Scalable Data Analytics Scalable Algorithms, Software Frameworks and Visualisation ICT-2013.4.2a

Project **FP7-619435 / SPEEDD**

Deliverable **D4.3**

Distribution **Public**

http://speedd-project.eu/

Decision Making III

C. Canudas de Wit (CNRS), F. Garin (CNRS), P. Grandinetti (CNRS), M. Keller (ETH), J. Lygeros (ETH), C. Ramesh (ETH), M. Schmitt (ETH)

Status: Final version. (Version 2.0)

October 2016

Project

Project ref.no.	FP7-619435
Project acronym	SPEEDD
Project full title	Scalable ProactivE Event-Driven Decision making
Porject site	http://speedd-project.eu/
Project start	February 2014
Project duration	3 years
EC Project Officer	Stefano Bertolo

Deliverable

Deliverabe type	report
Distribution level	Public
Deliverable Number	D4.3
Deliverable title	Decision Making III
Contractual date of delivery	M32 (September 2016)
Actual date of delivery	October 2016
Relevant Task(s)	WP4 / Tasks 4.1, 4.2, 4.3, 4.4 & 4.5
Partner Responsible	ETH
Other contributors	ETH,CNRS
Number of pages	138
Author(s)	C. Canudas de Wit (CNRS), F. Garin (CNRS), P. Grandinetti (CNRS), M. Keller (ETH), J. Lygeros (ETH), C. Ramesh (ETH), M. Schmitt (ETH)
Internal Reviewers	
Status & version	Final version.
Keywords	decision making; robust control; distributed control; optimality

Contents

List of Tables

Executive Summary

The purpose of this document is to describe the third and final version of the decision making component, which provides a set of proactive, event-driven decision-making tools for the SPEEDD use cases.

SPEEDD (Scalable ProactivE Event-Driven Decision making) has developed a system for proactive real-time decision-making. The decisions will be triggered by events, which are processed on-the-fly by monitoring extremely large noisy data streams from diverse geographical locations. Complex events that describe the state of the system are detected or forecasted and are used to find the best decisions in accordance with user-defined optimality criteria. The goal of WP4 (Real-Time Event-Based Decision-Making Under Uncertainty) is to provide innovative techniques for proactive event-driven decision-making to optimize the specified objective, while at the same time guaranteeing probabilistic constraint satisfaction or providing worst-case guarantees even in the presence of inherent, problem-specific uncertainty in the detected events. This goal is to be met by exploring different types of decision making algorithms, ranging from worst-case real-time decision making to randomized, scenario-based decision-making.

This report provides a complete and concise description of the final version of the decision making algorithms developed in WP4. As such, we do not only list the incremental changes since the last version, but also include previously described algorithms if they are relevant to the final version. As explained in past deliverables, the decision making algorithms are use-case specific. In the traffic use-case, decision making takes a prominent role in comparison to event processing since even a complete and uncertainty-free characterisation of the state of the traffic network does not render the question of optimal decisions trivial. In this report, we extensively describe algorithms that allow for optimal or practically efficient decision making for traffic control, while strictly following the distributed, scalable paradigm of SPEEDD. In the use-case of credit card fraud detection, the question of reliable and accurate detection of fraud is predominant and decision making is reduced to a supporting role, providing additional detection algorithms or addressing the question of how to automate the decision tasks that in current state-of-the-art solutions are left to human experts.

The derivation and implementation of suitable decision making algorithms is considered mostly complete with this deliverable. Future work includes extensive testing of the developed algorithms in real-world applications or realistic test-cases, depending on how a realistic but also controlled test environment can best be realized. This evaluation is not work-package-specific but concerns the entire project and the complete, integrated software. Therefore, future modification to the decision making algorithms are expected to be restricted to adapting the interface with other components, if necessary, or to the tuning of the underlying algorithms according to the needs of the end users.

1 Introduction

1.1 History of the Document

Version	Date	Author	Change Description
0.1	10/10/2016	Marius Schmitt (ETH)	First Draft
0.2	20/10/2015	Marius Schmitt (ETH)	Added contributions by various authors.
1.0	24/10/2015	Marius Schmitt (ETH)	Contents completed, ready for internal revision.
2.0	31/10/2015	Marius Schmitt (ETH)	Changes according to internal revision.

