Energy-Efficient Adaptive Cruise Control for BEVs in Urban Scenarios with Traffic Lights Negotiation

Energy-Efficient Adaptive Cruise Control for BEVs in Urban Scenarios with Traffic Lights Negotiation*

Chengyang Ye[1], **Stefano Favelli**[1,2], **Andrea Tonoli**[1,2]

[1]Dipartimento di Ingegneria Meccanica e Aerospaziale (DIMEAS), Politecnico di Torino
[2]Center for Automotive Research and Sustainable Mobility (CARS), Politecnico di Torino

***Abstract*—Optimizing energy consumption for small electric vehicles in urban areas is crucial for enhancing travel efficiency and environmental sustainability. This study presents a longitudinal vehicle dynamics controller based on Model Predictive Control (MPC) that integrates vehicle following and traffic light information to minimize energy consumption and optimize comfort. The proposed MPC dynamically adjusts the headway distance to the preceding vehicle, achieving efficient energy management in urban traffic conditions by leveraging the knowledge of the electric motor map. The impact of knowing the lead vehicle trajectory through Vehicle-to-Vehicle (V2V) communication versus its estimation is discussed. The integration of traffic light status through Vehicle-to-Infrastructure (V2I) communication is also introduced. The Signal Phasing and Timing (SPaT) information is assumed to be available, enabling the vehicle to anticipate traffic signals and adjust speed proactively. This paper details the design principles, algorithm implementation, and simulation results. The experiments demonstrate that the proposed controller reduces the energy consumption of a battery electric vehicle (BEV) during urban operation, providing a safe and reliable driving experience.**

***Index Terms*—Model Predictive Control, Adaptive Cruise Control, Vehicle-to-Infrastructure, Energy Efficiency**

I. Introduction

The rapid urbanization and increasing environmental concerns have driven the demand for efficient and sustainable transportation solutions. Small electric vehicles are becoming a popular choice for urban commuting due to their low emissions and cost-effectiveness. However, optimizing energy consumption remains a critical challenge for enhancing the overall efficiency and practicality of these vehicles in complex urban environments.

The significance of developing an advanced vehicle controller lies in its potential to address this challenge. Extensive research has been conducted on energy-efficient vehicle speed profiles for traffic lights negotiation [1] [2] [3] [4]. Fast acceleration or deceleration to the desired cruising speed helps save energy. Traffic light information can determine the cruising speed to avoid stopping, thus reducing energy consumption. In this paper, we reference these ideas by integrating the eco-approach and departure with classical vehicle-following theory. In [5] a complete review of the policies adopted for adaptive cruise control (ACC) is addressed, while [6] offers the necessary background on model predict control (MPC). Comfortable driving is also an important indicator for ACC evaluation and its acceptance from the users' perspective; [7] and [8] discuss the effects of longitudinal acceleration on comfort. The comfortable acceleration range is defined from -2 to 1.47 m/s^2, while jerk should remain in the range ±2 m/s^3, mostly around ± 0.9 m/s^3. By integrating ACC and traffic light information, the proposed predictive controller significantly reduces energy consumption and improves the driving experience in urban scenarios.

The activity has been partially financed by the European Union – Next Generation EU - PNRR M4C2, Investimento 1.4 - Avviso n. 3138 del 16/12/2021 - CN00000023 Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile)—CNMS—CUP E13C22000980001.

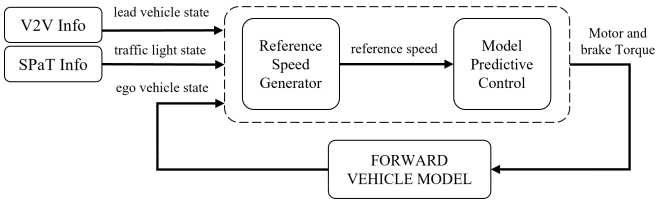

Fig. 1. Energy-efficient Adaptive Cruise Control structure. The reference speed generator is in charge of managing the external policies, while MPC is used to optimize consumption and account multiple constraints, also ensure smooth transitions between different driving scenarios.

The CasADi toolbox [9] coded in MATLAB is used for optimal control problem formulation. The algorithm is compiled to generate C-code and implemented in real-time in Simulink. Performance assessment is conducted in Software-in-the-Loop (SiL) simulation with a validated forward model of the vehicle.

