THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Distributed Optimization for the Optimal Control of Electric Vehicle Fleets

RÉMI LACOMBE

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2023

Distributed Optimization for the Optimal Control of Electric Vehicle Fleets

Rémi Lacombe
ISBN 978-91-7905-965-1

Copyright © 2023 Rémi Lacombe
All rights reserved.

Doktorsavhandlingar vid Chalmers tekniska högskola.
Ny series nr 5431
ISSN 0346-718X
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, December 2023

À mes parents.

"Hobbits … liked to have books filled with things that they already knew, set out fair and square with no contradictions."

J.R.R. Tolkien
The Lord of the Rings: The Fellowship of the Ring
Prologue

Abstract

Owing to their absence of tailpipe emissions and their independence from fossil fuels, Electric Vehicles (EVs) are currently experiencing a rapid deployment in an attempt to curb global greenhouse gas emissions. EV operation represents a technical challenge, however, as new control algorithms need to be developed to address their limited driving range and their longer charging times. Optimization-based control techniques offer a promising way to plan EV operation over a prediction horizon while including key operational constraints, but they can be prohibitively slow for real-time applications as they rely on solving computationally hard optimization problems. One way to address the computational complexity of these approaches is by deploying adapted decomposition methods with which the computational load of solving these optimization problems can be distributed across the vehicles involved, where most computations can then be carried out in parallel.

This thesis presents decomposition-based solution procedures for optimal control problems involving groups of EVs. In particular, the problems covered in this work are (i) the cooperative eco-driving control of a platoon of electric trucks, (ii) the eco-driving and operational control of an electric bus line, and (iii) the operational control and charging scheduling of an electric bus network. Even though their particular objective functions and constraints may differ, the coupling structures of these problems, i.e. how each vehicle's influence on the others is organized, share some similarities.

The platoon control problem is formulated as a Nonlinear Program (NLP) and solved with second-order optimization methods. The Riccati recursion is used as part of a decomposition scheme that exploits the chain-like coupling structure of a truck platoon and makes it possible to fully distribute all computations. Similarly, the bus line problem is formulated as an NLP. A primal decomposition scheme where the NLP is split into a master problem and independent bus subproblems is presented. The hierarchical control architecture obtained makes it possible to distribute most of the computations. Finally, the bus network problem is formulated as a Mixed-integer Linear Program (MILP). A dual decomposition scheme based on Lagrangian relaxation is deployed to relax the coupling constraints between the different bus lines.

Keywords: Electric Vehicles, Optimal Control, Distributed Optimization, Public Transit, Platooning.

List of Publications

This thesis is based on the following publications:

[A] **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár, "Distributed Eco-driving Control of a Platoon of Electric Vehicles Through Riccati Recursion". *IEEE Transactions on Intelligent Transportation Systems*, March 2023.

[B] **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár, "Bilevel Optimization for Bunching Mitigation and Eco-Driving of Electric Bus Lines". *IEEE Transactions on Intelligent Transportation Systems*, August 2022.

[C] **Rémi Lacombe**, Nikolce Murgovski, Sébastien Gros, and Balázs Kulcsár, "Integrated Charging Scheduling and Operational Control for an Electric Bus Network". Submitted to *Elsevier Transportation Research*.

Other publications by the author, not included in this thesis, are:

[D] **Rémi Lacombe**, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár, "Hierarchical Control of Electric Bus Lines". Published in *21st IFAC World Congress*, Berlin, July 2020.

[E] **Rémi Lacombe**, Nikolce Murgovski, Sébastien Gros, and Balázs Kulcsár, "Conflict-free Charging and Real-time Control for an Electric Bus Network". Published in *22nd IFAC World Congress*, Yokohama, July 2023.

Acknowledgments

First and foremost, I want to thank my main supervisor Professor Balázs Kulcsár. I am very grateful for your constant support and for providing me with valuable feedback at every step of the way. You always managed to find the time for me whenever I had questions or needed guidance despite your busy schedule. I think it is safe to say that this thesis would not have existed without you.

I also want to extend my deep gratitude to my co-supervisors Professor Nikolce Murgovski and Professor Sébastien Gros. I have learned so much from you during the countless technical discussions that we had over the years and I have been constantly impressed by the depth of your theoretical knowledge. Thank you for kindling my interest for optimal control and for showing me what it really means to be an expert in a field.

Next, Professor Bo Egardt and Professor Nikolce Murgovski deserve special mention as I have been lucky enough to serve as a teaching assistant in their courses at Chalmers. It has truly been a pleasure and an honor to work alongside such capable and talented teachers.

I am also grateful to Energimyndigheten for having financially supported this thesis through the OPNET project.

In addition, I wish to thank all my past and present colleagues at SYSCON, some of whom I now have the privilege of calling my friends. In order of appearance, thank you to Yixiao, Simon, Elena, Fredrik, Ivo, Ankit, Masoud, Albin, Angel, Ramin, Constantin, Mattias, Endre, Sabino, Rita, Ahad, Stefan, Lars, Maxime, Anand, Huang, Yao, Yizhou, Ze, Sten Elling, Maxi, Ektor, Hasith, Rikard, Carl-Johan, Sondre, Faris, Albert, Godwin, Gabriel, Ahmet, Alvin, Nishant, Johannes, Dan, Erik, Attila, Benedick, Lorenzo, and anyone else I might have accidentally forgotten, for making all these years at Chalmers so rich and full of colors. Thank you for all the late night afterworks, the countless climbing sessions, the various outings in and out of the city, and for being awesome human beings in general! I'm sorry that I always had to leave early to catch the last bus back to my forest. You are the ones who truly shaped my experience as a PhD student.

Last but not least, I want to thank my family for their unwavering love and support. Thank you to my parents, to my grandparents and to my brothers Antoine and Arthur for always believing in me. Même si j'ai souvent été loin ces dernières années, je n'ai jamais vraiment quitté Biot ni Rodez.

And finally, my biggest thanks go to Adeline. Thank you for being by my side all these years. Thank you for sharing my adventure in this great country in the North. Thank you for wanting to live a life of quiet tranquility with me among the birds and the trees. Thank you for being a continuous source of inspiration for me. Je t'aime Adeline.

Acronyms

BnB:	Branch and Bound
BVP:	Boundary Value Problem
EV:	Electric Vehicle
KKT:	Karush-Kuhn-Tucker
LP:	Linear Program
LQ:	Linear-quadratic
LICQ:	Linear Independence Constraint Qualification
MILP:	Mixed-integer Linear Program
MPC:	Model Predictive Control
NLP:	Nonlinear Program
OCP:	Optimal Control Problem
PDIP:	Primal-dual Interior Point
QP:	Quadratic Program
SOSC:	Second Order Sufficient Conditions
SQP:	Sequential Quadratic Programming

Contents

Part I

Overview

CHAPTER 1

Background and Outline

1.1 Introduction

The decarbonization of the transport sector constitutes a major challenge for our societies due to the strong dependency of the sector on fossil fuels. Contrary to other carbon-intensive activities, like the industry or the energy sector, greenhouse gas emissions from the transport sector have continued to increase steadily in both OECD and non-OECD countries, where they represented 30% of all CO_2 emissions in the former and 16% in the latter in 2016 [1]. This trend may very well continue in the future as the total transport demand is projected to keep increasing fast in the coming decades since both passenger transport and global freight demand are expected to triple between 2015 and 2050 [1]. In this context, Electric Vehicles (EVs) offer a promising way to break the oil dependency of the transport sector and curb its greenhouse gas emissions. EVs combine no tailpipe emissions with a lower carbon intensity than fossil fuel vehicles and are now being massively deployed on the roads of the world, with record sales in the last few years despite the coronavirus pandemic [2].

In terms of absolute numbers, the deployment of large EVs such as electric buses and electric trucks is lagging behind that of lighter vehicles. In 2020, the global fleet of electric buses was estimated to number around 600 000 vehicles and about 30 000 for electric trucks, whereas more than 6 million electric cars were estimated to be on the roads in the same year [2]. Electric bus fleets are now being rapidly adopted by municipalities around the world as measures to reduce both air and noise pollution in congested modern cities. Several of the world's largest cities have already pledged to reach specific electrification targets in the current decade [3]. However, both electric buses and electric trucks have to face technical challenges that do not exist for their conventional counterparts. Electric trucks are still struggling with their limited driving range for long-distance driving missions of several hundred kilometers. A limited driving range can also be an issue for electric buses, which may struggle to complete a full day of service on a single battery charge. The questions of when and where to charge these large service vehicles, and of the type and placement of the charging infrastructure, must also be addressed carefully. In light of these new technical challenges, innovative control algorithms are now needed to carry out the planning and operation of systems of electric trucks and buses.

One direct way of increasing the driving range of EVs in operation is by deploying control methods with a focus on energy efficiency. One such method that has become especially popular for heavy-duty vehicles is *platooning*, a method which consists of having vehicles drive in close succession in order to decrease the drag force acting on them and thus reduce their energy consumption. Platooning has been shown to lead to sizeable energy savings in field experiments carried out on truck platoons [4]–[6]. In addition, this method has the benefit of not needing any structural change in order to be implemented, but guaranteeing safety when trucks are operated at a close distance requires robust control algorithms. The platoon control problem has mostly been treated from the angle of string stability, i.e. the attenuation of deviations on positions and speed along the platoon, historically [7]–[9]. Driving proximity can then be used as a satisfactory proxy for energy savings: the closer trucks are driving to each other, the more the drag force is reduced. However, tracking this simple control objective is not always energy-optimal for truck platoons [10]. For instance, trucks driving as a platoon in a sharp downhill may have to use their friction brakes to avoid collisions and thereby waste energy.

Additional energy savings can be achieved for truck platoons, and for any EV in general, by adopting *eco-driving* strategies [11]. This term denotes the general idea of adapting vehicles' driving profiles to the road conditions with the goal of minimizing energy consumption. As a control strategy, eco-driving relies on predictive information about road gradients and expected traffic conditions, among other things, in order to estimate the energy optimal trajectory over a control horizon. Eco-driving control has been applied for driving missions of single trucks [12], [13] as well as truck platoons [14], [15], and has shown an encouraging potential for energy savings for both. It must be noted, however, that most of the works on the truck platoon control problem available in the literature have focused on diesel trucks so far, with hybrid electric and electric trucks only recently starting to attract attention [16].

Similarly, the specific challenges arising in the planning and operation of a network of electric buses have only recently gained widespread interest in the literature [17]. The electric bus planning process is usually decomposed into a *strategic*, a *tactical*, and an *operational* stage, depending on the time frame of the problem considered. The electrification of a bus network mostly introduces new planning problems at the strategic and tactical stages [18]. At the strategic stage, long-term decisions are taken on bus fleet and charging infrastructure investments [3], as well as on charging infrastructure placement [19]–[21]. At the tactical stage, the electric vehicle scheduling problem assigns individual electric buses to specific timetabled trips [22], [23], while decisions on when and for how long each bus should charge its battery can be obtained from the charging scheduling problem [21], [24]–[26].

At the operational stage, real-time decisions are taken to control the vehicles when they are in operation, usually with the aim of maximizing timetable adherence or minimizing passengers' waiting time. If left uncontrolled, bus lines are prone to instability as any delay gets amplified by the additional dwell time incurred by the increased passenger loads, thus acting as a positive feedback loop that further increases the delay [27]. Likewise, any bus traveling behind a delayed bus will encounter fewer passengers at stops and may eventually catch up with the delayed bus, thus resulting in *bus bunching*. To counteract the formation of bus bunching, several different intervention strategies have been studied in the literature. The simplest of these is bus holding, a method that consists of holding buses at predesignated control points to dissipate service irregularities [28]–[32]. Other station control strategies include stop-skipping

[33], [34] and limiting the number of passengers that are allowed to board each bus [35]. Inter-station control strategies have been studied less systematically by the scientific community, in part because they are usually harder to deploy in practice. Examples of inter-station control strategies include transit signal priority, where the phases of traffic lights at intersections are adjusted to favor bus traffic [36], or bus substitution, where some buses are kept in reserve by the transit agency and dispatched to replace buses who fall behind schedule too much [37]. Finally, a control strategy of particular importance in the context of this thesis is speed control, where the speed of each vehicle is adjusted in real-time to offset any potential delay in bus service [38], [39].

Most of the control problems discussed so far can be seen conceptually as aiming to minimize a certain cost (monetary or otherwise) over a given time horizon for a system evolving with dynamics that can be modeled mathematically and with operating conditions that can be defined as a set of constraints. As such, these problems can be considered as *optimal control problems* [40]. In fact, the eco-driving control problem mentioned earlier can be directly regarded as an optimal control problem [11].

A framework that has been steadily rising in popularity in the last few decades for treating optimal control problems is Model Predictive Control (MPC) [41]. MPC is an optimization-based approach where a discrete time approximation of the original optimal control problem is solved only over a (usually small) part of the time horizon of the original problem. Once one optimization problem has been solved, the optimal control action obtained for the current time step is applied and a new optimization problem is formed by moving the MPC horizon one discrete time step forward, thus earning MPC the alternative name of “receding horizon control”. The sequence of control actions obtained is then an approximation of the optimal control trajectory of the original problem. MPC has been successfully applied to the operational control of a bus line as part of a speed control strategy [42]–[44]. Similarly, the eco-driving control for a truck platoon has mostly been treated in the MPC framework in the literature so far [14]–[16].