1.2 Purpose and Scope of the Document

The purpose of this document is to describe the third and final version of the Decision-Making algorithms. We describe decision making algorithms for freeway traffic control, inner-city traffic control and credit card fraud detection. The focus of this report is on the theoretical derivations of the decision making algorithms. Typically, the description of each algorithm encompasses a description of the underlying model, objective and assumptions, the description of the actual algorithm and theoretical proof of properties like stability, robustness or optimality.

The decision making algorithms are use-case specific and thus, this report is divided into two main chapters. The structure of each of these chapters is detailed in the respective introduction. There, we also provide a summary of the work package tasks and indicate the individual contributions that address each of the tasks.

1.3 Relationship with Other Documents

This document presents the final version of the decision making algorithms and extends and completes the previous decision making deliverables **D4.1** and **D4.2**. As the final report, this document is self contained, that is, relevant parts of the previous documents that are part of the final decision making algorithms are repeated in this report. The focus of this report is on algorithms and the associated theory. Implementation and interaction with other components are briefly discussed, but a more detailed description can be found in the deliverable concerning the final, integrated prototype **D6.7**. The algorithms described in this document are use-case specific. The system requirements for the Proactive Traffic Management use case are described in **D8.1** and the ones for the Proactive Credit Card Fraud Management are described in **D7.1**. The respective final user evaluations will be presented in **D7.4** and **D8.6**, which will be conducted using the algorithms described in this deliverable as part of the final, integrated prototype.

Results presented in this document have been published or submitted for publication in Schmitt et al. (2015), Grandinetti et al. (2015b), Ramesh et al. (2015), Grandinetti et al. (2015c), Schmitt et al. (2016b) and Schmitt et al. (2016a).

2

DM for Traffic

This chapter describes the third and final version of the **Decision Making** module for the traffic use case. In accordance with the objectives of SPEEDD, we focus on distributed, scalable and event-driven algorithms. The initial architecture concept depicted in Figure 2.1 shows the module implemented in a distributed manner. The final internal structure of the DM component is depicted in Figure 2.2. We chose a *hierarchical* control approach, with a low-level control layer that provides local feedback in an entirely decentralized and therefore inherently scalable manner and a high-level coordination layer, which aims to coordinate the system. This is achieved by appropriately adapting the control targets for the low-level controllers such that the overall system optimizes a central control objective. The DM module is thus composed of three main parts:

(i) *State Estimation and System Identification* The main purpose of this component is the mitigation of the uncertainty in events and measurements. Sources of uncertainty fall into one of three categories: First, some forecasted, complex events might be uncertain. Quantification of this type of uncertainty is provided by the CEP module and falls out of the scope of this report. Second, measurements, e.g. of traffic flows, velocities or densities are typically noisy. We rely on standard state estimation techniques like Kalman filtering to obtain accurate estimations of the true network state. Third, traffic dynamics are inherently uncertain due to varying environmental conditions or arbitrary driver decisions. Quantification of this type of uncertainty is directly handled by the DM module and Sections 2.1.1 and 2.1.2 describe the chosen approaches.

(ii) *Low-level Control* The low-level control acts in a decentralized manner, that is, it chooses local control actions (e.g. traffic light phases) depending on local variables (e.g. flow and occupancy), estimates of which are provided by the State Estimation. It also directly uses the stochastic model identified by the system identification component. Its control target, i.e. the desired state, is modified externally by the coordination algorithms described next. The corresponding algorithms are described in Sections 2.2.1, 2.2.2 and 2.2.3.

(ii) *High-level coordination* The coordination algorithms ensure that the overall system behavior optimizes a system-wide objective. They do not decide on control actions directly but achieve their goal indirectly by modifying the control targets of the low-level feedback laws. These algorithms are also implemented in a distributed manner and rely on communication via events to realize