The controller consists of multiple modules, as shown in Figure 1.The inputs available for implementation are:

- *Ego vehicle states*: position, speed, and acceleration.
- *SPaT information*: actuated signals, which include traffic light phases and counter to the next phase change.
- *V2V information*: lead vehicle's planned speed in the prediction horizon, estimated if not available.

The paper is organized as follows: in Section II, the reference speed generator is presented. This module generates the tracking reference according to the lead vehicle and SPaT information. In Section III, the MPC is formulated for its online implementation, and it is combined with a low-level controller for torque and brake commands' actuation. The results of the SiL simulation are reported in Section IV for 3 case studies, while the conclusions are drawn in Section V.

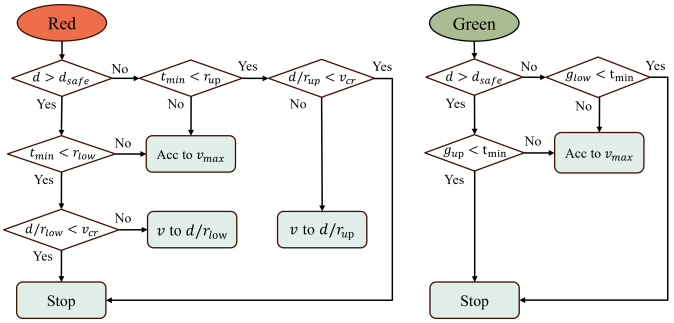

Fig. 2. Signal Phasing and Timing (SPaT) is used as decision-making information to address the challenge of traffic lights negotiation.

II. Reference Speed Generator

The reference speed generator is used to provide the desired speed profile to be tracked by the MPC in the prediction horizon. It first generates the profile according to SPaT information. If SPaT information does not exist, the lead vehicle is tracked. If both SPaT information and the lead vehicle are absent, the maximum speed allowed is the reference. Additionally, the priority between the policies can be changed to improve vehicle-following capabilities and avoid cut-ins.

A. Speed reference for traffic light negotiation

The primary task of the reference speed generator is to make decisions based on SPaT information. It has been demonstrated in [1] and [10] that for a given travel time and distance (such as passing a traffic light), the most energy-efficient speed profile involves accelerating or decelerating as quickly as possible and maintaining a constant speed in between.

Figure 2 shows the reference speed generator logic when SPaT is available. r_{up}, g_{up} is the upper SPaT red/greed phase countdown timer of the actuated signal, and r_{low}, g_{low} is the lower SPaT red/green phase countdown timer. v_{cruise} is the minimum cruising speed; if the vehicle were to cruise at a lower speed, stopping is preferable to avoid the impact on traffic flow.

The minimum time to pass the traffic light is defined as:

$$t_{min} = \begin{cases} \frac{(dis2tl+0.5\cdot(v_{max}-v)^2}{v_{max}a_{max}} & \text{if } dis2tl > \frac{v_{max}^2-v^2}{2a_{max}} \\ \frac{-v+\sqrt{v^2+2a_{max}\cdot dis2tl}}{v_{max}} & \text{if } dis2tl < \frac{v_{max}^2-v^2}{2a_{max}} \end{cases} \quad (1)$$

where $dis2tl$ is the distance to the traffic light, v is the current speed of the ego vehicle, while v_{max} and a_{max} are constraints on maximum speed and acceleration.

The safety distance refers to the space required for a vehicle to stop comfortably. If the vehicle is far from the traffic light ($d > d_{safe}$), a more aggressive strategy is employed. The reference speed is generated based on the lower-bound red phase and the upper-bound green phase countdown timers. Conversely, if the vehicle is closer to the traffic light, the opposite strategy is adopted. Either way, d_{safe} is defined as:

$$d_{safe} = max(\frac{v^2}{2a_{max}}, 20) \quad (2)$$

The equations (3)-(6) present the formulas for speed profile generation. In every one, $dis2tl$ is the vehicle to traffic light distance, v_{max} is the maximum speed, a_{max} is the maximum allowed acceleration set to 1.47 m/s^2 according to the ISO standard, and v is the current vehicle speed.