However, the downside of optimization-based control methods is their reliance on being able to solve potentially large optimization problems. Depending on the application, the optimization problems considered can involve the control of many vehicles on possibly long horizons, and even the enormous increase in the available computing power witnessed in the past decades may still not be enough to solve these large-scale problems in a reasonable time. Thankfully, distributed optimization methods can, in certain cases, provide significant computational improvements by making use of parallel computing hardware [45]. By distributing the computations required to solve a given optimization problem, some of the computational load can be parallelized, which can potentially result in significant computational speedups. This might not always be possible though, and it usually depends on the specific structure of the optimization problem considered as well as on the type of optimization algorithm used to solve it.

It is often necessary to first decompose an optimization problem before being able to distribute the computations required to solve it. The core idea of decomposition methods in optimization is to identify smaller *subproblems* in a larger optimization problem. An intuitive example of this is when an optimization problem is formulated for a group of agents (e.g. vehicles) or subsystems, in which case it is frequent to form subproblems corresponding to the individual agents or subsystems. Subproblems usually have some degree of *coupling*, i.e. they share some of the problem variables or constraints. When they exist, these coupling variables or constraints make it impossible to directly separate an optimization problem into independent subproblems. A decomposition method then normally proceeds by extracting the subproblems from the original problem and by finding its solution (or an approximation of it) through an iterative procedure that involves solving the subproblems and coordinating their solutions. Decomposition methods are usually classified into primal or dual methods, depending on whether they aim to solve the primal optimization problem or the associated dual problem [46]. The exact decomposition method that is best adapted to solve a given problem depends on what type of problem it is, and also on the specific coupling structure that this problem has [47].

1.2 Research gaps

A few studies have focused on eco-driving for truck platoons, but all the control methods that have been proposed in the literature are centralized or partially centralized approaches. As such, they rely on a central control unit to communicate reference trajectories to vehicles. Using such methods might prove difficult for truck platoons where vehicles belong to different transportation companies, where the need to have a de facto “leader” in the platoon who bears the responsibility of computing the final commands for everybody else might be unacceptable. However, a fully distributed approach for solving the eco-driving control problem of a truck platoon has so far been lacking in the literature. In addition, no work has studied eco-driving for platoons of fully electric trucks yet.

Most fully distributed solution algorithms for optimization problems arising in control applications tend to rely on first-order optimization methods, that is iterative solution methods that only use gradient information. By contrast, second-order optimization methods are similar optimization methods that use both gradient and Hessian information. Consequently, second-order methods tend to have greater convergence speeds (up to quadratic convergence rates when taking full Newton steps) and thus require fewer iterations to reach an optimal solution. However, fully distributed algorithms for second-order methods have received far less attention in the literature, despite their stronger convergence properties. Part of the reason is that second-order methods are usually hard to distribute and require specific problem structures with sparse coupling in order to do so. In this thesis, a problem with such a structure is presented together with a fully distributed second-order method to solve it.

Speed control strategies have not been studied as extensively as other control strategies in the operational bus line control literature. The few works that combine speed control strategies with optimal control tend to use very simplistic driving dynamics and energy consumption models. The main reason for this is that detailed models make for complicated optimization problems that can be hard to solve in almost real-time, which is a point that this thesis tries to address.

Furthermore, very few works in the literature have explored the operational control of electric buses. Most of the work carried out so far for the electric bus network planning process has been at the strategic and tactical stages, to generate fixed charging schedules for buses for example. However, these schedules might not be robust to operational disturbances, and robust charging scheduling methods have so far been lacking in the literature. In addition, no work at the operational stage has ever tried including eco-driving control strategies for bus line control.

In light of these research gaps, this thesis explores the following research questions:

- **RQ1:** How can the operational control of platoons of electric trucks and electric bus lines be formulated as optimization problems?
- **RQ2:** How can the specific coupling structure of a truck platoon, a bus line, or a bus network be exploited with decomposition schemes to distribute most of the computations when solving the optimization problems formulated in **RQ1**?
- **RQ3:** How can the characteristic chain-like structure of a truck platoon be leveraged to solve a shared optimal control problem with second-order optimization methods in a fully distributed and privacy-preserving fashion?

1.3 Contributions

This thesis presents several control problems involving the operation of groups of electric vehicles, namely platoons of electric trucks and networks of electric buses. Each problem is formulated as an optimization problem in which the coupling structure between the vehicles is clearly exhibited. Depending on the exact nature of each problem, vehicles may share a common objective, and they may have similar constraints. What all these problems have in common, though, is that the coupling terms are relatively few compared with the rest of the terms, which have to do with the dynamics and decisions of individual vehicles for the most part. For all the problems studied, it is therefore possible to decouple individual vehicle subproblems, where most of the complexity is, and solve them separately. This thesis shows how specific decomposition schemes can be deployed on the particular coupling structure of each of the problems treated. These decomposition schemes make it possible to distribute most of the computations required when solving these problems with traditional optimization algorithms.

The focus of this thesis is on finding suitable decomposition schemes for optimization problems with a specific coupling structure. The main research limitation of the thesis is therefore that all the algorithms and decomposition methods presented are meant to be seen as proofs of concept rather than full-fledged implementations ready to be deployed in practical situations. In particular, an efficient real-time capable implementation of these algorithms is left outside the scope of this thesis. In addition, all the optimal control prob-

Table 1.1: Overview of the papers.

	Paper A	**Paper B**	**Paper C**
System	Truck platoon	Bus line	Bus network
Application	Eco-driving	Eco-driving + service regularity	Charging + service regularity
Optimization problem	Nonlinear program	Nonlinear program	Mixed-integer linear program
Coupling	Constraints	Objective	Constraints
Decomposition method	Riccati recursion	Primal decomposition	Lagrangian decomposition
Control architecture	Distributed	Hierarchical	Hierarchical
Exact solution	Yes	Yes	No

lems presented are formulated over relatively long control horizons and with economic objectives. The optimal solutions obtained should then be seen as high-level reference trajectories, and should therefore be complemented with low-level tracking control layers in a practical implementation. The design of these low-level control layers is also left outside the scope of this thesis.

The main contributions of this thesis are:

- A fully distributed second-order optimization procedure is presented to solve the cooperative eco-driving control problem for a platoon of vehicles organized in a chain-like structure (**RQ3**).

- An optimal control problem with detailed dynamics and energy consumption models is formulated for the eco-driving and operational control of an electric bus line (**RQ1**).

- A primal decomposition scheme is proposed to decouple individual bus problems and solve them in parallel when solving the bus line control problem (**RQ2**).

- An optimization problem integrating both the tactical level charging scheduling problem and the operational control problem for an electric bus network is formulated (**RQ1**).

- A dual decomposition scheme is developed for the bus network control problem that relaxes the coupling between bus lines at the shared charging infrastructure and makes it possible to solve each individual bus line problem in parallel (**RQ2**).

Since the scope of the problems, algorithms, and decomposition methods treated in the thesis is rather wide, Table 1.1 presents a succinct overview of the main features of each of the appended papers.

1.4 Outline

This thesis is divided into two parts. Part I contains seven chapters that offer an introduction to and an overview of the research papers appended in Part II.

Part I is structured as follows. Chapter 2 introduces fundamental concepts in optimal control and numerical optimization and lays the theoretical foundation for understanding the subsequent chapters. The cooperative eco-driving platoon control problem is briefly introduced in Chapter 3. There, it is shown how solving this problem can be done in a distributed fashion by using the Riccati recursion. Chapter 4 then formulates the eco-driving and operational control problem for a line of electric buses and proposes a primal decomposition scheme aiming at decoupling individual bus subproblems. The focus of Chapter 5 is on the operational control and charging scheduling problem for an electric bus network. It is detailed there how a dual decomposition scheme based on Lagrangian relaxation can be deployed on this problem. A succinct summary for each of the publications included in Part II is then offered in Chapter 6. Finally, Chapter 7 presents some concluding remarks as well as suggestions for future research directions.

CHAPTER 2

Preliminaries

This chapter provides some mathematical background useful for understanding this thesis. In particular, a brief introduction to optimal control and numerical optimization is given. Popular algorithms for smooth optimization like Sequential Quadratic Programming (SQP) algorithms and Primal-dual Interior Point (PDIP) algorithms are briefly described. This presentation draws upon [40], [48], [49], among other works.

2.1 Optimal control

Optimal control combines the control of *dynamical systems*, i.e. processes described by states and evolving over time according to some deterministic or stochastic rule, with optimization. Solving an Optimal Control Problem (OCP) amounts to finding the control trajectory that minimizes a certain cost functional subject to the evolution rule of the dynamical system and to some additional path constraints. In this thesis, we consider deterministic dynamical systems whose evolution rule can be formulated as *ordinary differential equations*.

An OCP between an initial time t_0 and a final time t_f has the general form:

$$
\begin{aligned}
\min_{x(.),u(.)} \quad & V(x(t_f), t_f) + \int_{t_0}^{t_f} l(x(t), u(t), t)\mathrm{d}t & \text{(2.1a)} \\
\text{s.t.} \quad & x(t_0) = \bar{x}_0, & \text{(2.1b)} \\
& \forall t \in [t_0, t_f] : \\
& \dot{x}(t) = f(x(t), u(t), t), & \text{(2.1c)} \\
& g(x(t), u(t), t) \leq 0, & \text{(2.1d)}
\end{aligned}
$$

where x and u are the state and control input trajectories, respectively, V is the terminal cost, l is the stage cost, $\bar{x}_0$ is the initial state, f represents the system dynamics, and g represents the path constraints. In this version, no endpoint conditions are considered.

There are two main ways of solving OCPs of the form (2.1): direct and indirect approaches.

Indirect approaches

These approaches have been popular to solve OCPs historically, before the advent of modern computers. They stem from *Pontryagin's maximum principle*, which establishes necessary conditions for the optimal control input trajectory and the associated state trajectory [40]. These conditions can be expressed to form a two-point Boundary Value Problem (BVP). However, the BVP can only be solved analytically for simple systems and must be solved numerically in most practical applications, especially when considering path constraints. This is why indirect methods are sometimes referred to as the "first optimize, then discretize" approach to optimal control: some optimality conditions are first derived, and then the resulting BVP is discretized in order to be solved numerically. Indirect approaches have nowadays mostly been supplanted by direct approaches for large-scale OCPs or OCPs with many path constraints.

Direct approaches

Direct methods take the opposite approach than indirect methods, which has earned them the name of the "first discretize, then optimize" approach to optimal control. They proceed by first forming an optimization problem whose solution is an approximation of the solution of the original OCP, and then by solving it. For continuous-time OCPs of the form (2.1), feasible sets for solution candidates consist of infinite-dimensional function spaces. By contrast, feasible sets in regular optimization problems only consist of finite-dimensional vector spaces, and effective numerical algorithms are today available to solve such problems. This will be discussed later in this chapter.

Since only direct approaches have been used in this thesis, we now detail how the discretization step can be carried out. A parametrized approximation of the control input trajectory is usually searched for in the discretized problem. A standard input parametrization choice is that of a piecewise constant input on a uniform time grid, $t_0, t_1, ..., t_{N_k}$, where N_k is the number of grid intervals and where $t_{N_k} = t_f$. In this case, $u(t) = u_k$, $\forall t \in [t_k, t_{k+1}[$, with $k \in [\![0, N_k - 1]\!]$. The control input trajectory would then be approximated by the vector of control inputs $u = [u_0, ..., u_{N_k-1}]^\top$.

Similarly, the state trajectory can be approximated through *multiple shooting* by the state vector $x = [x_0, ..., x_{N_k}]^\top$, where each x_k, $k \in [\![0, N_k]\!]$, denotes the state value at the k-th grid point. Numerical integration is necessary to express the stage cost (2.1a) and the dynamics (2.1c) in the discretized problem. Numerical integrators can be either explicit or implicit, and there are many different methods available, the most prominent of which are Euler methods, Runge-Kutta methods, and collocation methods [48]. Regardless of which numerical integrator is chosen, a discretized version of OCP (2.1) is:

$$
\begin{aligned}
\min_{x,u} \quad & V(x_{N_k}) + \sum_{k=0}^{N_k-1} L_k(x_k, u_k) && \text{(2.2a)}\\
\text{s.t.} \quad & x_0 = \bar{x}_0, && \text{(2.2b)}\\
& \forall k \in [\![0, N_k - 1]\!] : \\
& x_{k+1} = F_k(x_k, u_k), && \text{(2.2c)}\\
& g_k(x_k, u_k) \leq 0, && \text{(2.2d)}
\end{aligned}
$$

where L_k and F_k carry out the numerical integration on the interval $[t_k, t_{k+1}[$ of the stage cost l and system dynamics f, respectively, and where g_k enforces the path constraints g at the k-th grid point.

The optimization problem (2.2) obtained has a nonlinear objective function and nonlinear constraints in the general case, which makes it a Nonlinear Program (NLP). The next section shows how optimization problems of this type can be solved numerically.

