CHALMERS
UNIVERSITY OF TECHNOLOGY

Bilevel optimization for bunching mitigation and eco-driving of electric bus lines

Downloaded from: https://research.chalmers.se, 2026-02-28 11:51 UTC

Citation for the original published paper (version of record):
Lacombe, R., Gros, S., Murgovski, N. et al (2022). Bilevel optimization for bunching mitigation and eco-driving of electric bus lines. IEEE Transactions on Intelligent Transportation Systems, 23(8): 10662-10679. http://dx.doi.org/10.1109/TITS.2021.3095267

N.B. When citing this work, cite the original published paper.

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Bilevel Optimization for Bunching Mitigation and Eco-Driving of Electric Bus Lines

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár

***Abstract*— The problems of bus bunching mitigation and the energy management of groups of vehicles have traditionally been treated separately in the literature and been formulated in two different frameworks. The present work bridges this gap by formulating the optimal control problem of the bus line eco-driving and regularity control as a smooth, multi-objective *nonlinear program*. Since this nonlinear program has only a few coupling variables, it is shown how it can be solved in parallel aboard each bus, such that only a marginal amount of computations need to be carried out centrally. This procedure leverages the structure of the bus line by enabling parallel computations and reducing the communication loads between the buses, which makes the problem resolution scalable in terms of the number of buses. Closed-loop control is then achieved by embedding this procedure in a *model predictive control*. Stochastic simulations based on real passengers and travel times data are realized for several scenarios with different levels of bunching for a line of electric buses. Our method achieves fast recoveries to regular headways as well as energy savings of up to 9.3% when compared with traditional holding or speed control baselines.**

***Index Terms*— Bus bunching, electric buses, optimal control, nonlinear programming.**

I. Introduction

ELECTRIC vehicles offer a promising way to mitigate the increasing greenhouse gases emissions of the transport sector. Electric buses in particular combine no tailpipe emissions and lower energy consumption than other types of city buses [1] with the lower marginal emissions that urban public transit has in general [2]. However, bus lines are inherently unstable systems, and they have long been known to develop bus bunching if left uncontrolled [3], [4]. One late bus may cause the accumulation of passengers at stops downstream, which acts as a positive feedback loop on the bus and further increases its delay. Likewise, an early bus encounters fewer passengers at stops than expected, and may ultimately catch up with the preceding bus at which point the buses start bunching. The increased service delays incurred by this so-called bus bunching phenomenon may in turn significantly increase the passenger delays, which may eventually discourage users from choosing to use public transport [5].

Networks of electric buses potentially present an additional challenge for any control strategy, namely that electric buses may have charging constraints due to the limited autonomy of their batteries, whereas traditional diesel buses usually manage to complete an entire day in operation with one full tank [6]. Consequently, a model for the energy consumption might be needed in order to anticipate how control actions affect the battery state of charge of any controlled bus. This type of predictive model-based control for vehicles has been extensively used in the literature to minimize the energy consumption during driving missions in various types of environment [7]-[11]. In particular, some authors have designed controllers for the energy management of plug-in hybrid electric buses [12], [13]. However, these works focus mainly on the energy consumption of each individual bus, and as such overlook the operational aspects of the bus line and the issue of bus bunching.

Traditional methods to mitigate bus bunching rely mostly on station-based interventions such as stop skipping or holding buses at bus stops. Transit agencies often implement the latter strategy in an ad-hoc manner to maintain their buses on schedule, but better trade-offs between holding time and commercial speed can usually be achieved [14]. The control strategy developed in that paper, which is based only on real-time information of the bus line, has even be extended to a full bus network [15]. Other authors have further investigated the benefits of the bus holding strategy in various types of settings. The effects of overtaking among buses have been studied in detail in [16] and [17], while [18] explored the impact of including information about the phases of signalized intersections, and [19] modeled the merging of independent bus lines. In addition, control performances have been shown to be further improved for the bus line control problem in [20] and [21] when also including some predictive information about the expected bus travel times. This information can for example be leveraged to reduce large gaps in service by holding buses longer than is possible based on real-time information only. That being said, the aforementioned papers only include limited amounts of predictive information, and it is only used to compute the next control step.

Manuscript received 24 November 2020; revised 26 March 2021 and 9 June 2021; accepted 29 June 2021. Date of publication 21 July 2021; date of current version 9 August 2022. This work was supported in part by the Swedish Electromobility Centre and in part by the Swedish Energy Agency through the Project "Operational Network Energy Management for Electrified Buses" under Grant 46365-1. The work of Balázs Kulcsár was supported by the Transport Area of Advance at the Chalmers University of Technology. The Associate Editor for this article was S. Siri. *(Corresponding author: Rémi Lacombe.)*

Rémi Lacombe, Nikolce Murgovski, and Balázs Kulcsár are with the Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden (e-mail: lacombe@chalmers.se; nikolce.murgovski@chalmers.se; kulcsar@chalmers.se).

Sébastien Gros is with the Department of Engineering Cybernetics, Norges teknisk-naturvitenskapelige universitet (NTNU), 7491 Trondheim, Norway (e-mail: sebastien.gros@ntnu.no).

Digital Object Identifier 10.1109/TITS.2021.3095267

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

In recent years, many authors have opted for model-based rolling horizon control strategies for the bus line problem. These methods operate by solving a problem formulated in the mathematical programming framework to choose a set of control actions over a given time (or space) horizon. Most papers in this vein choose bus holding at stops as their main intervention method. Some research has been done on complementing this strategy with the ability to skip bus stops [22] or to limit passengers boarding [23], and on applying it to multiple bus lines [24], [25]. In these articles, control commands are updated in an event-based fashion, typically when one bus leaves a stop, and the mathematical programs are solved centrally for the whole network. Since the focus is put on station control, the inter-station bus dynamics are most often ignored. In addition to that, but holding strategies might suffer from a lack of space where buses can be held in urban settings, and the absence of inter-station command updates does not leverage fully the potential of real-time communications between buses.

Another vein of research for the model-based predictive control of buses explores inter-station intervention, which often takes the form of speed control. In [26], a linear-quadratic Gaussian control scheme is developed to adjust the speed of a bus to that of the preceding bus in operation. The authors in [27] assemble a model predictive control (MPC) to compute the optimal velocity profile of each bus on a receding horizon that extends to the next bus stop. This controller aims to minimize deviations from the time-table and to enforce regular headways. In [28], an hybrid MPC is used to regularize bus spacings instead, while maintaining a high commercial speed. But in all those papers, the aim is to fulfill a service-oriented objective, and no attention is paid to the eco-driving of the bus fleet as a result. To the best of our knowledge, only [29] includes an energy minimization aspect to the bus line control problem. In that paper, the authors develop a multi-objective MPC scheme with includes energy and service-related cost terms. However, the electric machine model used and the sampling in time of the MPC warrant the inclusion of integer variables in the objective function, thus resulting in a non-smooth optimization problem which needs to be solved centrally, and over short prediction horizons.

This paper extends and develops the MPC-based velocity control strategy outlined in [30], which operates by adjusting predicted time headways to improve both the regularity and energy efficiency of the bus service. Some notable modeling improvements over this reference are presented here. Namely, the predicted bus mass is no longer assumed to be static, but rather to evolve dynamically as a function of the passenger demand at stops. In addition, the MPC is not made to track a static target headway, but is now free to use adaptive headways in order to enforce a regular bus service in any situation. The speed of the surrounding traffic is now also explicitly included in the predictive framework as a constraint on the bus speed.

The main contribution of this paper is to formulate the bus line regularity control problem and the bus fleet eco-driving problem in the same framework, and as a smooth nonlinear program (NLP) with no integer variables. A resolution procedure for the bilevel optimization problem obtained from the decomposition of this NLP is proposed for the real-time implementation of the control strategy in an MPC scheme. This strategy is meant to be deployable in most bus line settings since it alleviates the need for prior timetabling or scheduling through the use of adaptive headways, and scalable since it relies on computations carried out in parallel aboard individual buses. The originality of our work lies in the fine-grained modeling of the inter-station bus travels over long prediction horizons. No other study on bus line control focuses on the eco-driving of individual buses to the best of our knowledge, nor include such a detailed model of the bus dynamics and energy consumption.

Note that the control strategy proposed in this paper has a hierarchical architecture [31], since the MPC scheme operates centrally to compute optimized reference trajectories. Independent local bus controllers are then assumed to track these references, but they are left outside the scope of this paper.

This article is organized as follows. The modeling of buses and passengers is presented in Section II. In Section III, it is explained how the general optimal control problem can be reformulated and solved as a bilevel NLP, and embedded in an MPC. Simulations results are shown and analyzed in Section IV. Finally, the paper closes on some concluding remarks in Section V.

II. Bus Line Modeling and Control

In this section and in the one that follows, we present a predictive control algorithm based on a deterministic model of the bus line problem. The decisions taken by the algorithm are therefore based on a representation of the *averaged* behavior of the system. This algorithm is later applied in a stochastic simulation environment in Section IV.

In order to ease the comprehension of the following modeling steps, we invite the reader to refer to Table VI and Table VII in the Appendix, which offer a detailed summary of all the notations introduced throughout the paper.

A. Modeling Assumptions

We consider n buses that travel continuously on a circular route of length L with q bus stops. The buses are indexed from 1 to n, where the bus with index 1 is the last one that drove through the origin of the route. The bus route layout is represented in Fig. 1. In what follows, we use modular notations to account for the circular aspect of the route. Every bus or stop index is written modulo n or q, respectively, and every position is written modulo L.

We assume that no overtaking among buses can take place, and that their onboard capacity is not limited. This first assumption is not very restrictive as our control strategy aims to keep regular headways, which makes overtaking events unlikely. Considering an infinite bus capacity is also a reasonable assumption in the case of a prior tactical planning phase, as motivated in [33]. The transit agency can be assumed to dispatch sufficiently many buses, such that their capacity is rarely exceeded. Note also that explicitly taking capacity constraints into account would result in adding integer variables to our mathematical model, thus destroying the scalability of

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

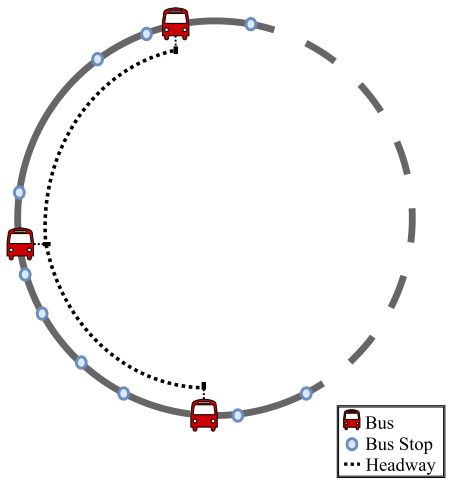

Fig. 1. Illustration of a circular bus route with a nonhomogeneous distribution of bus stops.

our approach. Finally, the charging problem is not addressed here.

In this paper, we consider that no intervention strategy other than speed control can be applied by the controller. In particular, it cannot hold buses at stops longer than needed for the boarding and alighting operations to complete, and it is assumed that these operations can only take place at designated bus stops. Note that augmenting the controller with e.g. a holding strategy or the ability to limit passengers boarding is possible in theory. However, we chose to restrict our analysis to the case where only speed control is allowed, both for addressing situations where station-based interventions cannot be implemented (e.g. in dense urban environments) and as a way to evaluate the benefits that can be expected from the speed control strategy itself.

We further assume that the bus line is operated without any prior timetable, such that only headway regularity is of interest. This fits urban settings well, where the high service frequency results in uncoordinated arrivals of passengers at bus stops [34]. As a result, this is a common assumption in the bus bunching literature [21], [23]. We consider perfect communications of the relevant information between the buses and a central node (which can itself be a bus) when computing control trajectories. When a control trajectory is generated by the algorithm, it is assumed that the buses implement it as such, as would be the case with e.g. autonomous buses. The issue of the lack of precision or compliance from the drivers is outside the scope of this paper.