Case 1 Acceleration to maximum speed (acc to v_{max}):

$$v_{ref} = \begin{cases} v + a_{max}t & \text{if } t > \frac{v_{max}-v}{a_{max}} \\ v_{max} & \text{if } t > \frac{v_{max}-v}{a_{max}} \end{cases} \quad (3)$$

Case 2 Tracing a specific speed (v to $v_h = \frac{dis2tl}{t_{tl}}$):

$$v_{ref} = \begin{cases} v + a_{max}t \cdot sign(v_h) & \text{if } t < \frac{abs(v_h-v)}{a_{max}} \\ v_h & \text{if } t > \frac{abs(v_h-v)}{a_{max}} \end{cases} \quad (4)$$

Case 3 Stopping before the traffic light (Stop):

$$v_{ref} = \begin{cases} v - \frac{v^2}{2dis2tl\cdot t} & \text{if } t < \frac{v^2}{2dis2tl\cdot t} \\ 0 & \text{if } t > \frac{v^2}{2dis2tl\cdot t} \end{cases} \quad (5)$$

B. Speed reference for vehicle following

When a vehicle in front is detected, the reference speed is generated based on the prediction of the lead vehicle's speed, obtained either from V2V communication or by estimation. Since the SPaT logic and vehicle-following logic share the same weight factor, the speed-tracking error should remain similar to achieve good performance while maintaining the desired headway distance. In SPaT logic, the speed tracking error is low in the early steps of the prediction because acceleration and deceleration are smooth. The reference speed in the vehicle-following logic is set to be just 1 m/s higher than the lead vehicle's speed in the prediction horizon. However, this approach can result in longer times for the ego vehicle to reach the desired headway distance if the current relative distance is too large. Therefore, if the headway distance h exceeds 1.1 times the desired value h_{des}, the reference speed will be increased accordingly.

$$v_{ref} = \begin{cases} v_{lead} + 1 & \text{if } h < 1.1h_{des} \\ 1.2v_{lead} + 1 & \text{if } h > 1.1h_{des} \end{cases} \quad (6)$$

III. MPC Problem Formulation

The MPC is initially formulated as a Nonlinear Programming (NLP) problem and then transitioned to a Quadratic Programming (QP) formulation to achieve real-time performance. The sampling time of the controller t_s is 0.3 s, and the prediction horizon is 20 steps, resulting in a prediction time of 6 s, compatible with the application.

A. Lead vehicle trajectory

To build the MPC controller, the trajectory of the lead vehicle speed in the prediction horizon is needed. As previously discussed, this information can be precisely obtained using V2V technology or only estimated online. Two cases of lead vehicle speed estimation are considered:

1) Constant Speed: the lead vehicle's speed is assumed to be constant in the prediction horizon.

$$s_{lead,k} = s_{lead,0} + kt_s v_{lead,0} \quad (7)$$

2) Constant Acceleration: the lead vehicle's acceleration is assumed to be constant in the prediction horizon.

$$s_{lead,k} = s_{lead,0} + kt_s v_{lead,0} + \frac{1}{2} a_{lead,0} (kt_s)^2 \tag{8}$$

B. State function

Two states represent the internal model of the MPC, the longitudinal position s and speed v. The command is the torque request at the electric machine T_m.

The state evolution is the following:

$$\begin{cases} s_{k+1} = s_k + v_k t_s \\ v_{k+1} = (\frac{1}{r_w} T_{m,k} i_{gb} \eta_{gb}^{sign(T_{m,k})} - F_{r,k}) \frac{1}{m_{app}} t_s + v_k \end{cases} \tag{9}$$

where r_w is the wheel radius, i_{gb} and η_{gb} are the gearbox ratio and efficiency, and m_{app} is the apparent mass of the vehicle, the product of the mass m and the inertia factor λ. $F_{r,k}$ is the contribution of all the resistive forces, defined as:

$$F_{r,k} = \frac{1}{2} \rho C_x A_f v_k^2 - mgsin(\theta(s_k)) - f_0 mgcos(\theta(s_k)) \tag{10}$$

where ρ is the air density, C_x is the areodynamic coefficient, A_f is the vehicle front area, f_0 is the rolling resistance, and $\theta(s_k)$ is the road grade angle as function of space.

C. Constraints

1) Speed: the ego vehicle speed should be positive and constrained by the legal road limits:

$$0 < v_k < v_{max}(s_k) \tag{11}$$

2) Safety distance: the distance between ego and lead should be positive and always greater than the minimum distance fixed to $1m$ for the urban scenario:

$$s_k < s_{lead,k} - 1 \tag{12}$$

3) Intrusion into the lead vehicle desired headway distance: the slack variable $\epsilon_{h,k}$ is used in the constraints to implement a penalty on distance. In this paper, a human driver behavior policy (HDB) has been adopted as the vehicle-following policy for urban scenarios. As discussed in , non-linear increase in headway distance with ego vehicle speed can prevent other vehicles from cutting in, especially in urban traffic situations.

$$\epsilon_{h,k} + s_{lead,k} > s_k + A + T v_k + G v_k^2 \tag{13}$$

where the parameters of the HDB are set as follows:
$A = 2$, $T = 1.5$, $G = -0.0246T + 0.010819$.

