achieve a fuel consumption reduction of about 3.5% on the 120 km route without an increase in trip time. In contrast, studies [13], [14] avoid optimizing gear selection by using an approximate bivariate polynomial energy consumption function w.r.t. vehicle's velocity and apparent acceleration or power request. Then a slope-considered objective function is solved by establishing the Pontryagin's Maximum principle (PMP) optimal conditions for an OCP.

Several studies Consider the traffic information, either making the traffic average velocity the desired driving velocity or adjusting ego-car velocity with varying maximum velocity that is imposed by different traffic conditions. Study [15] integrates the traffic average speed in the objective function as the desired speed, using an approximated fuel model w.r.t. velocity and acceleration, The PMP method is applied to solve the OCP. Studies [16], [17] account for the speed limit changes due to varying traffic congestion, and Sequential Quadratic Programming (SQP) is used to optimize the energy consumption over a road slope profile for both an ICE vehicle and an EV. Furthermore, studies [18], [19] develop a two-stage method that initially determines a global speed profile and subsequently optimizes a local speed profile adapting to varying traffic flow speed, where Quadratic Programming (QP) and SQP are used to tackle quadratic and nonlinear problems defined for the local stage.

Exactly considering intervehicle distance-keeping instead of average traffic flow speed for safety critic requirements, the distance tracking error is formulated in the objective functions of Adaptive Cruise Control (ACC) system [20], [21], [22], [23]. Study [21] focuses on globally optimizing energy consumption for an electric vehicle (EV) by considering both car-following dynamics and road slope, using Pontryagin's Maximum Principle (PMP) to address the Optimal Control Problem (OCP). Initially, a safe-distance-unconstrained OCP is solved to circumvent the time-dependent boundary value

problem introduced by ACC. Subsequently, sections where the distance constraint is violated are identified, and a sub-OCP with a fixed end time is repeatedly solved. Study [23] optimize the eco-driving in a Model Predictive Control (MPC) framework, predicting slope profiling and leading car speed over a time horizon, with each timestep incorporating interve-hicle distance tracking error into the objective function. Using an approximate continuous ICE fuel model, the OSP over a prediction horizon is solved by computing PMP optimal conditions with a linear solver. Study [20] also integrates the ACC distance tracking error as an objective and formulates the DP method to optimize the energy consumption of a Hybrid Electric Vehicle (HEV) over a prediction horizon. Study [22] explores the merits of using SQP to optimize the energy of a HEV in an ACC condition, compared to the high computing costs of PMP, SQP achieves near-optimal results with higher efficiency.

From the experience of the abovementioned studies, we observe that by integrating a model-based energy savings strat-egy into AD trajectory planning, ECO-driving could provide a rich source of methodologies. However, several practical gaps exist when adapting ECO-driving techniques to real-world autonomous driving applications:

(i) Hardware Interfaces. Autonomous vehicles often rely on non-OEM components, meaning the autonomous driving controller may lack direct control over the engine and gear-box. Instead, a lower-level controller executes PID control on the Drive-By-Wire (DBW) systems for steering, gas, and brakes, as evidenced by DBW industry [24] and control commands such as the *AckermannControlCommand* used in Autoware [25]. Consequently, plans that involve direct manipulation of gear selection and torque or power requests [11], [12], [9] become impractical.

(ii) Onboard Real-time Long Horizon Prediction and Re-planning. Most onboard AD trajectory planning methodolo-gies necessitate frequent iterative predictions and replanning to adapt to rapidly changing conditions and ensure safety. Techniques such as Dynamic Programming (DP) or Pon-tryagin's Maximum Principle (PMP) are challenged by long prediction horizons and high replanning frequencies [16], [22]. Although some studies [22], [18] have utilized Sequential Quadratic Programming (SQP) and Quadratic Programming (QP) to achieve higher computing efficiency, both methods still struggle to meet real-time processing needs, with computation times ranging from 80–496 and 9–28 secondes [18].

(iii) Safety Critic. In autonomous driving operations, safety constraints should be strictly adhered to at all times. Studies that only consider average traffic flow speeds or speed limits as references [15], [17], [19] fail to implement stringent safety measures. Although other research incorporates ACC operations to include distance-keeping errors in the objective function [20], [21], [22], [23], the enforcement of hard safety constraints over the prediction horizon is generally lacking. Methods based on PMP struggle with the computational complexity of enforcing hard constraints, often resorting to incorporating safety measures into the cost function rather than

directly constraining state variables [21]. This approach, while reducing computational demand, potentially compromises the global safety guarantees [16].

To address the identified gaps in applying ECO-driving tech-niques to real-world autonomous driving, we present several key contributions:

(i) Offline Fuel Model Fitting and Gear Optimization Method. We propose a method that accurately translates dis-crete engine characteristics and fuel map data into a bivariate polynomial function w.r.t. velocity and traction acceleration. This approach uses a Non-Linear Programming (NLP) method to quickly solve min-square fitting problems. Unlike previous works [13], [23] that fit fuel data relative to velocity and apparent acceleration, this study leverages traction acceleration to provide a more accurate fuel rate estimate on rolling roads.

(ii) Online NLP-based Trajectory Optimization Method. We introduce a fast and near-optimal direct NLP optimization method under the ePAC framework that incorporates the en-ergy model and ACC safety constraints into every state along the trajectory, ensuring global safety. This method integrates horizon-flexible traffic-slope predictions into the optimization program, which enables iteratively replanning with real-time frequency on an embedded onboard device.

(iii) Comprehensive Testing. We conduct extensive tests comparing popular online trajectory optimization meth-ods such as Quadratic Programming (QP) and Sequential Quadratic Programming (SQP) across six driving cycles and three different road slope profiles in both a sedan and a truck. Our tests not only measure fuel consumption but also evaluate prediction horizon length, usage of slope prediction, and solving time, which are rarely emphasized previously.