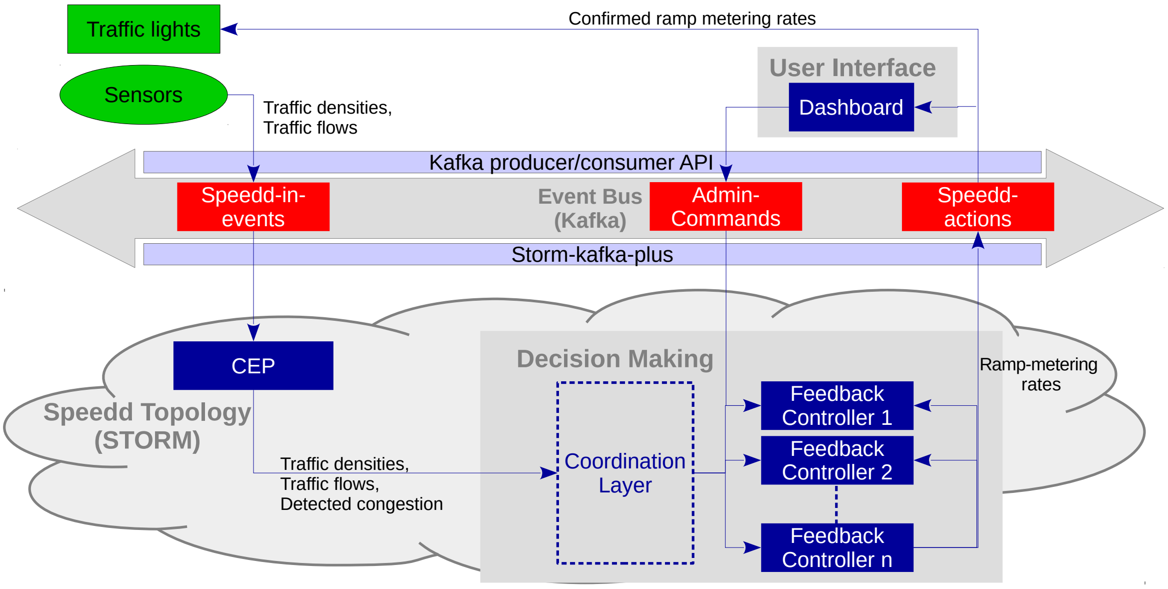


Figure 2.1: Integration concept of DM into the SPEEDD prototype in previous versions (D4.1 and D4.2).

coordination. They also provide the option to modify the behavior externally via administrator command from some human operator, e.g. as an emergency overwrite of the automatic control. The corresponding algorithms are described in Sections 2.3.2 and 2.3.3. A result concerning a particular scenario, in which no coordination is required to achieve optimality, is described in Section 2.3.1.

The scope of this report encompasses all results regarding Decision Making (WP 4) from the entire 30 month duration of the project. This includes the intermediate results presented in the first (Deliverable 4.1) and second (D 4.2) version of the deliverables describing decision making algorithms, in so far as they did not become outdated by later results. This work package is divided into five distinct tasks:

T4.1 *Event-Driven Proactive Decision-Making* This task is concerned with developing criteria for determining how and when decisions should be selected in response to specific events recognized or forecasted by the methods developed in WP3. A key finding is that in a fast-changing environment such as road traffic, small corrective decisions made with high frequency are vital in stabilizing desirable operating conditions. This conclusion is reflected in the decomposition of the DM algorithms into low-level *feedback laws* and high-level *coordination algorithms* Sections 2.2.1 and 2.2.3 both describe low-level, local feedback laws that are meant to be executed periodically with high frequency. These feedback laws are extended with distributed coordination strategies (Section 2.3.3) which describe the communication pattern between local controllers.

T4.(2,3,4) *Decision-Making Methods* Tasks T4.2, T4.3 and T4.4 are all concerned with decision making under uncertainty, depending on which particular approach is chosen. However, some of the developed algorithms are hard to categorize according to this scheme, since they achieve mitigation of system or disturbance uncertainty by virtue of feedback and not by explicitly modeling uncertainty in an optimization problem. This applies to the algorithms described in Sections 2.2.1, 2.2.3 and 2.3.3.