4) Speed tracking error: $\epsilon_{v,k}$ is introduced to punish the offset between current and reference speed:

$$\begin{aligned} v_{ref,k} &< v_k + \epsilon_{v,k} \\ v_{ref,k} &> v_k - \epsilon_{v,k} \end{aligned} \tag{14}$$

5) Traffic light stop: the vehicle should stop if the traffic light state is red. The result is a position constraint:

$$s_k < s_{stop} + pass \cdot 10^8 + phase \cdot 10^8 \tag{15}$$

where s_{stop} is the stop line position. When the traffic light phase is green, $phase = 0$, and when it is red, $phase = 1$. If the ego vehicle passes the traffic light in the predicted horizon, $pass = 1$; otherwise, $pass = 0$.

6) Comfort: acceleration and jerk are defined as:

$$\begin{cases} a_k = \frac{1}{m_{app}} (\frac{1}{r_w} T_{m,k} i_{gb} \eta^{sign(T_{m,k})} - F_{r,k}) \\ j_k = (a_{k+1} - a_k) \frac{1}{t_s} \end{cases} \tag{16}$$

They are constrained to remain within ISO standards and comfortable limits, relaxed with a proper slack variable $\epsilon_{s,k}$:

$$\begin{aligned} a_{min,iso}(v_k) < a_k < a_{max,iso}(v_k) \\ j_{min,iso}(v_k) < j_k < j_{max,iso}(v_k) \end{aligned} \tag{17}$$

$$\begin{aligned} -2 - \epsilon_{s,k} < a_k < 2 + \epsilon_{s,k} \\ -2 - \epsilon_{s,k} < j_k < 2 + \epsilon_{s,k} \end{aligned} \tag{18}$$

7) Motor and tires limits: physical limits on the powertrain performance and vehicle dynamics are also included:

$$\begin{aligned} T_{m,k} &< T_{max}(\omega_{m,k}) \\ |T_{m,k} i_{gb} \tfrac{1}{r_w} \eta_{gb}^{sign(Tm,k)}| &< \mu mg cos(\theta(s_k)) \\ |T_{m,k} i_{gb} \tfrac{1}{r_w} \eta_{gb}^{sign(Tm,k)}| &< \tfrac{\mu_x}{L} \big[mg \big(L_f cos(\theta(s_k)) \\ &+ h_g sin(\theta(s_k)) \big) + h_g (\tfrac{1}{2} \rho C_x A_f v_k^2 + m_{app} a) \big] \end{aligned} \tag{19}$$

where μ_x is the longitudinal friction coefficient, h_g is the center gravity height, L is the wheelbase, and L_f is the front overhang.

8) Initial state: the first state of the optimization vector is constrained as the ego initial state augmented with a, the acceleration used to compute the first value of jerk:

$$x_0 = [s \quad v \quad a]' \tag{20}$$

9) Slack variables: all the slacks should be positive.

D. Optimization vector

The resulting optimization vector z is:

$$z = [x_0, u_0, \epsilon_0, u_1, \epsilon_1 ...., u_{N-1}, \epsilon_{N-1}] \tag{21}$$

where x is the state vector which includes s, v, and a, u is the control vector which includes only T_m, and ϵ is the slack variables' vector, including ϵ_h, ϵ_v, and ϵ_s.

E. Objective function

The objective function of the MPC is defined as follows:

$$\begin{aligned} \min_{z,u} \sum_{k=0}^{N-1} & w_p P_b(\omega_{m,k}, T_{m,k}) + w_a a_k^2 + w_j j_k^2 + \\ & w_v \epsilon_{v,k}^2 + w_h \epsilon_{h,k}^2 + w_s \epsilon_{s,k} \end{aligned} \tag{22}$$

subj. to

initial state constraint	(20)
speed tracking error	(14)
traffic light stop constraint	(15)
intrusion to desired headway distance	(13)
acceleration and jerk bounds from ISO	(17)
acceleration and jerk bounds from comfort	(18)
road speed limit constraint	(11)
safety distance constraint	(12)
electric motor and friction constraints	(19)