(iv) Open-source and Reproducible Implementation. All our engineering programs are open-sourced (see Appendix), allow-ing our results to be reproduced without further tuning. This transparency provides a reliable benchmark for the community, fostering further research and development in this area.
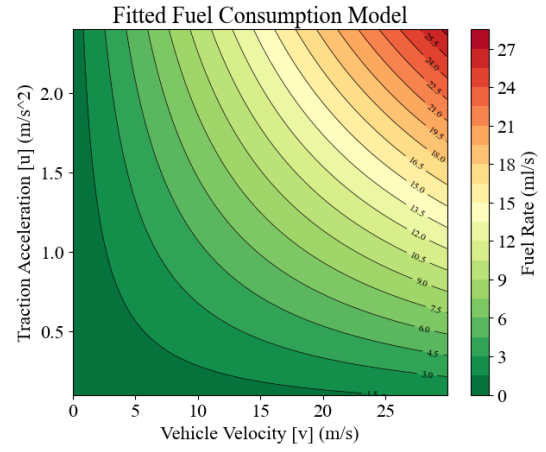
These initiatives aim to bridge the current gaps between ECO-driving and energy-model-aware autonomous driving. The rest of the paper is arranged as follows. Offline fuel mod-eling and gear optimization are introduced in Section II. The online Nonlinear Trajectory planning method is illustrated in section III. Experiments and Results are shown in Section IV. The conclusion is drawn in Section V.

## II. OFFLINE FUEL MODELING AND GEAR OPTIMIZATION

When observing the requirements for AD software plat-forms like Autoware, control commands sent to the vehicle are usually generalized kinematic signals like steering angle and the linear vehicle speed. For the sake of incorporating an energy model into the AD trajectory planning, the originally discrete engine fuel map w.r.t. torque, engine speed, and gear ratios, will need to be approximated by a function w.r.t. vehicle speed and acceleration, such that even without real-time torque feedback, the upper-level AD controller can achieve a model-based energy optimization.
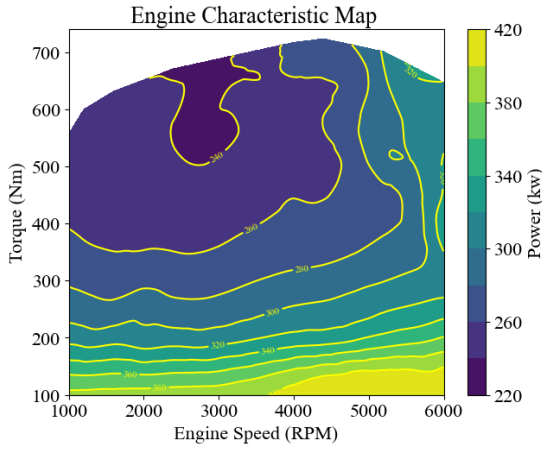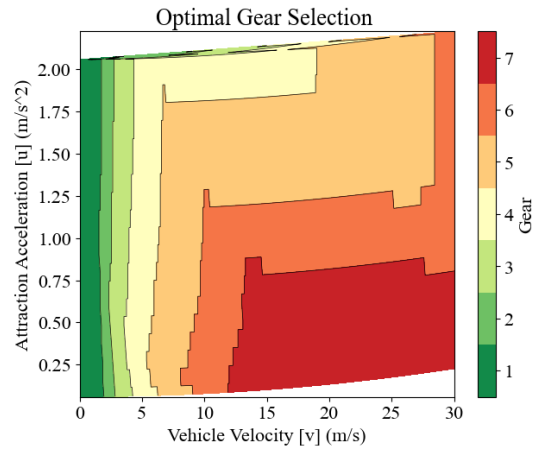
(a) Engine characteristic map
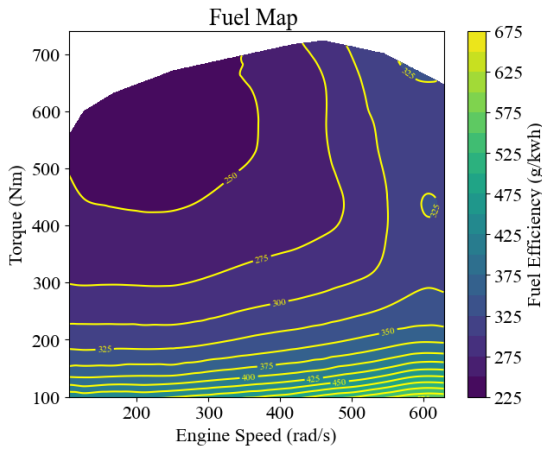
(b) Fuel efficiency map

Fig. 1: Engine characteristic and fuel efficiency maps of the diesel truck used in this study.

(a) Fitted fuel model

(b) Optimized gear selection

Fig. 2: Fitted fuel model and optimized gear selection with respect to u and v for the 7-speed truck.