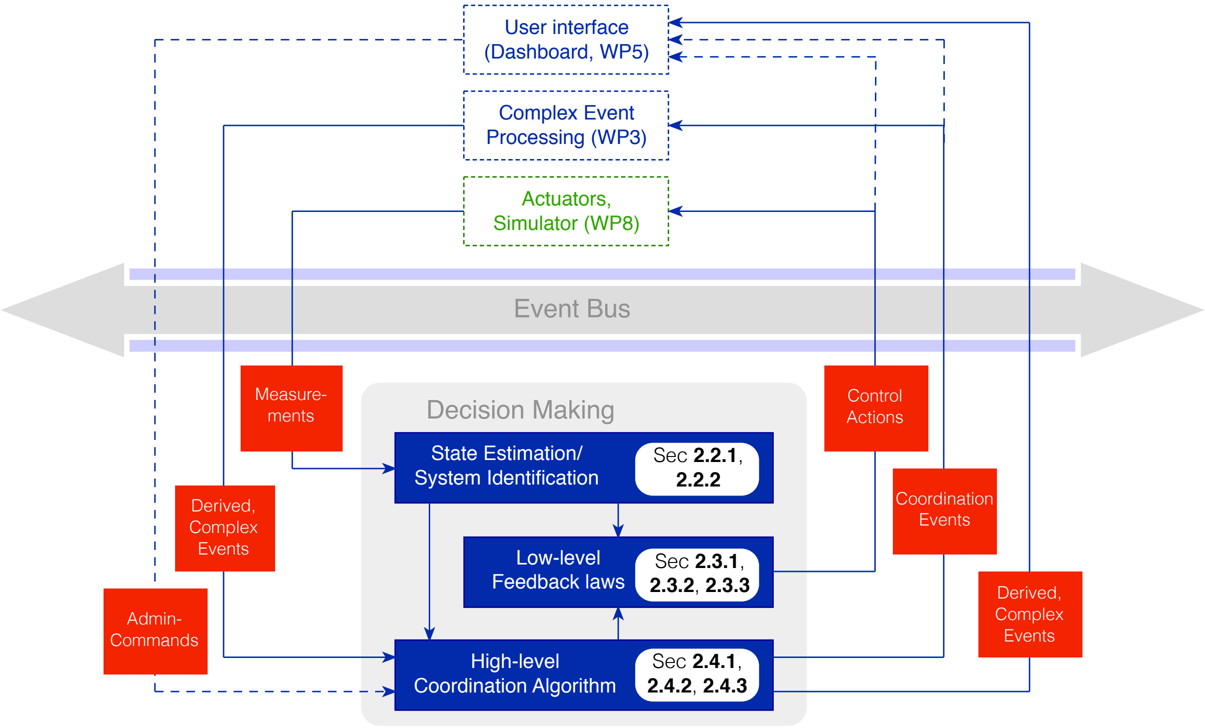


Figure 2.2: Final concept of the Decision making module, using a distributed, scalable structure. The sections which describe the results related to the respective (sub-) components of DM are indicated in the figure.

T4.2 *Worst-Case Decision-Making Methods* In this task, optimization methods for computing decisions in a worst-case or robust manner are considered. This paradigm is mainly reflected in the algorithm described in Section 2.2.1, which is provably robust to a wide range of traffic demand uncertainty.

T4.3 *Stochastic Decision-Making Methods* The methods developed in this task are based on an optimization over expected outcomes given an assumed model of the time-evolution of the system and some data-based model of the uncertainty. This approach is chosen in the data-driven quantification of uncertainty in traffic models via Gaussian Processes described in Sections 2.1.1 and the control strategy building upon these models outlined in Section 2.1.2.

T4.4 *Randomised Decision-Making Methods* For those cases in which the models have very little exploitable internal structure, an approach entirely different from those developed in Tasks 4.2 and 4.3 is required. For such cases, Task 4.4 proposes to use optimization methods based on randomized optimization. This approach should be considered as a method of last resort. The algorithm described in Section 2.3.2 describes a generic approach for randomized decision making in the presence of uncertainty and its application to freeway ramp metering. In some sense, one could argue that the method described in 2.3.3 also shares elements of this approach, as its convergence to an approximate solution of a non-convex optimization problem ultimately depends on the (at least partly random) choice of the starting point of the coordination algorithm.