The MPC aims to minimize the objective function formed by a polynomial. When the problem is formulated with NLP, the battery power P_b is expressed using a 5×5 polynomial of motor speed ω and motor torque T_m, as shown in (23):

$$P_b = b_{00} + b_{10}T_m + b_{01}\omega + ... + b_{50}T_m^5 + ... + b_{14}T_m\omega^4 + b_{05}\omega^5 \tag{23}$$

CasADi is used to sample points on the power map and choose the offset between the polynomial and the real map as the objective function. To avoid local minima, the following constraints are applied: 1) Efficiency is always less than one; 2) Power consumption always increases when output torque increases. The resulting goodness of fit among the sampling points is 98.74 %.

F. Quadratic Programming (QP) Formulation

To achieve real-time performance, the NLP problem is rewritten as a QP. The QP formulation is a direct derivation of the NLP, with the following changes:

1) Sparsity: in QP, we use a sparse optimization vector, where the interaction between x_k and x_{k-1} is included in the constraints. The optimization vector z is rewritten as:

$$z = [x_0, u_0, \epsilon_0, x_1, u_1, \epsilon_1 ..... x_{N-1}, u_{N-1}, \epsilon_{N-1}, x_N] \tag{24}$$

2) Quadratic cost: the cost function can only use quadratic terms, so the fitting of the electric machine efficiency map is proposed with a 2nd-order polynomial. The resulting goodness of fit among the sampling points is 98.13 %.

$$P_b = b_{00} + b_{10}T_m + b_{01}\omega + b_{11}T_m\omega + b_{20}T_m^2 + b_{02}\omega^2 \tag{25}$$

3) Linear constraints: the quadratic terms present in aerodynamic calculations and desired headway distance, should use predefined estimated values. Additionally, terms and limitations that should be derived from a lookup table of vehicle speed and distance, such as traffic light status, motor maximum torque, and road angle, should remain the same. Different ego vehicle speeds and distance estimation methods have been tested. Ultimately, the previous MPC solution has been chosen as the estimation method, as reported below:

$$s_{ego,k} = \begin{cases} s_{ego,pre,k+1} & \text{if } k = 1, ..., N-1 \\ s_{ego,pre,k} + v_{ego,k}t_s & \text{if } k = N \end{cases} \tag{26}$$

$$v_{ego,k} = \begin{cases} v_{ego,pre,k+1} & \text{if } k = 1, ..., N-1 \\ v_{ego,pre,k} + a_{ego,k}t_s & \text{if } k = N \end{cases} \tag{27}$$

IV. Results

A. Simulation environment

The controller is tested on a real urban driving scenario reproduced in simulation. The driving cycle has been recorded in Turin, Italy, consisting of a total length of 3089 m and a travel time of 435 s. The recorded speed profile is used as the lead vehicle driven by a human, without any optimization. The SPaT information is set by assumption for each traffic light in the scenario, as shown in Figure 3. The x-axis represents the elapsed time of the current phase, while the y-axis represents

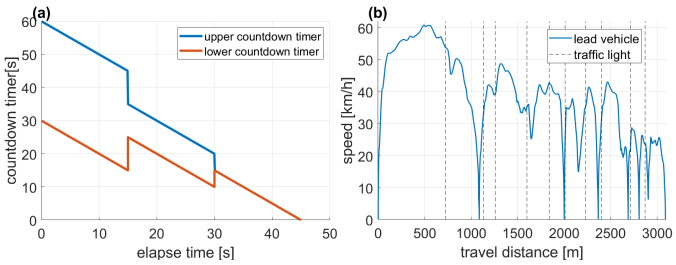

Fig. 3. Urban driving scenario with traffic lights. (a) The SPaT information is represented by the countdown counters. (b) A common speed profile on the scenario is depicted.

TABLE I
MAIN BEV PARAMETERS

Test Mass	1400 kg	Inertia factor	1.05
Wheelbase	2.3 m	Center gravity height	0.5 m
Wheel radius	0.3 m	Rolling resistance	0.006
Front area	2.15 m^2	Aerodynamic drag	0.33
Gearbox ratio	9.6	Gearbox efficiency	0.97
Maximum power	87 kW	Battery size	42 kWh

the upper and lower actuated signal countdown timer The SPaT information is only available within 300 m before the traffic light. The signals delivered to the vehicle include only the two countdown timers and the phase.