B. Longitudinal Bus Dynamics

The longitudinal dynamics of a bus $i \in \mathbb{I}_{[1,n]}$ along a fixed route can be written with the position s_i and the bus speed v_i as state variables:

$$\dot{s}_i(t) = v_i(t), \tag{1a}$$

$$m_i(s_i,t)\dot{v}_i(t) = F_{m,i}(t) - F_{b,i}(t) - F_{d,i}(v_i) - F_{r,i}(s_i), \tag{1b}$$

where $F_{m,i}$ is the motor force at the wheels, $F_{b,i}$ is the force generated by the friction brakes, $F_{d,i}$ is the aerodynamic drag, and $F_{r,i}$ gathers the rolling resistance and the gravitational pull. The explicit dependence in t has been omitted from $F_{d,i}$, and $F_{r,i}$ in (1) for simplicity. The mass of the bus m_i is considered to be a function of both space and time to account for the influence of the travel time on the amounts of passengers encountered at the stops. The exact model used for the mass is discussed later in this section. Furthermore,

$$F_{d,i}(v_i,t) = \frac{1}{2}\rho A_{\text{bus}} c_a v_i(t)^2, \tag{2a}$$

$$F_{r,i}(s_i,t) = g\, m_i(s_i,t)\big(\sin\theta(s_i) + c_r\cos\theta(s_i)\big), \tag{2b}$$

where ρ is the air density, A_{bus} is the frontal area of the vehicle, c_a is the aerodynamic air drag coefficient, c_r is the rolling resistance coefficient, and θ is the road gradient [35].

In order to ease the modeling of bus stops, as becomes clear in the next section, a change of the independent variable in (1)-(2) is proposed. The dynamics are now considered with respect to the position s, which means that e.g. the bus speed v_i now denotes a function of the variable s. This variable change to the space domain is common in the predictive cruise control literature, and additional details can be found in [8]-[10]. One of the immediate benefits of this transformation is that the nonlinearities coming from the space-dependent road gradient in (2b) are removed, since the position is no longer a state. Instead, the travel time t_i is now chosen as a state variable. Similarly, the quadratic nonlinearity in (2a) can be removed with an extra variable change, namely by choosing the kinetic energy per mass unit $E_i(s) = \frac{1}{2}v_i^2(s)$ as a state variable instead of the velocity v_i. As a result, the state-space representation of bus i is:

$$\frac{dE_i(s)}{ds} = \frac{1}{m_i(s,t_i)}(F_{m,i}(s) - F_{b,i}(s) - \rho A_{\text{bus}} c_a E_i(s)) - g(\sin\theta(s) + c_r\cos\theta(s)), \tag{3a}$$

$$\frac{dt_i(s)}{ds} = \frac{1}{\sqrt{2E_i(s)}} + \Delta_{\text{stop},i}(s,t_i), \tag{3b}$$

where the states and control inputs can be assembled as $x_i(s) = [E_i(s), t_i(s)]^\top$ and $u_i(s) = [F_{m,i}(s), F_{b,i}(s)]^\top$, respectively. A delay term $\Delta_{\text{stop},i}$ is added to the travel time dynamics in order to capture the dwell times of bus i at bus stops. Its exact expression is presented later in this section, when passengers modeling is discussed. Perfect state measurement is assumed in the rest of this paper.

There is a price to pay for the space domain variable change however, which is that the vehicles cannot have zero speed, as imposed by (3b). This is usually not an issue for cruise control on highways, but it becomes problematic for city buses. One way to address this limitation is to enforce a lower bound constraint on the speed everywhere on the route, which is henceforth noted v_{entry}. This bound is assumed to have a very low, non-zero value, and to be the speed at which buses drive when entering and exiting bus stops. This way, (3) is able to capture the bus dynamics properly, and any additional delay entailed when reaching zero speed at stops is added through $\Delta_{\text{stop},i}$.

In addition, the bus velocity can be limited by e.g. the surrounding traffic or the legal speed limits. Such constraints

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

are formulated on the kinetic energy instead, as:

$$\frac{1}{2}v_{\min}^2(s,t_i) \le E_i(s) \le \frac{1}{2}v_{\max}^2(s,t_i), \tag{4}$$

where $v_{\min}$ and $v_{\max}$ are the lower and upper bounds on the speed, respectively. In order to enforce a non-zero bus velocity, the lower speed bound verifies $v_{\min}(s,t_i) \ge v_{\text{entry}} > 0, \forall s, \forall t_i$. Likewise, the bus stops can be accounted for by imposing $v_{\min}(s_l,t_i) = v_{\max}(s_l,t_i) = v_{\text{entry}}, \forall l \in \mathbb{I}_{[1,q]}, \forall t_i$, where s_l is the location of the bus stop with index l.

Remark 1: The speed bounds can have any general smooth shape. They may be chosen to have large spatial variations, to account for different traffic conditions at different places of the route for example, or large temporal variations, to model different traffic regimes at different times of the day for instance. Note that $v_{\min}$ and $v_{\max}$ can even be updated in operation to include real-time traffic speed information, e.g. if each bus communicates the current state of traffic downstream to the following buses.

C. Energy Consumption Model

The motor speed $\omega_{m,i}$ and torque $T_{m,i}$ of a bus $i \in \mathbb{I}_{[1,n]}$ can be related to its longitudinal force and speed through:

$$T_{m,i}(s) = \frac{r_w \eta(F_{m,i})}{M_f} F_{m,i}(s), \quad \omega_{m,i}(s) = \frac{M_f}{r_w}\sqrt{2E_i(s)}, \tag{5}$$

where r_w is the wheel radius and M_f is the final gear ratio. The transmission efficiency η captures the feature of an electric motor (EM) to be able to operate both in traction and in generation. It models the transmission losses by taking value $1/\eta_f$ when $F_{m,i}(s) \ge 0$ and η_f when $F_{m,i}(s) < 0$, where η_f is the efficiency coefficient of the final gear.

Due to the power limitations of the motor, the torque has to satisfy the constraint:

$$|T_{m,i}(s)| \le \min(T_{\max}, P_{\max}/\omega_{m,i}(s)), \tag{6}$$

where $T_{\max}$ is the maximum motor torque and $P_{\max}$ is the maximum power that the motor can supply continuously [35].

In this paper, the battery is modeled as an open circuit voltage connected in series to an internal resistance. Then, the internal battery power $P_{b,i}(T_{m,i},\omega_{m,i})$ balances the power dissipated over the internal resistance, a constant load consumed by auxiliary devices and the electrical power of the EM [7], [12]. The EM electrical power is modeled by fitting a polynomial function to the data shown in Fig. 2, including second order terms in $T_{m,i}$ and up to fifth order terms in $\omega_{m,i}$ [8]. Other models may be used for the battery and the EM, but in the general case, the battery power can be considered as a nonlinear and monotonically increasing function in $T_{m,i}$ and $\omega_{m,i}$ [8], [13].

D. Bus Stops and Passengers

Let p_i be the current position of each bus $i \in \mathbb{I}_{[1,n]}$ on the route. We assume that each bus is controlled over a finite spatial horizon, which we refer to as prediction horizon, or simply horizon, hereafter. Since part of our goal is to enforce a regular bus service, we choose to let the horizon

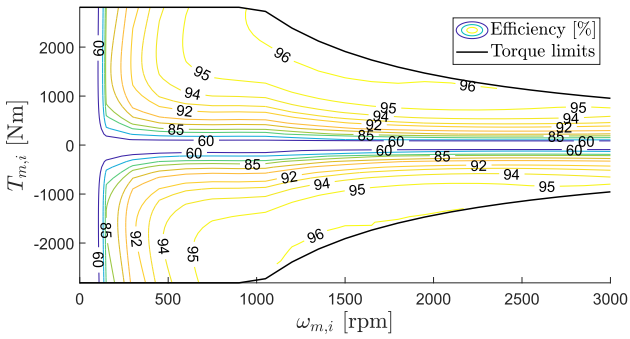

Fig. 2. Efficiency map of the motor, as a function of its speed and torque. The black lines denote the torque constraints.

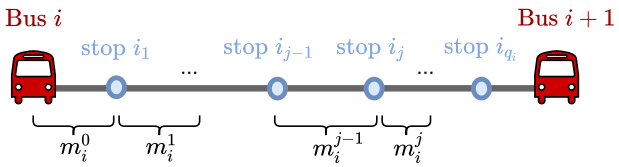

Fig. 3. Prediction horizon for bus i, where the successive mass values at the stops downstream are shown.

of each bus stretch all the way to the preceding bus. The horizon for bus i is thus the interval $[p_i, p_{i+1}]$. This way, the predicted forward headway with bus $i+1$, noted H_i, is by construction nothing else than the difference between the terminal travel time $t_i(p_{i+1})$ and the current simulation time t^0. This avoids the need to resort to an indirect proxy, such as e.g. bus spacings [28], to enforce headway regularity. Indeed, this particular proxy might not be adapted to certain settings, such as a route with a non-homogeneous distribution of bus stops. Note that this means that the horizons are not overlapping here, and that their union covers the full route.[1] This comes at a price, however, since bus overtaking cannot be captured with this choice of control horizons. Indeed, if one bus is about to overtake another, its prediction horizon shrinks to zero. But recall that it has been assumed previously that no overtaking event can take place here.

In these settings, the bus stops are distributed among the horizons of the buses. Let $\mathcal{S}_i$ be the subset of size q_i of the bus stops found on the horizon of bus i. For the sake of simplicity, we may consider a relative indexing of these stops, $\mathcal{S}_i = \{i_1, \ldots, i_{q_i}\}$, in the order that they are visited by the bus. The position $s_{i_j}, i_j \in \mathcal{S}_i$, at which any stop lies on the horizon is then reached by bus i at time $t_i(s_{i_j})$. An illustration of the complete horizon for bus i is displayed in Fig. 3.

It is assumed that the arrivals of passengers at any stop $i_j \in \mathcal{S}_i$ are modeled as a homogeneous Poisson process with parameter λ_{i_j} [36]. Since overtaking is not allowed, the last bus to have visited stop i_j is the one directly preceding bus i, i.e. bus $i+1$. Let t_{i+1}^j be the time at which it departed from stop i_j. Note that it does not refer to the travel time of bus

[1]Other horizon types are possible, such as shrinking horizons to the next stop [27]. By choosing to have long and variable inter-bus horizons we gain direct access to the time headways, at the expense of the problem's complexity.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

$i+1$ on its own horizon, but rather denotes a fixed scalar since it refers to a past event. Having introduced these notations, we can now write that bus i expects to find $\lambda_{i_j}(t_i(s_{i_j}) - t^j_{i+1})$ passengers on average when reaching stop i_j, i.e. that the amount of passengers increases linearly with respect to the travel time. For simplicity, we assume that the numbers of boarding and alighting passengers at stops are real variables. The delay term introduced in the travel time dynamics (3b) to capture the behavior of buses at stops can then be expressed as:

$$\Delta_{\text{stop},i}(s, t_i) = \begin{cases} 2t_s + b\lambda_{i_j}(t_i(s_{i_j}) - t^j_{i+1}) & \text{if } s = s_{i_j}, \\ & i_j \in \mathcal{S}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

In this equation, the delay for the bus to reach zero speed from the lowest allowed speed v_{entry} and open its doors (and vice-versa) is noted t_s, and the boarding time for each passenger is noted b. It is assumed that the boarding and alighting operations can be carried out in parallel (e.g. through different doors of the bus). Since the boarding operation usually takes longer, the delay caused by alighting passengers is not included [36]. Note that the travel time t_i is a piecewise continuous function in space as a result, due to the *jumps* caused by $\Delta_{\text{stop},i}$ when driving through bus stops.

E. Evolution of the Mass

Similarly to the travel time, the mass m_i of bus i is affected by passengers boarding and alighting from the bus. It is a piecewise constant function in space, since passengers exchange can only take place at bus stops, and it is also dependent on the travel time since this affects the passengers loads encountered by the bus at the stops downstream. To derive an expression for m_i, one can start by noticing that it can only take $q_i + 1$ distinct values over the horizon of bus i since q_i bus stops are encountered. Let $\{m^0_i, m^1_i \dots, m^{q_i}_i\}$ be the set of the successive values taken by m_i, where the dependency in the travel time has been dropped for notational brevity, and where m^0_i is the initial mass of bus i. The evolution of m_i over the horizon of bus i is illustrated in Fig. 3.

For any $j \in \mathbb{I}_{[1,q_i]}$, the new mass value m^j_i past stop i_j can be computed recursively from the previous one m^{j-1}_i as:

$$m^j_i = (1 - \mu_{i_j})(m^{j-1}_i - m_{\text{emp}}) + m_{\text{pax}}\lambda_{i_j}\left(t_i(s_{i_j}) - t^j_{i+1}\right), \quad (8)$$

where m_{pax} is the average passenger mass, m_{emp} is the mass of the empty bus, and where μ_{i_j} is the *alighting proportion* of onboard passengers at stop i_j [36]. Note that μ_{i_j} is then a fixed scalar in $[0, 1]$ which can be set from historical passenger flow data. The right-hand side of (8) thus models the onboard passengers staying on the bus (first term), and the new boarding passengers (second term), the load of which increases with the travel time needed to reach that stop. This expression prevents the mass from ever becoming smaller than m_{emp} since only a fraction of the onboard passengers alight at each stop.