A. Basic Dynamics

In this study, both a sedan and a truck are used for experiments and analysis, where we refer to study [13] for the sedan parameter settings, and we use a diesel truck's data to illustrate the modeling process since study [13] does not provide the original engine and fuel map data. Some necessary vehicle dynamic and fuel consumption formulas are introduced below for later fuel rate modeling.

$$k_1 = \frac{C_d \cdot \rho \cdot A_v}{2 \cdot M} \quad (1)$$

$$k_2 = \mu \cdot g, \quad k_3 = g \quad (2)$$

$$a_R = k_1 \cdot v^2 + k_2 \cdot cos(\theta(s)) + k_3 \cdot sin(\theta(s)) \quad (3)$$

$$\underbrace{u = a_V + a_R + a_B}_{\text{vehicle dynamic equation}} \quad (4)$$

$$\underbrace{v = v_0 + \int_{t_0}^{t_k} a\,\Delta t, \quad s = s_0 + \int_{t_0}^{t_k} v\,\Delta t}_{\text{vehicle kinematic equations}} \quad (5)$$

$$F_t = u \cdot M, \quad T_w = F_t \cdot r \quad (6)$$

$$T_e = \frac{T_w}{i_t \cdot \eta}, \quad \omega_e = \frac{v \cdot i_t}{r}, \quad i_t = i_g \cdot i_f \quad (7)$$

$$P_e = \mathcal{F}_P(\omega_e, T_e), \quad \eta_f = \mathcal{F}_{fe}(\omega_e, T_e) \quad (8)$$

$$f_r = \frac{P_e \cdot \eta_f}{c_u} \quad (9)$$

Where C_d, ρ, A_v, and M respectively denote air drag coefficient, air density frontal area, and equivalent mass; μ and g are the rolling resistance coefficient and gravity value; s, v, u, are the vehicle's path coordinate, velocity, and traction acceleration; a_V, a_R, a_B denote apparent; acceleration, resistance acceleration, and brake acceleration; Road slope/grade

is denoted by $\theta(s)$, which generally can be obtained from road profiling or GPS information from road coordinate s [13], [14].

To acquire the fuel rate, referring to Equation (6-9), engine torque T_e can be derived from wheel torque T_w with traction force F_t, wheel radius r, the transmission ratio i_t, and efficiency η_t, in which i_t equals to the product of gear ratio i_g and final drive ratio i_f, where traction force F_t is directly related to u; Engine speed ω_e can be calculated with vehicle speed v and transmission ratio i_t; By using two interpolated functions $\mathcal{F}_P$ and $\mathcal{F}_{fe}$, engine power P_e(w) and fuel efficiency η_f (g/kwh) can be quired form ω_e, T_e, as shown in Fig. 1a and 1b; A fuel rate fr at unit (ml/s) referring to study [13], can be derived from the product of P_e and η_f with a coefficient $c_u = \rho_g \cdot 1000 \cdot 3600$ that converts the consumption rate unit to (ml/s), where $\rho_g = 0.85$ (g/ml) represents the disel density.

B. Fuel Rate Modeling

Previous works tend to approximate the fuel consumption map into a bivariate quadralic [26] or polynomial [13], [23] functions with vehicle speed v and apparent acceleration a as two variables, achieving a differentiable numerical optimization. Considering the traction acceleration u would involve fuel consumption in a more direct way than apparent acceleration a as slope $\theta(\cdot)$ is also incorporated referring to equation (4), we fit the fuel model and optimize the gear w.r.t. v and u. With a similar fitting function [13] with a_V replaced by u in Equation (10).

$$\hat{f}_r(v, u) = o_0 + o_1 v + o_2 v^2 + o_3 v^3 + o_4 v^4 + (c_0 + c_1 v + c_2 v^2)u \tag{10}$$

To precisely fit a fuel rate function, the gear selection policy needs to be optimized in advance to eliminate the impact introduced by different engine working conditions caused by different gear selections, such that the optimized gear selection policy will let the engine work on high-efficiency points. The pseudocode of our implementation is described in Algorithm 1, where the returned $data\_list$ contains v, u sampling data with the determined optimal fuel rate and gear number opt_f_r and opt_gear. ω_e, T_e are bounded by $[62 \text{ (idling)}, 630]$ rad/s and $[0, 724]$ Nm.

With the optimized gear selection as Fig. 2b shows and v, u samples in $data\_list$, we now can design a min-square objective function that fits the parameters in Equation (10) as below to minimize the squared error over all samples:

$$J_{fit} = \min_{o_0,o_1,o_2,o_3,o_4,c_0,c_1,c_2} \sum_{i=1}^{N} \left(f_r^{(i)} - \hat{f}_r^{(i)} \right)^2 \tag{11}$$

where $f_r^{(i)}$ is the observed fuel rate in the i-th data tuple of $data\_list$, $\hat{f}_r^{(i)}$ is the estimated fuel rate using the model for the i-th data point, and N is the total number of data points. The min-square problem can be efficiently solved using the nonlinear optimization method IPOPT [27] within a Casadi [28] symbolic problem-setting interface. The fitted fuel estimated model can limit the average prediction error at 0.1047 (ml/s) on $data\_list$, especially for this truck, within the

more general working conditions $v \sim (5, 25)$ $u \sim (0.1, 1.0)$, the error will be controlled within 0.0596 (ml/s). The fuel rate w.r.t. u and v is mapped as Fig. 2a. The optimized parameters o_1-o_4, c_0-c_2 are recorded in Appendix.

Algorithm 1: Optimize Gear Selection

1 Initialize $data\_list \leftarrow \emptyset$;
2 **for** *each v in v_range* **do**
3   **for** *each a in a_range* **do**
4   Compute u, F_t, T_w (4) (6);
5   **for** *each gear in gear_list* **do**
6   Calculate T_e, ω_e, with gear ratio (7) ;
7   Calculate P_e and f_r (8) (9);
8   **if** $f_r < opt\_fr$ **then**
9          $opt\_f_r \leftarrow f_r$, $opt\_gear \leftarrow gear$;
10   **end**
11   **end**
12   **if** $opt\_gear \neq 0$ **then**
13   Append $(v, u, opt\_f_r, opt\_gear)$ to $data\_list$;
14   **end**
15   **end**
16 **end**
17 **return** $data\_list$;

III. NONLINEAR ONLINE TRAJECTORY OPTIMIZATION

In this section, we present an online trajectory optimization framework that considers the prediction of the leading vehicle's trajectory and the road slope profile over the prediction time horizon. Also, the proposed framework allows both convex/linear-quadratic and nonconvex/nonlinear problem setups without extra effort. For the convex setting, we set up a QP problem for the general-energy based trajectory planning method that is widely used in AD studies [7], [6]; For the nonconvex setting, we directly optimize the objective on the fuel rate model, and subsequently, an SQP and NLP settings are adapted to tackle the challenges posed by the highly-nonlinear fuel rate estimate model defined in Equation (9).