2.2 Numerical optimization

Let us consider a general NLP:

$$\min_{x} \quad f(x) \tag{2.3a}$$
$$\text{s.t.} \quad g_i(x) = 0, \qquad i = 1, ..., m, \tag{2.3b}$$
$$h_i(x) \leq 0, \qquad i = 1, ..., p, \tag{2.3c}$$

where $x \in \mathbb{R}^n$ is now the vector of decision variables, $f : \mathbb{R}^n \to \mathbb{R}$ is the cost function, $g = [g_1, ..., g_m]^\top$ are the equality constraints, with $g : \mathbb{R}^n \to \mathbb{R}^m$, and $h = [h_1, ..., h_p]^\top$ are the inequality constraints, with $h : \mathbb{R}^n \to \mathbb{R}^p$.

A smooth version of NLP (2.3) is considered here, where the functions f, g, and h are twice continuously differentiable at least. These functions are nonlinear in the general case, but there exist a few cases of interest which deserve special mention. If f and h are convex functions and g is an affine function, then (2.3) is a convex NLP. Similarly, if f is a quadratic function and g and h are both affine functions, (2.3) is a Quadratic Program (QP). In case f is affine as well, (2.3) reduces to a Linear Program (LP).

Optimality conditions

A few basic properties on the optimality of NLP (2.3) are now reviewed.

Definition 2.1 (Feasibility) A point $x \in \mathbb{R}^n$ is said to be *feasible* in (2.3) if $x \in \Omega$, where Ω is the *feasible set* of the problem and is defined as $\Omega = \{x \in \mathbb{R}^n |\, g(x) = 0,\ h(x) \leq 0\}$.

Definition 2.2 (Global minimum) A point $x^* \in \Omega$ is said to be a *global minimizer* of the minimization problem (2.3) if

$$f(x^*) \leq f(x), \quad \forall x \in \Omega. \tag{2.4}$$

In this case, $f(x^*)$ is said to be the *global minimum* of the problem.

Definition 2.3 (Local minimum) A point $x^* \in \Omega$ is said to be a *local minimizer* of the minimization problem (2.3) if there exists a neighborhood $\mathcal{N}$ of x^* (e.g. an open ball) such that:

$$f(x^*) \leq f(x), \quad \forall x \in \Omega \cap \mathcal{N}. \tag{2.5}$$

Definition 2.4 (Active constraints) An inequality constraint h_i is said to be *active* at a point $x \in \Omega$ if $h_i(x) = 0$, and *inactive* otherwise. The *active set* $\mathcal{A}(x) = \{i \in [\![1, p]\!] \,|\, h_i(x) = 0\}$ is then defined as the index set of the inequality constraints that are active at x.

Before we assemble the first-order optimality conditions of NLP (2.3), the concept of *regularity* must be briefly discussed. Feasible points are said to be *regular* if they verify some *constraint qualification* conditions. Here, we only present the one most commonly used in practice: Linear Independence Constraint Qualification (LICQ).

Definition 2.5 (LICQ) The LICQ condition is said to hold at a given local minimizer x^* of problem (2.3) if the gradients of the equality constraints $\{\nabla_x g_i\}_{i \in [\![1,m]\!]}$ and active inequality constraints $\{\nabla_x h_i\}_{i \in \mathcal{A}(x^*)}$ are linearly independent at x^*.

Definition 2.6 (Lagrangian function) The *Lagrangian function* of problem (2.3) is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x), \tag{2.6}$$

where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ are the *Lagrange multipliers* associated with the equality and inequality constraints, respectively.

We are now ready to state the first-order optimality conditions of NLP (2.3), also known as the Karush-Kuhn-Tucker (KKT) conditions.

Theorem 2.1 (KKT conditions) *If x^* is a regular local minimizer of* (2.3), *then* $\exists \lambda^*, \mu^*$ *such that:*

$$
\begin{aligned}
&\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0, && && \text{(2.7a)}\\
&g_i(x^*) = 0, && i = 1, ..., m, && \text{(2.7b)}\\
&h_i(x^*) \leq 0, && i = 1, ..., p, && \text{(2.7c)}\\
&\mu_i^* h_i(x^*) = 0, && i = 1, ..., p, && \text{(2.7d)}\\
&\mu_i^* \geq 0 && i = 1, ..., p. && \text{(2.7e)}
\end{aligned}
$$

In this case, x^* *is said to be a KKT point.*

In the KKT conditions, (2.7b) and (2.7c) are the *primal feasibility* conditions, simply ensuring that KKT points are part of the feasible set, and (2.7e) are the *dual feasibility* conditions. Next, condition (2.7a) is usually called the *stationarity* condition, and (2.7d) the *complementary slackness.*

Note that the KKT conditions are only necessary conditions for optimality: not all KKT points are local minimizers. Nonetheless, KKT conditions play an important role in continuous optimization as they form the basis on which many numerical solvers are built, as will be seen shortly. For now, we make the observation that convex problems constitute an interesting class of NLPs. Indeed, for convex problems, the necessary conditions for optimality become sufficient conditions, and local optimality leads to global optimality, as stated in the following theorem.

Theorem 2.2 (Optimality of convex problems) *If problem* (2.3) *is convex, then any KKT point is also a global minimizer.*

For non-convex problems, it is also of interest to have access to a set of sufficient conditions for optimality. The second-order optimality conditions of NLP (2.3), also known as the Second Order Sufficient Conditions (SOSC), give us just that.

Definition 2.7 (Set of linearized feasible directions) The *set of linearized feasible directions* at a feasible point $x \in \Omega$ is defined as:

$$\mathcal{F}(x) = \{d \in \mathbb{R}^n | \ \nabla g(x)^\top d = 0, \ \nabla h_i(x)^\top d \leq 0, \ \forall i \in \mathcal{A}(x)\}. \tag{2.8}$$

Theorem 2.3 (SOSC) *Let $\{x^*, \lambda^*, \mu^*\}$ satisfy the KKT conditions. If*

$$d^\top \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*, \mu^*) d > 0, \quad \forall d \in \mathcal{C}(x^*, \mu^*) \backslash \{0\}, \tag{2.9}$$

where $\mathcal{C}(x^*, \mu^*) = \{d \in \mathcal{F}(x^*) | \, \nabla h_i(x^*)^\top d = 0, \, \forall i \in \mathcal{A}(x^*) \text{ s.t. } \mu_i^* > 0\}$ is the critical cone, then the SOSC hold and x^* is a strict local minimizer of (2.3).

The SOSC use curvature information to guarantee that no feasible direction starting from a local minimizer x^* can generate any improvement in the cost function. The proofs of Theorems 2.1-2.3 can be found in [49].

We are now ready to present two of the most widely used methods for solving NLPs: the sequential quadratic programming method and the primal-dual interior point method. Both the SQP and PDIP methods fall into the category of *second-order optimization methods*, which get their name from the fact that they use information from both first-order and second-order derivatives of the objective function and constraints to operate. At their core lies the celebrated *Newton's method*.

Newton's method

Newton's method is a root-finding algorithm that iteratively refines a solution candidate by taking steps along the direction of the gradient of the function at the current candidate. To illustrate how Newton's method works in practice, consider the root-finding problem:

$$r(z) = 0, \tag{2.10}$$

where $z \in \mathbb{R}^n$ and $r : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function. Newton's method solves this problem by constructing a sequence of iterates, starting from an initial guess $z^{[0]}$. Each subsequent iterate is computed as:

$$z^{[j+1]} = z^{[j]} + \alpha^{[j]} \Delta z^{[j]}, \tag{2.11}$$

where $\alpha^{[j]}$ is the *step size* and $\Delta z^{[j]}$ is the *Newton direction*. The Newton direction is calculated as the solution of:

$$\nabla_z r\big(z^{[j]}\big) \Delta z^{[j]} + r(z^{[j]}) = 0, \tag{2.12}$$

where the Newton direction is well-defined only if the gradient $\nabla_z r(z^{[j]})$ is nonsingular.

The main strength of Newton's method is its very fast convergence speed to the roots of a function under good conditions. The following theorem states a strong convergence speed result when applying Newton's method to solve the system of KKT conditions of an equality-constrained NLP.

Theorem 2.4 (Convergence speed of Newton's method) *If LICQ and SOSC hold at the optimal solution and if Newton's method is initiated sufficiently close to the optimal solution, then the iterates generated by Newton's method converge quadratically to the optimal solution.*

The proof can be found in [49]. Here it must be pointed out how fast a quadratic convergence speed really is: the number of correct digits of the iterates roughly *doubles* at each iteration. Even when some of the assumptions of Theorem 2.4 do not hold, it is possible to show that Newton's method still benefits from a superlinear convergence speed. Both the SQP and PDIP methods apply Newton's method to solve NLP (2.3) and as such benefit from its powerful convergence properties.

Most of the numerical algorithms for solving NLPs are essentially "KKT solvers" in that they usually operate by generating a sequence of iterates converging to a KKT point. In the case of the SQP and PDIP methods, Newton's method is deployed to find solutions to the system of equations formed by the KKT conditions (2.7). However, the presence of inequality constraints complicates the root-finding procedure presented earlier since (2.7c) and (2.7e) are inequalities, and since the complementary slackness conditions (2.7d) are non-smooth. Next, we give a detailed presentation of each method and show how they handle these issues.

Sequential quadratic programming

The SQP method operates by iteratively generating a sequence of primal-dual solution candidates for NLP (2.3). The Newton directions are computed by solving local QP approximations of NLP (2.3) around each of the solution candidates. The Newton step is then selected as the best descent direction based on the local curvature information.

More specifically, the Newton direction at each SQP iteration j is calculated by solving the inequality-constrained QP:

$$\min_{\Delta x} \quad \frac{1}{2}\Delta x^\top \nabla^2_{xx}\mathcal{L}\big(x^{[j]}, \lambda^{[j]}, \mu^{[j]}\big)\Delta x + \nabla f(x^{[j]})^\top \Delta x \tag{2.13a}$$

$$\text{s.t.} \quad \nabla g_i(x^{[j]})^\top \Delta x + g_i(x^{[j]}) = 0, \qquad i = 1, ..., m, \tag{2.13b}$$

$$\nabla h_i(x^{[j]})^\top \Delta x + h_i(x^{[j]}) \leq 0, \qquad i = 1, ..., p, \tag{2.13c}$$

which involves gradients and Hessians of the functions of NLP (2.3) at the current solution candidate $\{x^{[j]}, \lambda^{[j]}, \mu^{[j]}\}$. Note that the exact Hessian $\nabla^2_{xx}\mathcal{L}$ is only used in (2.13a) when LICQ and SOSC hold at the current iterate. Otherwise, various Hessian approximations are available to choose from.

Let $\Delta x^{[j]}$ be the primal solution of (2.13) and $\lambda^{[j]}_{\mathrm{QP}}$ and $\mu^{[j]}_{\mathrm{QP}}$ be the dual solutions. The SQP method then takes a Newton step on the current primal-dual solution candidate as:

$$x^{[j+1]} = x^{[j]} + \alpha^{[j]}\Delta x^{[j]}, \tag{2.14a}$$

$$\lambda^{[j+1]} = \alpha^{[j]}\lambda^{[j]}_{\mathrm{QP}} + (1 - \alpha^{[j]})\lambda^{[j]}, \tag{2.14b}$$

$$\mu^{[j+1]} = \alpha^{[j]}\mu^{[j]}_{\mathrm{QP}} + (1 - \alpha^{[j]})\mu^{[j]}, \tag{2.14c}$$

where $\alpha^{[j]}$ is the step size. Various strategies can be used to design the step size in order to ensure progress when taking the Newton step. A *backtracking* procedure can be deployed to adjust the step size in order to guarantee that the next solution candidate is better than the previous one. The SQP method then continues iterating until a solution candidate satisfying the KKT conditions (2.7) up to a certain tolerance threshold is found.

Primal-dual interior point method

The PDIP method is similar to the SQP method in that it also generates a sequence of primal-dual solution candidates iteratively for NLP (2.3). However, instead of solving a sequence of local QP subproblems, the PDIP method indirectly solves a version of the NLP where the inequality constraints have been relaxed through the introduction of *slack variables*. PDIP algorithms tend to be faster than SQP algorithms on large NLPs, but their convergence is generally not as robust.

The PDIP method proceeds by iteratively solving a relaxed version of the

KKT conditions (2.7):

$$\nabla_x \mathcal{L}(x, \lambda, \mu) = 0, \tag{2.15a}$$
$$g_i(x) = 0, \qquad i = 1, ..., m, \tag{2.15b}$$
$$h_i(x) + s_i = 0, \qquad i = 1, ..., p, \tag{2.15c}$$
$$\mu_i s_i - \tau = 0, \qquad i = 1, ..., p, \tag{2.15d}$$
$$\mu_i \geq 0, \qquad i = 1, ..., p, \tag{2.15e}$$
$$s_i \geq 0, \qquad i = 1, ..., p, \tag{2.15f}$$

where τ is the *barrier parameter*, and $s = [s_1, ..., s_p]^\top$ is the vector of slack variables. Part of the rationale for considering these modified KKT conditions is that the complementary slackness conditions have now been smoothened in (2.15d), thanks to the barrier parameter. These modified conditions provide an approximation of the original KKT conditions (2.7) that gets gradually closer as $\tau \to 0$. Observe that the original KKT conditions are recovered when $\tau = 0$. PDIP solvers are usually initialized with a large barrier parameter τ, where the convergence of Newton's method is boosted by the smoother KKT conditions. As iterations progress, τ is then gradually decreased to approach the KKT conditions of the original problem, and the algorithms normally terminate when (2.15) has been solved for a small enough value of τ.