The test vehicle model is a class M battery electric vehicle (BEV), whose data is shown in Table I. The model is implemented in MATLAB Simulink using Simscape libraries. The controller used for closed-loop SiL simulation is based on QP formulation, and its refresh rate is set to 0.1 s.

B. Cost function weight factors

The tuning method for the MPC consists of randomly generating thousands of sets of weight factors and testing them using a short driving cycle. Then, the root mean square of acceleration, jerk, tracking error, and energy consumption is compared. The procedure allows the definition of the order of magnitude for the weight factors, which are ultimately fine-tuned to arrive at the optimal solution.

Their numerical values are reported below:

- w_p = 0.5 $1/W$ $\quad$ w_a = 3000 s^2/m
- w_j = 10000 s^3/m $\quad$ w_v = 4000 s/m
- w_h = 4000 $1/m$ $\quad$ w_s = 10^8

It is important to note that the lead vehicle's trajectory information becomes less precise in the latest steps of the prediction horizon. Additionally, the SPaT logic reference is generated based on the current vehicle state and does not account for external conditions. Therefore, in the second 10 steps of the prediction horizon, the weight factor of speed tracking is decreased to 1000 s/m. After testing, this adjustment has proven to improve the controller's performance.

C. ACC with SPaT information and stop

Figure 4 shows the simulation results for ACC with SPaT information available, assuming the lead vehicle's trajectory obtained through V2V. The simulation includes multiple states:

TABLE II
KPIs COMPARISON LEAD VS EGO VEHICLE

	Lead vehicle	Ego vehicle	Delta
Travel time $[s]$	435	435	-
Acc. rms $[m/s^2]$	0.677	0.494	-27%
Jerk rms $[m/s^3]$	0.857	0.158	-81%
Battery energy $[Wh]$	232.1	225.4	-2.9%

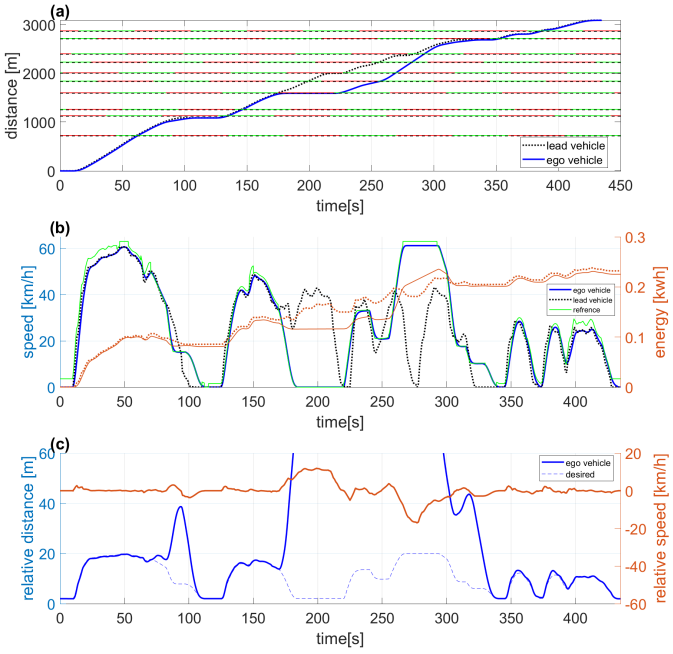

Fig. 4. Results of the ACC with SPaT information and stop. (a) The travel distance (b) The speed profile of the ego and lead vehicle, (c) vehicle-following metrics

from 0 to 43 s, and after 387 s, the vehicle is either far from the traffic light or has already passed it. No SPaT information is available in these periods, and the ego vehicle executes only the ACC task. At 170 s, the ego vehicle stops due to the traffic light turning red, while the lead vehicle passes in the last second of the green phase. Until 302 s, the lead vehicle is far ahead of the ego vehicle; during this period, the controller has no lead vehicle information. The ego vehicle follows only the SPaT to plan its speed, attempting to pass the intersections as soon as possible. Then, at 302 s, it catches up with the lead vehicle again.

In the other cases, the ego vehicle follows the lead vehicle with SPaT information. From 82 to 112 s, and from 306 to 340 s, the ego vehicle decelerates earlier than the lead vehicle and maintains a constant cruising speed to avoid stopping. The multiple speed steps are due to the sudden changes in the SPaT countdown timer information. However, since the lead vehicle still exists and does not stop exactly at the stop line, the ego vehicle still has to stop. Nonetheless, the ego speed profile benefits from energy savings because it reduces high-speed travel time. If the lead vehicle stops just before the stop line and accelerates quickly when the traffic light turns green, it is possible to avoid stopping.