From this recursive formulation, one can prove by induction that m^j_i can be written as an explicit function of the travel time and the initial mass:

$$m^j_i = m^0_i \prod_{l=1}^{j}(1 - \mu_{i_l}) - m_{\text{emp}} \sum_{l=1}^{j}\prod_{r=l}^{j}(1 - \mu_{i_r}) + m_{\text{pax}} \sum_{l=1}^{j}\prod_{r=l+1}^{j}(1 - \mu_{i_r})\lambda_{i_l}(t_i(s_{i_l}) - t^l_{i+1}). \quad (9)$$

It can be noted from this expression that the mass too depends linearly on the travel time.

Now that the set $\{m^0_i, m^1_i \dots, m^{q_i}_i\}$ is known, we may use it to assemble the mass function m_i as:

$$m_i(s, t_i) = m^j_i(t_i), \quad \forall s \in [s_{i_j}, s_{i_{j+1}}], \quad \forall j \in \mathbb{I}_{[0,q_i]}, \quad (10)$$

where indices i_0 and i_{q_i+1} are used to refer to the two stops bordering the horizon of bus i, and where the dependency of each m^j_i, $j \in \mathbb{I}_{[1,q_i]}$, in the travel time is now written explicitly.

III. Bilevel Optimization and Receding Horizon Control

In this section, a general optimal control problem (OCP) is assembled based on the model developed throughout the previous section. Since problems of this type are unpractical to treat as such, the OCP is first rewritten as a smooth NLP through a tight relaxation of some of the problem constraints. This NLP is then decomposed into a high-level problem and several independent bus-level problems. Since each bus-level problem only contains the information related to a single bus, one can envision a physically distributed resolution of this bilevel decomposition where buses can act as independent computing nodes. Lastly, we present how this resolution framework can be embedded in an MPC to address the challenges of real-time control in urban settings.

A. Optimal Control Formulation

A predictive bus line model can now be derived from the modeling steps taken in Section II. Recall that we noted H_i the predicted forward headway of bus i with the preceding bus $i + 1$, and that it also denotes the predicted travel time of bus i on its control horizon. Writing the state and control input vectors $x(s) = [x_1(s), \dots, x_n(s)]^\top$ and $u(s) = [u_1(s), \dots, u_n(s)]^\top$, where $x_i(s) = [E_i(s), t_i(s)]^\top$ and $u_i(s) = [F_{m,i}(s), F_{b,i}(s)]^\top$, the energy-aware bus line control problem can be formulated as the following OCP:

$$\min_{x(s),u(s)} \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha \sum_{i=1}^{n}(H_i - H_{i-1})^2 + \beta \sum_{i=1}^{n} \int_{p_i}^{p_{i+1}} \frac{P_{b,i}(T_{m,i}, \omega_{m,i})}{\sqrt{2E_i(s)}} ds, \quad (11a)$$

$$\text{s.t. } \forall i \in \mathbb{I}_{[1,n]}:$$

$$E_i(p_i) = E^0_i, \quad t_i(p_i) = t^0, \quad (11b)$$

$$H_i = t_i(p_{i+1}) - t^0, \quad (11c)$$

$$(3), (4), (6), \quad (11d)$$

where the state dynamics (3) and the constraints (4), (6) are enforced for all $s \in [p_i, p_{i+1}]$ for each bus i, and where

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

the bus mass can be computed explicitly from (9) and (10). The initial reduced kinetic energy of bus i is E_i^0, and the initial simulation time is noted t^0. The motor speed and torque are expressed in terms of the state and control variables through (5) in the expressions that use them. The objective function (11a) is weighted by Λ_i, which is defined as a ratio involving passengers arrival rates at stops downstream, $\Lambda_i = \sum_{i_j \in S_i} \lambda_{i_j} / \sum_{l=1}^{q} \lambda_l$, and by the positive coefficients α and β that account for the trade-off between the different objectives.

Note that (11) is based on the nominal bus line model. As a result, the optimal trajectories obtained when solving this OCP may not be tracked perfectly since the system may be subject to external disturbances in practice. This point is discussed further when introducing the receding horizon control idea at the end of this section.

In this formulation, the buses do not try to track a predefined service headway, but rather aim to adapt their predicted headways to whichever common headway is optimal. The rationale for this is that the desirable headway for service regularity might change depending on e.g. the amount of disturbances applied to the system [21]. The predictive information available can therefore be leveraged to try to find this optimal headway.

In the economic objective function (11a):

- The first term is a *look-ahead* term which rewards short headways proportionally to Λ_i for each bus i. These coefficients account for the differences in the passengers affluence at stops downstream among the prediction horizons, and are used as proxies to minimize passengers waiting times at stops. For instance, a bus entering the inner city center where many passengers might be waiting is given a higher incentive to have a short headway than one traveling towards the outskirts of the city.
- The second term introduces a *look-back* feature which penalizes the deviations of successive headways. In other words, it introduces some coupling between successive buses, such that each bus also adapts its driving behavior to the following bus.
- The third term is the amount of battery energy required for each bus to drive to the end of its prediction horizon. This sets an incentive for buses to adapt their driving behavior accordingly, and is motivated by the observation that different trajectories with similar travel times can have a vastly different energy consumption.

The first two objectives focus on bus headways and directly aim to improve the overall passenger experience, while the last objective focuses on the energy consumption, which is meaningful to the service provider. Therefore, these three objectives may promote opposed control actions since e.g. enforcing shorter headways usually requires a higher energy consumption. Hence, the trade-off coefficients α and β must be calibrated carefully depending on the application considered.

B. Direct Reformulation of the OCP

As a next step towards the resolution of the problem presented, we propose a direct optimal control reformulation of (11). The prediction horizon of each bus i is split into N uniform shooting intervals of varying length Δs_i, due to the unequal horizon lengths. We assume a piecewise constant input parametrization, i.e. $u_i(s) = u_{i,k}\ ,\ s \in [s_{i,k}, s_{i,k+1})$, where $s_{i,k} = p_i + k\Delta s_i$, and a multiple-shooting 'discretization' of the dynamics [37]. Since the shooting points might often 'miss' the exact locations of the bus stops, the latter are assumed to be located at the closest shooting point instead, i.e. $s_{i_j} = s_{i,k}\ ,\ i_j \in \mathcal{S}_i$, where the k-th shooting point is the one closest to stop i_j. The direct reformulation can now be written as an NLP:

$$\min_{X,U} \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha \sum_{i=1}^{n} (H_i - H_{i-1})^2 + \beta \sum_{i=1}^{n} \sum_{k=0}^{N-1} J(x_{i,k}, u_{i,k}), \tag{12a}$$

$$\text{s.t. } \forall i \in \mathbb{I}_{[1,n]}:$$

$$E_{i,0} = E_i^0, \quad t_{i,0} = t^0, \tag{12b}$$

$$H_i = t_{i,N} - t^0, \tag{12c}$$

$$x_{i,k+1} = F(x_{i,k}, u_{i,k}), \quad k \in \mathbb{I}_{[0,N-1]}, \tag{12d}$$

$$g(x_{i,k}, u_{i,k}) \leq 0, \quad k \in \mathbb{I}_{[0,N-1]}, \tag{12e}$$

where $X_i = [x_{i,0}, \ldots, x_{i,N}]^\top$, and $U_i = [u_{i,0}, \ldots, u_{i,N-1}]^\top$ are vectors gathering the optimization variables relative to bus i, and where $X = [X_1, \ldots, X_n]^\top$, and $U = [U_1, \ldots, U_n]^\top$. The numerical integration of the state dynamics (3) is carried out over $s \in [s_{i,k}, s_{i,k+1}]$ by the function $F(x_{i,k}, u_{i,k})$, starting from $x_{i,k}$ and with the input $u_{i,k}$. Similarly, the function $J(x_{i,k}, u_{i,k})$ carries out the numerical integration of the function $P_{b,i}/\sqrt{2E_i}$ over $s \in [s_{i,k}, s_{i,k+1}]$ to find the energy consumed. In both cases, the Runge-Kutta method is used. A discretized version of (10) can be obtained easily for the bus mass, as it is already piecewise constant in space, and used in (12d). Finally, the function g gathers the inequality constraints from (4) and (6).

Remark 2: One important feature of NLP (12) is that no additional integer variables need to be added to detect the bus stops. Indeed, each bus stop is automatically associated with a shooting point. If the problem had been formulated in time rather than in space however, integer variables would have been needed since the correspondence between shooting points and bus stops would have been dependent on the control inputs. Note also that this discretization step removes the previous discontinuities of the mass and the travel time at the stops. Formulating a smooth NLP is crucial for being able to deploy second-order optimization methods later on, which are a powerful tool to solve such problems.

The only non-smooth part in (12) now comes from the discretization of the torque constraints (6). Indeed, the motor torque is not continuously differentiable in zero due to the different transmission efficiency when the EM operates in traction or in generation, as can be seen from (5). This can be dealt with by *lifting* the NLP, i.e. by adding additional optimization variables in order to obtain a smooth modified version. Here, we introduce separate longitudinal force variables for each motor regime (traction or

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

generation), such that $F_{m,i,k} = F_{t,i,k} - F_{g,i,k}$, $\forall k \in \mathbb{I}_{[0,N-1]}$, $\forall i \in \mathbb{I}_{[1,n]}$. The discretized control input vector then becomes $u_{i,k} = [F_{t,i,k}, F_{g,i,k}, F_{b,i,k}]^\top$. The torque constraints (6) may now be rewritten as:

$$0 \leq F_{t,i,k} \leq \frac{\eta_f P_{\max}}{\sqrt{2E_{i,k}}}, \quad F_{t,i,k} \leq \frac{\eta_f M_f T_{\max}}{r_w}, \tag{13a}$$

$$0 \leq F_{g,i,k} \leq \frac{P_{\max}}{\eta_f \sqrt{2E_{i,k}}}, \quad F_{g,i,k} \leq \frac{M_f T_{\max}}{\eta_f r_w}, \tag{13b}$$

thus removing any non-smoothness from (12), since the motor torque $T_{m,i,k}$ can now be expressed as:

$$T_{m,i,k} = \frac{r_w}{\eta_f M_f} F_{t,i,k} - \frac{\eta_f r_w}{M_f} F_{g,i,k}. \tag{14}$$

With these new expressions, the lifted version of (12) is the smooth NLP:

$$\min_{X,U} \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha \sum_{i=1}^{n} (H_i - H_{i-1})^2 + \beta \sum_{i=1}^{n} \sum_{k=0}^{N-1} \tilde{J}(x_{i,k}, u_{i,k}), \tag{15a}$$

$$\text{s.t. } \forall i \in \mathbb{I}_{[1,n]}:$$

$$(12b), (12c), (12d), \tag{15b}$$

$$\tilde{g}(x_{i,k}, u_{i,k}) \leq 0, \quad k \in \mathbb{I}_{[0,N-1]}, \tag{15c}$$

where $\tilde{J}$ and $\tilde{g}$ are very similar to J and g in (12), except that the torque expression (14) is used to compute the energy consumption $\tilde{J}$, and that the inequality constraints $\tilde{g}$ include the modified torque constraints (13) instead of the original ones (5)-(6).

Now, it remains to show that this smooth relaxation (15) of the original problem (12) is tight, i.e. that both problems have the same solution (including the same optimal values for the longitudinal force). Fortunately, this can be proved rather easily with a mild assumption on the battery power $P_{b,i}$. This result is formalized in Proposition 1, the proof of which is given in Appendix A.

Proposition 1: The lifted version (15), where the torque constraints are enforced through (13) and where the torque is expressed as (14), has the same solution as the original problem (12).

C. Decomposition

Solving the lifted version (15) of the fully-centralized NLP as such presents some difficulties. The nonlinear dynamics from (3b) cause it to be non-convex, and the size of the problem might become large depending on the scenario size. In addition, the resolution of this NLP would have to be carried out centrally, which might make the method sensitive to communication issues with the vehicles in the case of a real-life implementation. As the goal is for the problem to be solved repeatedly in a receding horizon fashion thereafter, we propose to make it more tractable through a bilevel decomposition.

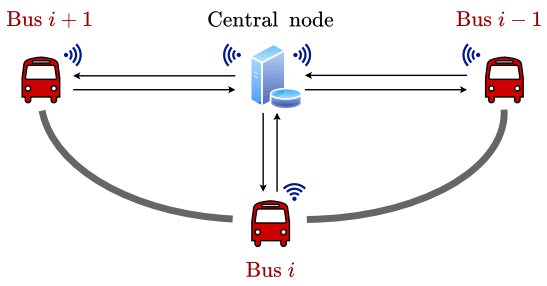

Fig. 4. Resolution scheme of the decomposed problem. The arrows denote the remote communication between the central node and the buses. Note that the buses do not need to share information with each other in these settings.

