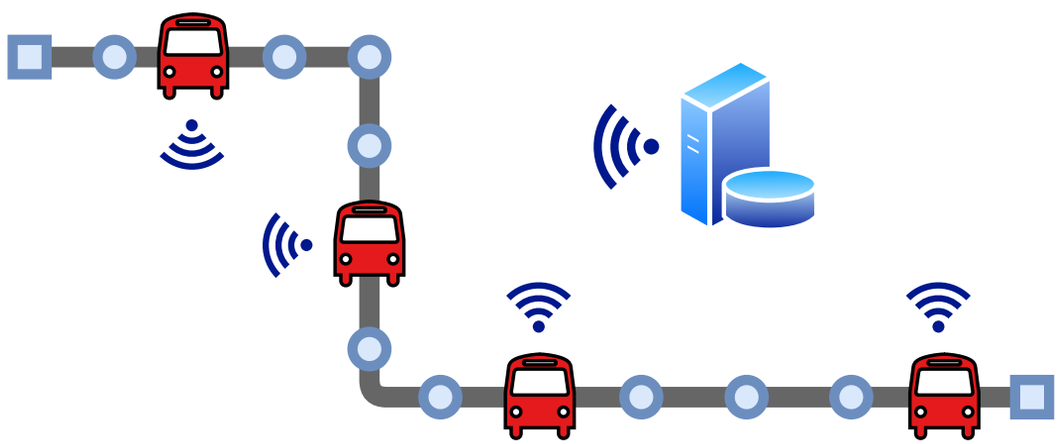
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

LICENTIATE THESIS

RÉMI LACOMBE • Optimal Control of Electric Bus Lines • 2021



Optimal Control of Electric Bus Lines

RÉMI LACOMBE

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Optimal Control of Electric Bus Lines

Rémi Lacombe

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2021

Optimal Control of Electric Bus Lines

Rémi Lacombe

This thesis has been prepared using LaTeX.

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, 2021

To my friends and family

Abstract

Bus lines are inherently unstable systems, where any delay tends to be further amplified by the accrued passenger loads encountered at stops downstream. This self-reinforcing mechanism, when combined with the multiple sources of disturbances of an urban environment, can lead to the problem of bus bunching. To mitigate this, various types of control strategies have been proposed and some are routinely employed by transit agencies around the globe to improve service regularity. They range from simple rule-based ad-hoc solutions, to elaborate real-time prediction-based bus velocity control. However, most of these strategies only focus on service-related objectives, and often disregard the potential energy savings that could be achieved through the control intervention. Velocity-based control, in particular, is very suitable for eco-driving strategies, which can increase the energy efficiency of the transit system by adjusting the planned velocity trajectories of the vehicles based on the road and traffic conditions.

This thesis proposes a scalable resolution method for the bus line regularity and eco-driving optimal control problem for electric buses. It is shown how this problem can be recast as a smooth nonlinear program by making some specific modelling choices, thus circumventing the need for integer decision variables to capture bus stop locations and avoiding the infamous complexity of mixed-integer programs. Since this nonlinear program is weakly coupled, a distributed optimization procedure can be used to solve it, through a bi-level decomposition of the optimization problem. As a result, the bulk of the computations needed can be carried out in parallel, possibly aboard each individual bus. The latter option reduces the communication loads as well as the amount of computations that need to be performed centrally, which makes the proposed resolution method scalable in the number of buses. Using the concept of receding horizons to introduce closed-loop control, the optimized control trajectories obtained were applied in a stochastic simulation environment and compared with classical holding and velocity control baselines. We report a faster dissipation of bus bunching by the proposed method as well as energy efficiency improvements of up to 9.3% over the baselines.

Keywords: Bus bunching, Electric buses, Optimal control, Eco-driving, Nonlinear programming, Model predictive control.

List of Publications

This thesis is based on the following publications:

[A] **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár, "Hierarchical control of electric bus lines". Accepted in 21st *IFAC World Congress*, Berlin, Germany, Jul. 2020.

[B] **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár, "Distributed optimization for bunching mitigation and eco-driving of electric bus lines". Submitted to *IEEE Transactions on Intelligent Transportation Systems (ITS)* in Nov. 2020.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Balázs Kulcsár, for his constant support, help and guidance throughout my PhD studies so far. I would also like to thank my two co-supervisors, Prof. Nikolce Murgovski and Prof. Sébastien Gros, for their significant help to my research work, through their great expertise and their frequent feedback. I am very grateful to all of you for the fruitful discussions that we have had over the course of this project.

Acronyms

OCP:	Optimal Control Problem
MPC:	Model Predictive Control
NLP:	Nonlinear Program
SQP:	Sequential Quadratic Programming
EM:	Electric Machine
QP:	Quadratic Program
PI:	Proportional-integral

Contents

Part I

Overview

CHAPTER 1

Introduction

1.1 Motivation

Due to the estimated continued increase in the global population and in urban sprawl, passenger-kilometers in urban areas are projected to double between 2015 and 2050 [1]. At the same time, if the current observed trends in urban planning continue, the share of private vehicles used in urban mobility is projected to decrease from 70% in 2015 to around 40% in 2050 [1]. Private vehicles are responsible for an overwhelmingly large share of the total CO_2 emissions of the urban transport sector. Since shared mobility and traditional public transports are projected to gradually take over as the dominant urban transit option, the CO_2 emissions per passenger-kilometer are expected to decrease as well [1]. These projections rely partly on expected improvements in public transport services, to explain their growing share in urban mobility. In this context, it seems desirable to pursue the goal of improving the performances of public transit in order to accelerate the rate at which private vehicles phase out of cities.

In the case of urban bus transit, electric buses appear today as the most promising way to mitigate greenhouse gas emissions and reduce energy con-

sumption. Several lifecycle assessment studies have found fully-electric buses to perform better on these metrics than buses powered by other fuel types. As with other types of electric vehicles, the carbon-intensity of the electricity grid mix used to power the buses plays an important role in their environmental impacts [2]. However, electric buses manage to achieve lower emission levels and a higher energy efficiency in regions with both a high and low carbon-intensity of the electricity grid mix like China [3], and like Finland or California [4], respectively. In addition, the collapse in lithium-ion battery prices throughout the last decade has now made electric buses economically competitive with diesel buses [4], [5].

The energy efficiency of a fleet of vehicles in operation can be further improved by control strategies with a focus on *eco-driving* [6]. The latter denotes the procedure of following the driving profiles which minimize the energy consumption of the vehicles. Eco-driving has the advantage of not requiring any structural changes of the vehicles while still having the potential of yielding significant energy savings, and is usually treated as an *optimal control problem* (OCP) [6]. As the possible trajectories of the vehicles are predicted over a certain horizon, the available information on the road gradient or the speed limits can be used to choose the best driving profiles. With this information, the speed of the vehicles can be adjusted to avoid dissipating energy through unnecessary braking or to harness the acceleration from the gravitational pull in downhill sections. Eco-driving controllers are more frequently used for driving missions on highways, where vehicles are subject to relatively few disturbances [7]–[10]. Urban settings, on the other hand, are considerably more difficult to handle for eco-driving controllers since intersections, pedestrians, and the surrounding traffic represent significant sources of disturbances [11], [12]. In particular, eco-driving strategies have not yet been studied for a line of city buses.

Bus lines have long been known to be unstable systems [13]. If left uncontrolled, they can be strongly affected by the multiple sources of disturbances found in a typical urban environment. 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 than expected at stops, and may eventually catch up with the preceding bus, at which point the buses start traveling together. The increased service delays incurred by this so-called *bus bunching* phenomenon

may then significantly increase the passenger delays, which may eventually discourage users from choosing to use public transport [14]. Part of this instability problem can be mitigated upstream of the operational stage, during the transit network planning. For instance, transit agencies can adjust the service frequency, or the scheduling of the vehicles, during a prior tactical planning phase [15]. This is not enough, however, to fully compensate for the stochastic nature of urban bus operations and avoid bus bunching. To this end, operational control strategies must also be deployed.

Station control strategies are the operational control strategies which have received the most attention from the research community by far [16]. Among them, holding strategies are the most studied and the most implemented. As their name suggests, they consist of holding one or more buses at a subset of bus stops, called control points, in order to prevent or dissipate service irregularities. By doing so, early buses can for example be prevented from catching up with the bus preceding them. The control law according to which holding commands are computed can be schedule-based [17], [18], headway-based [19]–[21], or based on a direct estimate of the passenger waiting times [22]–[24]. Regardless of which type they are, holding strategies incur additional dwell times at stops, and as such decrease the average commercial speed of the bus line, while also increasing the travel times of the subset of passengers aboard the buses being held. This consideration illustrates that, in general, the operational bus line control problem is a multi-objective problem which involves a trade-off between objectives such as service regularity, commercial speed, passenger waiting times, and potentially energy consumption. These objectives can be conflicting, and may be valued differently by the transit users and by the transit operator [25]. It is therefore not always straightforward what the optimal response to this trade-off is.

Holding strategies can sometimes be augmented with other types of station control strategies, such as stop-skipping strategies for instance [22], [26]. In these settings, any late bus can take the decision of skipping one or a few upcoming stops in order to bridge the service gap with the preceding bus. This favors the service regularity and commercial speed, but at the expense of accrued waiting times for passengers at the stops being skipped. A similar intervention method is to limit the number of passengers that are allowed to board buses at each stop [23]. This strategy is especially relevant when capacity constraints are considered. Besides that, it can improve service regularity

in a way similar to the stop-skipping strategies.

Another approach to solve the operational bus line control problem is inter-station control strategies. This family of control strategies encompasses types of intervention which take place directly when buses are traveling. Such interventions can be deployed in combination with station control strategies in general, but they may also be a good substitute if the latter cannot be implemented, for instance due to a lack of space to hold buses at stops. One example of inter-station control strategies is transit signal priority, where the phases of traffic lights at intersections are adjusted to favor bus traffic. This can be done by granting priority to late buses at signalized intersections through green light phase extension [27], but the adverse effects it can have on the traffic of other vehicles should also be taken into account. Another inter-station intervention method is bus substitution. Transit providers can keep one or several buses in reserve, and dispatch them to replace buses which fall behind schedule too much [28]. This method entails additional costs for the transit agency, but avoids generating frustration among passengers like some of the station control strategies.

In addition to these two strategies, which both involve elements external to the operating buses, inter-station control can be carried out by adjusting the dynamics of the buses themselves. Such speed control strategies represent a small proportion of the studies on the operational bus line control problem, but their ability to provide commands continuously and in real-time makes them very attractive for the future developments of this field. Explicit speed control laws have been proposed to regularize bus spacings in operation [29], [30]. However, they hinge on an idealized bus route representation where stops and passengers are not modeled explicitly. Some authors have adopted a different approach instead and treated the operational bus line control problem as an optimal control problem [31]–[33]. The speed control laws are then defined implicitly in the framework of *model predictive control* (MPC). In these works, the control horizons are designed to be very short, such that only the next stop appears on the horizon of each bus. The rationale is to avoid intractable optimization problems, as some of the decision variables are integer-valued. In addition, these works use simple approximations of the longitudinal bus dynamics, as double integrator models are used for the bus motion.

In light of these observations of the literature treating the operational bus line control problem, the research questions investigated in this thesis are:

- How can the operational control of a bus line by means of speed adjustment be combined with the eco-driving of electric buses, and formulated as an optimal control problem?

- How can one exploit the particular structure of this optimal control problem to simplify its resolution?

- What performances, both in terms of service regularity and energy consumption, can be expected when implementing the solutions of this optimal control problem in realistic simulations?

1.2 Contributions

The focus of this thesis is the presentation of a scalable resolution method for the bus line regularity and eco-driving optimal control problem for electric buses. In particular, the main contributions of this thesis are:

- An extensive modelling framework for the inter-station dynamics, energy consumption, and longitudinal force control of the bus line. The dynamics are modeled in the space domain as a way to avoid integer variables and remove some of the nonlinearities.

- The formulation of this optimal control problem as a smooth *nonlinear program* (NLP), which includes predictive route and traffic information over long horizons, as well as a dynamical bus mass model. Service regularity is enforced through the use of adaptive time headways, meaning that the proposed method can be deployed without needing much prior line-specific information.

- A scalable resolution method based on the bi-level decomposition of this NLP. It hinges on the *sequential quadratic programming* (SQP) method and enables most of the computations to be carried out in parallel, while only requiring a minimal amount of information to be exchanged.

- A realistic case study to quantify the potential improvements in energy efficiency and service regularity of the proposed method when compared with traditional holding and speed control baselines.

1.3 Outline

The thesis is structured as follows. Chapter 2 introduces the models used for the route, the buses, the bus stops, and the passengers. Chapter 3 assembles an OCP for the bus line regularity and eco-driving problem. It is then explained how this OCP can be decomposed and solved in a scalable way. Some simulation results are also presented and discussed. Chapter 4 summarizes the content of the papers appended to the second part of the thesis. Chapter 5 concludes the thesis and discusses the possible future extensions of the research presented.

CHAPTER 2

Modelling

In this chapter, we introduce a modelling framework for a line of electric buses. This includes a nominal model of the bus dynamics and energy consumption, as well as a nominal model of the passenger exchanges that take place at bus stops.

2.1 Bus route

We assume that n buses travel continuously on a circular route, such as the one represented in Figure 2.1. Equivalently, this loop may represent the two travel directions of a linear bus route. The important aspect is that a fixed set of buses operate on this route, instead of being dispatched from the first stop and exiting the system at the last one, as assumed in some studies [34].

One of the bus stops is chosen as the first stop, and set as the origin of the route. The buses are indexed from 1 to n, where the bus with index 1 is the last one to have reached the first stop. In what follows, we therefore use modular notations to account for the circular aspect of the route. For instance, the bus with index 0 is in fact the bus with index n.

In addition, it is assumed that the bus line is operated with a high fre-

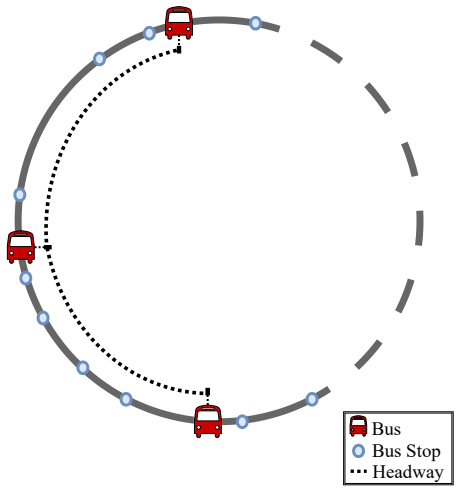


Figure 2.1: Illustration of a circular bus route.

quency of service, with typical time headways of 10 minutes or less between successive buses. This assumption is not very restrictive since it applies to most bus lines operating in urban environments. In this case, passengers tend to not coordinate their arrivals at stops [35], meaning that headway regularity is more relevant to their waiting times than the adherence to a fixed timetable. Therefore, the control methods explored in this thesis focus on service regularity in high frequency lines.

2.2 Bus dynamics and speed constraints

According to Newton's second law of motion, the longitudinal dynamics of a bus $i \in \mathbb{I}_{[1,n]}$ along a fixed route can be written as:

$$\dot{s}_i(t) = v_i(t), \tag{2.1a}$$

$$m_i(s_i,t)\dot{v}_i(t) = F_{m,i}(t) - F_{b,i}(t) - F_{d,i}(v_i,t) - F_{r,i}(s_i,t), \tag{2.1b}$$

where s_i, v_i, and m_i denote the bus position, speed, and mass, respectively. The forces acting upon the bus include the motor force at the wheels $F_{m,i}$, the force generated by the friction brakes $F_{b,i}$, the aerodynamic drag $F_{d,i}$, and

the rolling resistance and gravitational pull $F_{r,i}$. The two former terms can be seen as control inputs, while the two latter terms can be expressed as:

$$F_{d,i}(v_i, t) = \frac{1}{2}\rho A_{\text{bus}} c_a v_i(t)^2, \tag{2.2a}$$

$$F_{r,i}(s_i, t) = g\, m_i(s_i, t)\big(\sin\theta(s_i(t)) + c_r \cos\theta(s_i(t))\big), \tag{2.2b}$$

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 [36].