Newton's method can be applied directly on the subset of conditions (2.15a)-(2.15d) since they are all equalities. The Newton direction at iteration j is then given by solving the system:

$$\begin{bmatrix} H^{[j]} & \nabla g(x^{[j]}) & \nabla h(x^{[j]}) & 0 \\ \nabla g(x^{[j]})^\top & 0 & 0 & 0 \\ \nabla h(x^{[j]})^\top & 0 & 0 & I \\ 0 & 0 & \Sigma^{[j]} & \Lambda^{[j]} \end{bmatrix} \begin{bmatrix} \Delta x^{[j]} \\ \Delta \lambda^{[j]} \\ \Delta \mu^{[j]} \\ \Delta s^{[j]} \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(x^{[j]}, \lambda^{[j]}, \mu^{[j]}) \\ g(x^{[j]}) \\ h(x^{[j]}) + s^{[j]} \\ \Sigma^{[j]} \mu^{[j]} - \tau^{[j]} e \end{bmatrix} \tag{2.16}$$

where $H^{[j]} = \nabla^2_{xx} \mathcal{L}(x^{[j]}, \lambda^{[j]}, \mu^{[j]})$ is the exact Hessian, where $\Lambda^{[j]} = \text{diag}(\mu^{[j]})$ and $\Sigma^{[j]} = \text{diag}(s^{[j]})$, where I is the identity matrix, and where $e = [1, ..., 1]^\top$. Note that (2.16) is sometimes called the *KKT system*, and the large matrix on the left-hand side the *KKT matrix*.

Taking a step along the Newton direction is similar to the SQP method:

$$x^{[j+1]} = x^{[j]} + \alpha^{[j]} \Delta x^{[j]}, \tag{2.17a}$$

$$\lambda^{[j+1]} = \lambda^{[j]} + \alpha^{[j]} \Delta \lambda^{[j]}, \tag{2.17b}$$

$$\mu^{[j+1]} = \mu^{[j]} + \alpha^{[j]} \Delta \mu^{[j]}, \tag{2.17c}$$

$$s^{[j+1]} = s^{[j]} + \alpha^{[j]} \Delta s^{[j]}. \tag{2.17d}$$

Here, however, some backtracking on $\mu^{[j+1]}$ and $s^{[j+1]}$ is needed in order to ensure that the inequality conditions (2.15e) and (2.15f) hold at the next solution candidate. This is pretty straightforward since backtracking on μ and s does not require any function evaluation, and the step size can simply be chosen as $\alpha^{[j]} = \min(\alpha_\mu^{[j]}, \alpha_s^{[j]})$, where $\alpha_\mu^{[j]} = \max(\alpha \in]0,1]|\, \mu^{[j]} + \alpha \Delta \mu^{[j]} \geq 0)$ and $\alpha_s^{[j]} = \max(\alpha \in]0,1]|\, s^{[j]} + \alpha \Delta s^{[j]} \geq 0)$. An additional backtracking step may then be carried out in order to ensure progress.

Finally, note that the PDIP method, together with another optimization method called the active-set method, often forms the basis of modern numerical QP solvers, which can for instance be used to solve the QP subproblems (2.13) arising in the SQP method.

Mixed-integer problems

We have so far presented numerical methods for solving continuous optimization problems, a general form of which is given as NLP (2.3). The presentation is now extended to include optimization problems where some of the decision variables are restricted to take integer values, so-called *mixed-integer* problems.

Without loss of generality, a general form for a mixed-integer optimization problem where all integer variables are represented as binary variables is:

$$\min_{x_c, x_b} \quad f(x_c, x_b) \tag{2.18a}$$

$$\text{s.t.} \quad g(x_c, x_b) = 0, \tag{2.18b}$$

$$h(x_c, x_b) \leq 0, \tag{2.18c}$$

$$x_b \in \{0,1\}^{n_b}, \tag{2.18d}$$

where $x_c \in \mathbb{R}^{n_c}$ denotes the vector of continuous variables, x_b is the vector of binary variables, n_c and n_b are the number of continuous and binary variables, respectively, f is the objective function, g is the vector of equality constraints and h is the vector of inequality constraints. Note that if the vector of binary variables is empty, i.e. if $x_b = \varnothing$, then problem (2.18) essentially reduces to NLP (2.3).

A popular general framework for solving mixed-integer problems of the form (2.18) is the family of Branch and Bound (BnB) methods. The core idea behind these methods is to iteratively explore a search tree in which each branch represents a subset of solutions for the integer variables of the problem. The tree search is carried out by gradually improving an upper and lower bound on the optimal value of the problem. The upper bound is normally improved by finding better feasible solutions, while the lower bound is usually improved by solving continuous subproblems at tree nodes in which the integrality constraints have been relaxed. There exist many BnB variants, differing mainly in their search strategy, but this general procedure lies at the heart of many well-established solvers for mixed-integer problems today.

In the rest of this thesis, the general form (2.18) and its associated notation are used when formulating optimization problems.

CHAPTER 3

Truck Platoon Control

Each of the next three chapters is centered on one of the control problems treated in this thesis. Each problem is first formulated as a general optimization problem where the coupling structure between vehicles is made apparent. Then, a decomposition scheme is presented to exploit the specific coupling structure of each problem with the goal of distributing most of the computations required to solve it.

The cooperative eco-driving control problem for a platoon of electric trucks is the focus of this chapter. It is shown how the *Riccati recursion* can be deployed to fully distribute second-order optimization algorithms when solving this problem.

3.1 Problem formulation

The platoon eco-driving control problem consists of finding speed profiles for all vehicles of the platoon over a given control horizon such that the overall energy consumption of the platoon is minimized. In the version of the problem formulated in Paper A, a continuous model for the longitudinal dynamics of the vehicles is assembled in the space domain. In this problem, the state variables are the speed (or, more precisely, the kinetic energy) and travel time of each vehicle, and the control inputs are the longitudinal and braking force of each vehicle.

No binary variables are needed when assembling the cooperative eco-driving control problem of a truck platoon. Therefore, $x_b = \varnothing$ and x_c is noted as x in this section for the sake of clarity. Here, an index i denotes the i-th truck of the platoon, where it is assumed that the platoon leader has an index 1. The optimization problem formulated in Paper A can be written conceptually as:

$$\min_{x} \quad \sum_{i=1}^{N} f_i(x_i) \tag{3.1a}$$
$$\text{s.t.} \quad g_1(x_1) = 0, \tag{3.1b}$$
$$h_1(x_1) \leq 0, \tag{3.1c}$$
$$\forall i \in [\![2, N]\!] :$$
$$g_i(x_{i-1}, x_i) = 0, \tag{3.1d}$$
$$h_i(x_{i-1}, x_i) \leq 0, \tag{3.1e}$$

where N is the number of trucks in the platoon, where the optimization variables are organized as $x = [x_1, ..., x_N]^\top$, where the constraint functions are organized as $g = [g_1, ..., g_N]^\top$ and $h = [h_1, ..., h_N]^\top$, and where the objective function term for each truck i is noted f_i.

It can be observed in (3.1) that the constraints of the platoon leader have a different form than the constraints of the other platoon members. Indeed, since the platoon leader is not closely trailing behind any other truck, its equality constraints (3.1b) consist of its own longitudinal dynamics, and its inequality constraints (3.1c) include speed limits, the power limitations of its electric motor, and a final travel time constraint. Constraints for the other trucks contain all these terms as well, but the equality constraints (3.1d) also include a drag reduction term dependent on the distance with the preceding truck, and the inequality constraints (3.1e) include safety constraints that specify a minimum acceptable distance to the preceding truck. Since (3.1) is an eco-driving problem, the objective function is simply the overall energy consumption of the platoon.

The objective function of the problem is separable since the overall energy consumption is the sum of the energy consumption of individual trucks.

However, both the drag reduction and collision avoidance terms introduce constraint coupling. The coupling structure is particular as each vehicle is only coupled to its immediate neighbors in the platoon, resulting in a so-called *chain-like* structure. This coupling structure has interesting properties for solving the problem, as will be explained in a moment.

The version of problem (3.1) formulated in Paper A has a nonlinear objective function and nonlinear constraints and is therefore an NLP. No decomposition method is applied directly to this NLP. Instead, if one deploys second-order optimization algorithms, like an SQP or PDIP algorithm, to solve NLP (3.1), one can in fact decompose the computations required by the algorithms due to the particular chain-like structure of a vehicle platoon. Here, the decomposition step takes place directly in the low-level computations rather than upstream when formulating the problem. The proposed decomposition scheme is based on the concept of Riccati recursion, to which we now give a brief introduction.

3.2 The Riccati recursion

Let us consider the finite-horizon, discrete-time Linear-quadratic (LQ) control problem, which can be written as the generic QP:

$$
\begin{aligned}
\min_{X,U} \quad & \sum_{k=0}^{N_k-1} \left(\frac{1}{2} \begin{bmatrix} X_k \\ U_k \end{bmatrix}^\top \begin{bmatrix} Q_k & M_k \\ M_k^\top & R_k \end{bmatrix} \begin{bmatrix} X_k \\ U_k \end{bmatrix} + \begin{bmatrix} q_k \\ r_k \end{bmatrix}^\top \begin{bmatrix} X_k \\ U_k \end{bmatrix} \right) \\
& + \frac{1}{2} X_{N_k}^\top Q_{N_k} X_{N_k} + q_{N_k}^\top X_{N_k} && \text{(3.2a)} \\
\text{s.t.} \quad & X_0 = \bar{X}_0, && \text{(3.2b)} \\
& \forall k \in [\![0, N_k - 1]\!] : \\
& X_{k+1} = A_k X_k + B_k U_k + c_k, && \text{(3.2c)}
\end{aligned}
$$

where $X = [X_0, ..., X_{N_k}]^\top \in \mathbb{R}^{n(N_k+1)}$ is the vector of state variables and $U = [U_0, ..., U_{N_k-1}]^\top \in \mathbb{R}^{mN_k}$ is the vector of control inputs, $\bar{X}_0 \in \mathbb{R}^n$ is the vector of initial conditions, N_k is the horizon length, Q_k,R_k,M_k are penalty matrices and q_k,r_k are penalty vectors, and the matrices A_k,B_k and the vector c_k are used to write the dynamics.

This type of LQ problem often appears in MPC-related applications [41], in particular when solving MPC-type optimization problems with second-order iterative methods, where subproblems with the form (3.2) usually need to be solved at each iteration [50]–[53]. It is standard to assume that the matrices:

$$\begin{bmatrix} Q_k & M_k \\ M_k^\top & R_k \end{bmatrix}, \quad \forall k \in [\![0, N_k - 1]\!], \tag{3.3}$$

are positive semidefinite, and that all matrices R_k, $k \in [\![0, N_k - 1]\!]$, are positive definite. These assumptions constitute a sufficient condition for problem (3.2) to have a unique solution [51].

The optimal solution of (3.2) can be obtained by solving the KKT conditions:

$$Q_k X_k + M_k U_k + q_k + A_k^\top \lambda_{k+1} - \lambda_k = 0, \quad \forall k \in [\![0, N_k - 1]\!], \tag{3.4a}$$
$$M_k^\top X_k + R_k U_k + r_k + B_k^\top \lambda_{k+1} = 0, \quad \forall k \in [\![0, N_k - 1]\!], \tag{3.4b}$$
$$Q_{N_k} X_k + q_{N_k} - \lambda_{N_k} = 0, \tag{3.4c}$$
$$X_0 = \bar{X}_0, \tag{3.4d}$$
$$X_{k+1} = A_k X_k + B_k U_k + c_k, \quad \forall k \in [\![0, N_k - 1]\!], \tag{3.4e}$$

where $\lambda_0 \in \mathbb{R}^n$ are the Lagrange multipliers associated with the initial conditions constraint (3.2b), and each $\lambda_k \in \mathbb{R}^n$, $k \in [\![1, N_k]\!]$, are the Lagrange multipliers associated with the equality constraints (3.2c).