The problem can be split into a line-level (or high-level) problem, and bus-level subproblems:

$$\min_{H} \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha (H_i - H_{i-1})^2 + \beta V_i(H_i), \tag{16a}$$

$$\text{s.t. } H_i \in \text{dom}(V_i), \quad i \in \mathbb{I}_{[1,n]}, \tag{16b}$$

$$V_i(H_i) = \min_{X_i, U_i} \sum_{k=0}^{N-1} \tilde{J}(x_{i,k}, u_{i,k}), \tag{17a}$$

$$\text{s.t. } E_{i,0} = E_i^0, \quad t_{i,0} = t^0, \tag{17b}$$

$$t_{i,N} = t^0 + H_i, \tag{17c}$$

$$x_{i,k+1} = F(x_{i,k}, u_{i,k}), \quad k \in \mathbb{I}_{[0,N-1]}, \tag{17d}$$

$$\tilde{g}(x_{i,k}, u_{i,k}) \leq 0, \quad k \in \mathbb{I}_{[0,N-1]}, \tag{17e}$$

where $H = [H_1, \ldots, H_n]^\top$, where $V_i(H_i)$ is the optimal cost of the bus-level NLP (17) for a given H_i and where $\text{dom}(V_i)$ at the high-level denotes the feasible set of (17) for bus i. Note here that $\tilde{g}$ includes the inequality constraints (13), and that the motor torque appearing in the expression of $\tilde{J}$ is computed according to (14), as mentioned previously.

Remark 3: Due to the non-convexity of the bus-level subproblems (17), no guarantees of global optimality can be obtained, in general. However, this type of decomposition is known to conserve global optimality in the convex case [38]. The proof proposed in that work can be adapted to the non-convex case to show that the decomposed problem (16)-(17) has the same set of KKT points [39] as the original problem (12). Under some mild assumptions, this is equivalent to saying that these two problems have the same set of local minima [39].

Remark 4: Some of the computations of (16)-(17) can be carried out in parallel since the only coupling terms between the buses have been gathered at the high-level. This opens the door to a physically distributed resolution where the bus-level NLPs (17) could be solved independently aboard each bus, while only (16) would be solved centrally, as illustrated in Fig. 4. The amount of information that would need to be exchanged between the central node and the buses in that case is quite low, as it is motivated when discussing the resolution procedure in Appendix B.

Constraint (16b) is important to guarantee the feasibility of each bus-level problem with respect to the terminal

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

constraint (17c). Since the feasible set $\mathrm{dom}(V_i)$ gathers the possible terminal travel times for bus i, it is in fact an interval with static bounds, and can be expressed as $\mathrm{dom}(V_i) = [H_i^{\min}, H_i^{\max}]$. Therefore, (16b) may simply be rewritten as a set of linear inequality constraints. The two bounds of each feasible set can be computed as the solutions to the optimization problems:

$$H_i^{\min} = \min_{X_i, U_i} \; t_{i,N}, \tag{18a}$$

$$\text{s.t. (17b), (17d), (17e)}, \quad \forall k \in \mathbb{I}_{[0,N-1]}, \tag{18b}$$

and,

$$H_i^{\max} = \min_{X_i, U_i} \; -t_{i,N}, \tag{19a}$$

$$\text{s.t. (17b), (17d), (17e)}, \quad \forall k \in \mathbb{I}_{[0,N-1]}, \tag{19b}$$

which are the minimum and the maximum time problems, respectively. They too may be solved in parallel, aboard the concerned buses.

This bilevel decomposition of the original problem is not a panacea, however. The high-level problem (16) remains a non-convex NLP, with an objective function defined implicitly through (17). Different tools can be deployed to solve it. Here, we chose to combine some results from parametric optimization [40] with a second-order optimization method, which was in part motivated by the better convergence rates of such methods. The details of the Sequential Quadratic Programming (SQP) [39] algorithm implemented can be found in Appendix B. In what follows, we assume that the resolution of the decomposed problem has been carried out.

D. Receding Horizon Control

Urban buses typically evolve in an environment which can be highly dynamic and uncertain since many different types of actors interact in a restricted space. However, the control decisions obtained from solving (16)-(17) are based on deterministic predictions, which ignore the stochastic disturbances coming from the real system. In order to reject these disturbances, closed-loop control is introduced by using an MPC [32].

Despite having expressed the commands and the dynamics in the space domain up to that point, we choose to sample the MPC in time. The rationale for this is that the time needed for each bus to travel a given distance may change a lot depending on where the bus is located on the route. For instance, running an MPC sampled in space would result in no command update for any bus dwelling at a stop, thus potentially ignoring new information coming from the other buses. Opting for synchronous updates instead makes it possible to be computing new commands constantly. The MPC sampling time, which is noted T hereafter, could for example be calibrated on the computation time needed to solve the decomposed problem (16)-(17) in order to apply command updates as frequently as possible.

By sampling the MPC in time however, a bus might travel through a fractional number of shooting points during any MPC stage. In this case, its new states can be interpolated from the previous state trajectories when the next stage begins.

Algorithm 1: MPC for the Bus Line Problem

```
initialize state, time ← 0, data ← {}
while time − time_end < 0 do
    X, U ← solve (16)-(17)
    state ← system_evolution(state, X, U, T)
    data ← {data, state}
    time ← time + T
end
return data
```

Since the buses move relatively to each other between MPC stages, their prediction horizon length changes constantly. Using a constant number of shooting points N for each horizon guarantees that each NLP (17) has a constant size over time, regardless of the horizon length. Since these NLPs are the main bottleneck in terms of computation time, having a constant N ensures that T can be chosen in a way that guarantees that the full problem (16)-(17) can always be solved before the next MPC update.

A summarized pseudo-code representation of the MPC is given in Algorithm 1. The *state* variable is to be understood as containing the current information available about the whole system, including e.g. the last departure times from stops or the horizon lengths. The *system_evolution* function implements during T time units the control trajectories U obtained by solving the decomposed problem. It then returns the updated state of the bus line based on the evolution of the real system, at which point new commands can be computed. A data structure containing the complete state history is finally returned when the control has been applied during the desired period of time.

In this algorithm, the *system_evolution* function is assumed to represent some local bus controllers which can track the optimized reference trajectories X and U generated at each sampling instant. As a result, the entire bus line control structure can be said to have a hierarchical architecture [31]. The tracking control layer can be assumed to operate at a higher frequency than that of the MPC, and to guarantee critical safety constraints such as e.g. collision avoidance with surrounding vehicles, including the other controlled buses. The implementation of this additional control layer is outside the scope of this paper, however.

IV. Simulations

In this section, the proposed MPC strategy outlined in Algorithm 1 is tested in simulations capturing realistic bus operations.

A. Simulations Setup and Route Layout

Historical data from bus line 17 in Gothenburg, Sweden, is used to calibrate the simulations. This urban bus line serves a total of 28 stops during one full trip, several of which are located in the inner city center. In addition, this route makes the buses drive through a hilly terrain, as shown in Fig. 5, which means that their driving profiles must be adapted

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

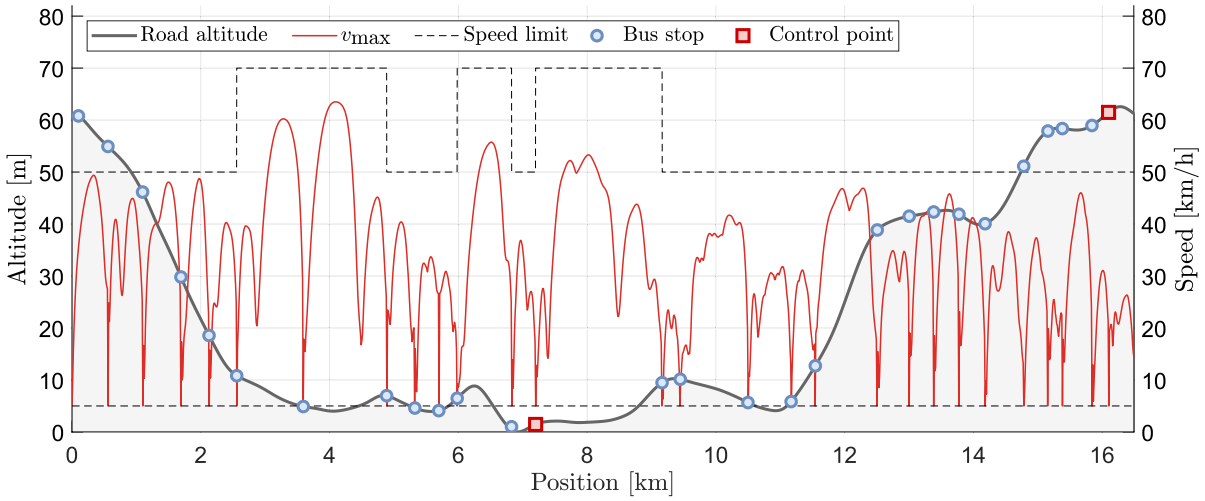

Fig. 5. Route layout of bus line 17. The bus stops are placed according to their altitude on the route, and it is indicated which of them serve as control points for the holding baseline. The solid red line is the average maximum speed obtained from historical bus driving profiles, and is used as the upper velocity bound $v_{\max}$. The dashed lines indicate the speed limit on each road segment, as well as the lower velocity bound v_{entry}.

accordingly by the MPC in order to be energy-efficient [8]. We focus on transit operations during rush hour in the simulations as this presents the biggest challenge from a control point of view. The upper bound $v_{\max}$ for the bus velocity used in the predictions is extrapolated from several real driving profiles in order to represent the disturbances coming from the surrounding traffic. Fig. 5 displays how $v_{\max}$ changes at different places of the route, depending e.g. on the speed limit. Note that $v_{\max}$ is assumed to be only space-dependent here. The advantage of having a detailed velocity profile instead of e.g. a simple piecewise constant function is that it enables fine-grained predictions of the energy consumption over the route. The passengers arrival rates λ_l and alighting proportions μ_l at every stop $l \in \mathbb{I}_{[1,q]}$ are directly inferred from the historical data.

Each simulation run consists of two hours of bus operation during rush hour, and each starts with 8 buses in total. In these simulations, the MPC is sampled with $T = 30\,\text{s}$. Unlike the prediction framework used in the MPC, which is fully deterministic, the simulations include several sources of stochastic disturbances to account for the unpredictability of a real transit system. (i) The accumulation of passengers at each bus stop $l \in \mathbb{I}_{[1,q]}$ is sampled from a Poisson process with parameter λ_l. (ii) Similarly, the number of alighting passengers at l is sampled from a binomial distribution depending on the alighting proportion μ_l and the load of the bus arriving at l [36]. (iii) Lastly, the maximum velocity at which buses can travel is increased (or decreased) by a certain percentage of $v_{\max}$ on each inter-stop segment separately. The deviation percentages are sampled from a normal distribution centered around 0 and with a constant variance $\sigma^2_{\text{traffic}}$. They are meant to model the fluctuations of the real traffic conditions around their historical average, as well as the differences in traffic conditions across different segments of the route. In order to account for the evolution of traffic conditions over time, these disturbances are resampled every two and a half minutes. Note that the speed upper bound obtained may have to be adapted *ex post facto* to comply with the legal speed limit on each segment (which is either 50 km/h or 70 km/h). This means that larger values of σ_{traffic} tend to slow down the system on average since $v_{\max}$ is not usually much lower than the legal speed limit. It the simulations, we chose to set $\sigma^2_{\text{traffic}} = 10\,\text{m}^2/\text{s}^2$ in order to model moderate deviations from the historical average.

In [30], it was observed that the speed constraint functions in (4) can affect significantly the performances of the control algorithm. In particular, choosing a high value for the lower bound $v_{\min}$ leads to limited speed adjustments possibilities, and might impair the ability of the controller to regularize bus operations. Therefore, we chose to set $v_{\min}(s) = v_{\text{entry}}$, $\forall s \in [0, L]$, in order to investigate the full potential of the proposed method, and where it is assumed that $v_{\min}$ does not depend on time either. Note that the buses are never made to travel at such a low speed in practice, except in the case of extreme bunching.

The modeling and simulation aspects are implemented in MATLAB. The symbolic framework CasADI [41] is used to assemble the NLPs (17) and (22), which are then solved with the primal-dual interior point solver IPOPT and with the active-set solver qpOASES, respectively.

B. Baselines

We compare the proposed MPC strategy with two baselines relying on different intervention strategies, namely a classical headway-based holding baseline, and a proportional-integral (PI) controller. The former method relies on simple rule-based control actions at a subset of bus stops, named control points, where each bus may be held for long periods of times in order to compensate unstable headways. Holding methods in general are the ones most commonly implemented by transit agencies in practice, which motivates the inclusion of one as a baseline. On the other hand, the PI-controller has an intervention strategy similar to that of the proposed MPC since it computes longitudinal force commands to adjust the bus speed in operation. Like the MPC, it is not allowed to hold

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

buses at the control points. Each baseline is presented in-depth below.