The previous set of equations can be rewritten in the space domain instead. This is a common transformation in the eco-driving literature [8]–[10], with the goal of removing some of the nonlinearities from the longitudinal dynamics. We take a similar approach here, and consider the position s to be the independent variable. We further consider the travel time t_i and the kinetic energy per mass unit E_i as the state variables for bus i. The kinetic energy per mass unit is defined as $E_i(s) = 1/2\, v_i(s)^2$. With this transformation, the expressions (2.2a) and (2.2b) are now linear in the state variables.

The space domain transformation is especially meaningful in the case of a bus line, and its main advantage goes beyond the removal of some nonlinearities. Indeed, since the position is now the independent variable, bus stops can be modeled independently from state variables, which circumvents the need to introduce integer variables when formulating the control problem. This point will be further clarified in the next chapter.

A new problem arises with the formulation of the dynamics in the space domain though, as it prevents vehicles from having zero speed. Indeed, the dynamics of the travel time t_i are given by the formula $\mathrm{d}t_i(s)/\mathrm{d}s = (\sqrt{2E_i(s)})^{-1}$. While this is usually not an issue for cruise control on highway, it becomes one for city buses as they need to stop every few minutes to pick up passengers.

This problem can be addressed when setting up constraints on the bus speed. It is assumed that the bus speed cannot exceed the speed limits nor the speed of the surrounding traffic, and that it must be higher than a certain minimum value. These speed constraints can be written as:

$$\frac{1}{2}v_{\min}^2(s, t_i) \leq E_i(s) \leq \frac{1}{2}v_{\max}^2(s, t_i), \tag{2.3}$$

where $v_{\min}$ and $v_{\max}$ are the lower and upper bounds on the speed, respec-

tively. By assuming that $v_{\min}$ is strictly positive everywhere, the singularity of the travel time dynamics at zero speed is no longer an issue.

However, it remains to see how bus stops can be included in the description of the dynamics, since the bus speed cannot be zero. As proposed in Paper A, one way to do so is to impose a very low velocity v_{entry} to buses at the bus stop locations through the constraints (2.3), with the goal of approximating a complete stop of the vehicles. The time needed to reach zero speed from v_{entry}, as well as the dwell times at stops due to the passenger exchanges, can be modeled as an additional delay term $\Delta_{\text{stop},i}$ added to the travel time dynamics. Note that this delay term is then nonzero only at bus stops. The exact expression of $\Delta_{\text{stop},i}$ is discussed below, when modeling bus stops.

Finally, the longitudinal dynamics of bus i in the space domain can be expressed as:

$$\frac{\mathrm{d}E_i(s)}{\mathrm{d}s} = \frac{1}{m_i(s,t_i)}(F_{m,i}(s) - F_{b,i}(s) - \rho A_{\text{bus}} c_a E_i(s)) - g(\sin\theta(s) + c_r\cos\theta(s)), \tag{2.4a}$$

$$\frac{\mathrm{d}t_i(s)}{\mathrm{d}s} = \frac{1}{\sqrt{2E_i(s)}} + \Delta_{\text{stop},i}(s,t_i), \tag{2.4b}$$

where E_i is strictly positive due to the constraints (2.3). The state vector for bus i can be assembled as $x_i(s) = [E_i(s), t_i(s)]^\top$, and the control input vector as $u_i(s) = [F_{m,i}(s), F_{b,i}(s)]^\top$.

Remark 1: *The speed bounds v_{min} and v_{max} in (2.3) can have any general shape on the bus route, as long as they verify the conditions mentioned previously. In particular, they need not be smooth functions of the position, meaning that legal speed limits can be modeled quite easily. The speed bounds 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.*

2.3 Bus stops and passengers

The influence of a bus stop j on bus i can be modeled by the travel time delay it causes, which is captured in the dynamics by the term $\Delta_{\text{stop},i}$, and by the change in bus mass due to the passenger exchange at the stop. Both aspects are directly dependent on the number of passengers waiting at the stop.

As is often assumed in the bus bunching literature, the arrivals of passengers at any stop j are modeled as a homogeneous Poisson process with parameter λ_j [37]. This static parameter denotes the arrival rate of passengers, and may differ from stop to stop. When reaching stop j, the number of passengers waiting for bus i is then proportional to the amount of time elapsed since the previous bus (with index $i+1$) left stop j. The time at which this event occurred is assumed to be known, and is noted t_{i+1}^{j}.

Another common assumption is to model the dwell times of buses at stops proportionally to the number of boarding passengers only. This is motivated by the observation that the boarding operation usually takes longer than the alighting operation [37], and that both can in general take place in parallel through different doors of the bus. Finally, the number of boarding passengers is modeled as a real number.

Based on these assumptions, the increase in the travel time of bus i at stop j can be modeled as:

$$\Delta_{\text{stop},i}(s, t_i) = \begin{cases} 2t_s + b\lambda_j(t_i(s_j) - t_{i+1}^{j}) & \text{if } s = s_j, \\ 0 & \text{otherwise,} \end{cases} \tag{2.5}$$

where t_s denotes the time needed for the bus to reach zero speed from v_{entry} and open its doors (and vice-versa), and where b is the boarding time for each passenger. The location of stop j on the route is noted s_j here.

The change in the mass of bus i when reaching stop j can be modeled in a similar way, but the number of alighting passengers must be considered this time. To this end, it is assumed that a given proportion μ_j of the passengers aboard bus i alight when stop j is reached [37]. The alighting proportion μ_j is a fixed scalar in $[0, 1]$, and can be set from historical passenger flow data.

By noting $m_i(s_j^-)$ and $m_i(s_j^+)$ the mass of bus i when approaching and leaving stop j, respectively, we have:

$$m_i(s_j^+) = (1 - \mu_j)(m_i(s_j^-) - m_{\text{emp}}) + m_{\text{pax}}\lambda_j\big(t_i(s_j) - t_{i+1}^{j}\big), \tag{2.6}$$

where m_{pax} is the average passenger mass, and m_{emp} is the mass of the empty bus.

The right-hand side of (2.6) 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. It is furthermore assumed that passenger exchanges can only take place at the designated bus stops, such that the bus mass remains constant between two consecutive stops. The mass m_i of bus i is therefore entirely defined by the relation (2.6).

2.4 Energy consumption

The powertrain of an electric vehicle usually consists of a battery connected to an electric machine (EM), which is itself linked to the drive wheels through a mechanical transmission system. The EM transforms the electric energy stored in the battery into rotational energy, which is then translated into a traction force at the wheels by the transmission system. A schematic diagram of the powertrain is given in Figure 2.2.

In general, an EM can operate both as a motor, by drawing energy from the battery of the vehicle to generate a traction force at the wheels, and as a generator, by doing the opposite. The efficiency map of the EM model used in this thesis is provided in Figure 2.3, where the motoring and the generating modes of the EM can be clearly seen. Therefore, part of the kinetic energy of an electric bus can be stored back in the battery, when the bus decelerates before reaching a bus stop for example. On the other hand, the use of the friction brakes results in a loss of energy, and must be avoided when possible.

The rotational speed $\omega_{m,i}$ and torque $T_{m,i}$ of the EM of bus i can be related to its longitudinal force and kinetic energy through:

$$T_{m,i}(s) = \frac{r_w \eta(F_{m,i})}{M_f} F_{m,i}(s), \quad \omega_{m,i}(s) = \frac{M_f}{r_w} \sqrt{2E_i(s)}, \tag{2.7}$$

where r_w is the wheel radius and M_f is the final gear ratio. The transmission efficiency function η captures the losses occurring along the transmission system of the powertrain. It has value $1/\eta_f$ or η_f depending on whether the EM operates in traction or in generation, and where η_f is the efficiency coefficient

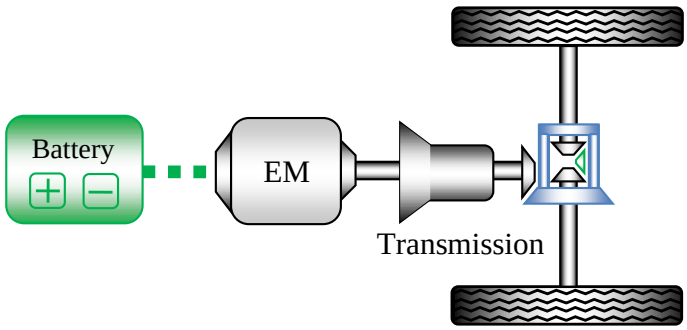


Figure 2.2: Schematic representation of the powertrain of an electric vehicle and its main components.

of the final gear.

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

$$|T_{m,i}(s)| \leq \min(T_{\max}, P_{\max}/\omega_{m,i}(s)), \tag{2.8}$$

where $T_{\max}$ is the maximum motor torque and $P_{\max}$ is the maximum power that the motor can supply continuously [36]. These bounds on the torque of the EM are displayed in Figure 2.3.

In this thesis, the battery is modeled as an open circuit voltage connected in series to an internal resistance. Then, the internal battery power balances the power dissipated over the internal resistance, a constant load consumed by auxiliary devices and the electrical power of the EM [38], [39]. The EM electrical power is modeled by fitting a polynomial function to the data shown in Fig. 2.3, including second order terms in $T_{m,i}$ and up to fifth order terms in $\omega_{m,i}$ [9]. 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 $\omega_{m,i}$ [9], [40].

Remark 2: *The torque $T_{m,i}$ is not continuously differentiable in 0, i.e. when switching between the motoring and generating mode, due to the discontinuity of the efficiency function η in* (2.7)*. This may require a special treatment, depending on the resolution method used for the optimal control problem presented in the next chapter. This point is developed at great length*

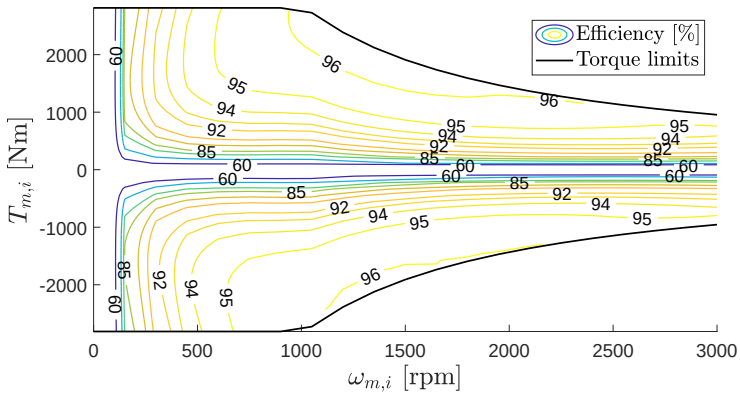


Figure 2.3: Steady-state efficiency map of the electric machine, together with the torque constraints.

in Paper B, where the control input vector has to be augmented with an additional term.

CHAPTER 3

Optimal control

In this chapter, the nominal models developed in the previous chapter are used to formulate the bus line regularity and eco-driving problem as an OCP. This OCP should have the general form:

$$\begin{aligned}
&\text{minimize} && \text{service regularity} + \text{energy consumption} && \text{(3.1a)}\\
&\text{such that} && \forall\, \text{bus} \in \mathbb{I}_{[1,n]} : && \\
& && \text{initial conditions,} && \text{(3.1b)}\\
& && \text{bus dynamics,} && \text{(3.1c)}\\
& && \text{velocity constraints,} && \text{(3.1d)}\\
& && \text{torque constraints,} && \text{(3.1e)}
\end{aligned}$$

where the bus dynamics, the velocity constraints and the torque constraints are here given by (2.4), (2.3), and (2.8), respectively.

A scalable resolution method for this OCP is then presented and discussed. Selected simulation results are included at the end of this chapter to illustrate how the proposed method can be applied in practice and what performances can be expected from it.

3.1 Problem formulation

Optimal control problem

Let the space-dependent vectors x and u be the state and control input trajectories of all buses, i.e. $x(s) = [x_1(s), ..., x_n(s)]^\top$ and $u(s) = [u_1(s), ..., u_n(s)]^\top$. Recall that the individual state and control input vectors for bus i have been defined previously as $x_i(s) = [E_i(s), t_i(s)]^\top$ and $u_i(s) = [F_{m,i}(s), F_{b,i}(s)]^\top$. The bus line regularity and eco-driving problem can be formulated as the following OCP:

$$\min_{u(s),x(s)} \quad \sum_{i=1}^{n} V_f(x_i(s_{f,i}), x_{i-1}(s_{f,i-1})) + \sum_{i=1}^{n} \int_{s_{0,i}}^{s_{f,i}} l(x_i(s), u_i(s), s)\mathrm{d}s \tag{3.2a}$$

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

$$x_i(s_{0,i}) = x_i^0, \tag{3.2b}$$

$$\frac{\mathrm{d}x_i(s)}{\mathrm{d}s} = f(x_i(s), u_i(s), s), \tag{3.2c}$$

$$g(x_i(s), u_i(s), s) \leq 0, \tag{3.2d}$$

where V_f and l are the terminal and stage cost, and are assumed to include some headway-related terms and the energy consumption, respectively. For each bus i, the initial and final positions are noted $s_{0,i}$ and $s_{f,i}$, and x_i^0 is its initial state. The function f denotes the nonlinear dynamics of bus motion (2.4), while the function g gathers the inequality constraints on the states and control inputs (2.3) and (2.8).

Since the states and control inputs considered in (3.2) are all continuous variables, this OCP can be expressed as a NLP. This requires an additional discretization step, which can be carried out by using the multiple shootings method for instance [41], and during which the continuous variables of the OCP are transformed into a finite number of scalar decision variables. The traditional Runge-Kutta method can then be used for the numerical integration of the dynamics and of the energy consumption. This step is not essential to our purpose in this thesis, however, and the interested reader may find more details in Paper B.

Remark 3: *While* (3.2) *is a fairly general formulation of the bus line regularity and eco-driving problem, it hinges on two important design choices: (i) the service-related aspects are only penalized at the end of the bus horizons*

since they only appear in the terminal cost, and (ii) only the coupling between pairs of successive buses is considered. These assumptions are reasonable, but they are restrictive in the space of all possible problem formulations. For instance, one could think of a stage cost including service-related terms at all bus stops on the horizons, or additional couplings between buses. However, we chose to present directly the particular problem structure on which the method proposed in this thesis operates, for the sake of simplicity.

Remark 4: *Before we continue, it must be emphasized how important the choice of the space domain is when formulating the OCP. This is namely a necessary condition for the discretized version of* (3.2) *to be a smooth NLP. The key idea is that in the time domain, the control inputs affect the location of the bus stops on the control horizon, whereas their location is fixed and independent from the control inputs in the space domain. The consequence is that integer variables are needed to model bus stops in the former case, but not in the latter. Since this explanation might seem obscure with only the continuous version* (3.2) *in mind, we refer the reader to the discretized version in Paper B and to the discussion therein.*

Horizons and objective function

The service-related terminal cost in (3.2a) can take several forms. A classical choice is to opt for the minimization of passenger waiting times both at stops and inside buses, usually in combination with horizons covering the entire route [23], [24], [42]. Alternatively, headway regularity is often used as a proxy for the minimization of passenger waiting times at stops [21], [34], usually by tracking a predefined target headway provided by the local transit agency. Spacings (i.e. the distance between consecutive buses) are also used as a proxy for time headways occasionally [32].

In this thesis, we consider headway regularity as the service-related objective. On the one hand, it offers a better alternative than spacing regularity. Indeed, the non-homogeneous distribution of bus stops and average traffic conditions on a typical bus route means that buses can have very different average speeds on different parts on the route. The direct consequence is that homogeneous spacings may translate into inhomogeneous time headways. On the other hand, headway regularity does not require the long horizons that are needed when considering passenger waiting times. To see that, one may reflect on the fact that a regular bus service cannot be attained with a waiting

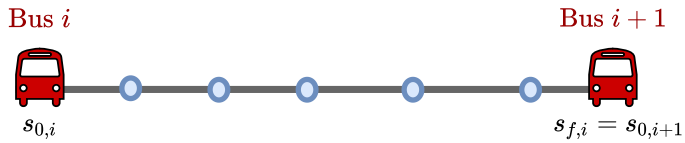


Figure 3.1: Control horizon of bus i.

times-driven objective if the horizons are not overlapping. In such a scenario of short horizons, each bus would simply be constantly traveling as fast as possible to minimize the waiting times of passengers appearing on its own horizon, which would ultimately result in an irregular service for the bus line as a whole.