TABLE III
PURE ACC AND ACC WITH SPAT

	Lead vehicle	Pure ACC	ACC with SPaT
Travel time $[s]$	435	435	435
Acc. rms $[m/s^2]$	0.677	0.530	0.492
Jerk rms $[m/s^3]$	0.857	0.158	0.182
Battery energy	-	-7.34%	-10.6%

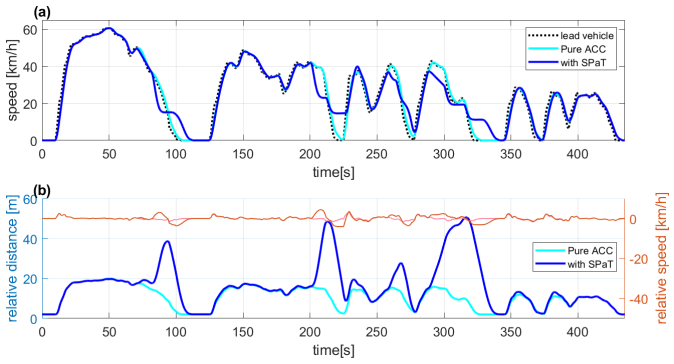

Fig. 5. Results of the pure ACC versus the ACC with SPaT information. In (a) the ego speed is compared to the lead speed, while in (b) the vehicle-following metrics are reported. Knowledge of the traffic light state helps to optimize the speed profile, maintaining missing timing while maximizing its efficiency. The relative distance is acceptable to prevent cut-ins.

The table II shows the overall KPIs of the total driving cycle. Compared to the lead vehicle, the maximum, minimum, and root mean square of acceleration and jerk are significantly decreased, providing a more comfortable driving experience. Before stopping due to the traffic light, the ego vehicle saves approximately 6% of energy compared to the lead vehicle. However, because the ego vehicle maintains a higher speed to catch up with the lead vehicle, this leads to higher energy consumption. Overall energy savings are not significant (about 3 $\%$) but the same travel time is achieved.

D. Comparison between pure ACC and ACC with SPaT

The controller also been tested in the following scenarios:

Pure ACC Performing the vehicle-following task without SPaT information.

ACC with SPaT Performing the vehicle-following task with traffic light information but change the fourth traffic light signal compared to the first scenario, allowing the ego vehicle to pass the traffic light during the green phase without stop.

Figure 5 and table III. shows the comparison between pure ACC and ACC with SPaT. The pure ACC scenario is considered the benchmark. Traditional ACC carries the risk of following the lead vehicle's potentially inefficient decisions. For instance, around the 300 s, the lead vehicle unnecessarily accelerated, and in pure ACC mode, the ego vehicle followed suit. If the SPaT signal is available, the ego vehicle prevents the situation and determines that accelerating would still not allow it to pass the upcoming section. This insight enables the ego vehicle to avoid unnecessary acceleration. Also, the ego vehicle can decelerate earlier and maintain a constant cruising speed to avoid stopping and reduce the total resistance to

TABLE IV
COMPARISON BETWEEN V2V, CON-A AND CON-V

	V2V	Constant acc.	Constant speed
Acc. rms $[m/s^2]$	0.492	0.522	0.561
Jerk rms $[m/s^3]$	0.182	0.264	0.230
Battery energy	-10.61%	-8.22%	-3.88%

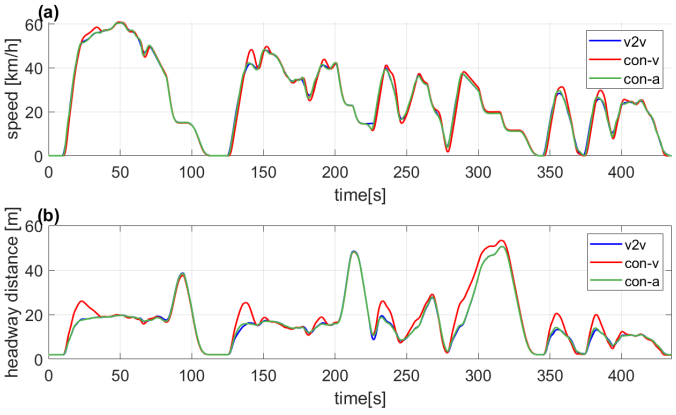

Fig. 6. Analysis of the influence of different lead vehicle trajectory estimation methods, namely V2V (blue), constant speed (red), and constant acceleration (green). Impact on (a) ego vehicle speed and (b) vehicle-following metrics.

further save energy. However, inflection points between the deceleration and the constant cruising phases can result in a higher jerk.