Holding baseline: This control method makes the buses travel at the maximum possible speed between stops. At the control points, it holds buses if necessary, until they can be dispatched from each control point according to a predefined target headway. In other words, when a bus is ready to leave a control point, two situations can occur:

- If the last bus departure from that control point occurred more than one target headway ago, the bus leaves the control point immediately.
- Otherwise, the bus is held at the control point until the time where the last departure occurred precisely one target headway ago, at which point the bus leaves the control point.

Based on the rush hour timetables for bus line 17, the target headway is set to 5 minutes in the simulations. Note that it is only used by the two baselines, but not by the proposed MPC, which is based on adaptive headways, and as such does not need any predefined target. It is assumed that the route has two control points and that the holding baseline can hold the buses there without any time constraint. In order to mimic real-life operations, they are chosen as stops where the transit agency already performs bus holding in practice. One control point is chosen to be the central station of Gothenburg, as it is the stop with the highest passenger flow in our dataset, and the other is chosen to be the stop which is the farthest away from the city center, as it corresponds to the actual bus terminus of the line. The location of these control points on the route is shown in Fig. 5.

PI-controller: This controller operates along the same lines as the PI-controller presented in [28]. The main difference here is that instead of spacing errors, we consider the error between the current position of a given bus and the shifted position of the preceding bus. More precisely, the position of the preceding bus is shifted backward in time by one target headway, i.e. 5 minutes here. More details on this error term can be found in [27]. The control inputs are then updated in discrete-time proportionally to the current error, with proportionality constant K_I, and to the current rate of change of this error, with proportionality constant K_P. In other words, K_I is the integral gain and K_P is the proportional gain of the controller. These parameters can be tuned to decide how fast and with what amplitude the PI-controller updates its control input based on the errors observed. We invite the reader to refer to [28] for further information on the design of this controller. In order to make the comparison with the MPC accurate, we consider that the control input provided by the PI-controller is the longitudinal force too. In the simulations, these control inputs are updated with a period of 1s.

Contrary to the MPC, the PI-controller does not include any predictive information about the route or the passengers, but can only apply reactive control based on the current errors observed. Note also that the state constraints introduced in the modelling section cannot be directly included in the design of this controller. Instead, they are enforced in the simulation framework, in case the PI-controller returns a control input which is not feasible given the current state of traffic.

C. Performance Metrics

In order to assess the regularity of bus service, we use the squared coefficient of variation of headways, noted CV^2, as a performance indicator. It can be defined as:

$$CV^2 = \frac{\sigma_{hw}^2}{\mu_{hw}^2} \tag{20}$$

for any given set of headways, where μ_{hw} and σ_{hw} denote the sample mean and sample standard deviation of this set, respectively. Lower values of CV^2 therefore correspond to stable headways, up to perfectly balanced headways if $CV^2 = 0$. In addition, one can show that the average passengers waiting time is directly proportional to CV^2 if the passengers arrive at stops following a Poisson process [42], so that CV^2 can be used as a proxy to monitor passengers waiting times at stops. Since transit agencies ultimately aim to provide good service to their passengers, we focus on the set of headways observed at each bus stop in the simulations, i.e. what the passengers would be experiencing in practice. This set of observed headways is used to compute CV^2 according to (20).

Transit agencies might also value the commercial speed of their vehicles as it can e.g. affect the waiting times of passengers already on board. In fact, there is a trade-off between regular and short headways since the intervention methods studied are built around preventing some buses from traveling as fast as possible along the route, either by slowing them down or by holding them at stops. The commercial speed is monitored by looking at the sample mean μ_{hw} of the observed headways, as it correlates directly with the average speed of the buses.

In addition to these two passengers-related metrics, the amount of battery energy consumed to enforce the different control methods is also studied.

D. Experiments

Several scenarios are generated, each with different initial spacings of the buses, as a way to evaluate the control strategies in different operational settings. Namely, these 10 scenarios are meant to cover most levels of bus bunching. They are ordered from 1 to 10 based on the mean deviation from homogeneous spacings that the buses have at the beginning of the simulations. Scenarios 1 and 10 represent two extreme situations: buses with homogeneous spacings and buses bunched into one broad cluster, respectively. The initial state of the bus line at the start of the simulations for every scenario can be found in Table I. Each scenario is simulated 5 times in order to provide an averaged representation. For each simulation, the same sample of traffic-related disturbances is used by all control methods to compare them on an equal footing.

The number of shooting points for each MPC horizon is chosen as $N = 200$. This is enough for the Runge-Kutta method to simulate the bus dynamics with a good accuracy over potentially long horizons, such as those that come up in the last few scenarios. Increasing the number of shootings

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

TABLE I

INITIAL POSITIONS OF THE BUSES AND CORRESPONDING AVERAGE DEVIATIONS FROM HOMOGENEOUS SPACINGS FOR ALL SCENARIOS. ALL THE VALUES BELOW ARE GIVEN IN METERS

	Bus 1	Bus 2	Bus 3	Bus 4	Bus 5	Bus 6	Bus 7	Bus 8	Mean deviation
Scenario 1	50	2111.5	4173	6234.5	8296	10357.5	12419	14480.5	0
Scenario 2	100	2300	4300	6200	8500	10600	12600	14500	46
Scenario 3	1000	3500	5250	6900	8800	12945	14745	15635	314
Scenario 4	1000	3000	4500	6000	9700	14000	14900	15700	421
Scenario 5	100	1000	5000	7300	11000	13900	14900	15500	446
Scenario 6	1350	5800	6350	7820	9300	9750	15000	15950	596
Scenario 7	1755	8305	9025	13200	15300	15500	15900	16385	741
Scenario 8	100	900	1350	1725	2060	9350	10650	14325	796
Scenario 9	250	650	4000	4250	4920	13100	13420	13620	924
Scenario 10	3250	4700	5200	5800	6200	6950	7300	7850	1319

TABLE II

PERFORMANCE INDICATORS FOR DETERMINISTIC SIMULATIONS OF SCENARIO 5 WITH DIFFERENT PARAMETERS α AND β FOR THE MPC. THE HEADWAY-RELATED INDICATORS ARE AVERAGED OVER ALL BUS STOPS

α	β	CV^2	μ_{hw}[s]	Energy [kWh]
0.5	0.278	0.053	325	267
	2.78	0.054	328	260
	27.8	0.127	330	256
2	0.278	0.029	331	264
	2.78	0.025	330	262
	27.8	0.033	337	250
20	0.278	0.029	347	249
	2.78	0.029	348	249
	27.8	0.021	351	244
Holding		0.056	320	287

points in these extreme cases only marginally improves the accuracy, while increasing the computation time, thus motivating our choice for N.

Likewise, the parameters α and β, which regulate the trade-off between the objective terms in the optimization problem solved by the MPC, are kept fixed for all simulations. They are calibrated by running the MPC in an intermediately bunched environment (scenario 5) and in fully deterministic settings for several different values of these parameters. Table II provides the performances obtained for several parameters pairs. The one which achieves the best CV^2 score while keeping μ_{hw} within 10 seconds of what is achieved by the holding baseline is chosen. This sets a bound on how slowly the MPC is allowed to operate the buses on average, since better CV^2 scores and better energy savings could potentially be achieved at the price of slowly-traveling buses. Transit agencies may adjust these parameters accordingly, depending on what they value the most. In what follows, we then assume that $\alpha = 2$ and $\beta = 2.78$ s/kW.

The gains K_I and K_P of the PI-controller are tuned in a similar way. The same deterministic simulation settings are used, with the same requirement on the commercial speed of the buses. It was found that $K_I = 10$ and $K_P = 10$ achieved the best regularity performances. Note that the trade-off between headway regularity, commercial speed, and energy consumption is not reflected as explicitly when tuning these gains as it is when setting the weights in the objective function of the MPC, where each weight has a clear physical meaning. This in turn might make the design phase harder to handle for transit agencies when using a PI-controller. Table VII in the Appendix gathers the rest of the numerical values for the parameters used in the simulations.

E. Results

Fig. 6 presents the evolution of the performance indicators for scenario 5, which corresponds to an intermediate level of initial bunching. Note that unlike Fig. 6c and Fig. 6b, Fig. 6a displays the sample mean and sample standard deviation of the bus headways for only one simulation instance of scenario 5. This is meant to showcase the convergence profile of each method, but the observations below can be easily generalized to all other simulation instances.

As can be seen in Fig. 6a, all control strategies manage to dissipate the initial service irregularities and to reach roughly homogeneous headways after some time, despite the permanent disturbances coming from the passengers and from traffic. The MPC algorithm converges naturally to headways of around 5 minutes once regular service is restored, thereby confirming that a target headway of 5 minutes was a sensible choice for the baselines.

It can be observed on this figure that the controllers have different convergence profiles. Indeed, the MPC first leverages its adaptive feature to set higher headway commands to the buses during the first 30 minutes of simulation. This results in a strong initial increase of the average headways observed, but it enables a faster convergence to homogeneous headways as the standard deviation decreases much earlier than for the baseline buses. Since the holding controller makes the buses travel at maximum speed, their headways remain low on average. However, it takes a longer time to dissipate the initial bunching since all the buses must reach the control points first, and wait there for possibly long periods of time. In addition, the standard deviation of the headways of the MPC-controlled buses is consistently lower than that of the baseline buses during the last hour of simulation. One of the main reasons for this is that the MPC relies on its models to anticipate the expected quantities of passengers at upcoming stops. It can thereby take preemptive action to slow down or speed up buses accordingly, hence keeping low headway variations. The PI-controller, on the other hand, only reacts to current observations, which results in a higher headway variability.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

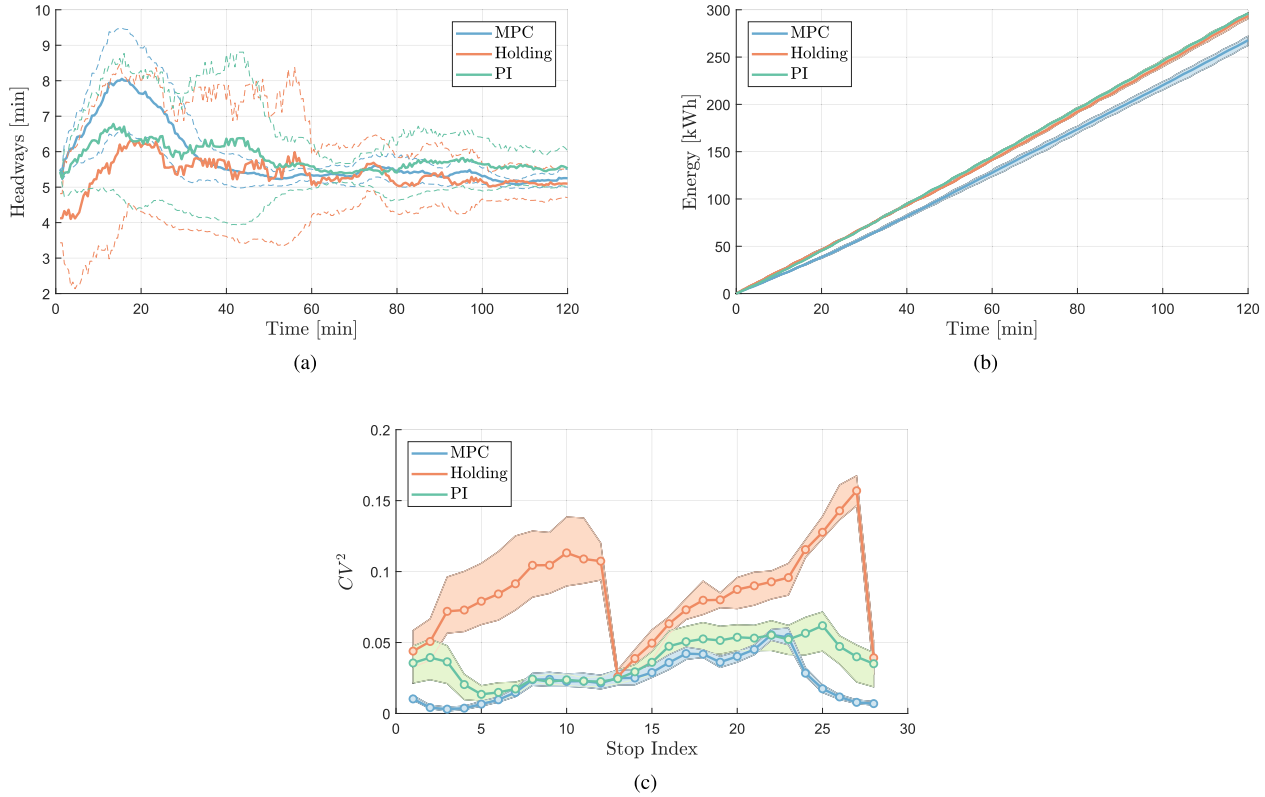

Fig. 6. Results for scenario 5. In these figures, the shaded areas are bounded above and below by the maximum and minimum metrics values observed among all the simulations of scenario 5 for the control strategy concerned. (a) Temporal evolution of the average headways at stops, for one simulation instance of this scenario (solid lines). The dashed lines are plotted one sample standard deviation away from the average lines to indicate the current dispersion of headways in this simulation instance. (b) Average total energy consumption of the buses over time (solid lines). (c) Average squared coefficient of variation of headways CV^2 at each of the stops (solid lines).