Now, it remains to choose an adequate design for the bus horizons in order to capture predicted bus headways. Given our modeling assumptions up to that point, the natural choice for the horizon of any bus i is the complete route segment between bus i and the preceding bus $i+1$. In other words, the control horizon for each bus $i \in \mathbb{I}_{[1,n]}$ is chosen to be the interval $[s_{0,i}, s_{0,i+1}]$, i.e. $s_{f,i} = s_{0,i+1}$. A schematic representation of this horizon for bus i is given in Figure 3.1. The rationale behind this choice is to be able to express the travel time needed to reach the preceding bus as the terminal state, as the travel time is part of the state vector. Since the travel time to the preceding bus is also the predicted time headway between these two buses, headway regularity can now be directly expressed as a function of the terminal states, as required in (3.2a), with this choice of bus horizons.

Consequently, the entire bus route is divided into n non-overlapping control horizons. This means that each bus stop and each inter-stop link are accounted for in the OCP (3.2). The downside, however, is that some horizons can be rather long, depending on the length of the bus route considered and on the number of buses. This motivates the need for scalable resolution methods, which will be discussed shortly.

In Paper A, headway regularity is enforced by minimizing deviations with respect to a predefined target headway. We adopted a different approach in Paper B, where adaptive headways are used instead. With adaptive headways, 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. Indeed, the desirable headway for service regularity might change depending

on e.g. the amount of disturbances applied to the system [20], and having some flexibility on the objective to be tracked is therefore desirable. In light of this, the terminal cost in (3.2a) can be express as:

$$V_f(x_i(s_{f,i}), x_{i-1}(s_{f,i-1})) = (t_i(s_{f,i}) - t_{i-1}(s_{f,i-1}))^2 + t_i(s_{f,i})^2, \tag{3.3}$$

where the terminal travel times denote the predicted time headways, as discussed above.

It is worth noting here that the second term in (3.3) is needed to incentivize shorter headways, in addition to the first term which penalizes headway deviations of consecutive buses. Indeed, headway deviations can always be reduced by forcing buses to travel at an arbitrarily slow speed, but this may not be desirable since the commercial speed affects passenger waiting times too. As a result, both aspects need to be included in the adaptive headways objective (3.3). Note also that this commercial speed requirement is implicitly embedded in predefined target headways, such that only the headway regularity term usually needs to appear explicitly when tracking a target headway.

Remark 5: *The bus line regularity and eco-driving problem is by essence a multi-objective problem. With the choice of service-related cost* (3.3)*, the economic objective function of the OCP* (3.2a) *contains three, possibly conflicting, objectives: energy consumption, headway minimization and minimization of successive headway deviations. How each objective is weighted in comparison with the others must therefore be carefully investigated depending on the application considered. In particular, constant weights can be included in* (3.2a) *to handle this trade-off, but they are not represented here for the sake of brevity. We direct the interested reader to Paper B for a more detailed presentation of the objective function.*

3.2 Resolution

Due to our modelling choices, (3.2) describes a non-convex NLP which can include a significant number of decision variables due to the potentially long control horizons. Even if solving this NLP directly with fully-centralized computations may still be doable, it is certainly not scalable in the number of buses and may result in long computation times. This section presents a bi-level decomposition of (3.2), which makes the resolution of the corresponding NLP

scalable.

Primal decomposition

It can be observed in (3.2) that buses are only coupled through the terminal cost in the objective function, due to the term penalizing the headway deviations of successive bus pairs in (3.3). In other words, only the terminal travel times prevent (3.2) from being separable into n independent sub-problems, one for each bus. For problems with such a weakly-coupled structure, it can be meaningful to consider a primal decomposition [43], [44]. This type of decomposition provides an equivalent bi-level reformulation of the original problem, where the coupling variables (also called complicating variables) are gathered in a higher-level master problem, and where the lower-level sub-problems are independent.

Here, the coupling variables are the terminal travel times, which correspond to the predicted time headways as discussed in the previous section. Therefore, we introduce the variables $\{H_i\}_{i\in\mathbb{I}_{[1,n]}}$ to denote the predicted time headways in the decomposed problem. The primal decomposition of (3.2) consists in the line-level master problem:

$$\min_{H} \quad \sum_{i=1}^{n} V_f(H_i, H_{i-1}) + \sum_{i=1}^{n} V_i(H_i) \tag{3.4a}$$

$$\text{s.t.} \quad H_i \in \text{dom}(V_i), \qquad i \in \mathbb{I}_{[1,n]}, \tag{3.4b}$$

and in the bus-level sub-problems:

$$V_i(H_i) = \min_{u_i(s), x_i(s)} \quad \int_{s_{0,i}}^{s_{f,i}} l(x_i(s), u_i(s), s)\mathrm{d}s \tag{3.5a}$$

$$\text{s.t.} \quad x_i(s_{0,i}) = x_i^0, \tag{3.5b}$$

$$t_i(s_{f,i}) = H_i, \tag{3.5c}$$

$$\frac{\mathrm{d}x_i(s)}{\mathrm{d}s} = f(x_i(s), u_i(s), s), \tag{3.5d}$$

$$g(x_i(s), u_i(s), s) \leq 0, \tag{3.5e}$$

where $H = [H_1, ..., H_n]^\top$ gathers the coupling variables. In this decomposed problem, V_i and $\text{dom}(V_i)$ denote the optimal cost and the feasible set of the

sub-problem for bus i, respectively. This feasible set gathers all possible terminal travel times for bus i, given the speed constraints imposed on its horizon. Therefore, this set can be described by linear inequalities since it is bounded by the minimum and maximum travel times.

Remark 6: *In the case of a convex NLP, the primal decomposition process is known to conserve global optimality [45]. In other words, both the original problem and the decomposed problem have the same global solution. This result cannot be obtained for a non-convex NLP however, like the one considered here, due to fact that the global optimality of a local minimum cannot be guaranteed for non-convex problems in general [46]. That being said, the proof proposed in [45] can be adapted to the non-convex case to show that, under some mild assumptions, the original and the decomposed problems have the same set of local minima.*

Distributed optimization

The sequential quadratic programming method [46] can be deployed to solve the master problem (3.4). In a nutshell, this resolution method iteratively solves a *quadratic program* (QP) representing a local second-order approximation of the NLP. With (3.3) in mind, it can be observed that only the implicit function V_i needs to be approximated by a linear or quadratic function in order for (3.4) to become a QP, since the constraints are linear. Some results from parametric optimization on sensitivity analysis readily deliver a local quadratic approximation for V_i, provided that the corresponding sub-problem (3.5) is solved first [43], [47]. The exact procedure is rather involved and is not vital to the purpose of this thesis, so it is not reproduced here. A complete treatment is given in the Appendix section of Paper B, however.

Consequently, at each SQP iteration, the n sub-problems (3.5) are solved first, followed by the QP approximation of the master problem (3.4). This process is repeated until convergence of the SQP algorithm. Dedicated solvers can be used to handle the bus-level sub-problems and the line-level QPs. In this thesis, the primal-dual interior point solver IPOPT [48] was used for the non-convex NLPs (3.5), and the active-set solver qpOASES [49] for the QP approximations of (3.4). Unsurprisingly, the amount of computations needed to solve each QP (with only n decision variables) is negligible in comparison with that needed to solve any of the sub-problems.

As a direct result of the decomposition of the original problem, the sub-

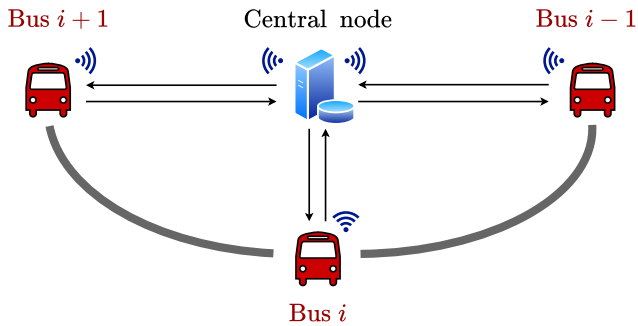


Figure 3.2: Distribution resolution scheme of the decomposed problem. The arrows denote the remote communication between the central node and the buses.

problems (3.5) are separable and can be solved independently at each iteration of the SQP algorithm. Since solving the sub-problems represents the bulk of the computations needed to run the SQP algorithm, it could be very beneficial, in terms of computation time, to solve them in parallel. One way to do so is to leverage the computing power installed aboard each bus and solve each sub-problem on the corresponding bus, as illustrated in Figure 3.2. Note, however, that the QP approximation of the master problem (3.4) includes a local approximation of the implicit function V_i for each bus i, and must therefore be solved centrally, i.e. in common for all buses, at each SQP iteration. It can be assumed that this part of the algorithm is handled by a central computing node, which could for example be a roadside unit, or even one of the buses.

At this point, one could ask how much communication a distributed application of the SQP algorithm, such as the one presented in Figure 3.2, would require. To this end, it must be noted that each local quadratic approximation of V_i is entirely defined by 3 scalars. These can be directly computed aboard the bus concerned since they only depend on the primal-dual optimal solution of the corresponding sub-problem. Therefore, at each SQP iteration, the central node needs to send 1 scalar value (the current estimate of H_i) to each bus i, which then only needs to send 3 scalar values back. Note also that, in this scenario, the bus-level trajectories obtained from the sub-problems at

each SQP iteration are directly accessible by the buses themselves. Not only does this alleviate the need for the central node to broadcast the possibly long optimal trajectories upon termination of the SQP algorithm, but it also makes the control scheme more robust to communication issues since buses can always use their most recent local trajectories.

Based on the previous analysis, it seems that a physically distributed resolution of the decomposed problem (3.4)-(3.5), where each sub-problem is solved in parallel aboard the corresponding bus, could greatly improve the computation times and decrease the communication loads. Adding new buses to the system would only marginally increase the computation times since the additional sub-problems can be solved in parallel. The proposed resolution scheme is therefore scalable in the number of buses.

Remark 7: *Due to the non-convexity of the decomposed problem* (3.4)-(3.5)*, the SQP method deployed in the proposed resolution scheme converges to a local minimum, which may or may not be the global solution of the problem. In practice, we observed that this SQP algorithm systematically managed to reach convergence by taking full Newton steps [46]. Under some mild assumptions, this means that the algorithm has a quadratic convergence rate on this problem [46]. Note that this is in general the best that can be expected from any optimization method on non-convex NLPs of this type.*

Receding horizon control

Up to this point, the focus of this section has been on assembling and solving the bus line OCP (3.2) (and its equivalent decomposed version). This OCP provides optimized state and control input trajectories for each bus based on the nominal bus line model assembled in the previous chapter.

For practical applications in uncertain environments, these optimized trajectories can be considered as high-level references. It can be assumed that aboard each bus is a local controller which aims to track the references provided by solving the OCP. This low-level control layer would need to operate at a high frequency to reject the disturbances coming from the system, and could thus be charged to guarantee critical safety constraints, such as collision avoidance with surrounding vehicles (including other buses). The implementation of this low-level control layer is outside the scope of this thesis, however.

In order to update the reference trajectories frequently and introduce closed-loop control, the OCP can be solved periodically, in a receding horizon fashion.

This means that only the first T time units of each control input trajectory are actually applied, before new reference trajectories are generated. In other words, the high-level controller generating the reference trajectories is an MPC [50]. Algorithm 1 offers an overview of how this MPC operates. Note that the control structure for the bus line is therefore hierarchical [51].

Algorithm 1 MPC algorithm. *state* contains the current information available about the buses, *local_control* models the tracking control layer and the system evolution, and *data* gathers the complete state history.

initialize state, time $\leftarrow 0$, data $\leftarrow \{\}$
while time $-$ time_end < 0 **do**
  $x(s)$, $u(s) \leftarrow$ solve (3.4)-(3.5)
  state $\leftarrow$ local_control (state, $x(s)$, $u(s)$, T)
  data $\leftarrow \{\text{data}, \text{state}\}$
  time $\leftarrow$ time $+ T$
end
return data

3.3 Selected results

The results presented here are a subset of the more detailed case study from Paper B. We focus only on a specific scenario in this section, for pedagogical purpose.

Experiment

Historical data from bus line 17 in Gothenburg, Sweden, is used to calibrate the parameters of our modelling framework. These include namely: stop locations, route altitude profile, boarding and alighting rates, speed limits, and historical traffic speeds. A condensed representation of the entire bus line is given in Figure 3.3.

Each simulation run consists of two hours of bus operation during rush hours, and each starts with 8 buses in total. The stochastic elements in the simulation framework come from the boarding and alighting of passengers at stops, as well as from the speed of the surrounding traffic. Since the reference trajectories of the MPC are based on a nominal model of the system, they

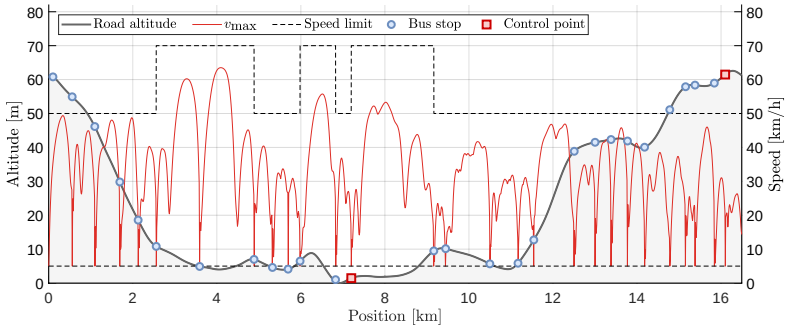


Figure 3.3: Route layout of bus line 17. The solid red line is the average maximum speed obtained from historical bus driving profiles, and is used as the upper velocity bound $v_{\max}$. The dashed lines indicate the speed limit on each road segment, as well as the lower velocity bound $v_{\min}$.

may sometimes become infeasible due to the traffic speed. In such cases, the commands implemented are adjusted accordingly in the simulation framework.

In this experiment, it is investigated how the control algorithm manages to restore regular headways if the transit system is initially in a state of intermediate bus bunching. To evaluate this, three performance metrics are monitored during the simulations:

- The squared coefficient of variations, noted CV^2, which is proportional to the headway variance, and can be used as a proxy for passenger waiting times at stops [52]. This term represents the headway regularity.
- The mean headway value, which can be used as a proxy for the commercial speed of the vehicles.
- The total energy consumption.

The modeling and simulation aspects are implemented in MATLAB. The symbolic framework CasADI [53] is used to assemble the NLPs (3.4) and (3.5).

Baselines

Two baselines are compared with our MPC. Both of them track a predefined headway target, which is initially set to 5 minutes. The MPC, on the other

hand, relies on adaptive headways. The baselines are:

- *Holding control.* This method relies on simple rule-based holding control actions, which can only be performed at two of the bus stops, named control points (see Figure 3.3). The holding actions are headway-based, meaning that a bus is held at a control point only if the previous bus left it less than 5 minutes before. Between control points, buses travel at the highest possible speed.

- *PI control.* This simple *proportional-integral* (PI) controller operates in a way similar to the MPC, i.e. by computing longitudinal force commands to adjust the bus speeds in operation. The error tracked by the PI controller for each bus is the trajectory of the preceding bus shifted backwards by 5 minutes [31].

Note that only the holding baseline is allowed to hold buses at control points, whereas only the MPC and the PI controller can adjust the bus speeds between stops. For more details about the simulation settings, the tuning of the controllers, and the numerical parameter values, see Paper B.

Results

The figures below present the evolution of the performance metrics considered. Due to the stochastic aspect of the simulations, several simulation runs are carried out for this experiment. The shaded areas in the figures represent how the values obtained are distributed across all simulation runs.

Figure 3.4 is the most representative of the behavior of each control method. It displays the evolution of the headways observed at bus stops over time. Due to the initial service irregularities, the headway variance is highest at the beginning of the simulation. It can be observed that all control strategies manage to dissipate the initial bunching over time and to restore headway regularity.