A. Traffic and Road Profile

In this study, we focus on trajectory planning in the longitudinal dimension, where the traffic information is considered as the leading car profile, such the ego agent should plan a safe trajectory to keep an appropriate distance from the leading car to avoid potential collisions. This study define the leading car profile as a driving cycle [21], [20], including travel distance profile, speed profile, and acceleration profile. The leading agent and eco-agent states can be modeled as follows.

$$\underbrace{x_l = [s_l, v_l, a_{Vl}]}_{\text{leading agent states}}, \quad \underbrace{x = [s, v, u, a_V, a_B]}_{\text{ego agent states}} \tag{12}$$

$$\mathbf{X} = \left\{ x^{t_i} = \begin{bmatrix} \mathbf{S} = \{s^{t_i}\} \\ \mathbf{V} = \{v^{t_i}\} \\ \mathbf{U} = \{u^{t_i}\} \\ \mathbf{A} = \{a_V^{t_i}\} \\ \mathbf{B} = \{a_B^{t_i}\} \end{bmatrix} \middle| i \in [0, N_T], t_{i+1} - t_i = \Delta t \right\}$$
(13)

Where the leading agent state is defined as a 3-tuple, and the ego agent consists of 5-tuple states since dynamic states like traction and brake acceleration u, and a_B are also included (12). Given a prediction horizon H with a second unit, the discretized timestep total number $N_T = \text{int}(\frac{H}{\Delta t})$ with a minimum time granularity Δt, the predicted trajectory $\mathbf{X}$ of the ego car consists of the $\mathbf{S}, \mathbf{V}, \mathbf{U}, \mathbf{A}, \mathbf{B}$ trajectories of their respective variable over the prediction horizon as Equation (13) shows. Similarly, the predicted leading car trajectory $\mathbf{X_l}$ can be formulated with its state trajectories $\mathbf{S_l}, \mathbf{V_l}, \mathbf{A_l}$ correspondingly. The intervehicle safety ACC constraint can be formulated by the below.

$$d_{max} > \mathbf{S_l} - (\mathbf{S} + t_h \cdot \mathbf{V}) > d_{min} \quad \text{(ACC constraints)} \quad (14)$$

Where t_h is a time headway, d_{min} is the minimum distance that two agents should keep at zero-speed conditions, whereas d_{max} is also imposed to constrain the travel distance of two cars within a proximate range, and facilitate a consistent basis for the comparison of different methods. For road profiling, we leverage a mixed trigonometric equation to formulate both slope and elevation profiles shown below.

$$\theta(s) = \theta_0 + \sum_{i=1}^{N} a_{gi} \cdot sin(\frac{2\pi s}{l_{wi}}) \quad (15)$$

$$h = h_0 + \int_{s_s}^{s_e} \tan\theta(s)\, ds \quad (16)$$

Road slope profiles are formulated along road longitudinal coordinate s by a static slope constant to simulate constant up or down hills, and a N-component mixture of $sin(\cdot)$ waves that i-th wave is parameterized by an amplitude a_{gi} and a wavelength l_{wi}; The elevation is profiled with an initial constant h_0 and integration of $tan\,\theta(s)$ along the s.

B. Optimization Problem Setup

As mentioned above, we set up the trajectory planning problem as an optimization problem in three ways, i.e., QP, SQP, and NLP, which will be illustrated subsequently. Some commonly used kinematic constraints are formulated first.

$$\begin{cases} \mathbf{S}[i+1] = \mathbf{S}[i] + \mathbf{V}[i]\,\Delta t + \frac{1}{2}\,\mathbf{A}[i]\,\Delta t^2 \\ \mathbf{V}[i+1] = \mathbf{V}[i] + \mathbf{A}[i]\,\Delta t \quad \text{(kinematic constraints)} \end{cases} \quad (17)$$

Dynamic constraints should also be considered for SQP and NLP settings, which can be formulated as below.

$$\begin{cases} \mathbf{R} = k_1 \cdot \mathbf{V}^2 + k_2 \cdot \cos(\mathbf{G}) + k_3 \cdot \sin(\mathbf{G}) \\ \mathbf{U} = \mathbf{A} + \mathbf{R} + \mathbf{B} \quad \text{(dynamic constraints)} \end{cases} \quad (18)$$

Where $\mathbf{R}$ is a trajectory array of resistance acceleration a_R, and $\mathbf{G}$ is a predicted road slope profile shown below.

$$\mathbf{G} = \{\theta(s^{t_i}) \,|\, i \in [0, N_T]\} \quad (19)$$

1) QP: A QP problem is set up for the general-energy based objectives that are popularly used in autonomous driving trajectory planning.

$$\begin{aligned} \min_{\mathbf{A}} \quad & J = w_1 \cdot \|\mathbf{V_l} - \mathbf{V}\|_2^2 + w_2 \cdot \|\mathbf{A}\|_2^2 \\ \text{s.t.} \quad & \text{Constraints (14) (17)} \\ & \text{and} \quad \mathbf{V}, \mathbf{A} \quad \text{bounds} \end{aligned} \quad (20)$$

Where w_1, w_2 are the weights of two objectives, and the first item is the velocity reference objective expecting the ego agent could keep a similar velocity with the leading agent. Making the reference speed the maximum speed could force the value of the first item to be comparatively too large when the leading is running in a low-speed zone, such that the ACC safety would solely rely on the hard constraints, and the fuel economy would be harmed. The second item is the general energy objective aimed at minimizing the acceleration costs to improve both smoothness and fuel economy.