In addition, it must be noted that these two control methods are able to react immediately if any bus starts falling behind schedule, while the holding controller has to wait for the concerned bus to reach the next control point. This accounts in part for the better headway regularity performances of the MPC and PI-controller over the holding baseline.

Fig. 6c comes as the logical consequence of the previous observations. It displays the CV^2 score at each bus stop for the controllers. The CV^2 score of the MPC is consistently lower than that of the holding baseline at nearly every bus stop across all simulation instances, and is lower than that of the PI-controller at about half of the bus stops, both CV^2 scores being roughly the same at the other half. In other words, the MPC algorithm achieves more stable headways at stops, according to (20), which translates directly into lower average passengers waiting times, as explained previously. It can also be seen on this figure that the holding baseline dispatches buses most regularly at the two control points (which have indices 13 and 28), as expected. The few stops located right after the control points benefit from the regular incoming flow of buses, and have relatively low CV^2 scores too. The regularity of the uncontrolled baseline buses eventually worsens until the next control point is reached, hence making CV^2 adopt a characteristic sawtooth pattern. Similarly, the CV^2 scores of the PI-controller have a similar shape than that of the CV^2 scores of the MPC, indicating that headway regularity may be harder to enforce locally at some of the stops. Finally, Fig. 6b showcases the better energy efficiency of our control algorithm when compared with both baselines.

TABLE III

HEADWAY-RELATED INDICATORS, AVERAGED OVER ALL BUS STOPS, FOR ALL SCENARIOS

	CV^2			μ_{hw}[s]		
Scenario	MPC	PI	Holding	MPC	PI	Holding
1	0.005	0.005	0.017	319	331	309
2	0.004	0.004	0.017	316	330	308
3	0.020	0.033	0.064	332	338	318
4	0.027	0.035	0.063	336	342	320
5	0.024	0.037	0.087	341	346	328
6	0.037	0.052	0.148	339	345	329
7	0.058	0.097	0.233	356	354	346
8	0.107	0.210	0.421	376	372	363
9	0.135	0.187	0.355	362	363	359
10	0.408	0.540	0.801	428	406	408

The same general patterns, in terms of headway regularization and energy consumption, are also found when investigating the other scenarios. Table III gathers the CV^2 scores and the average headways for all the scenarios. As expected, these two indicators increase with the strength of the initial bunching since the controllers need to exert a stronger slowing control on the buses in that case, and regular headways are thereby restored later in the simulation. The CV^2 scores observed

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

are orders of magnitude apart across scenarios, showing how strongly bunching can affect the quality of the service provided by the bus line.

Another clear pattern emerges from these results: buses controlled by the MPC algorithm are slightly slower on average than those controlled by the holding method, but they manage to achieve much better CV^2 scores. Indeed, the MPC-controlled buses may have 10 to 20 seconds longer headways on average, but their CV^2 scores are 2 to 4 times lower. This essentially means that the MPC regularizes headways faster and more consistently than the holding baseline. Ultimately, this results in lower passengers waiting times at the price of a slightly lower commercial speed. The conclusion is a bit different when comparing the MPC with the PI-controller. In scenarios with weak bunching, the two controllers have similar CV^2 scores, but the baseline buses are slower. The situation is reversed for scenarios with intermediate to strong bunching, where the commercial speed of the buses is comparable, but where the MPC achieves a better headway regularity than the PI-controller. Finally, it should noticed that the PI-controller too outperforms the holding baseline in terms of headway regularity, but at the price of a lower commercial speed. The MPC and PI-controller display clear similarities here again, but the predictive feature of the MPC enables it to systematically outcompete the PI-controller in terms of service performances.

Note that the MPC-controlled buses could have been made to travel faster with a different choice of trade-off parameters, albeit increasing their CV^2 score in doing so. However, it was observed that the general conclusion would have remained the same, even with different sets of parameter values.

Table IV displays the energy consumed by the buses for all control methods. The values presented are averaged over all simulation instances for any given scenario. It can be observed that the MPC algorithm consistently has a better energy efficiency than the two baselines, and enables energy savings of up to 9.3% over the best performing baseline. In addition, no baseline is clearly better than the other in terms of energy consumption. The PI-controller has a more intensive energy consumption for scenarios with strong bunching, while the reverse is true for scenarios with weak bunching. This observation may seem surprising since the holding-controlled buses always travel at the maximum speed, while the PI-controlled has the ability to adjust the speed of the buses in operation. However, it is useful to remember here that the holding baseline is allowed to have buses dwell indefinitely at stops, where they do not consume any energy. On the other hand, both the MPC algorithm and the PI-controller require buses to constantly be on the move when not picking passengers up. This difference in the intervention strategy of each method explains our previous observation. This is also why the performance indicator monitored is the total energy consumption rather than e.g. the energy consumption per distance unit. It accounts for the intervention difference between the two classes of methods, and aims at giving some indications of the actual energy savings that can be expected for the transit agency when implementing an energy-aware velocity control method over generic baselines.

TABLE IV

TOTAL ENERGY CONSUMED BY THE BUSES IN ALL SCENARIOS (IN kWh). THE ENERGY SAVINGS DENOTE THE ENERGY EFFICIENCY IMPROVEMENT OF THE MPC OVER THE BEST PERFORMING BASELINE IN EACH SCENARIO

Scenario	MPC	PI	Holding	Energy savings
1	275	295	302	6.8%
2	278	297	304	6.4%
3	265	291	294	8.9%
4	263	290	292	9.3%
5	257	285	282	8.9%
6	264	289	284	7.0%
7	251	282	271	7.4%
8	247	277	261	5.4%
9	252	279	267	5.6%
10	242	273	246	1.6%

In the simulations, the MPC leverages its predictive feature to generate energy-efficient driving profiles for the buses, e.g. by decreasing their speed before steep downhill sections in order to empty their kinetic energy buffers, thus avoiding unnecessary braking. By doing so, it manages to compensate the longer dwell times of the holding-controlled buses, and to surpass the myopic commands of the PI-controller, even when it operates buses at a lower commercial speed, since it lacks an eco-driving component. In addition, Table IV suggests that the expected energy savings depend on the initial bunching strength. For scenarios with strong bus bunching, the holding controller holds buses for a longer part of the simulation in order to regularize headways, thus consuming less energy compared with the MPC. This also explains why the holding controller starts having a lower energy consumption than the PI-controller as the bunching strength increases. Therefore, it can be concluded that the expected energy savings from the MPC are the largest in the case of intermediate to no bunching.

F. Reaction to a Major Perturbation in the Service

The previous series of experiments investigated how the control methods fare when recovering from bus bunching during normal operations, but one may go one step further and see how they each adapt to a major disturbance (one bus breaking down) during the recovery process.

Some simulations of scenario 5 in which the last bus of the line is removed after one hour of simulation time are run. The control methods have mostly stabilized the headways by that time, so that the effects of the bus breakdown can be clearly seen. The simulations are then carried on as before for one more hour. To be fair to the holding strategy and to the PI-controller, the target headway is increased to 6 minutes when the breakdown occurs. This is meant to provide them with a reasonable goal, since only 7 buses are then available. This scenario is simulated 5 times for each controller, and the averaged results are presented in Table V.

It can be observed from Fig. 7a that all control methods slow the buses down right after the breakdown occurs. As in the previous experiments, the MPC temporarily increases the headway commands of the buses. It then decreases them again until the headways are stabilized around the new natural

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

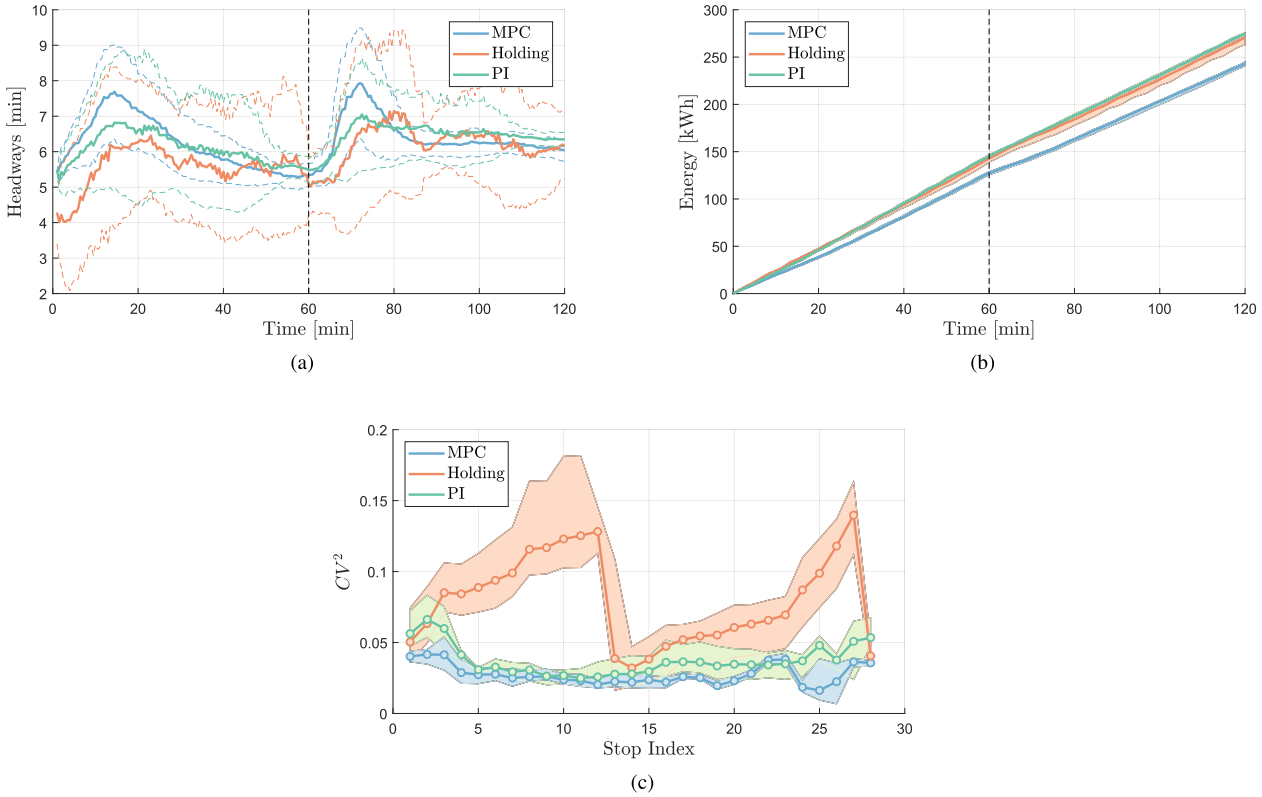

Fig. 7. Results for scenario 5 in the case where one bus breaks down after one hour of simulation (dashed vertical lines). As before, the shaded areas are bounded by the most extreme metrics values observed. (a) Average headways at stops for one simulation instance (solid lines). (b) Average total energy consumption. (c) Average squared coefficient of variation of headways CV^2 at each of the stops (solid lines).

TABLE V

AVERAGE PERFORMANCE INDICATORS FOR SCENARIO 5 IN THE CASE OF A BUS BREAKDOWN HALFWAY THROUGH THE SIMULATION

	MPC	PI	Holding
CV^2	0.027	0.037	0.087
μ_{hw} [s]	376	376	357
Energy [kWh]	243	275	269

headway of the bus line (around 6 minutes). It must also be noted from this figure that all methods manage to converge to stabilized headways faster than when dissipating the initial bunching. The reason for this is that the breakdown happens when buses already have roughly homogeneous headways, thus creating only one large gap in service, which is then easier to bridge. Fig. 7b and Fig. 7c present the energy consumption over time and the CV^2 scores at the stops, respectively. The patterns are similar to those of Fig. 6b and Fig. 6c, with the exception that the overall energy consumption now increases more slowly after one bus is removed from the line.