Equivalently, these KKT conditions can be assembled to form the KKT system:

$$TZ = L, \tag{3.5}$$

where T is the KKT matrix, where Z gathers the primal-dual optimization variables and L the constant terms, such that:

$$T = \begin{bmatrix} & -I & & & & & & & & & \\ -I & Q_0 & M_0 & A_0^\top & & & & & & & \\ & M_0^\top & R_0 & B_0^\top & & & & & & & \\ & A_0 & B_0 & & -I & & & & & & \\ & & & -I & Q_1 & M_1 & & & & & \\ & & & & M_1^\top & \ddots & \ddots & & & & \\ & & & & & \ddots & Q_{N_k-1} & M_{N_k-1} & A_{N_k-1}^\top & \\ & & & & & & M_{N_k-1}^\top & R_{N_k-1} & B_{N_k-1}^\top & \\ & & & & & & A_{N_k-1}^\top & B_{N_k-1} & & -I \\ & & & & & & & & -I & Q_{N_k} \end{bmatrix} \tag{3.6a}$$

$$Z = \begin{bmatrix}\lambda_0 & X_0 & U_0 & \lambda_1 & X_1 & \dots & X_{N_k-1} & U_{N_k-1} & \lambda_{N_k} & X_{N_k}\end{bmatrix}^\top \tag{3.6b}$$

$$L = \begin{bmatrix}-\bar{X}_0 & -q_0 & -r_0 & -c_0 & \dots & -r_{N_k-1} & -c_{N_k-1} & -q_{N_k}\end{bmatrix}^\top \tag{3.6c}$$

It can be observed that the KKT matrix for the LQ problem (3.6a) is sparse and has a particular *banded* structure. The Riccati recursion is an algorithm that is designed to exploit this particular structure and provide the solution of the KKT system (3.5). Instead of directly inverting the KKT matrix, the Riccati recursion factorizes it during a backward sweep and then computes the optimal solution during a forward sweep. Performing these two successive sweeps has a lower computational complexity than performing a direct inversion of the KKT matrix [50]. For this reason, the Riccati recursion is often used today in high-performance solvers for MPC applications [54].

The complete derivation of the Riccati recursion for the LQ problem (3.2) is left outside the scope of this thesis. Instead, we refer the interested reader to [53], where a comprehensive treatment of the topic is given. Only an adapted version of the Riccati recursion algorithms for the standard problem (3.2) is reproduced here.

Algorithm 1 presents the operations that take place during the backward sweep of the Riccati recursion. A matrix P_k and a vector ψ_k are computed for each stage of the LQ problem. Then, the optimal solution of the problem is computed iteratively during the forward sweep, which is detailed in Algorithm 2. Note that the entire Riccati recursion procedure for solving the discrete-

Algorithm 1: Backward sweep of the Riccati recursion

1 $P_{N_k} \leftarrow Q_{N_k}, \quad \psi_{N_k} \leftarrow -q_{N_k}$
2 **for** $k = N_k - 1, ..., 0$ **do**
3 $\quad \Lambda_{k+1} \leftarrow (R_k + B_k^\top P_{k+1} B_k)^{-1}$
4 $\quad P_k \leftarrow Q_k + A_k^\top P_{k+1} A_k - (A_k^\top P_{k+1} B_k + M_k)\Lambda_{k+1}(B_k^\top P_{k+1} A_k + M_k^\top)$
5 $\quad \Psi_k \leftarrow -(M_k + A_k^\top P_{k+1} B_k)\Lambda_{k+1}(B_k^\top \Psi_{k+1} - r_k - B_k^\top P_{k+1} c_k) +$
$\qquad A_k^\top \Psi_{k+1} - q_k - A_k^\top P_{k+1} c_k$
6 **end**

Algorithm 2: Forward sweep of the Riccati recursion

1 $X_0 \leftarrow \bar{X}_0$
2 **for** $k = 0, ..., N_k - 1$ **do**
3 $\quad U_k \leftarrow \Lambda_{k+1}(B_k^\top \Psi_{k+1} - r_k - B_k^\top P_{k+1} c_k) - \Lambda_{k+1}(B_k^\top P_{k+1} A_k + M_k^\top) X_k$
4 $\quad \lambda_k \leftarrow P_k X_k - \psi_k$
5 $\quad X_{k+1} \leftarrow A_k X_k + B_k U_k + c_k$
6 **end**
7 $\lambda_{N_k} \leftarrow P_{N_k} X_{N_k} - \psi_{N_k}$

time LQ problem (3.2) is conceptually very close to the standard *dynamic programming* solution, see e.g. [41] and [55]. In fact, the operation on Line 4 of Algorithm 1 is none other than the discrete-time Riccati equation. By solving the KKT system (3.5), the Riccati recursion provides the optimal solution of the QP (3.2).

3.3 Decomposition method

We now turn our attention back to the platoon control problem. As mentioned before, the optimization problem (3.1) is an NLP and can be solved with iterative second-order methods, like SQP or PDIP. Doing so ultimately requires having to solve subproblems at each iteration, regardless of the method chosen. Adapting the notation a bit, it is shown in Paper A that these subproblems have the general form:

$$
\begin{aligned}
\min_{X,U} \quad & \sum_{i=1}^{N} \left(\frac{1}{2} \begin{bmatrix} X_{i-1} \\ U_i \end{bmatrix}^\top \begin{bmatrix} Q_{i-1} & M_i \\ M_i^\top & R_i \end{bmatrix} \begin{bmatrix} X_{i-1} \\ U_i \end{bmatrix} + \begin{bmatrix} q_{i-1} \\ r_i \end{bmatrix}^\top \begin{bmatrix} X_{i-1} \\ U_i \end{bmatrix} \right) \\
& + \frac{1}{2} X_N^\top Q_N X_N + q_N^\top X_N && \text{(3.7a)} \\
\text{s.t.} \quad & X_0 = 0, && \text{(3.7b)} \\
& \forall i \in [\![1, N]\!] : \\
& X_i = A_i X_{i-1} + B_i U_i + c_i. && \text{(3.7c)}
\end{aligned}
$$

Here, the notation introduced at the beginning of this chapter has been readopted: an index i denotes the i-th truck of the platoon, and N is the total number of trucks. In this problem, the subscript i denotes local information that only the i-th truck has access to. Since the end goal is to design a fully distributed control algorithm for the platoon, it is crucial to keep track of exactly what information needs to be exchanged between vehicles when solving the subproblems.

The version of the subproblems given in (3.7) has been adapted from (A.16) in Paper A. The original version is obtained by using the SQP method to solve (3.1) and also includes linear inequality constraints. These constraints are not reproduced here, however, for the sake of keeping this presentation concise. In fact, it is shown in Paper A that the general structure of the KKT matrix of the subproblems is not modified when linear inequality constraints are present. We refer the interested reader to Paper A for more details on how inequality constraints can be addressed in that case.

At first sight, the QP (3.7) appears to have a very similar structure to the standard LQ problem (3.2). In fact, they can be shown to be exact same problem when carrying out the following variable and index changes in (3.7):

$$
\begin{aligned}
X_{i-1} &\leftarrow \tilde{X}_i, && \forall i \in [\![1, N+1]\!], && \text{(3.8a)} \\
Q_{i-1} &\leftarrow \tilde{Q}_i, && \forall i \in [\![1, N+1]\!], && \text{(3.8b)} \\
q_{i-1} &\leftarrow \tilde{q}_i, && \forall i \in [\![1, N+1]\!], && \text{(3.8c)} \\
i &\leftarrow i-1, && \forall i \in [\![1, N+1]\!], && \text{(3.8d)}
\end{aligned}
$$

where $\tilde{X}_i$ would then be the new name given to the state vectors. In other words, problems (3.2) and (3.7) have the same structure, but the variables have been indexed differently. The reason for this is that the standard LQ control problem (3.2) is assembled stage-by-stage, whereas the subproblem (3.7) has been organized vehicle-by-vehicle. More precisely, the individual stage-by-stage dynamics of each vehicle in (3.7) have been "hidden" away in their local state vector in order for the coupling structure between vehicles to appear in the QP. The coupling between the trucks in the platoon is clearly visible in the equality constraints (3.7c), as the states of truck i depend on its own control inputs, but also on the states of the preceding truck $i-1$ due to the distance-dependent drag reduction that comes from driving close to another truck. The zero initial conditions in (3.7b) model the fact that the platoon leader (with index 1) does not benefit from any drag reduction since it is not trailing behind another truck.

Since the subproblems (3.7) have the same structure as the standard LQ control problem (3.2), their KKT matrix has the same banded structure as in (3.6a), and the Riccati recursion can thus be applied to solve them. This time, the larger blocks inside the KKT matrix each contain the matrices and local parameters corresponding to one vehicle. Consequently, the operations carried out at each step of the backward and forward sweeps in the Riccati recursion mostly involve the local data of a single truck. This particular way to solve the truck subproblems opens the door to a fully distributed solution method, as explained in the next section.

3.4 Distributed optimization

Algorithm 1 and Algorithm 2, which present the Riccati recursion for a standard LQ control problem, are now modified to adopt the notation used in the truck platoon subproblems.

Algorithm 3 and Algorithm 4 present the Riccati recursion algorithms for the truck platoon subproblems (3.7). The order in which the operations are presented has been slightly modified compared to Algorithm 1 and Algorithm 2 in order to clearly highlight the role of each vehicle and the information exchange needed to solve the subproblems. Remember that the notations have been chosen such that each subscript i denotes the local information of truck i. In these algorithms, the index of the truck involved in the computations is given at the beginning of each line. Variables and coefficients with subscript

Algorithm 3: Backward sweep of the Riccati recursion for the truck platoon subproblem (3.7)

1 N: $P_N \leftarrow Q_N, \quad \psi_N \leftarrow -q_N$
2 **for** $i = N, ..., 2$ **do**
3 $\quad i$: $\Lambda_i \leftarrow (R_i + B_i^\top P_i B_i)^{-1}$
4 $\quad i$: $P_{i-1} \leftarrow A_i^\top P_i A_i - (A_i^\top P_i B_i + M_i)\Lambda_i(B_i^\top P_i A_i + M_i^\top)$
5 $\quad i$: $\Psi_{i-1} \leftarrow -(M_i + A_i^\top P_i B_i)\Lambda_i(B_i^\top \Psi_i - r_i - B_i^\top P_i c_i) - A_i^\top P_i c_i$
$\qquad\qquad + A_i^\top \Psi_i$
6 $\quad i$: **send** P_{i-1} and ψ_{i-1} to bus $i-1$
7 $\quad i-1$: $P_{i-1} \leftarrow P_{i-1} + Q_{i-1}$
8 $\quad i-1$: $\psi_{i-1} \leftarrow \psi_{i-1} - q_{i-1}$
9 **end**

Algorithm 4: Forward sweep of the Riccati recursion for the truck platoon subproblem (3.7)

1 1: $X_0 \leftarrow 0, \quad \lambda_0 \leftarrow P_0 X_0 - \psi_0$
2 **for** $i = 1, ..., N$ **do**
3 $\quad i$: $U_i \leftarrow \Lambda_i(B_i^\top \Psi_i - r_i - B_i^\top P_i c_i) - \Lambda_i(B_i^\top P_i A_i + M_i^\top)X_{i-1}$
4 $\quad i$: $X_i \leftarrow A_i X_{i-1} + B_i U_i + c_i$
5 $\quad i$: $\lambda_i \leftarrow P_i X_i - \psi_i$
6 $\quad i$: **send** X_i to bus $i+1$
7 **end**

0 have been included in order to make comparisons with the standard LQ problem easier, but they do not matter in the computations. Note also that Line 6 in the forward sweep algorithm does not apply to the last truck of the platoon with index N.

Thanks to the chain-like structure of the truck platoon, the backward and forward sweeps of the Riccati recursion can be carried out physically up and down the platoon by letting each truck take care of its own computations. To do so, each intermediary truck i needs to receive matrix P_i and vector ψ_i from the following truck during the backward sweep (Line 6 in Algorithm 3), and the state vector X_{i-1} from the preceding truck during the forward sweep (Line 6 in Algorithm 4), as symbolized in Figure 3.1. This means that the local information of each truck does not need to be communicated to other

Figure 3.1: Representation of the inter-truck communications needed during the Riccati recursion.

trucks, thus making this procedure *privacy-preserving*. Note that additional information related to the overarching optimization algorithm would also need to be exchanged when solving the platoon control problem (3.1). This includes step size and termination criterion information of the SQP or PDIP algorithm deployed for instance, none of which compromises the privacy-preserving nature of the proposed method. We refer the interested reader to Paper A for an in-depth presentation of the entire solution method. Additionally, some level of parallelization could in principle be achieved in Algorithm 3 and Algorithm 4. Even though the operations are performed sequentially in these algorithms, vehicles could start precomputing the local linear algebra operations required while waiting for the missing terms to be communicated to them by the other vehicles.

As we have seen, the proposed algorithms do not require centralized computations. We believe that having a fully distributed control method for the cooperative eco-driving of a truck platoon has promising practical implications. Since trucks do not need to communicate their local information with other trucks and are ultimately responsible for computing their own solution, it is conceivable that trucks belonging to different transportation companies agree to form a platoon together for their mutual benefit. Such platoons could be formed spontaneously while driving by trucks that happen to find themselves close to each other. However, it remains to address the question of electing a platoon leader, since the leader does not benefit from a decreased drag reduction and has thereby no immediate interest in being part of a platoon. In Paper A, a leader compensation mechanism is proposed where an equivalent monetary value is estimated from a parametric NLP in which a virtual drag reduction term has been added for the platoon leader. It is unnecessary to solve this NLP explicitly once the solution to the original problem (3.1) is known, and the proposed leader compensation value is therefore cheap to compute. This could contribute to making the role of platoon leader more attractive and thereby facilitate the spontaneous formation of platoons with the proposed fully distributed control method.

CHAPTER 4

Bus Line Control

This chapter focuses on the eco-driving and operational control problem for an electric bus line. A formulation as an optimization problem is given, with a particular emphasis on the coupling structure of the problem. A *primal decomposition* scheme exploiting the weakly coupled structure of the problem can be deployed to solve it in a scalable and computationally effective way. The next few sections explain how.