2) SQP and NLP: SQP converts a nonlinear/nonconvex problem into iterative convex problems that can be solved by QP, while a direct NLP method can handle the nonlinear/nonconvex objectives and dynamics without convex approximation like SQP. The original nonlinear problem setting tackled by a direct NLP method is expressed below.

$$\begin{aligned} \mathbf{F}(\mathbf{V}, \mathbf{U}) &= o_0 + o_1\mathbf{V} + o_2\mathbf{V}^2 + o_3\mathbf{V}^3 + o_4\mathbf{V}^4 \\ &\quad + (c_0 + c_1\mathbf{V} + c_2\mathbf{V}^2) \cdot \mathbf{U} \\ &= \{\hat{f_r}^{t_i}(\cdot) \,|\, i \in [0, N_T]\} \end{aligned} \quad (21)$$

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{B}} \quad & J = w_1 \cdot \|\mathbf{V_l} - \mathbf{V}\|_2^2 + w_2 \cdot \|\mathbf{A}\|_2^2 \\ & \quad + w_2 \cdot \|\mathbf{B}\|_2^2 + w_3 \cdot \|\mathbf{F}(\cdot)\|_1 \\ \text{s.t.} \quad & \text{Constraints (14) (17) (18)} \\ & \text{and} \quad \mathbf{V}, \mathbf{A}, \mathbf{U}, \mathbf{B} \quad \text{bounds} \end{aligned} \quad (22)$$

Where the 1-norm of fuel rate trajectory $\mathbf{F}(\cdot)$ is integrated into the objective function referring to Equation (21)(22) as a model-aware paradigm, and a new control variable $\mathbf{B}$ is introduced in objectives to comply with vehicle dynamics. To leverage the SQP method, the original NLP problem setting needs to be approximated to a QP setting by Taylor Expansion such that the QP solver can iteratively solve the approximated QP problems. Where the nonlinear fuel rate trajectory array $\mathbf{F}$ needs to be approximate by $\tilde{\mathbf{F}}$, a quadratic or linear form based on the Taylor Expansion on the reference trajectory $\bar{\mathbf{V}}, \bar{\mathbf{U}}$. Note that in the expression below, we omit the subscripts in partial derivatives e.g., $\frac{\partial \mathbf{F}}{\partial \mathbf{V} \partial \mathbf{U}}|_{\bar{\mathbf{V}}, \bar{\mathbf{U}}}$.

$$\begin{aligned} \tilde{\mathbf{F}}(\mathbf{V}, \mathbf{U}) &= \mathbf{F}(\bar{\mathbf{V}}, \bar{\mathbf{U}}) + \frac{\partial \mathbf{F}}{\partial \mathbf{V}}(\mathbf{V} - \bar{\mathbf{V}}) + \frac{\partial \mathbf{F}}{\partial \mathbf{U}}(\mathbf{U} - \bar{\mathbf{U}}) \\ &\quad + \frac{\partial^2 \mathbf{F}}{\partial \mathbf{V} \partial \mathbf{U}}(\mathbf{V} - \bar{\mathbf{V}})(\mathbf{U} - \bar{\mathbf{U}}) \\ &\quad + \frac{1}{2}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{U}^2}(\mathbf{U} - \bar{\mathbf{U}})^2 + \frac{1}{2}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{V}^2}(\mathbf{V} - \bar{\mathbf{V}})^2 \end{aligned} \quad (23)$$

Moreover, on the constraint side, the QP setting requires a linear state transition, consequently $\mathbf{R}$ needs to be replaced by an approximation on reference $\bar{\mathbf{V}}$ that $\tilde{\mathbf{R}} = k_1(\bar{\mathbf{V}}^2 + 2\bar{\mathbf{V}}(\mathbf{V} - \bar{\mathbf{V}})) + (...)$. Then given an initial reference $\bar{\mathbf{V}}, \bar{\mathbf{U}}$ guess, SQP gets a warm start and iteratively updates the reference with the solution solved last iteration, such the whole optimization process resembles a Newton method. Note although jerk is not included in the optimization for generalizability when updating the agent's states, jerk is hard constrained within $1 m/s^3$.