The controllers have rather different convergence profiles, however, as can be seen in Figure 3.4. The MPC leverages its adaptive feature to set higher headway commands to the buses during the first 30 minutes of simulation. This results in much higher headways initially, but it enables a faster convergence to homogeneous headways as the standard deviation decreases much earlier than for the baseline buses. As can be seen, headway regularity is achieved by

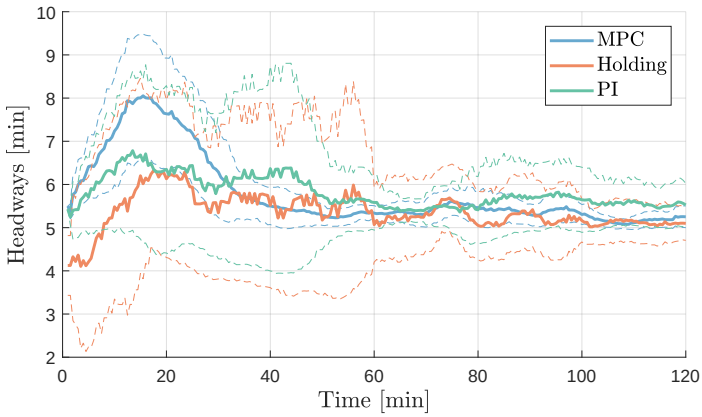


Figure 3.4: Sample mean (solid lines) and standard deviation (dashed lines) of headways at stops.

the MPC at around 35-40 minutes of simulation time, while it takes more than one hour for the other control methods. The holding baseline is particularly slow, as it struggles to dissipate the initial service irregularities since each bus must first reach a control point before any control can be applied. However, it is worth noting that the commercial speed of the holding-controlled buses is higher. The PI controller fares a bit better in terms of headway regularity, but does not manage to achieve the same low headway variance as the MPC, even during the latter stages of the simulation. The predictive feature of the MPC enables it to take preemptive control actions to slow down or speed up buses, based on the expected number of passengers at the upcoming stops for example. The PI controller, on the other hand, only reacts to current observations, which results in a higher headway variability.

Similar conclusions can be drawn from Figure 3.5, which displays the average CV^2 scores of the control methods at the end of the simulations. The CV^2 score of the MPC is consistently lower than that of the baselines across bus stops, with the notable exception of the PI controller, which manages to achieve CV^2 scores similar to that of the MPC at some of the bus stops. In other words, the MPC algorithm achieves more stable headways at stops than the other control methods. The difference with the holding control method is particularly striking, as the CV^2 scores of this baseline display a characteris-

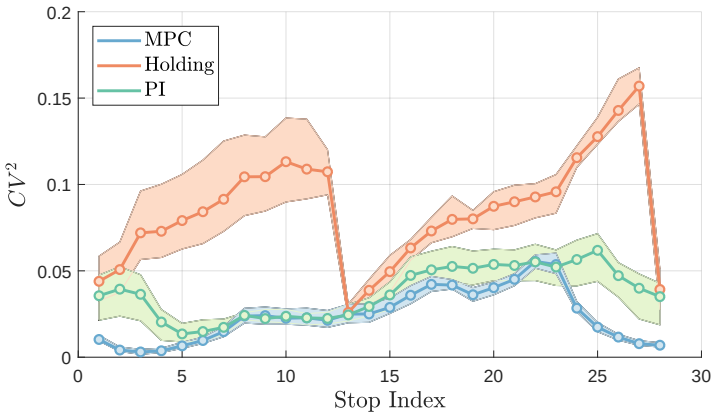


Figure 3.5: Headway regularity index (CV^2) at stops.

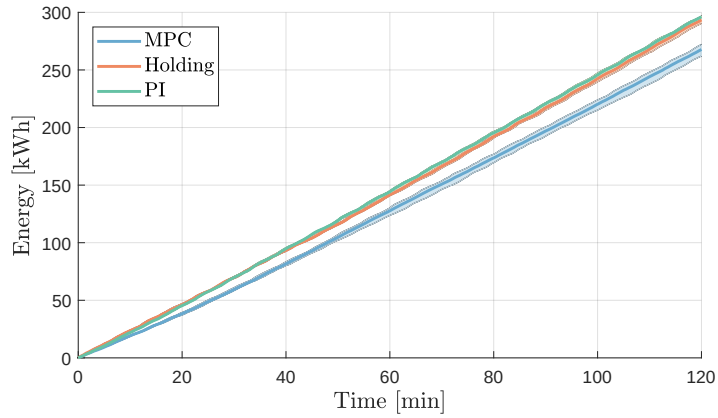


Figure 3.6: Total energy consumption.

tic sawtooth pattern, with the control points (which have indices 13 and 28) at its base. Finally, the similar CV^2 score patterns of the MPC and the PI controller suggests that headway regularity may be harder to enforce locally at some of the stops.

The energy consumption over time for all control methods is plotted in

Figure. 3.6. It can be observed that the MPC-controlled buses have a lower total energy consumption compared with that of the two baselines. In this experiment, the MPC realizes energy savings of 8.9% over the best performing baseline, which is the holding controller in this case. Note that the holding baseline manages to outcompete the PI controller in this particular experiment due to the fact that it is allowed to have buses dwell at control points, where they do not need any energy, without time limit. On the other hand, the buses controlled by the PI controller and the MPC must constantly be on the move when not picking passengers up. The superior performances of the MPC can be explained by its predictive feature, which it can leverage to generate energy-efficient driving profiles for the buses. For instance, the MPC can decrease the bus speeds before steep downhill sections in order to empty their kinetic energy buffers, thus avoiding unnecessary braking.

Reaction to a major disturbance

The control methods investigated show promising performances in a scenario with regular operational disturbances coming from the passengers and the traffic. Here, we investigate how the controllers fare when the bus line is subject to a major disturbance. Namely, one of the buses is removed after one hour of simulation, as a way to simulate a breakdown. It is assumed that this bus is not replaced by the transit agency. Note that the target headway tracked by the baseline methods is then increased to 6 minutes, as a way to incorporate the effect of the breakdown in the control.

It can be observed in Figure. 3.7 that the average headways increase right after one of the buses is removed. The convergence profile of each method is similar to that of Figure 3.4. As an example, the MPC temporarily increases the headway commands of the buses right after the breakdown, which enables it to recover homogeneous headways faster than the baselines. Here too, all control methods manage to dissipate the service irregularities caused by the bus breakdown, and their headways eventually stabilize around the new value of 6 minutes. It is interesting to note that the convergence to homogeneous headways happens somewhat faster than when dissipating the initial bunching. Indeed, the breakdown occurs when buses already have roughly homogeneous headways, thus creating only one large gap in service which is then easier to bridge.

Likewise, the patterns for the CV^2 scores in Figure 3.8 and the energy

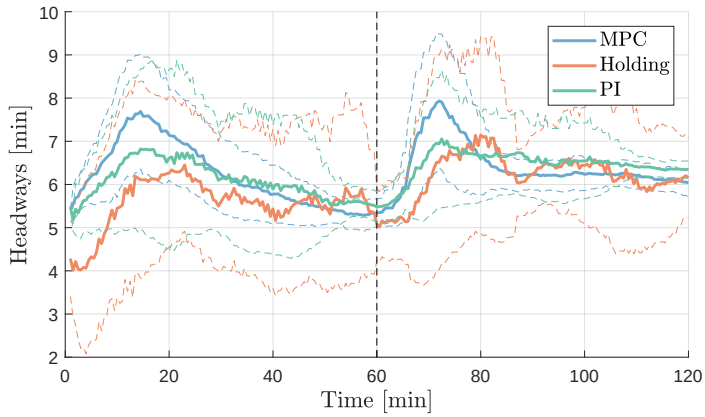


Figure 3.7: Sample mean (solid lines) and standard deviation (dashed horizontal lines) of headways at stops. Scenario with one bus breakdown (dashed vertical line).

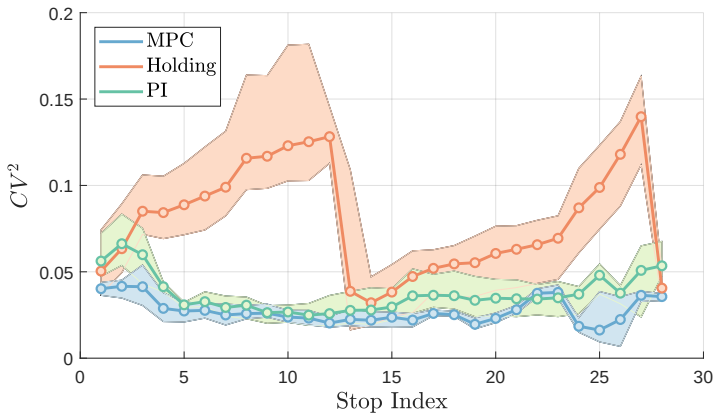


Figure 3.8: Headway regularity index (CV^2) at stops. Scenario with one bus breakdown.

consumption in Figure 3.9 are similar to what was previously observed. The energy savings of the MPC over the baselines are in the same order of magnitude as previously, as they now amount to 9.7%.

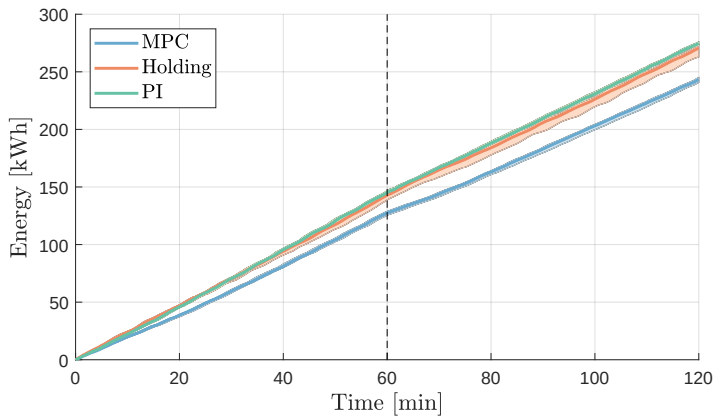


Figure 3.9: Total energy consumption. Scenario with one bus breakdown (dashed vertical line).

CHAPTER 4

Summary of included papers

This chapter provides a summary of the included papers.

4.1 Paper A

> **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár
> Hierarchical control of electric bus lines
> Accepted in 21st *IFAC World Congress*, Berlin, Germany, Jul. 2020.

In this paper, the formulation of the bus line regularity and eco-driving problem as an OCP and the scalable resolution method discussed in this thesis are presented for the first time. The state-space model for the buses is formulated in space rather than in time, which alleviates the need for integer decision variables to capture their behavior at bus stops. The effects of the surrounding traffic are not included, as the bus speeds are constrained to remain within a static speed corridor. A fixed target headway, assumed to be provided by the transit agency, is tracked in the objective function of the OCP to enforce service regularity. A bi-level decomposition of the OCP is then carried out in order to solve it. The two control layers discussed in

the paper correspond to what were referred to as the bus-level subproblems and the line-level master problem in this thesis. The high-level layer operates based on the sensitivity analysis information obtained when solving the NLPs of the low-level layer. Finally, deterministic simulations are run to evaluate the energy savings of the proposed controller over a classical headway-based holding baseline.

4.2 Paper B

> **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár
> Distributed optimization for bunching mitigation and eco-driving of electric bus lines
> Submitted to *IEEE Transactions on Intelligent Transportation Systems (ITS)* in Nov. 2020.

This paper extends and develops the resolution method presented in Paper A. A similar OCP is formulated, but based on a much more detailed model of the bus line. Namely, the historical traffic velocity is now embedded in the constraints on the bus speeds, the OCP no longer tracks a predefined target headway but can adapt the bus headways to the natural headway of the line, and the bus mass is no longer assumed to be static but to change dynamically as a function of the passenger demands at stops. It is explained, based on an energetic argument, how the OCP can be lifted to remove the non-smoothness of the motor torque. A smooth NLP is then obtained by discretizing the original OCP. It is shown how the equivalent decomposed NLP can be solved with the SQP method. In particular, the exact computations and communication needed to carry out the resolution are discussed. It is argued that most of the computations can be done in parallel aboard individual buses, with very little communication requirements. In addition, it is found that the SQP method manages to achieve a quadratic convergence rate when solving the decomposed problem in practice. The optimized reference trajectories provided by the resolution of this problem are then implemented in a receding horizon controller, and compared with a holding baseline and a PI controller in an extensive case study based on real data from bus line 17 in Gothenburg, Sweden. It is found that the proposed MPC consistently outperforms the baselines in terms of headway regularity and energy efficiency. Energy savings of up to 9.3% were reported over the best performing baseline.

CHAPTER 5

Conclusion and future work

5.1 Discussion and conclusion

This thesis explored how the operational control problem of a bus line, where the goal is to provide a regular service to passengers, could be combined with the objective of minimizing the energy consumption of a bus fleet in operation.

The approach chosen was to adopt a velocity-based control strategy. While this type of intervention strategy is usually harder to model than station-based strategies, it is particularly well adapted for increasing the energy efficiency of the bus fleet. In general, a given travel time for a specified trip can be achieved by many speed profiles which can differ greatly in terms of energy consumption. This is especially true in the case of electric vehicles, where careless driving may result in a more frequent use of friction brakes than needed, thus wasting energy that could otherwise have been stored back into the battery. Hence, adjusting the bus speeds to match the energy-optimal speed profiles for a given level of service can address both the regularity and energetic aspects at the same time.

In order to express the regularity and eco-driving control problems for a bus line in a common modelling framework, an OCP was formulated. With a

particular choice of control horizon for each bus, the regularity of time headways between successive pairs of buses could be addressed directly. Predictive information on road gradient, bus stop locations, passenger arrival rates at stops, and average traffic speed was included in this OCP in order to generate optimized reference trajectories for the buses. Since this problem was formulated in the space domain, it could be expressed as a smooth NLP with no integer variables.

In this thesis, we proposed a resolution procedure for this NLP. It is centered on the idea of decomposing the NLP and carrying out most of the computations in parallel. Since the structure of the problem is such that only a few coupling terms between buses appear in the objective function, an equivalent bi-level decomposition of the NLP was found to be easier to treat. The SQP method was applied to solve the decomposed problem. At each iteration of the method, the bus-level sub-problems could be solved independently and in parallel, each aboard the corresponding bus. A QP approximation of the line-level master problem would then be solved at a central node, based on the sensitivity information transmitted by each of the buses. Therefore, the proposed resolution method is scalable in the number of buses, thanks to the parallel computations, and only requires few bytes of data to be exchanged between the central node and the buses when solving the decomposed problem.

In order to evaluate the proposed method in a practical context, some simulation results were presented at the end of Chapter 3, which are a part of the larger case study realized in Paper B. To obtain these results, the optimal solutions of the decomposed problem were implemented in a receding horizon controller, where it was assumed that they acted as high-level reference trajectories for local tracking controllers. In the complete case study, this MPC was compared with two traditional baselines in scenarios with increasing levels of bus bunching. The final conclusions were similar to the observations drawn from the selected results presented in this thesis. Namely, the MPC was consistently the most efficient controller for the dissipation of the initial service irregularities. It was also generally able to outperform both baselines in terms of headway regularity, but at the price of a lower commercial speed on average compared with the holding baseline. In addition, the MPC managed to achieve good energy saving, while providing a similar level of service. The energy efficiency of the MPC was generally in the order of 5-9% better than that of the best performing baseline, depending on the strength of the

initial bunching.

5.2 Future work

We consider that the following constitute promising research directions to extend and complement the research presented in this thesis.

Bus capacity constraints

In this thesis, as well as in the appended papers, it is assumed that the onboard capacity of buses is unlimited. While this assumption is reasonable in the case of weak operational disturbances, due to the prior tactical planning phase usually carried out by transit agencies, it becomes more questionable in the case of large service irregularities. However, capacity constraints are hard to model in a mathematical programming framework without resorting to integer decision variables. A way to circumvent this could have been through a similar relaxation of the OCP as the one performed in Paper B to handle the torque discontinuity. The issue is that the energetic argument on which this relaxation hinges no longer applies for a relaxation of the bus mass. Indeed, it is not always optimal for the bus to have a low mass from an energetic point of view, mainly due to the fact that part of its kinetic energy can be transferred back into its battery. Therefore, the relaxation of the bus mass would not be tight in downhill sections, for example. Another, perhaps more promising, approach would be to resort to additional control inputs to allow buses to deny passenger boarding. The capacity constraint could then be modeled as a soft constraint in the objective function of the OCP. It is yet unclear how this new term would affect the trade-off between the other terms, however.

Charging and state of charge constraints

Certain models of commercially available electric buses are designed to be able to complete a full day in operation with only one battery charge, but most electric buses have smaller battery packs and are designed to be charged periodically during normal operation. In the latter case, the state of charge of each bus should be carefully monitored to avoid running out of energy due to the limited driving range. For station-based charging, the charging decisions could be added as continuous decision variables to the OCP presented

in this thesis. These decisions could then been made based on the current operational state of the bus line, by e.g. limiting the charging of late buses. A more detailed disturbance model might be needed, however, to reliably estimate the evolution of the state of charge. In order to guarantee the trip feasibility in a robust way for each bus, a stochastic MPC could be designed and implemented.