4.1 Problem formulation

The bus line eco-driving and operational control problem is very similar to the platoon eco-driving control problem in that it also consists of finding speed profiles for all vehicles over a given control horizon. The cost function in the bus line control problem includes the overall energy consumption of the vehicles as well as a service regularity term that penalizes irregular bus headways. Here too a continuous model for the longitudinal dynamics of the vehicles is formulated in the space domain. The state variables also include the speed (or rather, the kinetic energy) and travel time of the vehicles, and the control variables are the longitudinal and braking force of the vehicles.

Electric buses are quite similar to electric trucks in how their dynamics and energy consumption are modeled, and the main difference between the bus line control problem and the truck platoon control problem presented in the previous chapter comes from the additional constraints related to passenger boarding and the service regularity objective.

Note that the bus line model assembled in Paper B relies on the following modeling assumptions: buses cannot dwell at bus stops longer than needed for the passenger exchange operation to complete, passenger arrivals at stops are described by a homogeneous Poisson process, the onboard capacity of buses is not limited, and overtaking is not allowed.

The formulation of the eco-driving and operational control problem of an electric bus line is modeled without binary variables in Paper B. Therefore, here too we have $x_b = \varnothing$, and use the notation $x = x_c$. An index i now denotes the i-th bus of the line. The optimization problem assembled in Paper B has the general form:

$$\min_{x} \quad \sum_{i=1}^{N} f_{s,i}(x_i) + f_c(x_{i-1}, x_i) \tag{4.1a}$$

$$\text{s.t.} \quad \forall i \in [\![1, N]\!] :$$

$$g_i(x_i) = 0, \tag{4.1b}$$

$$h_i(x_i) \leq 0, \tag{4.1c}$$

where N is the total number of buses, where the optimization variables are such that $x = [x_1, ..., x_N]^\top$, and where the constraints are such that $g = [g_1, ..., g_N]^\top$ and $h = [h_1, ..., h_N]^\top$. In the objective function, $f_{s,i}$ denotes the separable objective term which only depends on the variables of bus i, while f_c represents the coupling objective terms. Note that the notation $x_0 = x_N$ is used in the objective function (4.1a).

In this problem, the equality constraints (4.1b) represent the longitudinal dynamics of each vehicle. These dynamics include a model for passenger accumulation at bus stops in order to capture bus dwell times at stops, since buses have to stop and pick up passengers at bus stops. Changes in bus mass resulting from these passenger exchange operations are also captured in the equality constraints. The inequality constraints (4.1c) include speed limits and the power limitations of the electric motor of each bus. As for the objective function (4.1a), the separable objective terms represent the energy consumption, which is computed from a detailed model of the electric motor and battery of each bus. Finally, the coupling objective terms capture the headways between consecutive buses, hence the coupling between each bus i and the bus $i-1$ preceding it. The objective of this problem is thus to minimize the overall energy consumption (eco-driving) and to incentivize all headways to have a similar value (service regularity). Since the longitudinal dynamics of the vehicles and the energy consumption model used include many nonlinear terms, the optimization problem (4.1) is an NLP.

4.2 Decomposition method

All constraints are separable in problem (4.1) since the dynamics, speed limits, and motor power limitations are specific to every bus. Similarly, each energy consumption term only involves variables associated with a single bus. Only the service regularity terms in the objective function involve more than one vehicle, as they introduce some coupling between all pairs of neighboring buses.

The inter-bus coupling through f_c in the objective function (4.1a) comes from only a subset of variables, referred to as *coupling variables* hereafter. For each $i \in [\![1, N]\!]$, let $x_i = [y_i, z_i]^\top$, where z_i are the coupling variables of bus i. The objective function (4.1a) can thus be rewritten as:

$$\sum_{i=1}^{N} f_{s,i}(y_i, z_i) + f_c(z_{i-1}, z_i). \tag{4.2}$$

It turns out that there is only a single coupling variable per pair of neighboring buses for the bus line problem (4.1). This problem is therefore weakly coupled as the coupling variables are very few compared with the total number of variables, which motivates the use of primal decomposition methods [46], [56]. This type of decomposition provides an equivalent bilevel reformulation of the original problem, where all the coupling variables are gathered in a higher-level *master problem*, and where the rest of the terms are separated into lower-level subproblems that are independent.

The primal decomposition of problem (4.1) results in the master problem:

$$\min_{z} \quad \sum_{i=1}^{N} f_c(z_{i-1}, z_i) + f_{\mathrm{sub},i}(z_i) \tag{4.3a}$$
$$\text{s.t.} \quad \forall i \in [\![1, N]\!] :$$
$$h_{\mathrm{feas},i}(z_i) \leq 0, \tag{4.3b}$$

and in the subproblems:

$$f_{\mathrm{sub},i}(z_i) = \min_{y_i} \quad f_{s,i}(y_i, z_i) \tag{4.4a}$$
$$\text{s.t.} \quad \forall i \in [\![1, N]\!] :$$
$$g_i(y_i, z_i) = 0, \tag{4.4b}$$
$$h_i(y_i, z_i) \leq 0, \tag{4.4c}$$

where $z = [z_1, ..., z_N]^\top$ is the vector of coupling variables. Additional constraints $h_{\mathrm{feas},i}$ need to be added in the master problem in order to guarantee that the values taken by each z_i are feasible in the subproblems (4.4). This additional set of constraints can be defined implicitly as the solution of small optimization problems, and we refer the interested reader to (B.18) and (B.19) in Paper B for more details.

4.3 Distributed optimization

The bus line control problem can now be solved by solving the master problem (4.3) in the primal decomposition. The master problem is an NLP where part of the objective function is defined implicitly as the solution of the subproblems (4.4), which are also NLPs, but which are each much larger than the master problem, in light of what has been said previously.

The key benefit of the primal decomposition scheme is that all subproblems can now be solved independently since each of them only involves variables connected to a single bus. In Paper B, it is proposed to solve the master problem with the SQP method. In the SQP algorithm proposed, computations at the level of the master problem are few but must be centralized, whereas the bulk of the computations, which come from having to solve the subproblems, can be distributed. This makes it possible to design a hierarchical control

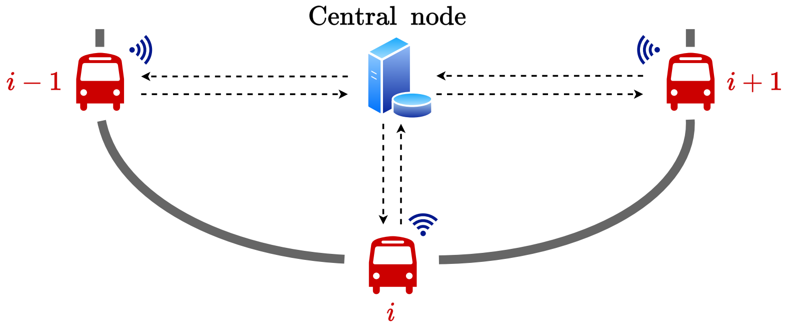

Figure 4.1: Representation of the communication structure between the vehicles and a central computing node in the proposed hierarchical control architecture.

architecture where computations for solving the subproblems could be physically distributed. One option could be to solve each individual bus subproblem directly onboard the bus in question, while centralized computations could be performed by a central computing node, as symbolized in Figure 4.1. This way, the communication requirements between this central node and the buses would be minimal, since buses would already have access to the solution for their own variables and only updates on the coupling variables would need to be exchanged. Most of the computations could be carried out in parallel with this procedure, thus potentially leading to significant computational speedups.

Finally, it must be noted that the vehicles are coupled in a chain-like structure here as well since each bus is only coupled to bus directly preceding and following it. In the context of this problem, this indicates that the number of coupling variables scales only linearly with the number of vehicles. Since the centralized computations involve only the coupling variables, and since the rest of the computations can be carried out in parallel, the proposed hierarchical control architecture scales very well with the number of buses.

CHAPTER 5

Bus Network Control

This chapter presents a decomposition method for the operational control and charging scheduling problem for an electric bus network.

5.1 Problem formulation

The control problem assembled in Paper C combines the tactical level charging scheduling problem, where decisions on when and for how long buses should charge are taken, and the operational control problem where decisions on bus speed and holding time are taken. Both types of decisions are taken over a common control horizon. In order to be able to anticipate bus charging needs several hours ahead, the horizons considered for this bus network control problem are significantly longer than they are for the bus line control problem presented in the previous chapter. As a result, only simplified discrete dynamics are considered for the vehicles. Contrary to the bus line control problem, bus holding is now allowed at the common bus terminal in addition to the speed control of the vehicles. The state variables now include the travel time, mass, and state-of-charge of each vehicle, and the control inputs are now speed commands, bus holding times, binary charging decisions, and charging times.

The bus network model considered in Paper C is built with the following modeling assumptions: bus lines only have a single stop in common which is also where all the shared charging infrastructure is located, bus holding can only be carried out at this shared terminal, vehicles can not be redeployed on a different bus line upon reaching the terminal, and overtaking is not allowed between buses of the same line.

The bus network operational control and charging scheduling problem, as it is formulated in Paper C, can then be represented conceptually as:

$$\begin{aligned}
\min_{x_c, x_b} \quad & \sum_{l=1}^{L} f_l(x_{c,l}, x_{b,l}) && (5.1a)\\
\text{s.t.} \quad & h_c(x_c, x_b) \leq 0, && (5.1b)\\
& x_{b,c} \in \{0,1\}^{n_{b,c}}, && (5.1c)\\
& \forall l \in [\![1, L]\!] : && \\
& g_l(x_{c,l}, x_{b,l}) = 0, && (5.1d)\\
& h_{s,l}(x_{c,l}, x_{b,l}) \leq 0, && (5.1e)\\
& x_{b,l} \in \{0,1\}^{n_{b,l}}, && (5.1f)
\end{aligned}$$

where an index l is used to denote the l-th bus line of the bus network considered, where L is the number of bus lines considered, where the continuous variables have been organized as $x_c = [x_{c,1}, ..., x_{c,L}]^\top$, where $g = [g_1, ..., g_L]^\top$, and where f_l is the objective function term for each line l. The inequality constraints have been split into separable line-specific terms $h_{s,l}$ and a term h_c gathering all coupling constraints, such that $h = [h_{s,1}, ..., h_{s,L}, h_c]^\top$. Similarly, the $n_{b,l}$ binary variables that are specific to bus line l only are noted $x_{b,l}$, and the $n_{b,c}$ coupling binary variables are noted $x_{b,c}$. The total vector of binary variables can then be written as $x_b = [x_{b,1}, ..., x_{b,L}, x_{b,c}]^\top$. Note that the coupling variables only appear in the coupling constraints (5.1b).

Contrary to the other problems presented so far in this thesis, charging decisions must now be taken for the vehicles. These decisions are modeled with the binary variables $x_{b,l}$ here, with the consequence of making (5.1) a mixed-integer problem. The state-of-charge evolution of each vehicle in the network, including battery charging and energy consumption when driving, is captured in the equality constraints (5.1d). Additionally, the equality constraints model the dynamics of the vehicles and the evolution of their mass, which is determined by the passenger exchange operations. Next, the inequality constraints (5.1e) include travel time bounds on each inter-stop link of the bus routes, bus capacity constraints, no-overtaking constraints, and, crucially, constraints on the minimum state-of-charge that each bus must have when leaving the chargers in order to be able to complete one full trip.

The economic objective function (5.1a) consists of a service regularity cost term, which penalizes deviations from a desired target headway, and a charging cost term for each bus line. The coupling constraints (5.1b) are *charger exclusion constraints* that emerge from the assumption that the available charging capacity is limited. This assumption is made here since the process of charging buses in operation is usually carried out with high-power chargers which are individually very expensive, with the consequence that transit agencies rarely over-invest in them. These charger exclusion constraints model the fact that two vehicles can not be using the same charger at the same time. However, modeling this requires additional binary variables, which are noted $x_{b,c}$ here. Finally, it must be noted that the objective function and all constraints are linear in (5.1). This problem is therefore a Mixed-integer Linear Program (MILP).

5.2 Decomposition method

In (5.1), the objective function and most of the constraints are separable across bus lines. However, since the charging infrastructure is assumed to be shared by all buses, the charger exclusion constraints (5.1b) involve all vehicles of the network, thereby introducing coupling between vehicles from different bus lines. Similarly, the binary variables $x_{b,c}$, which appear in these constraints, are also shared across bus lines. This means that problem (5.1) contains both coupling variables and coupling constraints. Note that, if one were to zoom in on the internal coupling structure of each individual bus line, one would find a chain-like coupling structure between vehicles similar to the one investigated in the previous chapter. However, only coupling between bus lines has been exploited for the present problem and it is therefore the sole focus of the rest of this chapter.

Being an MILP, the bus network problem formulated in (5.1) is NP-complete [57], and thus suffers from combinatorial explosion. Solving such problems to optimality can prove to be difficult in practice, even for relatively small problem instances. In order for our problem to be tractable, even when considering larger bus networks, we now present a heuristic solution method based on *Lagrangian relaxation.*

In a nutshell, the Lagrangian relaxation is a procedure that proposes to solve a dual problem of the original MILP [58], [59]. This so-called *Lagrangian dual problem* is formed by choosing to relax a subset of constraints from the original problem, usually constraints that would make the problem easier in some sense if they were removed. However, solving this dual problem is no guarantee of finding the optimal solution of the original MILP since there might be a nonzero *duality gap*, i.e. a difference between the optimal value of the original problem and of its dual version. Fortunately, the Lagrangian relaxation is known to generate very small duality gaps in general [58], which in part motivates its use in the proposed heuristic solution method.