IV. EXPERIMENTS
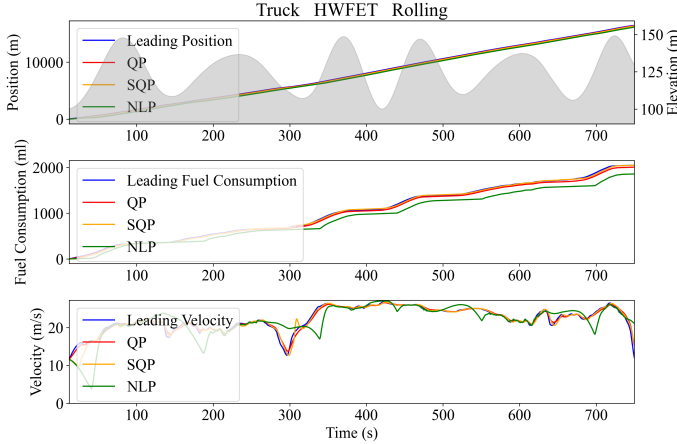
(a) An experiment running on "HWFET-Rolling" profile.
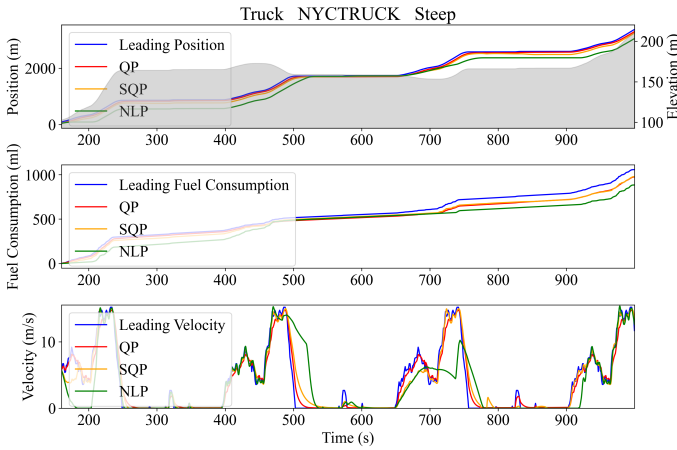
(b) An experiment running on "NYC_TRUCK-Steep" profile.

Fig. 3: Example experiments, in the velocity graph, the leading agent in the blue line represents the standard driving cycle speed profile, and shaded fill-in denotes the elevation of the leading agent w.r.t. time coordinate. Fuel consumption of QP, SQP, and NLP agents are compared by lines in different colors, where NLP agents could achieve lower consumption.

A. Online Trajectory Optimization Framework

We follow the previous studies [16], [20] to set an MPC-based online trajectory replanning framework that at each time step. Our proposed framework not only integrates a linear

prediction of the leading car with $[s_l, v_l, a_{Vl}]$ at the current time step over a time horizon $H = N_T \cdot \Delta t$, but also predicts a $\theta(\cdot)$ trajectory over H. Adapting to AD online applications, slope profile is predicted w.r.t. time t instead of s used in the previous studies [16], [18]. Our proposed online trajectory framework is illustrated in Algorithm 2.

Algorithm 2: Online Trajectory Optimization

---

1 Initialize $\mathbf{G}, \mathbf{X_l}, \mathbf{X}, \mathbf{U} \leftarrow \emptyset$
2 Slover.$type \in \{NLP, SQP, QP\}$
3 **for** t *in* $\{t_i \mid i \in [0, k], t_{i+1} - t_i = \Delta t\}$ **do**
4     $x_l^t, x^t \leftarrow UpdateStates(t)$
5     $\mathbf{X_l} \leftarrow PredictLeading(x_l)$ (5), $\mathbf{X_l}[0] = x_l^t$;
6   **if** $\mathbf{X} = \emptyset$ **then**
7         $d_{init} \leftarrow s_l^t - s^t$, s.t. $s_l^t \in x_l^t \quad s^t \in x^t$
8         $\mathbf{G} \leftarrow GetSlopeProfile(\mathbf{S_l} - d_{init})$, $\mathbf{S_l} \in \mathbf{X_l}$
9   **else**
10         $\mathbf{S_G} \leftarrow Concat(\mathbf{S}[1 : N_T], \mathbf{S}[N_T])$
11         $\mathbf{G} \leftarrow GetSlopeProfile(\mathbf{S_G})$
12   Solver.$Update(X_l, G, x(t))$
13     $\mathbf{X} \leftarrow$ Solver.$Solve()$, s.t. (17)(18)(14), $\mathbf{X}$ limits
14   **if** *Solver.type* = *QP* **then**
15   Execute $\mathbf{A}[0]$
16   **else**
17   Execute $\mathbf{U}[0]$ and $\mathbf{B}[0]$
18   Agent states transition.

In line 5, we use the current leading agent states to linearly predict their trajectory by the kinematic equation (5). To predict the slope from lines 6-11, we first initialize a slope reference with s_l and an initial intervehicle distance d_{init}, then update this reference by rolling out the solved ego trajectory X. With this s reference, slope profile $\mathbf{G}$ can be precisely predicted w.r.t. time. Note that in line 13, since the state variables are overlapped with the control variables, we would return the whole state trajectory. In lines 15 and 17, different control commands for different settings are executed with its first element based on an MPC policy, then the program will update agent states and restart the loop.

B. Experiment Setup

To comprehensively test our proposed online trajectory optimization framework with three methods "QP", "SQP", and "NLP", we build up a testbed consisting of six driving cycles and three slope profiles. Two highway scenarios including "HWFET", "INDIA HWY", and four urban scenarios including "INDIA URBAN", "NYCC", "NYC TRUCK", "MANHATTAN" driving cycles are incorporated; By using Equation (15), "flat","rolling", and "steep" road profiles are set up with $(\theta_0 = 0, a_g = 0)$, $(\theta_0 = 0, a_{gi} \in \{0.04, 0.02\}, l_{wi} \in \{2870, 2136\})$, and $(\theta_0 = 0.02, a_{gi} \in \{0.05, 0.02, 0.01\}, l_{wi} \in \{2380, 1860, 1430\})$. An example of individual experiments running on a single driving cycle and road profile is shown in Fig. 3. Comprehensive tests are conducted with six-cycle,