Bus networks

A natural extension of the proposed method, which solves the regularity and eco-driving problem for a single bus line, is to solve that same problem for an entire bus network. The main difficulty which arises when several interacting bus lines are considered is that of merging them in shared corridors. Indeed, if no fixed order is assumed between buses belonging to different lines, the problem becomes combinatorial in nature since it must now be decided in what order buses merge in shared sections. In addition to this, it becomes unclear how the horizon of each bus should be defined in this OCP. The choice of control horizons presented in this thesis might have to be modified accordingly for this new system. Finally, it is also unclear how headway regularity should be defined, especially in shared sections, if the bus lines have different operating frequencies. A possible alternative would be to assess service regularity directly through passenger waiting times. In light of these considerations, scaling up the control method proposed in this thesis to a bus network might prove to be a significant research challenge.

References

[1] I. T. Forum, *ITF Transport Outlook 2019*. OECD Publishing, 2019.

[2] A. Nordelöf, M. Romare, and J. Tivander, "Life cycle assessment of city buses powered by electricity, hydrogenated vegetable oil or diesel," *Transportation Research Part D: Transport and Environment*, vol. 75, pp. 211–222, 2019.

[3] B. Zhou, Y. Wu, B. Zhou, R. Wang, W. Ke, S. Zhang, and J. Hao, "Real-world performance of battery electric buses and their life-cycle benefits with respect to energy consumption and carbon dioxide emissions," *Energy*, vol. 96, pp. 603–613, 2016.

[4] 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, 2016.

[5] Y. Wang, Y. Huang, J. Xu, and N. Barclay, "Optimal recharging scheduling for urban electric buses: A case study in davis," *Transportation Research Part E: Logistics and Transportation Review*, vol. 100, pp. 115–132, 2017.

[6] A. Sciarretta, G. De Nunzio, and L. L. Ojeda, "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem," *IEEE Control Systems Magazine*, vol. 35, no. 5, pp. 71–90, 2015.

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

[8] N. Murgovski, B. Egardt, and M. Nilsson, "Cooperative energy management of automated vehicles," *Control Engineering Practice*, vol. 57, pp. 84–98, 2016.

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

[10] A. Hamednia, N. Murgovski, and J. Fredriksson, "Predictive velocity control in a hilly terrain over a long look-ahead horizon," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 485–492, 2018.

[11] M. Held, O. Flärdh, and J. Mårtensson, "Optimal speed control of a heavy-duty vehicle in urban driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1562–1573, 2018.

[12] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Model predictive control of vehicles on urban roads for improved fuel economy," *IEEE Transactions on control systems technology*, vol. 21, no. 3, pp. 831–841, 2012.

[13] G. F. Newell and R. B. Potts, "Maintaining a bus schedule," in *Australian Road Research Board (ARRB) Conference, 2nd, Melbourne*, vol. 2, 1964.

[14] A. Bar-Yosef, K. Martens, and I. Benenson, "A model of the vicious cycle of a bus line," *Transportation Research Part B: Methodological*, vol. 54, pp. 37–50, 2013.

[15] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz, "Planning, operation, and control of bus transport systems: A literature review," *Transportation Research Part B: Methodological*, vol. 77, pp. 38–75, 2015.

[16] K. Gkiotsalitis and O. Cats, "At-stop control measures in public transport: Literature review and research agenda," *Transportation Research Part E: Logistics and Transportation Review*, vol. 145, p. 102 176, 2021.

[17] Y. Xuan, J. Argote, and C. F. Daganzo, "Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis," *Transportation Research Part B: Methodological*, vol. 45, no. 10, pp. 1831–1845, 2011.

[18] K. Gkiotsalitis and O. Cats, "Multi-constrained bus holding control in time windows with branch and bound and alternating minimization," *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 1258–1285, 2019.

[19] C. F. Daganzo, "A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons," *Transportation Research Part B: Methodological*, vol. 43, no. 10, pp. 913–921, 2009.

[20] J. J. Bartholdi III and D. D. Eisenstein, "A self-coördinating bus route to resist bus bunching," *Transportation Research Part B: Methodological*, vol. 46, no. 4, pp. 481–491, 2012.

[21] S. J. Berrebi, K. E. Watkins, and J. A. Laval, "A real-time bus dispatching policy to minimize passenger wait on a high frequency route," *Transportation Research Part B: Methodological*, vol. 81, pp. 377–389, 2015.

[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," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 757–769, 2010.

[23] F. Delgado, J. C. Muñoz, and R. Giesen, "How much can holding and/or limiting boarding improve transit performance?" *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1202–1217, 2012.

[24] D. Hernández, J. C. Muñoz, R. Giesen, and F. Delgado, "Analysis of real-time control strategies in a corridor with multiple bus services," *Transportation Research Part B: Methodological*, vol. 78, pp. 83–105, 2015.

[25] O. Cats, A. N. Larijani, H. N. Koutsopoulos, and W. Burghout, "Impacts of holding control strategies on transit performance: Bus simulation model analysis," *Transportation Research Record*, vol. 2216, no. 1, pp. 51–58, 2011.

[26] Z. Liu, Y. Yan, X. Qu, and Y. Zhang, "Bus stop-skipping scheme with random travel time," *Transportation Research Part C: Emerging Technologies*, vol. 35, pp. 46–56, 2013.

[27] M. Estrada, J. Mensión, J. M. Aymamı, and L. Torres, "Bus control strategies in corridors with signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 500–520, 2016.

[28] A. Petit, Y. Ouyang, and C. Lei, "Dynamic bus substitution strategy for bunching intervention," *Transportation Research Part B: Methodological*, vol. 115, pp. 1–16, 2018.

[29] C. F. Daganzo and J. Pilachowski, "Reducing bunching with bus-to-bus cooperation," *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 267–277, 2011.

[30] K. Ampountolas and M. Kring, "Mitigating bunching with bus-following models and bus-to-bus cooperation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[31] B. Varga, T. Tettamanti, and B. Kulcsár, "Optimally combined headway and timetable reliable public transport system," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 1–26, 2018.

[32] I. I. Sirmatel and N. Geroliminis, "Mixed logical dynamical modeling and hybrid model predictive control of public transport operations," *Transportation Research Part B: Methodological*, vol. 114, pp. 325–345, 2018.

[33] B. Varga, T. Tettamanti, and B. Kulcsár, "Energy-aware predictive control for electrified bus networks," *Applied Energy*, vol. 252, p. 113 477, 2019.

[34] K. Gkiotsalitis and E. Van Berkum, "An exact method for the bus dispatching problem in rolling horizons," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 143–165, 2020.

[35] W. Fan and R. B. Machemehl, "Do transit users just wait for buses or wait with strategies? some numerical results that transit planners should see," *Transportation Research Record*, vol. 2111, no. 1, pp. 169–176, 2009.

[36] L. Guzzella and A. Sciarretta, *Vehicle propulsion systems*. Springer-Verlag, 2013, vol. 3.

[37] E. Hans, N. Chiabaut, and L. Leclercq, "Investigating the irregularity of bus routes: Highlighting how underlying assumptions of bus models impact the regularity results," *Journal of Advanced Transportation*, vol. 49, no. 3, pp. 358–370, 2015.

[38] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 71–83, 2007.

[39] 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, 2012.

[40] N. Murgovski, L. M. Johannesson, and B. Egardt, "Optimal battery dimensioning and control of a cvt phev powertrain," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2151–2161, 2013.

[41] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

[42] L. O. Seman, L. A. Koehler, E. Camponogara, L. Zimmermann, and W. Kraus, "Headway control in bus transit corridors served by multiple lines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4680–4692, 2019.

[43] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 2567–2573.

[44] R. Hult, "Optimization-based coordination strategies for connected and autonomous vehicles," PhD thesis, Chalmers University of Technology, 2019.

[45] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *American Control Conference (ACC)*, IEEE, 2015, pp. 763–768.

[46] J. Nocedal and S. Wright, *Numerical optimization*. Springer, 2006.

[47] G. Still, "Lectures on parametric optimization: An introduction," *Optimization Online*, 2018.

[48] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[49] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "Qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[50] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[51] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of process control*, vol. 19, no. 5, pp. 723–731, 2009.

[52] 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," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 197–211, 2018.

[53] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

Part II

Papers

PAPER A

Hierarchical control of electric bus lines

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár

Accepted in 21st *IFAC World Congress*, Berlin, Germany, Jul. 2020

The layout has been revised.

Abstract

In this paper, we propose a hierarchical control strategy for a line of electric buses with the double objective of minimizing energy consumption and providing regular service to the passengers. The state-space model for the buses is formulated in space rather than in time, which alleviates the need for integer decision variables to capture their behavior at bus stops. This enables us to first assemble a fully-centralized multi-objective line problem in the continuous nonlinear optimization framework. It is then reassembled into a hierarchical structure with two levels of control in order to improve on scalability and reliability. This new supervisory structure consists of a centralized line level controller which handles the time headway regularity of the buses, and of decentralized bus level controllers which simultaneously manage the energy consumption of each individual bus. Our method demonstrates good battery energy savings and regularity performances when compared to a classical holding strategy.

1 Introduction

Bus networks in general are known to be very vulnerable to disturbances from their surroundings, which can in turn substantially increase passengers waiting times. Newell and Potts (1964) first proved that buses have a natural tendency to bunch if delayed, primarily due to the accumulation of passengers at stops. More recently, online control strategies have been proposed, benefiting from the rapid growth of GPS technology and vehicle-to-vehicle communications. Some are primarily based on real-time information, such as those which elaborate on Daganzo (2009) and Bartholdi and Eisenstein (2012). Other approaches rely on model-based predictions instead (Varga et al., 2018). However, most of these works focus solely on reducing bus bunching, and as such run the risk of sacrificing potential savings in energy consumption.

Due to a reduced dependence on fossil fuels, electric buses are a promising solution to reduce the environmental impact of transport systems (Lajunen et al., 2016). But battery and charging constraints are still hindering their

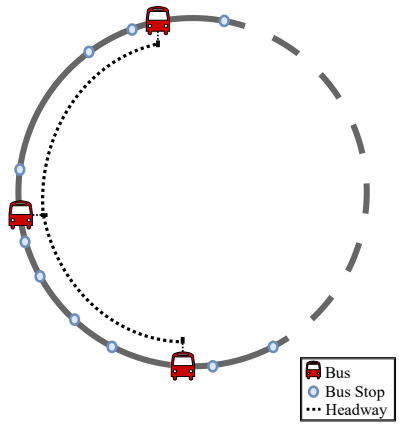


Figure 1: Representation of the bus line considered in this paper. The electric buses are running on a circular route and the controllers can estimate and adapt their predicted time headways.

market penetration today as they entail additional cost for transit operators. One way to mitigate this is to reduce the energy consumed by the buses when operated. Energy-saving driving strategies have been extensively investigated for heavy-duty vehicles in all kind of driving environments (Murgovski et al., 2016; Held et al., 2018). However, works on this topic have focused almost exclusively on trucks, and to the best of our knowledge only Varga et al. (2019) have addressed the energy-optimal bus line control problem. In their work, they formulate the line problem in the time domain and in a fully-centralized way. This may cause some reliability and scalability issues, which we try to address here.

In this paper, we focus on a circular route which is serviced by a single line of electric buses. These buses are made to keep a constant time headway provided by the transit agency between them. Such regularity-based operating policies are common in an urban context where small headways between consecutive buses mean that passenger arrivals can be considered as random (Fan and Machemel, 2009). Buses are allowed to run at different speeds between stops, both as a way to absorb deviations from the goal headway and to adapt to local settings (location of bus stops, road topography) to spare battery energy when possible. They are however not allowed to dwell at stops longer than needed to pick passengers up in order to avoid disturbing the

surrounding traffic. In addition, we make the assumption that the buses are operated automatically, i.e. without the supervision of a human driver. This enables them to react faster and more accurately to the commands and thus improves their energy-saving potential.

The main contributions of this paper are twofold. The first contribution is to assemble the multi-objective bus line optimization problem as a nonlinear program (NLP) thanks to a spatial formulation of bus dynamics. Choosing a temporal formulation for the dynamics would have required adding integer variables for the decisions of buses at stops, thus resulting in mixed-integer programs, which are notoriously harder to solve. The second contribution is to formulate the bus line problem in a hierarchical control framework. The proposed hierarchical structure contains several parallelizable finite horizon sub-problems which are solved at the bus level, and a simple constrained quadratic program (QP) at the line level. This allows for lower computation times compared with a fully-centralized approach. It also improves the reliability of the control as the commands can be computed directly in each bus independently from the rest of the line, thus making the system more resilient to e.g. communication errors.

2 Bus dynamics

In this section, the longitudinal dynamics are first described at the bus level, and are then combined at the line level. For the rest of this paper, we consider a setting where n buses are running on a circular route of total length L with q bus stops.

2.1 Longitudinal dynamics

For an electric bus $i \in \mathbb{I}_{[1,n]}$, the equations of motion along a fixed route are

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

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

where s_i is the bus position and v_i its velocity at time t, m_i is its mass, which is space-dependent, and $F_{m,i}$ is the force created at the wheels by the electric motor. The force $F_{b,i}$ is generated by the friction brakes, while the

aerodynamic drag $F_{d,i}$ and a force $F_{r,i}$ gathering the rolling resistance and the gravitational effects can be expressed as

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

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

where ρ is the air density, A_{bus} is the frontal area of the vehicle, c_a is the aerodynamic air drag coefficient and c_r is the rolling resistance coefficient. The function α is the road gradient, which depends on the position of bus i.

Since m_i is also space-dependent, due to uneven passengers loads between stops, it is more practical to express (A.1) in the space coordinate system s in order to remove the nonlinearity in the dynamics (Murgovski et al., 2016). We now consider the travel time t_i of bus i as a state instead of its position. The kinetic energy for a unit mass $E_i(s) = \frac{1}{2}v_i^2(s)$ is chosen as a state instead of v_i to simplify the expressions. The modified dynamics read

$$\frac{dt_i}{ds} = \frac{1}{\sqrt{2E_i(s)}}, \tag{A.3a}$$

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

where the second equation is a reformulation of (A.1b). The state vector for bus i is then $x_i(s) = [t_i(s), E_i(s)]^\top$. Note here that equation (A.3a) imposes that the speed of bus i should be strictly positive, even though it needs to stop frequently in reality. This is addressed by enforcing very low positive speeds around bus stops, and by introducing additional delays, as explained in the rest of this section.

2.2 Speed corridor

Some constraints must be introduced for the speed of the buses in order to e.g. capture that they should slow down around bus stops and comply with the speed limits. To do so, we create a space-dependent speed corridor, similar to Held et al. (2018), inside which the speed of each bus is constrained to be. Its upper and lower bounds are noted $v_{\max}$ and $v_{\min}$ and the constraints for

a bus $i \in \mathbb{I}_{[1,n]}$ materialize as

$$\frac{1}{2}v_{\min}^2(s) \leq E_i(s) \leq \frac{1}{2}v_{\max}^2(s). \tag{A.4}$$

Both bounds are positive, even at stops, to keep the kinetic energy from becoming null. Fig. 2. displays the appearance of the speed corridor around one stop, and this can be extended to the whole route without loss of generality.

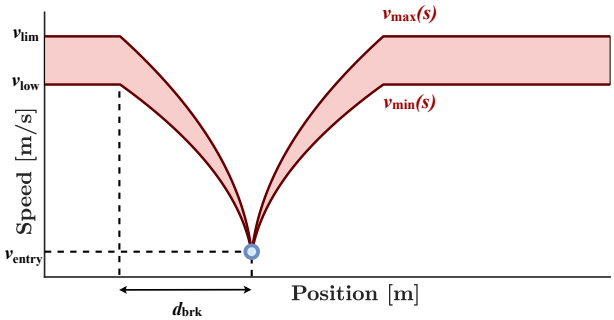


Figure 2: Illustration of the speed corridor around a stop. The shaded area represents the feasible speeds for the buses. At the stop, the buses are forced to slow down to a small velocity v_{entry}. When they are within a distance d_{brk} of a stop, it is their acceleration which is bounded to avoid disturbing the passengers on-board. Otherwise, the buses are limited by the speed limit of the road v_{lim}, and by a minimum acceptable speed v_{low} that prevents from going too slowly.