E. Lead Vehicle Trajectory Estimation

Since it is hard to have V2V information and obtain the actual lead vehicle's future trajectory within the prediction horizon, a common solution is estimation. Figure 6 and Table IV compare V2V with two estimation methods, namely constant speed and constant acceleration, based on the ACC with SPaT information scenario. These methods are called con-v and con-a respectively. Both yield reduced performance compared to V2V, with higher acceleration, jerk, and energy consumption. The con-v method significantly increases the energy consumed and tracking error, but produces a lower jerk when compared to con-a. Therefore, a trade-off between comfort and energy reduction should be made, while the benefits of leveraging V2V remain evident. Allowing a precise prediction of the lead vehicle's speed is a very active area of research, where machine learning algorithms and probabilistic approaches are emerging as concurrents of traditional estimation.

V. CONCLUSIONS AND FUTURE WORK

This study presented an MPC-based vehicle controller that integrates ACC and traffic light states to optimize energy consumption and enhance the driving experience of connected urban electric vehicles. The ACC dynamically adjusts the vehicle's speed based on the distance and speed of the preceding vehicle, ensuring safety and comfort while minimizing energy consumption. By leveraging V2I communication, the vehicle can anticipate traffic signal changes and adjust its speed proactively, reducing frequent acceleration and deceleration.

Simulation results demonstrated that the proposed controller performs well in different situations, significantly reducing the energy consumption of EVs, improving travel efficiency, and providing a safe and reliable driving experience. This research not only offers an effective solution for the application of EVs in urban environments but provides technical support and reference for the development of intelligent transportation systems (ITS).

The reference generator block in this paper is rule-based, but there is potential for algorithmic improvement. Future work will explore the use of machine learning techniques to achieve a more precise prediction of the lead vehicle's trajectory. Additionally, machine learning can be employed to make QP's results more similar to NLP's, further enhancing efficiency and accuracy. Further optimization of control algorithms will be pursued to improve the system's real-time performance and robustness. Expanding the scope of simulations to cover more diverse urban traffic conditions and scenarios is also important. Moreover, additional applications of V2I will be investigated to further enhance the intelligence and coordination capabilities of the vehicle controller.

REFERENCES

[1] Matthew Barth, Sindhura Mandava, Kanok Boriboonsomsin, and Haitao Xia. Dynamic eco-driving for arterial corridors. In *2011 IEEE forum on integrated and sustainable transportation systems*, pages 182–188. IEEE, 2011.

[2] Peng Hao, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J Barth. Developing a framework of eco-approach and departure application for actuated signal control. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 796–801. IEEE, 2015.

[3] Peng Hao, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J Barth. Eco-approach and departure (ead) application for actuated signals in real-world traffic. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):30–40, 2018.

[4] Guoyuan Wu, Peng Hao, Ziran Wang, Yu Jiang, Kanok Boriboonsomsin, Matthew Barth, Michael McConnell, Shuwei Qiang, and John Stark. Eco-approach and departure along signalized corridors considering powertrain characteristics. *SAE International Journal of Sustainable Transportation, Energy, Environment, & Policy*, 2(13-02-01-0002):25–40, 2021.

[5] Cunxue Wu, Zhongming Xu, Yang Liu, Chunyun Fu, Kuining Li, and Minghui Hu. Spacing policies for adaptive cruise control: A survey. *IEEE Access*, 8:50149–50162, 2020.

[6] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, 2021.

[7] Chang Wang, Xia Zhao, Rui Fu, and Zhen Li. Research on the comfort of vehicle passengers considering the vehicle motion state and passenger physiological characteristics: Improving the passenger comfort of autonomous vehicles. *International journal of environmental research and public health*, 17(18):6821, 2020.

[8] Il Bae, Jaeyoung Moon, and Jeongseok Seo. Toward a comfortable driving experience for a self-driving shuttle bus. *Electronics*, 8(9):943, 2019.

[9] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

[10] Alexander Koch, Tim Bürchner, Thomas Herrmann, and Markus Lienkamp. Eco-driving for different electric powertrain topologies considering motor efficiency. *World Electric Vehicle Journal*, 12(1):6, 2021.