The headway sample means presented in Table V are larger than those obtained in the previous version of scenario 5, which was expected since less buses are running during the second half of the simulation. Likewise, the overall energy consumption is lower than it was previously, but the energy savings from the MPC remain similar (9.7% against 8.9% before). However, the CV^2 scores are nearly the same as they were before. This may seem counter-intuitive since the system is now subject to additional disturbances, but it is good to recall here that the sample mean of the headways appears in the definition of CV^2 in (20). In this particular case, the increase of the headway variability thus seems to be offset by the increase of the headway sample mean. But as far as the comparison in the performances of the controllers is concerned, the same general conclusions as for the previous series of experiments can be drawn from this modified version of scenario 5.

V. CONCLUSION

This paper developed a model and a velocity control strategy for a line of electric buses. Thanks to a variable change to the space domain in the modeling step, the impractical complexity of mixed-integer problems could be avoided by modeling bus stops without resorting to additional integer variables. The optimal control problem assembled from the model could then be reformulated into a smooth NLP, after additional variables had been added to lift the problem and get rid of its discontinuities. In contrast to other studies, the choice of variable horizons to the preceding buses made it possible to regularize time headways directly in the NLP instead of having to use an indirect proxy for headway regularity, such as bus spacings. By leveraging some powerful results from parametric optimization, a bilevel optimization scheme was

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

proposed to decompose and solve this NLP, which was then embedded in an MPC to enforce closed-loop control.

Extensive simulations were carried out to investigate the headway regularization and energy savings performances of the proposed MPC strategy for different degrees of bus bunching in a real bus line. It was found that the MPC systematically achieved better headway regularity when compared with a classical holding baseline but had a slightly lower commercial speed on average. In addition, the predictive feature of the MPC enabled it to outcompete a simple PI-controller with a similar speed control intervention strategy. Consequently, the MPC was able to provide a more reliable service to passengers and to lower their waiting times at stops. The MPC was also able to recover from bunching faster than the baseline controllers by temporarily slowing down the buses to allow any late bus to catch up on schedule. These observations were found to be consistent across scenarios with various bunching strengths, and also in a scenario where the breakdown of one of the buses in operation acts as a major line disturbance. Due to the longer dwell times at stops of holding-controlled buses in highly-bunched settings, the energy savings of the MPC were highest for low and intermediate levels of bus bunching. We report savings of up to 9.3% in such favorable cases, which the MPC was able to achieve by adopting energy-efficient driving strategies when adjusting the bus velocities.

This work aimed to demonstrate that the bus fleet eco-driving problem could be treated in conjunction with the bus line regularity control problem, and formulated in a framework that enables its real-time implementation. Indeed, large-scale NLPs can generally be solved in real-time with purpose-built solvers, as demonstrated in [44]. The proposed control strategy is scalable, since the bus-level NLPs can be solved in parallel aboard buses, and adaptive, as it can be deployed on any bus line without requiring any prior target headway. In addition, the limited quantity of information that needs to be exchanged between the buses and the central node makes the approach robust to losses or noisy communication data.

The framework presented in this paper could be adapted to include additional charging-related constraints on top of the energy minimization objective. The scheduling of the charging decisions could then be investigated by including a limited driving range for the electric buses, which could be complemented well with a stochastic MPC to include a more thorough modeling of the external disturbances. The real-time implementability of the problem could also be addressed by designing an efficient solver tailored for that purpose. Other future research directions of interest include considering bus capacity constraints, and extending our approach to an entire bus network, with several lines interacting in shared corridors.

APPENDIX

A. Proof of Proposition 1

The lifted NLP (15) only differs from the original problem (12) through the modified torque constraints (13) and torque expression (14). A careful examination of the torque constraints (5)-(6) and (13) is enough to conclude that they define the same feasible set for the longitudinal force and the torque in both problems, and thus do not affect optimality. It remains to show that the different expressions for the torque (5) and (14) do not affect optimality either. A sufficient condition for this is if the same longitudinal force values yield the same torque values in both problems. In that case, both the original and the lifted problems would behave the same, and hence have the same optimal solution. This requires an additional assumption, however.

Assumption 1: The battery power $P_{b,i}$ is a monotonically increasing function in the motor torque $T_{m,i}$.

Note that Assumption 1 is not very restrictive, as it merely states that a higher motor torque systematically causes more energy to be drawn from, or supplied to, the battery. We can now prove the following proposition.

Proposition 2: Under Assumption 1, any value $F_{m,i,k}$, $k \in \mathbb{I}_{[0,N-1]}$, $i \in \mathbb{I}_{[1,n]}$, of the longitudinal force yields the same motor torque value in both the original and the lifted version of the NLP.

Proof: First, let us observe from (5) and (14) that the torque value is the same if and only if $F_{t,i,k}$ and $F_{g,i,k}$ are mutually exclusive (i.e. they cannot be non-zero simultaneously) in the lifted NLP, where $F_{m,i,k} = F_{t,i,k} - F_{g,i,k}$ holds.

Since $\eta_f < 1$, it can be noticed in the torque expression (14) that $r_w/(\eta_f M_f) > \eta_f r_w/M_f > 0$. It follows from this observation that:

$$T_{m,i,k} \geq \frac{r_w}{\eta_f M_f} F_{m,i,k} \quad \text{if } F_{m,i,k} > 0, \tag{21a}$$

$$T_{m,i,k} \geq \frac{\eta_f r_w}{M_f} F_{m,i,k} \quad \text{if } F_{m,i,k} \leq 0, \tag{21b}$$

where $T_{m,i,k}$ is defined according to (14). Note that the lower bounds in these inequalities are reached when either $F_{t,i,k}$ or $F_{g,i,k}$ is zero, depending on the scenario. According to Assumption 1, the battery power $P_{b,i}$ is minimized for the lowest feasible torque, which here corresponds to the case where $F_{t,i,k}$ or $F_{g,i,k}$ are mutually exclusive. In other words, any solution for which the inequality in (21) is strict would require more energy while delivering the same longitudinal force. Consequently, the torque value in the lifted NLP is set to the same value as in the original NLP for any $F_{m,i,k}$. □

Since Proposition 2 holds, the optimal solution of the two problems is the same, which concludes the proof of Proposition 1.

B. Resolution of the Decomposed Problem (16)-(17)

This appendix presents how the high-level NLP (16) can be solved by deploying an SQP algorithm. Under the assumptions that Linear Independent Constraint Qualification (LICQ) and Second Order Sufficient Condition (SOSC) hold [39], the Newton steps taken by the SQP algorithm can be computed by solving the following quadratic program (QP):

$$\min_{H^+} \sum_{i=1}^{n} \frac{1}{2}\Lambda_i {H_i^+}^2 + \alpha(H_i^+ - H_{i-1}^+)^2 + \beta \hat{V}_{i,H_i}(H_i^+), \tag{22a}$$

$$\text{s.t. } H_i^+ \in [H_i^{\min}, H_i^{\max}], \quad i \in \mathbb{I}_{[1,n]}, \tag{22b}$$

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

TABLE VI

SUMMARY OF THE MAIN NOTATIONS USED IN THE PAPER (APPENDICES EXCLUDED). THE SUBSCRIPTS i, l, AND k CONSISTENTLY REFER TO THE BUS WITH INDEX i, THE STOP WITH INDEX l, AND THE SHOOTING POINT WITH INDEX k, RESPECTIVELY

Variable	Definition
s_i	Bus position
v_i	Bus velocity
m_i	Bus mass
$F_{m,i}$	Longitudinal force
$F_{b,i}$	Braking force
θ	Road gradient
E_i	Kinetic energy per mass unit
t_i	Travel time
x_i, u_i	State and control input vectors for bus i
x, u	State and control input vectors for all buses
$\Delta_{\text{stop},i}$	Dwell time at stop
$v_{\min}$	Lower bound of the bus velocity
$v_{\max}$	Upper bound of the bus velocity
s_l	Stop position
$\omega_{m,i}$	Motor speed
$T_{m,i}$	Motor torque
$P_{b,i}$	Battery power
p_i	Initial bus position
H_i	Predicted headway with the preceding bus
t^0	Initial simulation time
$\mathcal{S}_i$	Set of stops on the horizon of bus i
q_i	Number of stops on the horizon of bus i
λ_l	Passengers arrival rate at stop l
t_{i+1}^j	Time when the preceding bus left stop i_j on the horizon of bus i
m_i^0	Initial bus mass
m_i^j	Mass of bus i when leaving stop j on its horizon
μ_l	Alighting proportion of passengers at stop l
Λ_i	Weighting ratio based on passengers rates on the horizon of bus i
E_i^0	Initial kinetic energy
Δs_i	Length of each shooting interval on the horizon of bus i
$s_{i,k}$	Position of the k-th shooting point on the horizon of bus i
X_i, U_i	Vectors of state and control optimization variables for bus i
X, U	Vectors of state and control optimization variables for all buses
F	Numerical integration of the state dynamics
J	Numerical integration of the battery power
g	Inequality constraints
$F_{t,i}$	Longitudinal force in traction
$F_{g,i}$	Longitudinal force in generation
$\tilde{J}$	Lifted version of J
$\tilde{g}$	Lifted version of g
V_i	Optimal cost of the bus-level NLP for bus i
$\text{dom}(V_i)$	Feasible set of the bus-level NLP for bus i
$H_i^{\min}$	Minimum terminal travel time of bus i on its horizon
$H_i^{\max}$	Maximum terminal travel time of bus i on its horizon
μ_{hw}	Sample mean of the observed headways
σ_{hw}	Sample standard deviation of the observed headways
CV^2	Squared coefficient of variation of headways

TABLE VII

NUMERICAL VALUES OF THE PARAMETERS USED IN THE SIMULATIONS IN SECTION IV

Parameter	Definition	Numerical value
n	Number of buses	8
L	Route length	16492 m
q	Number of bus stops	28
ρ	Air density	1.18 kg/m^3
A_{bus}	Bus frontal area	8.36 m^2
c_a	Aerodynamic drag coefficient	1
c_r	Rolling resistance coefficient	0.0047
v_{entry}	Bus speed around bus stops	1.39 m/s
r_w	Wheel radius	0.49 m
M_f	Final gear ratio	2.8
η_f	Efficiency coefficient of the final gear	0.98
$T_{\max}$	Maximum motor torque	5614 Nm
$P_{\max}$	Maximum power that the motor can supply continuously	290 kW
t_s	Time needed for a bus to stop at (or depart from) a stop	3 s
b	Passenger boarding time	1.5 s
m_{pax}	Average passenger mass	60 kg
m_{emp}	Mass of an empty bus	12000 kg
α	Trade-off coefficient weighing regular successive headways	2
β	Trade-off coefficient weighing energy consumption	2.78 s/kW
N	Number of sampling intervals on each bus horizon	200
T	MPC sampling time	30 s
$\sigma^2_{\text{traffic}}$	Variance of the deviation percentages from $v_{\max}$	10 m^2/s^2
K_I	Integral gain of the PI-controller	10
K_P	Proportional gain of the PI-controller	10

where $H^+ = [H_1^+, \ldots, H_n^+]^\top$ is the next primal solution, and where $\hat{V}_{i,H_i}$ is a quadratic approximation of V_i around the point H_i. Note that V_i is the only term that needs to be modified in (17) in order to obtain the local QP approximation (22), since the remaining terms in the objective are already quadratic functions, and all the constraints are linear.

In order to approximate V_i, let us first observe that each bus-level problem (17) is a parametric NLP, with a scalar parameter H_i, $i \in \mathbb{I}_{[1,n]}$. Therefore, the implicit function V_i is the parametric optimal cost function of the bus-level problem for bus i [40]. Likewise, we can define the primal-dual solution of (17) as an implicit function of H_i, and note it z_i. Some results from parametric optimization may now be used to find an expression for $\hat{V}_{i,H_i}$.