2.3 Electric motor and battery

We now elaborate a model to compute the energy consumed by the buses. Since electric motors can be used both for traction and for generation, the motor torque can be positive or negative depending on the operating scenario. For an electric bus $i \in \mathbb{I}_{[1,n]}$, the motor torque $T_{m,i}$ is related to the force at the wheels through $T_{m,i}(F_{m,i}) = r_w/(M_f\eta_f)F_{m,i}$ for the former case and $T_{m,i}(F_{m,i}) = (r_w\eta_f/M_f)F_{m,i}$ for the latter. Similarly, the motor speed $\omega_{m,i}$ is proportional to the bus speed with $\omega_{m,i}(E_i) = (M_f/r_w)\sqrt{2E_i}$. Here, r_w is the wheel radius, M_f is the final gear ratio and η_f is the efficiency coefficient of the final gear, with $\eta_f < 1$.

Due to the power limitation of the electric motor, the torque is constrained by

$$|T_{m,i}(F_{m,i})| \leq T_{\max}(E_i), \tag{A.5}$$

where $T_{\max}$ is the absolute value of the maximum torque that can be delivered for a given speed of the bus.

However, the motor torque is not continuously differentiable with respect to the command $F_{m,i}$, which can be problematic for solving optimization problems based on this model. A numerical work-around is to *lift* the model by adding separate force variables for each regime of the motor, such that

$$F_{m,i}(s) = F_{t,i}(s) - F_{g,i}(s), \tag{A.6a}$$

$$0 \leq F_{t,i}(s) \leq \frac{M_f \eta_f}{r_w} T_{\max}(E_i), \tag{A.6b}$$

$$0 \leq F_{g,i}(s) \leq \frac{M_f}{r_w \eta_f} T_{\max}(E_i), \tag{A.6c}$$

where $F_{t,i}$ and $F_{g,i}$ are the forces at the wheels for bus i when the motor is respectively working in traction or in generation. The command vector for bus i is thus $u_i(s) = [F_{t,i}(s), F_{g,i}(s), F_{b,i}(s)]^\top$. Note that $F_{t,i}$ and $F_{g,i}$ are mutually exclusive due to the subsequent problem design, as will be motivated in the next section. This yields a smoother expression for the motor torque

$$T_{m,i}(F_{t,i}, F_{g,i}) = \frac{r_w}{M_f \eta_f} F_{t,i} + \frac{r_w \eta_f}{M_f} F_{g,i}. \tag{A.7}$$

The internal battery power of bus i can be modeled as a nonlinear function $P_{b,i}(\omega_{m,i}, T_{m,i})$, as in e.g. Murgovski et al. (2014). The only assumption that we make on $P_{b,i}$ is that it is monotonously increasing with respect to $T_{m,i}$, i.e. that a higher motor torque draws more power from the battery, or equivalently supplies the battery with less power if $T_{m,i} < 0$.

2.4 Delays at stops

The dwell times of a bus $i \in \mathbb{I}_{[1,n]}$ at stops are not captured by the longitudinal dynamics model (A.3), but they can be added to the travel time t_i as extra delay terms D_j whenever the bus reaches a stop $j \in \mathbb{I}_{[1,q]}$. Each delay D_j consists of a constant term for stopping at and leaving stop j, and a variable

term for boarding the passengers currently waiting at that stop.

We note t_s the time it takes for a bus to come to a complete stop from the small velocity v_{entry} and to open its doors. It is assumed that it also takes t_s for it to close its doors and reach v_{entry} again when leaving the stop. In addition, we consider that the passengers arrive at stop j at a constant rate λ_j, and that it takes each passenger a time b to board the bus. The general expression for D_j is then

$$D_j(t_i, s) = \mathbf{1}_{s_j}(s)\left(2t_s + b\lambda_j T_j(t_i)\right) \tag{A.8}$$

where s_j is the position of stop j and T_j is the elapsed time since the preceding bus has left stop j. The indicator function $\mathbf{1}_{s_j}(s)$ is 1 when $s = s_j$ and 0 otherwise.

This expression for the delays ignores the influence of the alighting passengers, since alighting is usually faster than boarding (Petit et al., 2018). In addition, we use a normalized passengers model in order to keep the problem continuous. This means that the buses have to pick up passengers at each stop, which is a reasonable assumption for buses running in a dense urban environment. Other types of stops could be included, e.g. when buses have to halt due to traffic, but we choose to ignore those in this paper in order to keep a deterministic model.

3 Hierarchical bus line control

In this section, we motivate our choice of treating the line level optimization problem in a hierarchical framework. The fully-centralized line optimization problem is assembled and discussed in the first subsection. The following subsections then focus on detailing each step of the proposed hierarchical control formulation.

3.1 Fully-centralized line optimization problem

Let us assume that we are looking at a snapshot of the bus line at a given time, such as the one presented in Fig. 1. Each bus i is currently at a position p_i. The bus preceding it on the line has index $i+1$ and is at position p_{i+1}. In order to keep a schedule with a constant time headway H between the buses, each bus i should aim to travel from p_i to p_{i+1} as closely to H as possible. In

what follows, we note H_i the predicted time for bus i to travel that distance, which is defined as $H_i = t_i(p_{i+1})$. Note that H_i can be seen as the current headway between i and $i+1$. Therefore, the buses do not try to keep a headway H directly at the stops, but at whatever positions they have on the snapshot. The idea is that enforcing regular time headways at each point on the route will result in constant headways being enforced at the stops too. Based on the previous models, the centralized line control problem can now be written as

$$\begin{aligned}
\min_{U} \quad & \sum_{i=1}^{n}(H_i(x_i) - H)^2 \\
& + \alpha \sum_{i=1}^{n}(H_i(x_i) - H_{i-1}(x_{i-1}))^2 \\
& + \beta \sum_{i=1}^{n} \int_{p_i}^{p_{i+1}} \frac{P_{b,i}(x_i(s), u_i(s))}{\sqrt{2E_i(s)}} ds, && \text{(A.9a)} \\
\text{s.t.} \quad & \forall i \in \mathbb{I}_{[1,n]} : \\
& E_i(p_i) = \hat{E}_{0,i}, \quad t_i(p_i) = 0, && \text{(A.9b)} \\
& \forall s \in [p_i, p_{i+1}] : \\
& \frac{dx_i}{ds} = f_i\big(x_i(s), u_i(s)\big) + \sum_{j=1}^{q} D_j(t_i, s)\,[1, 0]^\top, && \text{(A.9c)} \\
& h_i\big(x_i(s), u_i(s)\big) \leq 0, && \text{(A.9d)}
\end{aligned}$$

where $U = [u_1, ..., u_n]^\top$ is the vector containing all the command functions, which are assumed to be piecewise constant. The initial kinetic energy of bus i is noted $\hat{E}_{0,i}$. The evolution function f_i corresponds to the dynamics (A.3), while h_i gathers the inequality constraints from (A.4), (A.6b) and (A.6c). Finally, α and β are trade-off parameters to weigh the relative importance of the different objectives in the cost function.

Remark 1: The second term in the cost function includes a *look-back* feature which constrains headways of neighboring buses to be similar. This ensures that the rate of service remains somewhat homogeneous locally, even as the headways are made to converge to H by the first term. The preponderance of each is adjusted with the parameter α. The competing objective of energy minimization is added as the third term and β regulates the trade-off.

However, the nonlinear optimal control problem (A.9) suffers from some practical issues. Indeed, centralized computations cause it to be unadapted to an on-line setting, and vulnerable to communication losses between the buses and the central coordinator. They also forbid to run any computations in parallel to ease the computational load. Therefore, we choose to structure (A.9) as a bi-level optimization problem instead by splitting it into a line level problem and bus level sub-problems to increase reliability. We respectively refer to each as the high-level or the low-level hereafter. At the high-level, the quadratic terms of the cost function (A.9a) are gathered into a static QP, which can be solved very quickly. The energy consumption is dealt with at the low-level, and can now be computed in parallel on each separate road segment $[p_i, p_{i+1}]$ instead of on the whole route at once. The *dialogue* between the high and the low-level is summarized in Fig. 3.

3.2 Low-level optimal control

Let us assume that a bus $i \in \mathbb{I}_{[1,n]}$ has received from the high-level controller a travel time goal $t_{f,i}$ to reach the preceding bus at p_{i+1}. This command concerns the predicted time headway H_i, and bus i should adapt its predicted trajectory such that $H_i(x_i) = t_{f,i}$ if possible. Note that $t_{f,i}$ can be quite different from the goal headway H, e.g. if the line is currently very disturbed.

The following NLP is assembled to find the energy-optimal control of bus i that meets the travel time goal $t_{f,i}$

$$
\begin{aligned}
\min_{u_i} \quad & J_i(x_i, u_i) = \int_{p_i}^{p_{i+1}} \frac{P_{b,i}(x_i(s), u_i(s))}{\sqrt{2E_i(s)}} ds + C\,\kappa_i \\
\text{s.t.} \quad & E_i(p_i) = \hat{E}_{0,i}, \qquad t_i(p_i) = 0, && \text{(A.10a)} \\
& \kappa_i \geq H_i(x_i) - t_{f,i}, \quad \kappa_i \geq t_{f,i} - H_i(x_i), && \text{(A.10b)} \\
& \forall s \in [p_i, p_{i+1}] : \\
& \frac{dx_i}{ds} = f_i\big(x_i(s), u_i(s)\big) + \sum_{j=1}^{q} D_j(t_i, s)[1, 0]^\top, && \text{(A.10c)} \\
& h_i\big(x_i(s), u_i(s)\big) \leq 0, && \text{(A.10d)}
\end{aligned}
$$

where most functions have already been introduced for problem (A.9). We note J_i the cost function of this problem, κ_i is a slack variable, and C is a large

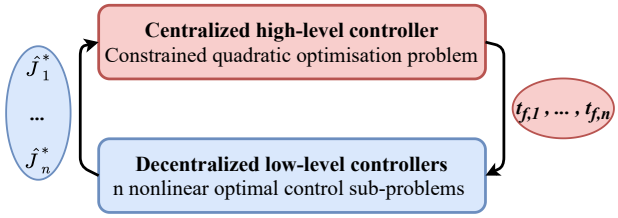


Figure 3: This diagram represents the information exchanged by the two control levels. The high-level controller sends a travel time command $t_{f,i}$ to each and every bus i of the line. Each low-level controller responds with the corresponding energy sensitivity function $\hat{J}_i^*$. Both control levels are operated with the same frequency until convergence of the loop, as they each need updated information from the other level.

constant coefficient. Note that this problem only needs to be solved locally, i.e. separately for each and every bus i. The low-level control can therefore always be carried out by using the latest available information, even for cases where commands from the high-level are temporarily unavailable.

Remark 2: The cost function J_i is the sum of the total electric energy consumed on the space-horizon and the slack variable. The motivation for introducing this last term is to guarantee that the problem is feasible, regardless of the travel time command $t_{f,i}$. Indeed, using this slack variable is akin to having a penalty term $|t_{f,i} - H_i(x_i)|$ in the objective function while keeping it smooth at the same time. And choosing a large coefficient C strongly penalizes solutions where the final time $H_i(x_i)$ is different from $t_{f,i}$, which imposes $H_i(x_i) = t_{f,i}$ whenever it is feasible.

Remark 3: Since (A.10) is an energy-minimization problem, the command function u_i is chosen to be as energy-efficient as possible. For a given feasible force requirement at the wheel, the lowest torque (and hence the lowest battery power) is achieved in (A.7) by having either $F_{t,i}$ or $F_{g,i}$ equal to 0 depending on the sign of the required force. This holds because we have assumed $P_{b,i}$ to be monotonously increasing with respect to the torque earlier. Therefore, $F_{t,i}$ and $F_{g,i}$ are mutually exclusive for this problem.

Remark 4: Numerical simulations are carried out to solve this optimal control problem. For bus i, we use multiple shootings to split the route segment $[p_i, p_{i+1}]$ into uniform shooting intervals. The standard Runge-Kutta method

is then used to integrate the dynamics of the discretized system on each shooting interval. The shooting points corresponding to the stops are known beforehand thanks to the sampling in space, so no integer variable needs to be added. In addition, even if the indicator function from (A.8) makes the NLP (A.10) discontinuous, its discretized counterpart becomes continuous in the optimization variables since the stops locations are known. Finally, having a decentralized control at the low-level means that these problems can be solved in parallel for each and every bus. In practice, they are found to be several time faster to solve than the fully-centralized problem (A.9).

3.3 Sensitivity analysis

Once the energy-optimal low-level control is known, we are interested in knowing the sensitivity of the energy consumption to the travel time command $t_{f,i}$ for a bus $i \in \mathbb{I}_{[1,n]}$. This information is crucial to address the trade-off between energy consumption and regular time headways, and will be used at the high-level of control.

Problem (A.10) can be seen as a parametric optimization problem, with the travel time command $t_{f,i}$ acting as a scalar parameter (Still, 2018). Let $J_i^*(t_{f,i}^0)$ be the parametric optimal cost for a given parameter value $t_{f,i}^0$. We know from *Remark 2* that this cost is equal to the energy consumption if and only if $t_{f,i}^0$ is a feasible time command. So the sensitivity analysis is only carried out if this condition holds, which is the case most of the time in practice.

Since there is no direct way to compute the implicit function J_i^*, we will approximate it with the function $\hat{J}_i^*$ based on the first two terms of its Taylor series instead

$$\begin{aligned}\hat{J}_i^*(t_{f,i}) = J_i^*(t_{f,i}^0) + \frac{\partial J_i^*}{\partial t_{f,i}}\big(t_{f,i}^0\big)\big(t_{f,i} - t_{f,i}^0\big) \\ + \frac{1}{2}\frac{\partial^2 J_i^*}{\partial t_{f,i}^2}\big(t_{f,i}^0\big)\big(t_{f,i} - t_{f,i}^0\big)^2.\end{aligned} \tag{A.11}$$

Let $\mathcal{L}_i$ be the Lagrange function of (A.10) and λ_i and μ_i the vectors of dual variables corresponding to the equality and inequality constraints respectively. Let $z_i = [w_i, \lambda_i, \mu_i]^\top$ be the solution vector, which gathers the primal and dual variables, with $w_i = [x_i, u_i]^\top$. By noting z_i^* the optimal solution, which is a

function of the parameter $t_{f,i}$, and applying the chain rule we have (Still, 2018)

$$\frac{\partial J_i^*}{\partial t_{f,i}}\big(t_{f,i}^0\big) = \left(\frac{\partial \mathcal{L}_i}{\partial t_{f,i}}\right)_{z_i^*(t_{f,i}^0),t_{f,i}^0} \tag{A.12a}$$

$$\frac{\partial^2 J_i^*}{\partial t_{f,i}^2}\big(t_{f,i}^0\big) = \left(\frac{\partial^2 \mathcal{L}_i}{\partial t_{f,i}^2} + \nabla^2_{t_{f,i}z_i}\mathcal{L}_i \frac{\partial z_i^*}{\partial t_{f,i}}\right)_{z_i^*(t_{f,i}^0),t_{f,i}^0} \tag{A.12b}$$

Note that the first two derivatives of J_i^* are not too expensive to obtain here since the parameter $t_{f,i}$ only appears in the constraints (A.10b). Let n_{z_i} and n_{μ_i} respectively be the length of the solution vector z_i, and the number of inequality constraints in problem (A.10). For simplicity, we assume that the constraints (A.10b) are the last two inequality constraints. Then it holds that

$$\frac{\partial J_i^*}{\partial t_{f,i}}\big(t_{f,i}^0\big) = \mu_i(n_{\mu_i}) - \mu_i(n_{\mu_i}-1), \tag{A.13a}$$

$$\frac{\partial^2 J_i^*}{\partial t_{f,i}^2}\big(t_{f,i}^0\big) = y_i(n_{z_i}-1) - y_i(n_{z_i}), \tag{A.13b}$$

where $y_i = R_{z,i}^{-1}\big(z_i^*(t_{f,i}^0), t_{f,i}^0\big)\, e$, $e = [0, ..., 0, -1, 1]^\top$, and where $R_{z,i}$ is the Jacobian of the KKT conditions vector with respect to the solution vector z_i (Hult et al., 2016).