TABLE I: Comprehensive Tests

Agent ID	Travel Time (s)	Travel Distance (m)	Fuel Consumption (ml)	Average Fuel Rate (ml/s)	Average Fuel Efficiency (L/100km)	Average Speed (m/s)	Fuel Efficiency Improvement (%)	Average Solving Time A (ms)	Average Solving Time B (ms)
Truck-L	13477.20	123177.83	20525.89	1.5230	16.6636	9.1397	0.00	-	-
Truck-QP-5	13477.20	123058.27	18861.42	1.3995	15.3272	**9.1308**	8.02	1.1828	4.4574
Truck-SQP-5	13477.20	121924.53	19398.65	1.4394	15.9104	9.0467	4.52	5.3067	20.0371
Truck-NLP-5	13477.20	119981.86	17074.13	1.2669	14.2306	8.9026	**14.60**	10.9702	40.7752
Truck-QP-10	13477.20	122871.33	19497.51	1.4467	15.8682	9.1170	4.77	1.6621	-
Truck-SQP-10	10942.50	104411.57	16726.72	1.5286	16.0199	9.54	3.86	8.6851	-
Truck-NLP-10	13477.20	119677.71	17439.35	1.2940	14.5719	8.8800	**12.55**	18.5562	-
Truck-QP-5F	13477.20	123058.27	18861.42	1.3995	15.3272	**9.1308**	8.02	1.1263	-
Truck-SQP-5F	4913.30	75300.29	11227.98	2.2852	14.9109	15.3258	10.52	4.7907	-
Truck-NLP-5F	13477.20	119987.92	17079.31	1.2673	14.2342	8.9030	**14.58**	10.8604	-
Sedan-L	13477.20	123177.83	8329.70	0.6181	6.7623	9.1397	0.00	-	-
Sedan-QP-5	13477.20	122265.88	7814.52	0.5798	6.3914	9.0721	5.49	1.1952	4.4082
Sedan-SQP-5	13477.20	120890.79	7867.88	0.5838	6.5083	8.9700	3.76	5.3042	20.1379
Sedan-NLP-5	13477.20	119719.42	7368.26	0.5467	6.1546	8.8831	**8.99**	10.4392	39.2654
Sedan-QP-10	13477.20	122912.78	7996.89	0.5934	6.5061	**9.1201**	3.79	1.6399	-
Sedan-SQP-10	13477.20	120802.82	7878.56	0.5846	6.5218	8.9635	3.56	9.0283	-
Sedan-NLP-10	13477.20	119437.12	7432.23	0.5515	6.2227	8.8622	**7.98**	18.5075	-
Sedan-QP-5F	13477.20	122265.88	7814.52	0.5798	6.3914	9.0721	5.49	1.1279	-
Sedan-SQP-5F	2213.70	48671.63	2935.05	1.3259	6.0303	21.99	10.82	6.1508	-
Sedan-NLP-5F	13477.20	119723.33	7368.78	0.5468	6.1548	8.8834	**8.98**	10.5095	-

three-road, two-vehicle, and three-method settings. Besides the common compared metrics like travel distance, fuel consumption, and fuel efficiency, this study also evaluates the energy-saving policy on the following factors. *(i) Planning and Prediction Horizon Length.* The planned trajectory for the ego agent and the prediction horizon of the traffic-slope profile share the same length, 5 and 10 seconds horizon (50 and 100 steps with Δt set as 0.1 second) are selected to compare which is reflected in the agent name in Table I. Horizons shorter like 2 seconds have been tested to have inefficient results such that it is skipped due to page limits. *(ii) Slope Prediction.* We also analyze the impact of usage of slope prediction or not, where only the current slope of the ego agent $\theta(s^{t_0})$ will be used for planning if not using a slope prediction $[\theta(s^{t_0}), \theta(s^{t_1}), ..., \theta(s^{t_{N_T}})]$. This is reflected in Table I where the agent name contains an "F" to denote a false boolean for using slope prediction. In Table I, original $6 \times 3 \times 2 \times 3 \times 2(\text{horizon}) \times 2(\text{slope prediction}) = 432$ tests are incorporated in a compact demonstration that results from various cycles and road profiles are summed and combined to do a total analysis due to page limits. QP, SQP, and NLP methods are implemented in Python with the OSQP [29] and IPOPT [27], within Casadi [28] symbolic expressions. Solving time tests are conducted on: A. Windows PC i7 CPU and B. Ubuntu Nvidia Xavier (embedded machine), and the corresponding solving time is also reflected in Table I. Weights for each method are preoptimized by extensive tests and are listed in the Appendix.

C. Results Analysis

1) Planning-Predction Horizon and Usage of Slope Prediction: The results show that a medium horizon of 5 seconds could outperform 10 seconds and 2 seconds (not shown) for both vehicles and all methods, where the model accuracy

TABLE II: NLP vs QP

Agent ID	Speed Lose (%)	Fuel Efficiency Improvement(%)	Solving Time Multiples A
Sedan-QP-5	0.00	0.00	1.00
Sedan-NLP-5	2.08	3.71	8.73
Truck-QP-5	0.00	0.00	1.00
Truck-NLP-5	2.49	**7.15**	9.28

is satisfied. Using slope prediction would slightly improve fuel efficiency but not trivial for NLP (slope not included in QP), however, missing the slope prediction could cause SQP a higher inaccuracy leading to safety constraints violations, which is reflected in the gray-marked rows (some SQP tests failed due to constraint violations).

2) Solving Time: All implemented methods can be solved efficiently on both devices, where SQP could run fast iterations with QP solver except for some fails, NLP takes 8-9 times of solving time of QP in Table II, while still maintaining an on-board (embedded device B) and online solving capability with a computation frequency of about 25 Hz. We believe this could satisfy an AD operation under normal working conditions. Device B requires approximately 3.76 times the computing time of Device A.

3) Velocity-Energy Efficiency Tradeoff: Here we compare the proposed NLP method with the QP method (model-aware vs model-agnostic) on speed loss and fuel efficiency improvement in Table II shows that both vehicles could improve fuel efficiency without significantly reducing average speed. Specifically, the truck achieves a more substantial improvement of 7.15% in energy savings. Given the total U.S. trucking industry's annual 327.48 billion trips [1], such an improvement could potentially generate $6.14 billion in revenue.