The bus-level NLP (17) for bus i can be solved for any fixed parameter value H_i by using primal-dual interior point algorithms [39]. The relaxation of the KKT conditions used by these algorithms creates a smooth modified version of the problem. If LICQ and SOSC also hold at the solution $z_i(H_i)$, then the parametric functions V_i and z_i are twice continuously differentiable locally. The implicit function theorem can now be applied to compute the first-order variations of V_i around H_i as:

$$\left.\frac{dV_i}{dH_i}\right|_{H_i} = \left.\frac{\partial \mathcal{L}_i}{\partial H_i}\right|_{z_i(H_i),H_i} \tag{23}$$

where $\mathcal{L}_i$ is the Lagrange function of the NLP [40]. The second-order variations of V_i can then be computed through a simple application of the chain rule to (23):

$$\left.\frac{d^2V_i}{dH_i^2}\right|_{H_i} = \left(\frac{\partial^2 \mathcal{L}_i}{\partial H_i^2} + \nabla^2_{H_i z_i}\mathcal{L}_i \frac{dz_i}{dH_i}\right)_{z_i(H_i),H_i} \tag{24}$$

where this function is well-defined everywhere thanks to the smoothing of the problem mentioned earlier. As a result, the following Taylor approximation of V_i may be used in (22):

$$\hat{V}_{i,H_i}(H_i^+) = V_i(H_i) + \left.\frac{dV_i}{dH_i}\right|_{H_i}\Delta H_i + \frac{1}{2}\left.\frac{d^2V_i}{dH_i^2}\right|_{H_i}\Delta {H_i}^2, \tag{25}$$

where $\Delta H_i = H_i^+ - H_i$.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: SQP Procedure for Solving the Decomposed Optimization Problem (16)-(17). C Denotes Centrally-Run Computations, While i Denotes Computations Carried Out Aboard Bus i

i: solve NLPs (18) and (19) and send $\{H_i^{\min}, H_i^{\max}\}$ to central node
C: initialize H, H^+
while $\|H^+ - H\|_2 >$ Tol **do**
  C: $H \leftarrow H^+$ and send H to buses
  for $i \in \mathbb{I}_{[1,n]}$ **do**
    i: $\hat{V}_{i,H_i} \leftarrow$ solve NLP (17), then (25)
    i: send $\hat{V}_{i,H_i}$ to central node
  end
  C: $H^+ \leftarrow$ solve QP (22)
end
C: $H \leftarrow H^+$ and send H to buses
$\forall i$: X_i, $U_i \leftarrow$ solve NLP (17)
return H, X_1, U_1, …, X_n, U_n

The computation of the term $\frac{dz_i}{dH_i}$ in (24) is not straightforward as it requires the first-order derivatives of the KKT conditions [43]. However, they can generally be obtained at a small computational cost. Computing the rest of the terms in (23) and (24) is significantly easier. The quadratic approximation (25) can consequently be used in (22) when running the SQP algorithm. Note that the primal-dual solution of the bus-level problem is needed each time (25) is computed around a new parameter value. Therefore, the NLP (17) must be solved again each time a Newton step is taken by solving (22). These two problems are then solved sequentially until convergence of the SQP method. The complete resolution procedure is summarized in Algorithm 2, where the computations are assumed to be distributed between the buses and a central node, as illustrated in Fig. 4.

As the main computational bottleneck lies in solving the bus-level NLPs, only few computations are needed at the central node. It can also be observed that very few variables need to be exchanged between the central node and the individual buses during the procedure. Consequently, the communication loads remain very small.

Note that since the high-level NLP (16) is non-convex, the SQP method deployed in Algorithm 2 converges to a local minimum, which may or may not be the global solution of the problem. In general, SQP methods have robust convergence properties, even from remote starting points. Their convergence can notably be improved by using e.g. quasi-Newton approximations, in case of ill-posed subproblems, or backtracking line search methods [39]. However, we observed that taking these precautions was not necessary to guarantee the convergence of Algorithm 2 in practice. As a result, the convergence rate of the SQP method in that case is quadratic, under some LICQ and SOSC assumptions [39], which is the best that can be expected when solving this type of problems. In the simulations of Section IV, Algorithm 2 systematically reached convergence in 2 to 3 iterations in each of the scenarios studied.

REFERENCES

[1] A. Lajunen and T. Lipman, "Lifecycle cost assessment and carbon dioxide emissions of diesel, natural gas, hybrid electric, fuel cell hybrid and electric transit buses," *Energy*, vol. 106, pp. 329–342, Jul. 2016.

[2] A. Bigazzi, "Comparison of marginal and average emission factors for passenger transportation modes," *Appl. Energy*, vol. 242, pp. 1460–1466, May 2019.

[3] G. F. Newell and R. B. Potts, "Maintaining a bus schedule," in *Proc. 2nd Austral. Road Res. Board Conf. (ARRB)*, Melbourne, VIC, Australia, vol. 2, 1964, pp. 388–393.

[4] C. F. Daganzo, *Fundamentals of Transportation and Traffic Operations*. New York, NY, USA: Pergamon, 1997.

[5] A. Bar-Yosef, K. Martens, and I. Benenson, "A model of the vicious cycle of a bus line," *Transp. Res. B, Methodol.*, vol. 54, pp. 37–50, Aug. 2013.

[6] M. Gallet, T. Massier, and T. Hamacher, "Estimation of the energy demand of electric buses based on real-world data for large-scale public transport networks," *Appl. Energy*, vol. 230, pp. 344–356, Nov. 2018.

[7] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 71–83, Mar. 2007.

[8] M. Hovgard, O. Jonsson, N. Murgovski, M. Sanfridson, and J. Fredriksson, "Cooperative energy management of electrified vehicles on hilly roads," *Control Eng. Pract.*, vol. 73, pp. 66–78, Apr. 2018.

[9] N. Murgovski, B. Egardt, and M. Nilsson, "Cooperative energy management of automated vehicles," *Control Eng. Pract.*, vol. 57, pp. 84–98, Dec. 2016.

[10] S. Uebel, N. Murgovski, B. Baker, and J. Sjoberg, "A two-level MPC for energy management including velocity control of hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5494–5505, Jun. 2019.

[11] M. Held, O. Flardh, and J. Martensson, "Optimal speed control of a heavy-duty vehicle in urban driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1562–1573, Apr. 2019.

[12] N. Murgovski, L. Johannesson, J. Sjöberg, and B. Egardt, "Component sizing of a plug-in hybrid electric powertrain via convex optimization," *Mechatronics*, vol. 22, no. 1, pp. 106–120, Feb. 2012.

[13] N. Murgovski, L. M. Johannesson, and B. Egardt, "Optimal battery dimensioning and control of a CVT PHEV powertrain," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2151–2161, Jun. 2014.

[14] C. F. Daganzo, "A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons," *Transp. Res. B, Methodol.*, vol. 43, no. 10, pp. 913–921, Dec. 2009.

[15] J. Argote-Cabanero, C. F. Daganzo, and J. W. Lynn, "Dynamic control of complex transit systems," *Transp. Res. B, Methodol.*, vol. 81, pp. 146–160, Nov. 2015.

[16] W. Wu, R. Liu, and W. Jin, "Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour," *Transp. Res. B, Methodol.*, vol. 104, pp. 175–197, Oct. 2017.

[17] W. Wu, R. Liu, W. Jin, and C. Ma, "Simulation-based robust optimization of limited-stop bus service with vehicle overtaking and dynamics: A response surface methodology," *Transp. Res. E, Logistics Transp. Rev.*, vol. 130, pp. 61–81, Oct. 2019.

[18] G. Laskaris, O. Cats, E. Jenelius, M. Rinaldi, and F. Viti, "Multiline holding based control for lines merging to a shared transit corridor," *Transportmetrica B, Transp. Dyn.*, vol. 7, no. 1, pp. 1062–1095, Dec. 2019.

[19] G. Laskaris, M. Seredynski, and F. Viti, "Enhancing bus holding control using cooperative ITS," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1767–1778, Apr. 2020.

[20] C. F. Daganzo and J. Pilachowski, "Reducing bunching with bus-to-bus cooperation," *Transp. Res. B, Methodol.*, vol. 45, no. 1, pp. 267–277, Jan. 2011.

[21] J. J. Bartholdi, III, and D. D. Eisenstein, "A self-coördinating bus route to resist bus bunching," *Transp. Res. B, Methodol.*, vol. 46, no. 4, pp. 481–491, 2012.

[22] C. E. Cortés, D. Sáez, F. Milla, A. Núñez, and M. Riquelme, "Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 5, pp. 757–769, Oct. 2010.

[23] F. Delgado, J. C. Munoz, and R. Giesen, "How much can holding and/or limiting boarding improve transit performance?" *Transp. Res. B, Methodol.*, vol. 46, no. 9, pp. 1202–1217, Nov. 2012.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.

[24] D. Hernández, J. C. Muñoz, R. Giesen, and F. Delgado, "Analysis of real-time control strategies in a corridor with multiple bus services," *Transp. Res. B, Methodol.*, vol. 78, pp. 83–105, Aug. 2015.
[25] L. O. Seman, L. A. Koehler, E. Camponogara, L. Zimmermann, and W. Kraus, "Headway control in bus transit corridors served by multiple lines," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4680–4692, Nov. 2020.
[26] K. Ampountolas and M. Kring, "Mitigating bunching with bus-following models and bus-to-bus cooperation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2637–2646, May 2021.
[27] B. Varga, T. Tettamanti, and B. Kulcsár, "Optimally combined headway and timetable reliable public transport system," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 1–26, Jul. 2018.
[28] I. I. Sirmatel and N. Geroliminis, "Mixed logical dynamical modeling and hybrid model predictive control of public transport operations," *Transp. Res. B, Methodol.*, vol. 114, pp. 325–345, Aug. 2018.
[29] B. Varga, T. Tettamanti, and B. Kulcsár, "Energy-aware predictive control for electrified bus networks," *Appl. Energy*, vol. 252, Oct. 2019, Art. no. 113477.
[30] R. Lacombe, S. Gros, N. Murgovski, and B. Kulcsár, "Hierarchical control of electric bus lines," in *Proc. 21st IFAC World Congr.*, 2020, pp. 1–7.
[31] R. Scattolini, "Architectures for distributed and hierarchical model predictive control—A review," *J. Process Control*, vol. 19, no. 5, pp. 723–731, May 2009.
[32] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI, USA: Nob Hill Publishing, 2017.
[33] K. Gkiotsalitis and E. C. van Berkum, "An exact method for the bus dispatching problem in rolling horizons," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 143–165, Jan. 2020.
[34] W. Fan and R. B. Machemehl, "Do transit users just wait for buses or wait with strategies?" *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2111, no. 1, pp. 169–176, Jan. 2009.
[35] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems*, 3rd ed. Berlin, Germany: Springer-Verlag, 2013.
[36] E. Hans, N. Chiabaut, and L. Leclercq, "Investigating the irregularity of bus routes: Highlighting how underlying assumptions of bus models impact the regularity results," *J. Adv. Transp.*, vol. 49, no. 3, pp. 358–370, Apr. 2015.
[37] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Philadelphia, PA, USA: SIAM, 2010.
[38] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 763–768.
[39] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.
[40] G. Still, "Lectures on parametric optimization: An introduction," *Optim. Online*, pp. 1–50, 2018.
[41] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi—A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.
[42] S. J. Berrebi, E. Hans, N. Chiabaut, J. A. Laval, L. Leclercq, and K. E. Watkins, "Comparing bus holding methods with and without real-time predictions," *Transp. Res. C, Emerg. Technol.*, vol. 87, pp. 197–211, Feb. 2018.
[43] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 2567–2573.
[44] M. Vukov *et al.*, "Real-time nonlinear MPC and MHE for a large-scale mechatronic application," *Control Eng. Pract.*, vol. 45, pp. 64–78, Dec. 2015.

Rémi Lacombe received the M.S. degree in applied mathematics jointly from ENSTA Paris, France, and from the KTH Royal Institute of Technology, Sweden, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Chalmers University of Technology, Sweden. His main research interests include optimal control, nonlinear optimization, and their application to transportation systems.

Sébastien Gros received the Ph.D. degree from EPFL, Switzerland, in 2007. After a journey by bicycle from Switzerland to the Everest base camp in full autonomy, he joined the Research and Development Group, Strathclyde University, focusing on wind turbine control. In 2011, he joined KU Leuven, where his main research focus was on optimal control and fast NMPC for complex mechanical systems. In 2013, he joined the Department of Signals and Systems, Chalmers University of Technology, Göteborg, where he became an Associate Professor in 2017. He is currently a Full Professor at NTNU, Norway, and an Affiliate Professor at Chalmers. His main research interests include numerical methods, real-time optimal control, reinforcement learning, and the optimal control of energy-related applications.

Nikolce Murgovski received the M.S. degree in software engineering from University West, Trollhättan, Sweden, in 2007, and the M.S. degree in applied physics and the Ph.D. degree in systems and control from the Chalmers University of Technology, Gothenburg, Sweden, in 2007 and 2012, respectively. He is currently an Associate Professor with the Department of Electrical Engineering, Chalmers University of Technology. His research interests include optimization and optimal control in the automotive area.

Balázs Kulcsár received the M.Sc. degree in traffic engineering and the Ph.D. degree from the Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 1999 and 2006, respectively. He has been a Researcher/Post-Doctor with the Department of Control for Transportation and Vehicle Systems, BUTE, the Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN, USA, and the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden. His main research interest focuses on traffic flow modeling and control.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on October 17,2022 at 08:59:53 UTC from IEEE Xplore. Restrictions apply.