Therefore, only one matrix inversion is needed to get a quadratic approximation $\hat{J}_i^*$ of the energy consumption sensitivity for bus i. The n sensitivity functions obtained are then passed as inputs to the high-level.

3.4 High-level controller

The role of the high-level controller is to find the best travel time commands $t_{f,1}$,...,$t_{f,n}$. This is formulated as a supervisory control problem with a static constrained QP:

$$\min_{t_{f,1},\ldots,t_{f,n}} \sum_{i=1}^{n} (t_{f,i} - H)^2 + \alpha(t_{f,i} - t_{f,i-1})^2 + \beta \hat{J}_i^*(t_{f,i}), \tag{A.14a}$$

$$\text{s.t.} \quad t_{f,i}^{\min} \le t_{f,i} \le t_{f,i}^{\max}, \quad \forall i \in \mathbb{I}_{[1,n]}. \tag{A.14b}$$

The approximate minimum and maximum feasible time commands $t_{f,i}^{\min}$ and $t_{f,i}^{\max}$ for any bus i are estimated from the solutions obtained by forcing the velocity of the buses to be equal to $v_{\min}$ and to $v_{\max}$ respectively. Thus they can provide a good approximation of the feasible set of the QP, which can be solved efficiently using dedicated toolboxes.

The travel time commands found from solving this QP are then fed in to the low-level again, and the loop continues until convergence. Note that both control levels are run with the same frequency during this process. Once the optimal solution of the overall problem is found, the bus line is updated in a MPC-like fashion with a sampling time Δt by implementing the low-level solutions from (A.10). This sampling time is unrelated to the low-level programs as it is common for all buses, and denotes the frequency at which the overall line problem is solved.

4 Numerical results

In this section, we present some simulation results to assess how well our hierarchical control method is able to maintain a regular level of service in an energy-efficient way. To do so, we chose to compare it to a bus holding benchmark method, as is done in Varga et al. (2019). In this scenario, buses go at the maximum allowed speed in-between stops, and then wait at some predesignated control points until the headway regularity condition is met. Many transit systems operators enforce this control method in practice as it is relatively inexpensive and easy to deploy. Following our previous notations, this means that the benchmark buses are forced to run at $v_{\max}$.

The route considered is that of bus line 17 in Gothenburg, Sweden, with some minor adjustments. Both control methods operate from the same initial disturbed state of the bus line in 5 different scenarios, ordered from least to most disturbed (i.e. scenario 1 is the least disturbed). Since no external disturbances have been added to the model, the controlled bus line will converge back to a state of regular service in all the scenarios. Two metrics are used to monitor each method's performances: the energy consumed per distance unit ΔE and the mean value $\Delta \hat{H}$ of the absolute headway deviations last observed at each of the stops. Both of them are computed at the line level. The numerical values for the relevant parameters are summarized in Table 1. The space-dependent functions α and m_i, for $i \in \mathbb{I}_{[1,n]}$, as well as the rates

λ_j, for $j \in \mathbb{I}_{[1,q]}$, are obtained directly from bus line 17 data.

Table 1: Simulation parameters. SI units: H, L, v_{lim}, v_{low}, v_{entry}, ρ, A_{bus}, r_w, d_{brk}, t_s, b, g. Unitless: n, q, c_a, c_r, M_f and η_f.

Parameter	n	q	H	L	v_{lim}	v_{low}	v_{entry}
Value	5	28	360	16492	13.89	11.11	1.39

Parameter	ρ	A_{bus}	c_a	c_r	r_w	M_f	η_f
Value	1.18	5.14	1	0.0047	0.49	2.8	0.98

Parameter	d_{brk}	t_s	b	g
Value	95.5	3	1.5	9.81

For each set of initial conditions, the bus line is simulated for n_{iter} iterations with a sampling time $\Delta t = 40s$. Due to the different initial disturbance strengths of the scenarios, the time required to converge back to a state of regular service varies. So n_{iter} is set separately for each scenario as the number of iterations required for the deviations $\Delta\hat{H}$ of both methods to become smaller than a threshold value, which we set to 15 seconds. The trade-off parameters α and β are chosen accordingly to make the deviation of the hierarchical control method reach that threshold value in the same number of iterations as the bus holding method when possible, as is illustrated for scenario 3 in Fig. 4. The results for each scenario are presented in Table 2.

Table 2: Results.

Scenario	1	2	3	4	5
ΔE control [kWh/km]	0.649	0.635	0.646	0.639	0.627
ΔE holding [kWh/km]	0.668	0.658	0.666	0.672	0.677
Energy savings [%]	2.9	3.6	3.1	5.2	8
n_{iter}	33	39	43	61	75

The hierarchical control method consistently consumes less energy to reach the headway deviation threshold, regardless of the initial conditions of the line. More energy can in fact be saved for scenarios with larger initial disturbances. This may be the consequence of needing longer simulation times to reach a regular level of service, as some buses will reach the desired headway sooner and have more time to drive in an energy-aware fashion.

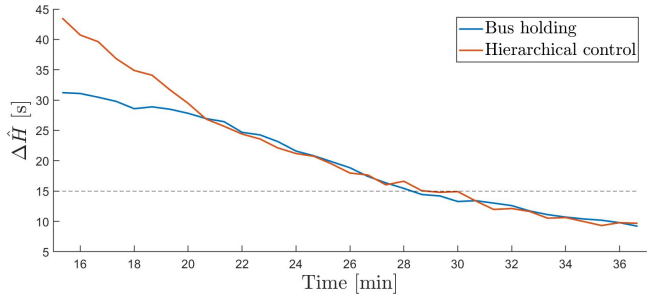


Figure 4: Average observed headway deviations at stops with respect to time for both methods in scenario 3. The dashed line is the 15 seconds threshold. This is obtained with $\alpha = 1$ and $\beta = 0.05$.

For the most disturbed cases (scenarios 4 and 5), the hierarchical control method is not able to reach the 15 seconds threshold as fast as the bus holding method, regardless of the trade-off parameters values. This is mainly due to an imbalance in what each method can do. Indeed, the bus holding method can rapidly absorb strong disturbances by having the buses wait at the control points, whereas the hierarchical control constrains the buses to be running at $v_{\min}$ or faster all the time. So it must be noted that this is not an inherent limitation of the hierarchical control strategy, as a bus-holding feature could be added to it to handle strongly disturbed lines.

5 Conclusion

In this paper we set to find a scalable and reliable control strategy that could solve a multi-objective bus line control problem. We chose to avoid formulating that problem as a mixed-integer program, but preferred a classical NLP formulation. It was first assembled as a fully-centralized problem based on the bus and line models developed before. But due to practical constraints (e.g. sensitivity to communication deficiencies), we split it into a bi-level hierarchical control structure. This new control structure only needs to solve independent sub-problems at the bus level, and a single quadratic problem at the line level. When implemented on a simple test case, it demonstrates a promising potential for sparing battery energy and keeping constant headways

in weakly disturbed situations.

However, it was also found that our approach should be augmented with an additional intervention method to absorb strong disturbances more quickly. Another direction for future works is to evaluate quantitatively the complexity of different problem formulations to find the most appropriate one for a future on-line implementation.

References

Bartholdi, J. and Eisenstein, D. (2012). A self-coördinating bus route to resist bus bunching. *Transportation Research B, Methodology*, 46, 481–491.

Daganzo, C.F. (2009). A headway-based approach to eliminate bus bunching. *Transportation Research B, Methodology*, 43, 913–921.

Fan, W. and Machemehl, R. (2009). Do transit users just wait for buses or wait with strategies? *Transportation Research Record*, 2111, 169–176.

Held, M., Flärdh, O., and Mårtensson, J. (2018). Optimal speed control of a heavy-duty vehicle in urban driving. *IEEE Transactions on Intelligent Transportation Systems*, 20, 1562–1573.

Hult, R., Zanon, M., Gros, S., and Falcone, P. (2016). Primal decomposition of the optimal coordination of vehicles at traffic intersections. In *Conference on Decision and Control (CDC)*, 2567–2573.

Lajunen, A. and Lipman, T. (2016). Lifecycle cost assessment and carbon dioxide emissions of diesel, natural gas, hybrid electric, fuel cell hybrid and electric transit buses. *Energy*, 106, 329–342.

Murgovski, N., Egardt, B., and Nilsson, M. (2016). Cooperative energy management of automated vehicles. *Control Engineering Practice*, 57, 84–98.

Murgovski, N., Johannesson, L., and Egardt, B. (2014). Optimal battery dimensioning and control of a CVT PHEV powertrain. *IEEE Transactions on Vehicular Technology*, 63, 2151–2161.

Newell, G.F. and Potts, R.B. (1964). Maintaining a bus schedule. In *Conference on Australian Road Research Board (ARRB)*. 2nd edition.

Petit, A., Ouyang, Y., and Lei, C. (2018). Dynamic bus substitution strategy for bunching intervention. *Transportation Research B, Methodology*, 115, 1–16.

Still, G. (2018). *Lectures on parametric optimization: an introduction*. Optimization Online.

Varga, B., Tettamanti, T., and Kulcsár, B. (2018). Optimally combined headway and timetable reliable public transport system. *Transportation Research C, Emerging Technologies*, 92, 1–26.

Varga, B., Tettamanti, T., and Kulcsár, B. (2019). Energy-aware predictive control for electrified bus networks. *Applied Energy*, 252.

PAPER B

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

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, Balázs Kulcsár

Submitted to *IEEE Transactions on Intelligent Transportation Systems (ITS)* in Nov. 2020

The layout has been revised.

Abstract

The problems of bus bunching mitigation and of the energy management of groups of vehicles are traditionally treated separately in the literature, and 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 only has 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 leverages the decentralized structure of a 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.

1 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.

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 distributed resolution procedure based on 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 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.

To clarify the terminology used in this paper, it must be noted that the control strategy proposed has a hierarchical architecture [31], since the MPC scheme operates centrally to computes optimized reference trajectories. Independent local bus controllers are then assumed to track these references, but they are left outside the scope of this paper. However, the decomposition of the NLP solved in the MPC enables its resolution to be carried out in parallel, for the most part. Hence, this decomposition procedure is referred to as distributed optimization, in the sense used in e.g. [32].

This article is organized as follows. The modeling of buses and passengers is presented in Section 2. In Section 3, it is explained how the general optimal control problem can be reformulated and solved in a distributed fashion, and embedded in an MPC. Simulations results are shown and analyzed in Section 4. Finally, the paper closes on some concluding remarks in Section 5.

2 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

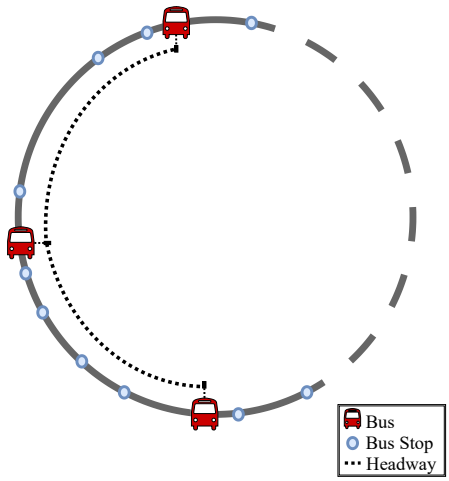


Figure 1: Illustration of a circular bus route with a nonhomogeneous distribution of bus stops.

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 4.

In order to ease the comprehension of the following modeling steps, we invite the reader to refer to Table 6 and Table 7 in the Appendix, which offer a detailed summary of all the notations introduced throughout the paper.

2.1 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 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.

2.2 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:

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

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

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 (B.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,

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

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

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 (B.1)-(B.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 (B.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 (B.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:

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

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

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 can not have zero speed, as imposed by (B.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, (B.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 are formulated on the kinetic energy instead, as:

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

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 $v_{\min}(s,t_i) \geq v_{\text{entry}} > 0$, $\forall s$, $\forall t_i$. 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.

2.3 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:

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

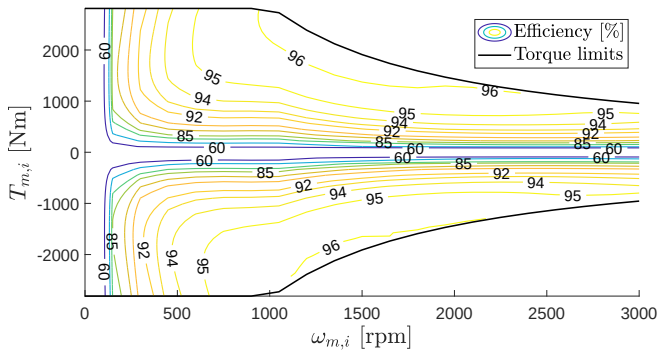


Figure 2: Efficiency map of the motor, as a function of its speed and torque. The black lines denote the torque constraints.

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 $F_{m,i}(s) \geq 0$ 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:

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

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].

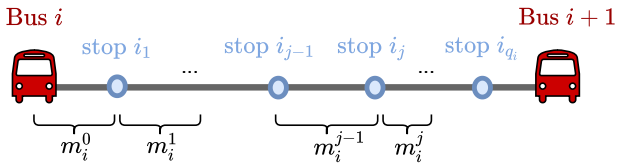


Figure 3: Prediction horizon for bus i, where the successive mass values at the stops downstream are shown.

2.4 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 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, $\mathcal{S}_i = \{i_1, ..., i_{q_i}\}$, 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

[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.

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 $i+1$ 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 $\lambda_{i_j}(t_i(s_{i_j}) - t_{i+1}^j)$ 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 (B.3b) to capture the behavior of buses at stops can then be expressed as:

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

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.

2.5 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 $\{m_i^0, m_i^1 ..., m_i^{q_i}\}$ 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_i^0 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_i^j past stop i_j can be computed recursively from the previous one m_i^{j-1} as:

$$m_i^j = (1-\mu_{i_j})(m_i^{j-1} - m_{\text{emp}}) + m_{\text{pax}}\lambda_{i_j}\big(t_i(s_{i_j}) - t_{i+1}^j\big), \tag{B.8}$$

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 (B.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_i^j can be written as an explicit function of the travel time and the initial mass:

$$\begin{aligned} m_i^j = m_i^0 \prod_{l=1}^{j}(1-\mu_{i_l}) - m_{\text{emp}} \sum_{l=1}^{j}\prod_{r=l}^{j}(1-\mu_{i_r}) \\ + m_{\text{pax}} \sum_{l=1}^{j} \prod_{r=l+1}^{j} (1-\mu_{i_r})\lambda_{i_l}(t_i(s_{i_l}) - t_{i+1}^l). \end{aligned} \tag{B.9}$$

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

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

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

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_i^j, $j \in \mathbb{I}_{[1,q_i]}$, in the travel time is now written explicitly.

3 Distributed 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 bi-level 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.

It must be observed that it is not question here of distributed control, in the sense given in [31], since parts of the computations need to be done centrally.

3.1 Optimal control formulation

A predictive bus line model can now be derived from the modeling steps taken in Section 2. 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 $x(s) = [x_1(s), ..., x_n(s)]^\top$ and $u(s) = [u_1(s), ..., u_n(s)]^\top$, 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:

$$\begin{aligned}
\min_{x(s),u(s)} \quad & \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha \sum_{i=1}^{n} (H_i - H_{i-1})^2 \\
& + \beta \sum_{i=1}^{n} \int_{p_i}^{p_{i+1}} \frac{P_{b,i}(T_{m,i}, \omega_{m,i})}{\sqrt{2E_i(s)}} ds, && \text{(B.11a)} \\
\text{s.t.} \quad & \forall i \in \mathbb{I}_{[1,n]} : \\
& E_i(p_i) = E_i^0, \quad t_i(p_i) = t^0, && \text{(B.11b)} \\
& H_i = t_i(p_{i+1}) - t^0, && \text{(B.11c)} \\
& \text{(B.3), (B.4), (B.6),} && \text{(B.11d)}
\end{aligned}$$

where the state dynamics (B.3) and the constraints (B.4), (B.6) are enforced for all $s \in [p_i, p_{i+1}]$ for each bus i, and where the bus mass can be computed explicitly from (B.9) and (B.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 (B.5) in the expressions that use them. The objective function (B.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 (B.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 (B.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.