As mentioned before, all the coupling inequality constraints are affine in the formulation of the operational control and charging scheduling problem presented in (5.1). By noting K the dimension of h_c in (5.1b), the k-th component of h_c, with $k \in [\![1, K]\!]$, can therefore be expressed as:

$$\sum_{l=1}^{L}(H_{c,l}^{k} x_{c,l} + H_{b,l}^{k} x_{b,l}) + H_{b,c}^{k} x_{b,c} + h^{k}, \tag{5.2}$$

where the constraint coefficients have been gathered in the vectors $H_{c,l}^{k}$, $H_{b,l}^{k}$, and $H_{b,c}^{k}$, and where h^k is a scalar coefficient. Since these constraints are responsible for the inter-line coupling in the problem, they are chosen to be relaxed in the proposed Lagrangian relaxation.

Having chosen the coupling constraints for the relaxation, the Lagrangian function of the problem (5.1) is defined as:

$$\mathcal{L}(x_c, x_b, \lambda) = \sum_{l=1}^{L} f_l(x_{c,l}, x_{b,l}) + \sum_{k=1}^{K} \lambda^k \left[\sum_{l=1}^{L}(H_{c,l}^{k} x_{c,l} + H_{b,l}^{k} x_{b,l}) + H_{b,c}^{k} x_{b,c} + h^{k} \right] \tag{5.3}$$

where λ is a vector containing the Lagrange multipliers associated with the relaxed constraints (5.2), such that $\lambda = [\lambda^1, ..., \lambda^K]^\top$.

Let us now make an observation on the role played by the coupling variables

$x_{b,c}$ in (5.3). First, let us consider the following minimization problem:

$$\min_{x_{b,c}} \quad \mathcal{L}(x_c, x_b, \lambda) \tag{5.4a}$$

$$\text{s.t.} \quad x_{b,c} \in \{0,1\}^{n_{b,c}} \tag{5.4b}$$

This problem consists in minimizing the Lagrangian function with respect to the coupling variables only. The optimal solution of this problem is quite straightforward since the variables $x_{b,c}$ only enter linearly in $\mathcal{L}$ in (5.3). The solution of (5.4) is given by:

$$x_{b,c} = \max\Big(- \operatorname{sign}\Big(\sum_{k=1}^{K} \lambda^k H_{b,c}^k \Big), 0 \Big). \tag{5.5}$$

In other words, when minimizing the Lagrangian function $\mathcal{L}$ for a fixed vector of Lagrange multipliers λ, each component of $x_{b,c}$ takes the value 0 or 1 depending on the sign of a fixed coefficient. This observation is useful for simplifying the Lagrangian relaxation form presented now.

The Lagrangian dual problem can then be formulated as:

$$\max_{\lambda \geq 0} \; f_{\text{dual}}(\lambda), \tag{5.6}$$

where:

$$f_{\text{dual}}(\lambda) = \min_{x_c, x_b} \quad \sum_{l=1}^{L} f_l(x_{c,l}, x_{b,l}) + \sum_{k=1}^{K} \sum_{l=1}^{L} \lambda^k (H_{c,l}^k x_{c,l} + H_{b,l}^k x_{b,l}) + \lambda^k \tilde{h}^k \tag{5.7a}$$

$$\text{s.t.} \quad \forall l \in [\![1, L]\!] :$$

$$g_l(x_{c,l}, x_{b,l}) = 0, \tag{5.7b}$$

$$h_{s,l}(x_{c,l}, x_{b,l}) \leq 0, \tag{5.7c}$$

$$x_{b,l} \in \{0,1\}^{n_{b,l}}. \tag{5.7d}$$

Since they did not appear in any of the constraints, the coupling variables have been removed from the optimization problem (5.7) by making use of the observation presented above in (5.4)-(5.5). The constant terms resulting from replacing $x_{b,c}$ by the expression given in (5.5) have been gathered in the terms

$\tilde{h}^k$ in (5.7a), together with the other constant terms initially present in the Lagrangian function.

5.3 Distributed optimization

It can be observed that the optimization problem (5.7) resulting from the Lagrangian relaxation procedure is separable into line-specific subproblems since all coupling constraints and variables have been removed. Doing so was in turn only possible because the coupling variables entered linearly in the coupling constraints, and were not present in any of the other constraints in problem (5.1).

In Paper C, it is proposed to solve the Lagrangian dual problem using an iterative optimization algorithm called the *subgradient algorithm*. As noted previously, however, solving (5.6) might not always return the optimal solution of the original MILP (5.1) and so it must not be forgotten that the proposed method is a heuristic only.

The process of solving (5.6) with the subgradient algorithm is similar in

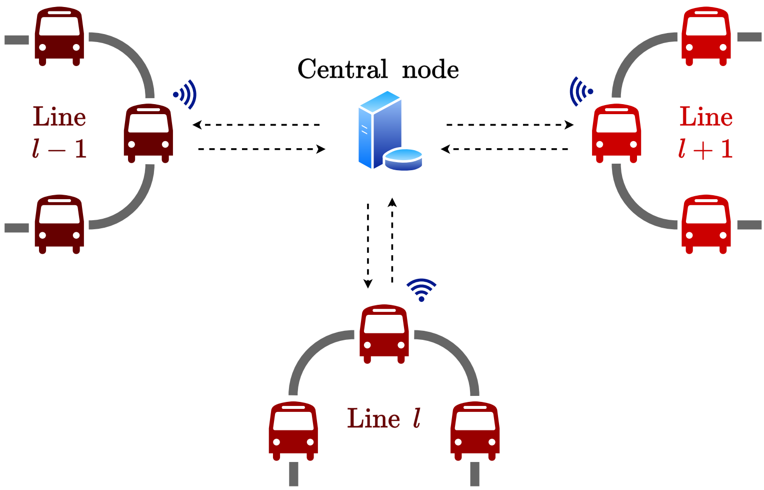

Figure 5.1: Representation of the communication structure between the bus lines and a central computing node in the proposed hierarchical control architecture.

substance to the process of solving the decomposed bus line problem with the SQP algorithm that was presented in the previous chapter. A few centralized computations must be carried out at the level of the dual problem (5.6), but most of the computations needed are for solving the MILP subproblems (5.7) and can be distributed across bus lines. A similar hierarchical control structure as for the bus line problem can thus be designed, as displayed in Figure 5.1. Individual bus line computations, which are responsible for most of the computation time of the algorithm, can now be carried out in parallel, and the few centralized computations required can be handled by a shared central computing node. This results in potentially large runtime improvements, and makes the proposed solution procedure scalable with the number of bus lines.

CHAPTER 6

Summary of included papers

This chapter provides a summary of the included papers.

6.1 Paper A

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár
Distributed Eco-driving Control of a Platoon of Electric Vehicles Through Riccati Recursion
IEEE Transactions on Intelligent Transportation Systems,
Vol. 24, pp. 3048-3063, March 2023.

This paper treats the eco-driving control problem for a platoon of electric trucks. First, the longitudinal dynamics of each vehicle are formulated in the space domain in order to include the influence of the road gradient and remove some nonlinear terms from the models. Since vehicles are driving as a platoon, safety constraints are added to impose a minimum time headway between successive vehicles to avoid collisions, and a nonlinear drag reduction term is added to the dynamics of each vehicle except for the platoon leader.

Next, the platoon control problem is formulated as an OCP with the goal of minimizing the total energy consumption of the platoon. A direct optimal control reformulation of this OCP is proposed and an SQP algorithm is deployed to solve the resulting NLP. At each SQP iteration, the Newton direction is obtained by solving a QP subproblem. By rearranging the decision variables over successive vehicles and carrying out a few algebraic operations, it is shown that each QP subproblem has the exact same structure as a standard inequality-constrained LQ problem. A PDIP algorithm is then proposed to solve each QP subproblem. By carrying out a few more algebraic operations, the KKT system to be solved at each PDIP iteration is shown to be the same as the KKT system of a standard LQ problem. Thanks to the particular banded structure of the KKT matrix, the Riccati recursion can be deployed to solve the KKT system. Since the decision variables are assembled vehicle-by-vehicle, the backward and forward sweeps of the Riccati recursion can be distributed physically by transmitting information up and down the platoon. By doing so, individual vehicles do not need to communicate their local information, thus making the procedure privacy-preserving. All the decisions related to the termination criteria and step size choices of the overarching algorithms can be similarly taken in a distributed fashion. This fully distributed control method is then evaluated in a case study involving a platoon of electric trucks driving on selected highway segments. It was found that, if all platoon members are electric trucks, a tighter platoon formation is always energy-optimal regardless of the road gradient, owing to the efficiency of the regenerative braking ability of EVs. The proposed cooperative eco-driving method manages to achieve energy savings in the order of 10% compared with trucks driving on their own, thus making a strong argument for independent trucks to spontaneously form platoons on the go with the proposed distributed control method. Finally, a compensation mechanism is proposed for the platoon leader whereby the leader receives a monetary compensation representing his share of the overall energy savings of the platoon.

6.2 Paper B

Rémi Lacombe, Sébastien Gros, Nikolce Murgovski, and Balázs Kulcsár
Bilevel Optimization for Bunching Mitigation and Eco-Driving of Elec-

tric Bus Lines
IEEE Transactions on Intelligent Transportation Systems,
Vol. 23, pp. 10662-10679, August 2022.

This paper treats the eco-driving and operational control problem for a line of electric buses. A detailed operational model of a bus line is first assembled, where the longitudinal bus dynamics are modeled in the space domain. This feature makes it possible to capture bus dynamics at bus stops without integer variables. Passenger accumulation at stops, and the resulting dwell times and mass variations of buses, are also captured in this model. Energy consumption is modeled as a nonlinear function based on realistic electric bus motor and battery models. It is assumed in this paper that a speed control strategy is used and that the aim is to minimize the energy consumption of the bus line and provide a good level of service through headway regularity. To this end, an OCP is assembled for the bus line control problem where the control horizon of each bus extends to the position of the preceding bus, so as to express headways as terminal states. This OCP is then converted into a general NLP. A primal decomposition scheme is deployed to decompose this NLP into a high-level master problem and low-level bus subproblems. In this formulation, the only coupling variables appearing in the master problem are time headways. By considering the bus subproblems as parametric NLPs for which the parameters are the time headways, an algorithm based on the SQP method is proposed to solve the master problem. In this algorithm, the bus subproblems can be solved independently and only a few centralized computations are needed at the high-level to update the parameters. This opens the door to a hierarchical control architecture where the subproblems can be solved in parallel onboard the concerned vehicle. Next, closed-loop control can be achieved by embedding the proposed solution algorithm into an MPC. A case study built on historical bus driving data is presented to evaluate the operational performance of the proposed MPC. It is shown that the MPC achieves a faster recovery to a regular level of service from initial scenarios with different degrees of bus bunching than do rule-based baselines implementing a simple holding or speed control strategy. In addition, energy savings of up to 9% are reported, thanks to the eco-driving feature of the proposed method.

6.3 Paper C

Rémi Lacombe, Nikolce Murgovski, Sébastien Gros, and Balázs Kulcsár
Integrated Charging Scheduling and Operational Control for an Electric Bus Network
Submitted in August 2023 to *Elsevier Transportation Research*.

This paper proposes to combine the charging scheduling problem with the operational control problem for an electric bus network. To this end, a detailed model of an electric bus network is assembled, one which includes travel time commands on individual inter-stop links, passenger queues at stops, capacity constraints, rush hour traffic, piecewise linear energy consumption functions for each individual inter-stop link, partial charging, limited charging capacity, and time-of-use electricity pricing. It is assumed that the transit agency relies on daytime opportunity charging for the bus network considered and that buses can only use a limited number of chargers that are all located at a common terminal shared by all bus lines. Speed control is the main control strategy deployed, but bus holding may also be applied at the shared terminal. The bus network operational control and charging scheduling problem is formulated over a long control horizon and is assembled as an MILP. An economic objective function is considered including the total charging cost as well as a monetary penalty for headway irregularities. A dual decomposition scheme based on Lagrangian relaxation is then proposed. The only coupling constraints between the different bus lines of the network are charger exclusion constraints preventing buses from accessing the same charger at the same time. These coupling constraints are relaxed to form the Lagrangian dual problem of the original MILP. The subgradient algorithm is deployed to solve the Lagrangian dual problem. Thanks to the proposed decomposition scheme, the individual bus line subproblems are separable at each subgradient iteration and can thus be solved in parallel, possibly by using physically distributed computing nodes. A simple local search heuristic is also proposed in order to generate feasible solutions to the original problem since there might be a nonzero duality gap. Next, a detailed case study based on historical data from the city of Chicago is presented. A small bus network is constructed with the high-fidelity microscopic traffic simulator Vissim. The proposed control framework is compared to two rule-based control baselines which reflect standard transit agency practices: either buses charge long enough to reach a desired state-of-charge, or they charge for a fixed amount of time at each visit to the chargers. As the proposed method integrates both charging and operational decisions, it is able to anticipate overlaps in buses' charging schedules and thus manages to avoid charging conflicts by slowing down or speeding up buses accordingly. As a result, the proposed control framework is found to consistently achieve lower overall costs over an entire day of bus operations compared with the baselines, which suffer from charging conflicts.