V. Conclusion

This study introduces a direct NLP trajectory optimization method within an energy-model-aware paradigm, effectively addressing the limitations observed in model-agnostic trajectory planning approaches. By incorporating a precise energy model directly into the trajectory planning, this method not only ensures enhanced safety during ACC operations but also demonstrates a significant increase in energy efficiency for both a tested sedan and truck. Specifically, the truck achieves a 7.15% improvement in fuel saving ($6.14 billion equivalently) compared to the conventional QP model-agnostic method prevalent in AD research, without substantial loss of time efficiency, highlighting the method's practical applicability.

Appendix

TABLE III: Vehicle Parameters

Param	Sedan	Truck	Param	Sedan	Truck
M	1200	4800	o_1	1.0254e-2	9.0901e-3
A_v	2.5	2.5	o_2	-9.2812e-4	3.7574e-8
ρ	1.184	1.184	o_3	2.154e-5	3.4935e-8
C_d	0.32	0.6	o_4	-4.2427e-7	2.4230e-4
μ	0.015	0.006	c_0	0.07224	1.6550e-1
g	9.81	9.81	c_1	0.09681	3.6070e-1
o_0	1.4627e-1	3.351e-1	c_2	1.0750e-3	2.4223e-4

TABLE IV: Optimzation Parameters

Param	QP	SQP	NLP	Param	Sedan	Truck
w_1	0.1	0.1	0.1	d_{init}	50	50
w_2	2	5	5	v_{max}	30	27
w_3	-	2	10	$a_{v\,max}$	2.0	2.0
d_{min}	10	10	10	b_{max}	5.0	5.0
d_{max}	100	200	100	u_{max}	9.0	3.0

Sedan fuel model variables in [13] are converted to v, u. View codes at this link.

References

[1] American Trucking Associations, "Economics and industry data," 2021, accessed: 2024-04-18. [Online]. Available: https://www.trucking.org/economics-and-industry-data

[2] L. Liu, W. Li, D. Wang, Y. Wu, R. Yang, and W. Shi, "Fuel rate prediction for heavy-duty trucks," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[3] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[4] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939–960, 2008.

[5] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4889–4895.

[6] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2061–2067.

[7] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 987–993.

[8] W. Cao, T. Yuno, and T. Kawabe, "Pulse and glide strategy analysis based on engine operating point during pulse mode," *European Journal of Control*, vol. 65, p. 100629, 2022.

[9] S. Xu, S. E. Li, X. Zhang, B. Cheng, and H. Peng, "Fuel-optimal cruising strategy for road vehicles with step-gear mechanical transmission," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3496–3507, 2015.

[10] Z. Tian, L. Liu, and W. Shi, "A pulse-and-glide-driven adaptive cruise control system for electric vehicle," *International Transactions on Electrical Energy Systems*, vol. 31, no. 11, p. e13054, 2021.

[11] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[12] E. Hellström, J. Åslund, and L. Nielsen, "Design of an efficient algorithm for fuel-optimal look-ahead control," *Control Engineering Practice*, vol. 18, no. 11, pp. 1318–1327, 2010.

[13] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Ecological vehicle control on roads with up-down slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, 2011.

[14] J. Ma, J. Hu, E. Leslie, F. Zhou, P. Huang, and J. Bared, "An eco-drive experiment on rolling terrains for fuel consumption optimization with connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 125–141, 2019.

[15] C. R. He, H. Maurer, and G. Orosz, "Fuel consumption optimization of heavy-duty vehicles with grade, wind, and traffic information," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 6, p. 061011, 2016.

[16] A. Hamednia, N. K. Sharma, N. Murgovski, and J. Fredriksson, "Computationally efficient algorithm for eco-driving over long look-ahead horizons," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 7, pp. 6556–6570, 2021.

[17] A. Hamednia, N. Murgovski, and J. Fredriksson, "Time optimal and eco-driving mission planning under traffic constraints," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

[18] H. Lim, W. Su, and C. C. Mi, "Distance-based ecological driving scheme using a two-stage hierarchy for long-term optimization and short-term adaptation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1940–1949, 2016.

[19] X. T. Yang, K. Huang, Z. Zhang, Z. A. Zhang, and F. Lin, "Eco-driving system for connected automated vehicles: Multi-objective trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7837–7849, 2020.

[20] Y. Luo, T. Chen, S. Zhang, and K. Li, "Intelligent hybrid electric vehicle acc with coordinated control of tracking ability, fuel economy, and ride comfort," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2303–2308, 2015.

[21] D. Shen, D. A. Karbowski, and A. Rousseau, "Highway eco-driving of an electric vehicle based on minimum principle," in *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2018, pp. 1–8.

[22] M. Vajedi and N. L. Azad, "Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 113–122, 2015.

[23] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "On board eco-driving system for varying road-traffic environments using model predictive control," in *2010 IEEE International Conference on Control Applications*. IEEE, 2010, pp. 1636–1641.

[24] Dataspeed Inc., "Adas by-wire system," https://www.dataspeedinc.com/adas-by-wire-system/, n.d., accessed: 2024-04-03.

[25] Autoware Foundation, "Autoware," https://github.com/autowarefoundation/autoware, 2024, accessed: 2024-04-03.

[26] A. A. Malikopoulos and J. P. Aguilar, "Optimization of driving styles for fuel economy improvement," in *2012 15th international IEEE conference on intelligent transportation systems*. IEEE, 2012, pp. 194–199.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[28] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[29] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.