3.2 Direct reformulation of the OCP

As a next step towards the resolution of the problem presented, we propose a direct optimal control reformulation of (B.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:

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

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

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

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

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

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

where $X_i = [x_{i,0}, ..., x_{i,N}]^\top$, and $U_i = [u_{i,0}, ..., u_{i,N-1}]^\top$ are vectors gathering the optimization variables relative to bus i, and where $X = [X_1, ..., X_n]^\top$, and

$U = [U_1, ..., U_n]^\top$. The numerical integration of the state dynamics (B.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 (B.10) can be obtained easily for the bus mass, as it is already piecewise constant in space, and used in (B.12d). Finally, the function g gathers the inequality constraints from (B.4) and (B.6).

Remark 2: One important feature of NLP (B.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 (B.12) now comes from the discretization of the torque constraints (B.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 (B.5). This can by 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 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 (B.6) may now be rewritten as:

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

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

thus removing any non-smoothness from (B.12), since the motor torque $T_{m,i,k}$

can now be expressed as:

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

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

$$\begin{aligned}
\min_{X,U} \quad & \sum_{i=1}^{n} \frac{1}{2}\Lambda_i H_i^2 + \alpha \sum_{i=1}^{n} (H_i - H_{i-1})^2 \\
& + \beta \sum_{i=1}^{n} \sum_{k=0}^{N-1} \tilde{J}(x_{i,k}, u_{i,k}), && \text{(B.15a)} \\
\text{s.t.} \quad & \forall i \in \mathbb{I}_{[1,n]} : \\
& (\text{B.12b}), (\text{B.12c}), (\text{B.12d}), && \text{(B.15b)} \\
& \tilde{g}(x_{i,k}, u_{i,k}) \leq 0, \qquad k \in \mathbb{I}_{[0,N-1]}, && \text{(B.15c)}
\end{aligned}$$

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

Now, it remains to show that this smooth relaxation (B.15) of the original problem (B.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 1.

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

3.3 Decomposition

Solving the lifted version (B.15) of the fully-centralized NLP as such presents some difficulties. The nonlinear dynamics from (B.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 bi-level decomposition.

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

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

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

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

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

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

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

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

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

Remark 3: Due to the non-convexity of the bus-level sub-problems (B.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 (B.16)-(B.17) has the same set of KKT points [39] as the original problem (B.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 (B.16)-(B.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 distributed resolution where the bus-level NLPs (B.17) could be solved independently aboard each bus, while

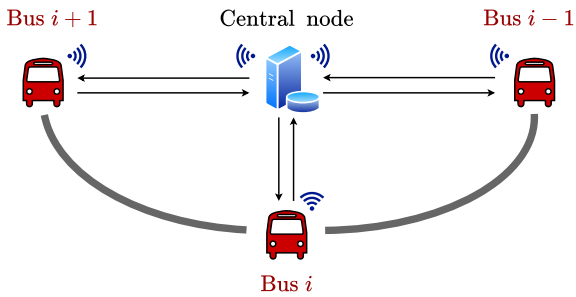


Figure 4: Resolution scheme of the decomposed problem. The arrows denote the remote communication between the central node and the buses. Note that the buses do not need to share information with each other in these settings.

only (B.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 2.

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

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

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

and,

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

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

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

This bi-level decomposition of the original problem is not a panacea, however. The high-level problem (B.16) remains a non-convex NLP, with an objective function defined implicitly through (B.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 2. In what follows, we assume that the resolution of the decomposed problem has been carried out.

3.4 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 (B.16)-(B.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 (B.16)-(B.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. 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

(B.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 (B.16)-(B.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.

Algorithm 1 MPC for the bus line problem

initialize state, time $\leftarrow 0$, data $\leftarrow \{\}$
while time $-$ time_end < 0 **do**
  X, $U \leftarrow$ solve (B.16)-(B.17)
  state $\leftarrow$ system_evolution (state, X, U, T)
  data $\leftarrow \{\text{data}, \text{state}\}$
  time $\leftarrow$ time $+ T$
end
return data

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.

4 Simulations

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

4.1 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 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 fluctu-

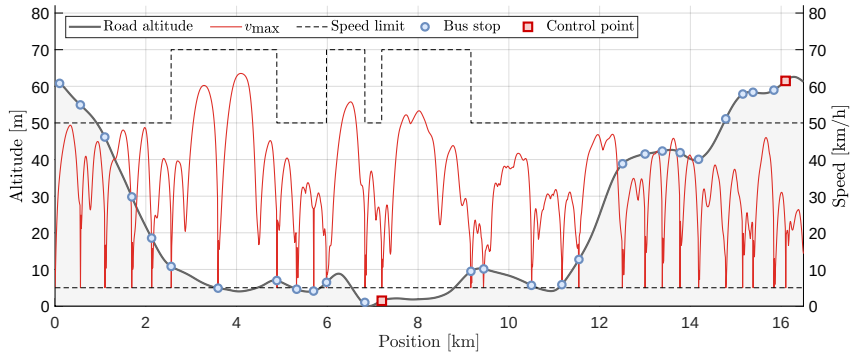


Figure 5: Route layout of bus line 17. The bus stops are placed according to their altitude on the route, and it is indicated which of them serve as control points for the holding baseline. The solid red line is the average maximum speed obtained from historical bus driving profiles, and is used as the 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}.

ations 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 (B.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 (B.17) and (B.22), which are then solved with the primal-dual interior point solver IPOPT and with the active-set solver qpOASES, respectively.

4.2 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 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.

4.3 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

Table 1: 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

defined as:

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

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 (B.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.

4.4 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 1. 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 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 2 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 $\beta = 2.78\,\mathrm{s/kW}$.

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

Table 2: 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

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 7 in the Appendix gathers the rest of the numerical values for the parameters used in the simulations.

4.5 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

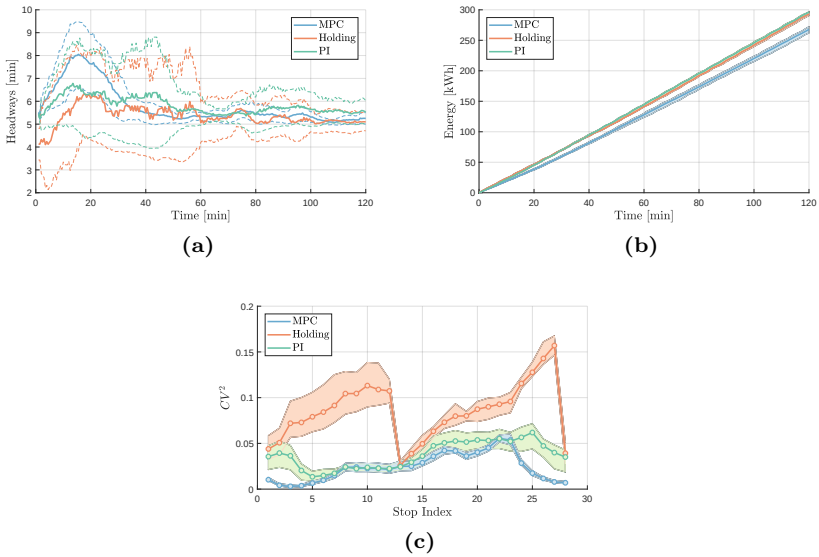


Figure 6: Results for scenario 5. In these figures, the shaded areas are bounded above and below by the maximum and minimum metrics values observed among all the simulations of scenario 5 for the control strategy concerned. (a) Temporal evolution of the average headways at stops, for one simulation instance 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).

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

Table 3: 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

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. 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 (B.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.

The same general patterns, in terms of headway regularization and energy consumption, are also found when investigating the other scenarios. Table 3 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 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

Table 4: 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%

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 4 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.

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 4 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.

4.6 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 5.

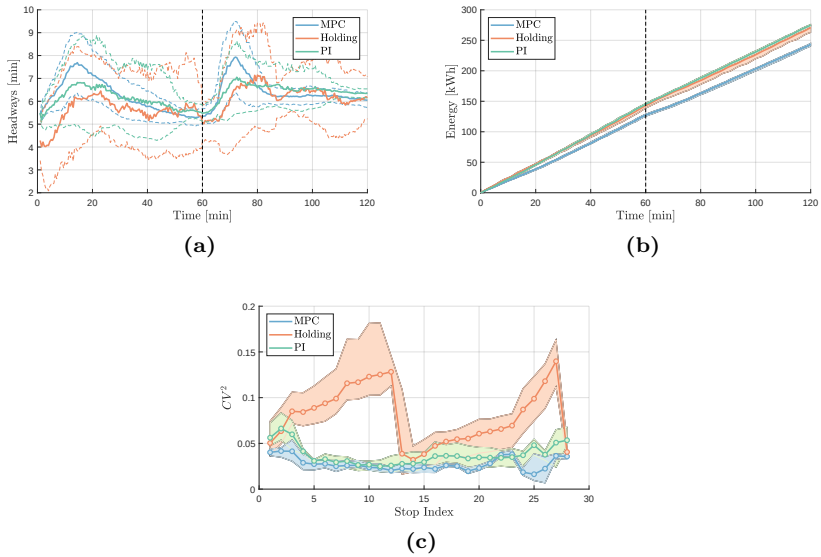


Figure 7: Results for scenario 5 in the case where one bus breaks down after one hour of simulation (dashed vertical lines). As before, the shaded areas are bounded by the most extreme metrics values observed. (a) Average headways at stops for one simulation instance (solid lines). (b) Average total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

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 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

Table 5: 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

more slowly after one bus is removed from the line.

The headway sample means presented in Table 5 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 (B.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.

5 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 works, 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 bi-level optimization scheme was

proposed to distribute 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. Likewise, 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 [43]. 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 e.g. 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 prob-

lem 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.

1 Proof of proposition 1

The lifted NLP (B.15) only differs from the original problem (B.12) through the modified torque constraints (B.13) and torque expression (B.14). A careful examination of the torque constraints (B.5)-(B.6) and (B.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 (B.5) and (B.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 (B.5) and (B.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 (B.14) that $r_w/(\eta_f M_f) > \eta_f r_w/M_f > 0$. It follows from this observation that:

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

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

where $T_{m,i,k}$ is defined according to (B.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 (B.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.

2 Resolution of the decomposed problem

This appendix presents how the high-level NLP (B.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):

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

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

where $H^+ = [H_1^+, ..., H_n^+]^\top$ 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 (B.16) in order to obtain the local QP approximation (B.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 (B.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 (B.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 (B.17) for bus i can be solved for any fixed parameter value H_i by using primal-dual interior point algorithms [39]. The relaxation

Table 6: 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 7: Numerical values of the parameters used in the simulations in Section 4.

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

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:

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

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 (B.23):

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

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 (B.22):

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

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

The computation of the term $\frac{dz_i}{dH_i}$ in (B.24) is not straightforward as it requires the first-order derivatives of the KKT conditions [44]. However, they can generally be obtained at a small computational cost. Computing the rest of the terms in (B.23) and (B.24) is significantly easier. The quadratic approximation (B.25) can consequently be used in (B.22) when running the SQP algorithm. Note that the primal-dual solution of the bus-level problem is needed each time (B.25) is computed around a new parameter value. Therefore, the NLP (B.17) must be solved again each time a Newton step is taken by solving (B.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 (B.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 sub-problems, or backtracking line search methods [39]. However, we observed that taking these precautions was not necessary to guarantee the

Algorithm 2 SQP procedure for solving the decomposed optimization problem (B.16)-(B.17). C denotes centrally-run computations, while i denotes computations carried out aboard bus i.

i: solve NLPs (B.18) and (B.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 (B.17), then (B.25)
    i: send $\hat{V}_{i,H_i}$ to central node
  end
  C: $H^+ \leftarrow$ solve QP (B.22)
end
C: $H \leftarrow H^+$ and send H to buses
$\forall i$: X_i, $U_i \leftarrow$ solve NLP (B.17)
return H, X_1, U_1, ..., X_n, U_n

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 4, 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, 2016.

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

[3] G. F. Newell and R. B. Potts, "Maintaining a bus schedule," in *Australian Road Research Board (ARRB) Conference, 2nd, Melbourne*, vol. 2, 1964.

[4] C. Daganzo, *Fundamentals of transportation and traffic operations*. Pergamon Oxford, 1997, vol. 30.

[5] A. Bar-Yosef, K. Martens, and I. Benenson, "A model of the vicious cycle of a bus line," *Transportation Research Part B: Methodological*, vol. 54, pp. 37–50, 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," *Applied energy*, vol. 230, pp. 344–356, 2018.

[7] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 71–83, 2007.

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

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

[10] S. Uebel, N. Murgovski, B. Bäker, and J. Sjöberg, "A two-level mpc for energy management including velocity control of hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5494–5505, 2019.

[11] M. Held, O. Flärdh, and J. Mårtensson, "Optimal speed control of a heavy-duty vehicle in urban driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1562–1573, 2018.

[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, 2012.

[13] N. Murgovski, L. M. Johannesson, and B. Egardt, "Optimal battery dimensioning and control of a cvt phev powertrain," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2151–2161, 2013.

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

[15] J. Argote-Cabanero, C. F. Daganzo, and J. W. Lynn, "Dynamic control of complex transit systems," *Transportation Research Part B: Methodological*, vol. 81, pp. 146–160, 2015.

[16] W. Wu, R. Liu, and W. Jin, "Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour," *Transportation Research Part B: Methodological*, vol. 104, pp. 175–197, 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," *Transportation Research Part E: Logistics and Transportation Review*, vol. 130, pp. 61–81, 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: Transport Dynamics*, vol. 7, no. 1, pp. 1062–1095, 2019.

[19] G. Laskaris, M. Seredynski, and F. Viti, "Enhancing bus holding control using cooperative its," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1767–1778, 2020.

[20] C. F. Daganzo and J. Pilachowski, "Reducing bunching with bus-to-bus cooperation," *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 267–277, 2011.

[21] J. J. Bartholdi III and D. D. Eisenstein, "A self-coördinating bus route to resist bus bunching," *Transportation Research Part B: Methodological*, 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," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 757–769, 2010.

[23] F. Delgado, J. C. Muñoz, and R. Giesen, "How much can holding and/or limiting boarding improve transit performance?" *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1202–1217, 2012.

[24] D. Hernández, J. C. Muñoz, R. Giesen, and F. Delgado, "Analysis of real-time control strategies in a corridor with multiple bus services," *Transportation Research Part B: Methodological*, vol. 78, pp. 83–105, 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 Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4680–4692, 2019.

[26] K. Ampountolas and M. Kring, "Mitigating bunching with bus-following models and bus-to-bus cooperation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] B. Varga, T. Tettamanti, and B. Kulcsár, "Optimally combined headway and timetable reliable public transport system," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 1–26, 2018.

[28] I. I. Sirmatel and N. Geroliminis, "Mixed logical dynamical modeling and hybrid model predictive control of public transport operations," *Transportation Research Part B: Methodological*, vol. 114, pp. 325–345, 2018.

[29] B. Varga, T. Tettamanti, and B. Kulcsár, "Energy-aware predictive control for electrified bus networks," *Applied Energy*, vol. 252, p. 113 477, 2019.

[30] R. Lacombe, S. Gros, N. Murgovski, and B. Kulcsár, "Hierarchical control of electric bus lines," in *IFAC World Congress, 21st, 2020, Berlin*, 2020.

[31] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of process control*, vol. 19, no. 5, pp. 723–731, 2009.

[32] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[33] K. Gkiotsalitis and E. Van Berkum, "An exact method for the bus dispatching problem in rolling horizons," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 143–165, 2020.

[34] W. Fan and R. B. Machemehl, "Do transit users just wait for buses or wait with strategies? some numerical results that transit planners should see," *Transportation Research Record*, vol. 2111, no. 1, pp. 169–176, 2009.

[35] L. Guzzella and A. Sciarretta, *Vehicle propulsion systems*. Springer-Verlag, 2013, vol. 3.

[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," *Journal of Advanced Transportation*, vol. 49, no. 3, pp. 358–370, 2015.

[37] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. 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 *American Control Conference (ACC)*, IEEE, 2015, pp. 763–768.

[39] J. Nocedal and S. Wright, *Numerical optimization*. Springer, 2006.

[40] G. Still, "Lectures on parametric optimization: An introduction," *Optimization Online*, 2018.

[41] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 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," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 197–211, 2018.

[43] M. Vukov, S. Gros, G. Horn, G. Frison, K. Geebelen, J. B. Jørgensen, J. Swevers, and M. Diehl, "Real-time nonlinear mpc and mhe for a large-scale mechatronic application," *Control Engineering Practice*, vol. 45, pp. 64–78, 2015.

[44] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 2567–2573.