CHAPTER 7

Conclusion

This chapter concludes the thesis by summarizing the main results and by suggesting future research directions that the Author believes would be meaningful to investigate.

7.1 Concluding remarks

This thesis has presented partially and fully distributed optimization-based methods for solving control problems in different applications involving groups of electric vehicles. These control problems have all been formulated as optimization problems, each with its own coupling structure depending on the type of application considered.

For instance, the bus network problem displays constraint coupling through a set of constraints that we dubbed the charger exclusion constraints, which prevent buses from different lines to access the same charger at the same time. These constraints are not present in the other two problems treated in this thesis since these problems have been assembled over shorter control horizons where the need for battery charging is not considered. As such, these problems both benefit from a particular chain-like coupling structure, where vehicles are organized in succession and each vehicle is only coupled to its immediate neighbors. The platoon control problem too has been assembled with coupling constraints, but constraints of a different type than for the bus network problem. Indeed, coupling between trucks comes from enforcing safety constraints to maintain trucks at a minimum distance from one another in that problem, and also from modeling the drag reduction resulting from trucks trailing closely behind one another. The coupling structure of the bus line problem is of a different nature, as it only involves some coupling between objective terms. This is the result of buses driving far away from one another, and thus not needing the same type of safety constraints, and of buses having a shared headway regularity objective. Headways being defined as the time difference between two consecutive buses reaching the same position, inter-bus coupling is unavoidable in the objective function in this problem.

What all the optimization problems assembled during the course of this thesis have in common, though, is that they are all characterized by having a weak coupling structure. By that, we mean that the number of coupling constraints and coupling terms in these problems is small compared with the total number of constraints and terms not involving any coupling. Consequently, specific decomposition schemes have been proposed in this thesis to exploit these weak coupling structures when solving each problem.

Here, the platoon control problem has been formulated as an NLP with coupling constraints. Different algorithms exist to solve this type of optimization problem, but we chose to focus on second-order algorithms due to their superior convergence properties. Thanks to the chain-like coupling structure of the platoon, the intermediate QP subproblems that must be solved as part of an SQP or PDIP procedure can be shown to be equivalent to a standard LQ problem, but one in which the problem would be assembled vehicle-by-vehicle instead of stage-by-stage. The Riccati recursion can then be applied to factorize the banded KKT matrices of these subproblems in a vehicle-by-vehicle fashion. By doing so, each truck is able to carry out most of the computations for its own solution and does not need to broadcast its local information to the rest of the platoon, thus making the entire procedure privacy-preserving. These features would make it conceivable for trucks from different transportation companies to spontaneously form a platoon on the go in order to benefit from air drag reduction without needing to disclose any private information. With this procedure, the platoon control problem can be solved in a fully distributed fashion, since information only needs to be communicated repeatedly up and down the platoon. In principle, this fully distributed solution procedure could also be adopted for any other type of vehicle platoon, and more generally for any control application involving agents organized in a chain-like structure.

The bus line problem, which is an eco-driving problem too, has also been formulated as an NLP, but the source of coupling is now the objective function. A primal decomposition scheme has been proposed in this thesis to split this NLP into a master problem, which contains the coupling objective terms, and independent bus line subproblems. This bilevel structure of the decomposed problem makes it possible to solve the subproblems in parallel. This opens the door to a hierarchical control architecture for the bus line problem, one in which most of the computations are distributed and carried out in parallel onboard the vehicles. The amount of centralized computations required is relatively small in comparison, and so too are the communication requirements between the central unit and the vehicles since each bus would be directly computing its own solution. In addition, this procedure makes the proposed control strategy scalable, since adding more buses to the system would not increase the overall computation time by much as the new subproblems could be solved in parallel as well.

The bus network problem had to be treated a bit differently than the other two since it has been formulated as an MILP and involves binary decision variables. Problems of this type are known to scale poorly due to combinatorial explosion. To remedy this, a dual decomposition scheme has been proposed in this thesis as a scalable heuristic for finding a good solution to the bus network problem within a reasonable time. In the proposed decomposition scheme, a Lagrangian relaxation procedure is carried out in which the coupling constraints are relaxed to form the Lagrangian dual problem. Relaxing the coupling constraints makes it possible to decouple individual bus line subproblems, which are responsible for most of the computational load, and to solve them in parallel during the solution procedure. A hierarchical control architecture can thereby be assembled to find a good solution to the bus network problem. In this control architecture, the bulk of the computations can be distributed across the different bus lines. Similarly, the proposed control architecture scales well with the number of bus lines since the individual bus line subproblems are solved in parallel.

7.2 Future work

Modeling

The optimization problems formulated in this thesis are based on a variety of modeling assumptions. In future work, some of these assumptions could be relaxed in order to increase the range of systems covered or to make the control algorithms perform better by having a more faithful representation of the underlying systems. The exact nature of these modeling extensions differs for each of the applications considered in this thesis.

The platoon control problem discussed in Chapter 3 could be extended to decide when a given group of trucks driving on the same road section would benefit from spontaneously forming a platoon. Since the proposed control method is privacy-preserving, it could in principle be used by trucks to form platoons with neighboring trucks on the go. The question of whether the extra energy required to form a platoon (e.g. by having to momentarily drive faster) would be offset by the energy savings from temporarily being in a platoon is fascinating and could be seen as an extension of the proposed control method.

The eco-driving trajectories generated by solving the bus line problem discussed in Chapter 4 could be improved by refining the model used for predicting driving conditions. A stochastic model could for example be used to account for the variability of traffic. In addition, any knowledge of the upcoming phases of traffic lights could also be included in the model.

As for the bus network problem discussed in Chapter 5, the way it is currently formulated relies on one central assumption: it is assumed that bus routes are not overlapping. While this may constitute a good approximation in many cases, a future research direction would be to model route overlaps in order to account for bus networks with strongly overlapping lines. Doing so would require more detailed passenger models and would likely change the coupling structure of the resulting optimization problem. Finding suitable decomposition methods to treat this new problem might be challenging and is left for future research.

Real-time implementation

The next logical step to develop the work in this thesis would be to create an effective implementation of the prototype algorithms presented. Since all these algorithms are destined to be run in real-time or near real-time settings, their implementation for deployment on real systems would have to satisfy these speed requirements.

This goal is not unrealistic. With the progress of onboard computational capacity and the constant improvements of optimization algorithms, it is today possible to solve large-scale NLPs extremely fast with adapted solvers, as has been observed both in simulations and on real systems [60]. In fact, experimental results suggest that vehicle-to-vehicle communications might be more of a bottleneck than solving NLPs in some automotive applications, even when vehicles are very close to each other [61].

For the prototype algorithms presented in this thesis, significant computational speedups could likely be obtained by using libraries and solvers specifically designed for high-performance computing. For instance, the HPIPM software [51], which contains hardware-tailored linear algebra routines specifically designed for the real-time execution of MPC algorithms, would be a good environment in which to implement the proposed algorithms. Deploying a real-time iteration scheme [62] could then lead to additional runtime improvements and more frequent MPC updates.

A lot of research and implementation work would likely be needed to carry out the steps suggested above and achieve near real-time performances. If that were to be done, however, the algorithms proposed during the course of this thesis could be evaluated on real test systems, and perhaps one day deployed in real commercial systems where they would be contributing to their safe and energy-efficient operation.

References

[1] International Transport Forum, *ITF Transport outlook 2019*. OECD Publishing, 2019, `https://doi.org/10.1787/transp_outlook-en-2019-en` Accessed 2023-10-05.

[2] International Energy Agency, *Global EV outlook 2021*. OECD Publishing, 2021, `https://www.iea.org/reports/global-ev-outlook-2021` Accessed 2023-10-05.

[3] S. Pelletier, O. Jabali, J. E. Mendoza, and G. Laporte, "The electric bus fleet transition problem," *Transportation Research Part C: Emerging Technologies*, vol. 109, pp. 174–193, 2019.

[4] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," SAE technical paper, Tech. Rep., 2000.

[5] A. Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *Proc. IEEE 13th Intel. Transp. Syst. Conf. (ITSC)*, 2010, pp. 306–311.

[6] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 68–77, 2016.

[7] W. Levine and M. Athans, "On the optimal error regulation of a string of moving vehicles," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 355–361, 1966.

[8] D. Swaroop and J. K. Hedrick, “String stability of interconnected systems,” *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 349–357, 1996.

[9] W. B. Dunbar and D. S. Caveney, “Distributed receding horizon control of vehicle platoons: Stability and string stability,” *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2011.

[10] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K. H. Johansson, “Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency,” *IEEE Control Systems Magazine*, vol. 35, no. 6, pp. 34–56, 2015.

[11] 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.

[12] 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.

[13] 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.

[14] V. Turri, B. Besselink, and K. H. Johansson, “Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 12–28, 2016.

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

[16] 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.

[17] S. S. Perumal, R. M. Lusby, and J. Larsen, “Electric bus planning & scheduling: A review of related problems and methodologies,” *European Journal of Operational Research*, vol. 301, no. 2, pp. 395–413, 2022.

[18] C. H. Häll, A. Ceder, J. Ekström, and N.-H. Quttineh, "Adjustments of public transit operations planning process for the use of electric buses," *Journal of Intelligent Transportation Systems*, vol. 23, no. 3, pp. 216–230, 2019.

[19] M. Xylia, S. Leduc, P. Patrizio, F. Kraxner, and S. Silveira, "Locating charging infrastructure for electric buses in stockholm," *Transportation Research Part C: Emerging Technologies*, vol. 78, pp. 183–200, 2017.

[20] K. An, "Battery electric bus infrastructure planning under demand uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 572–587, 2020.

[21] D. McCabe and X. J. Ban, "Optimal locations and sizes of layover charging stations for electric buses," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104 157, 2023.

[22] X. Tang, X. Lin, and F. He, "Robust scheduling strategies of electric buses under stochastic traffic conditions," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 163–182, 2019.

[23] M. Wen, E. Linde, S. Ropke, P. Mirchandani, and A. Larsen, "An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem," *Computers & Operations Research*, vol. 76, pp. 73–83, 2016.

[24] Y. He, Z. Liu, and Z. Song, "Optimal charging scheduling and management for a fast-charging battery electric bus system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, p. 102 056, 2020.

[25] 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.

[26] D. Huang, Y. Wang, S. Jia, Z. Liu, and S. Wang, "A lagrangian relaxation approach for the electric bus charging scheduling optimisation problem," *Transportmetrica A: Transport Science*, pp. 1–24, 2022.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] K. Ampountolas and M. Kring, "Mitigating bunching with bus-following models and bus-to-bus cooperation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2637–2646, 2020.

[40] D. S. Naidu, *Optimal control systems.* CRC press, 2018.

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

[42] 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.

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

[44] 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.

[45] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods.* Athena Scientific, 2015.

[46] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," Stanford University, Tech. Rep., 2008.

[47] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications.* Springer Science & Business Media, 2006.

[48] M. Diehl and S. Gros, "Numerical optimal control," *Optimization in Engineering Center (OPTEC)*, 2011.

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

[50] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *Journal of optimization theory and applications*, vol. 99, no. 3, pp. 723–757, 1998.

[51] G. Frison and M. Diehl, "Hpipm: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[52] D. Axehill and M. Morari, "An alternative use of the riccati recursion for efficient optimization," *Systems & Control Letters*, vol. 61, no. 1, pp. 37–40, 2012.

[53] I. Nielsen, "Structure-exploiting numerical algorithms for optimal control," PhD thesis, Linköping University, 2017.

[54] G. Frison and J. B. Jørgensen, "Efficient implementation of the riccati recursion for solving linear-quadratic control problems," in *IEEE Inter. Conf. on Control Appl. (CCA)*, 2013, pp. 1117–1122.

[55] D. P. Bertsekas, *Dynamic programming and optimal control.* Athena Scientific, Belmont, MA, 2012, vol. 1.

[56] 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.

[57] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the American Control Conference, 2005*, IEEE, 2005, pp. 2676–2683.

[58] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management science*, vol. 50, no. 12_supplement, pp. 1861–1871, 2004.

[59] M. Gaudioso, "A view of lagrangian relaxation and its applications," *Numerical Nonsmooth Optimization: State of the Art Algorithms*, pp. 579–617, 2020.

[60] M. Vukov, S. Gros, G. Horn, *et al.*, "Real-time nonlinear mpc and mhe for a large-scale mechatronic application," *Control Engineering Practice*, vol. 45, pp. 64–78, 2015.

[61] R. Hult, M. Zanon, G. Frison, S. Gros, and P. Falcone, "Experimental validation of a semi-distributed sequential quadratic programming method for optimal coordination of automated vehicles at intersections," *Optimal Control Applications and Methods*, vol. 41, no. 4, pp. 1068–1096, 2020.

[62] M. Diehl, "Real-time optimization for large scale nonlinear processes," PhD thesis, University of Heidelberg, 2001.