T4.5 *Determining the Right Level of Automation* This task considers the application of the results of Tasks 4.1-4.4 in enabling various levels of user decision support and determines which amount of

Section	Version	T4.1	T4.2	T4.3	T4.4	T4.5	Freeway	Inner City
2.1.1	v3			✓			✓	✓
2.1.2	v3			✓			✓	✓
2.2.1	v2	(✓)	✓			(✓)	✓	
2.2.2	v3					✓	✓	
2.2.3	v1	(✓)	(✓)	(✓)		(✓)		✓
2.3.1	v1		✓				✓	
2.3.2	v2				✓		✓	
2.3.3	v2	✓	(✓)	(✓)	(✓)			✓

Table 2.1: Overview over scientific results presented in this report and their main focus ✓and partial applicability (✓) to the individual tasks and use-cases in the SPEEDD project. Also, the version of the DM-deliverable (D4.1, D4.2 or D4.3) in which the respective algorithm was first described is stated.

automation is best suited for the user's needs. Similar to the conclusion mentioned for T4.1, local feedback laws as described in Sections 2.2.1 and 2.2.3 are meant to be executed fully automatically since the frequency and the nature of the required decision (determining optimal values for continuous variables like green-phase time with high precision) make them unsuitable for a human operator. However, the implemented algorithms allows a human operator to override decisions. This is required for safety reasons but effectively disables large parts of the algorithms.

A complete overview over all results presented in this report is given in Table 2.1.

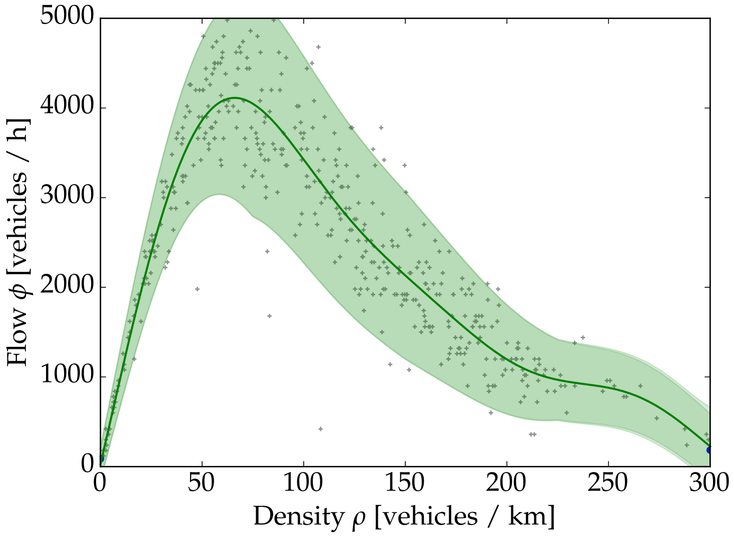


Figure 2.3: Data sample of real-world density-flow data pairs. These data can only poorly be represented by a single, deterministic function but a stochastic model, here depicted via its mean and 90% confidence interval, seems suitable.

2.1 Uncertainty in Traffic

The main purpose of this section is to address the question of handling the uncertainty present in traffic control problems. The importance of considering the effects of uncertainty in traffic control problems can easily be exemplified with real-world data as depicted in Figure 2.3. These data clearly cannot be described by a single, deterministic function but require a stochastic model instead[1]. Sources of uncertainty in traffic control fall into one of three categories:

(i) Sensor measurement are almost always corrupted by noise. For example, occupancy of a loop detector over an interval might not exactly correspond to the traffic density because of inhomogeneous traffic conditions or measured velocities might differ from average velocities if cars are accelerating or slowing down. This type of uncertainty is usually modeled as additive noise such that estimates of the true values can be obtained by a Kalman filter. This is the default approach and therefore does not require further explanation in this report.

(ii) Forecasted, convex events are uncertain in the sense that there is no guarantee that they will actually occur, but instead, the confidence in there occurrence is quantified in the event attributes. The quantification of the uncertainty is carried out by the Complex Event Processing module and out of the scope of this report. Section 2.1.2 describes how to modify control algorithms, e.g. the low-level feedback laws presented in Section 2.2 to take this uncertainty (and the model uncertainty described in (iii)) into account.

(iii) Traffic dynamics are inherently uncertain due to varying environmental conditions or arbitrary driver decisions. This type of uncertainty is hard to quantify a-priori, since it might even by time-varying in case of changing operation conditions. Therefore, we use a model-free, data-driven approach to estimate this uncertainty during operation based on the measured data. In Section 2.1.1, we describe a suitable approach based on Gaussian Processes.

[1] Note that this figure does not rule out a deterministic model outright, since dependencies on other variables might exist that explain the variance in the data in a deterministic fashion However, even higher order road traffic models, or models considering additional independent variables, do not reduce the unexplained noise in real-world traffic data significantly.

2.1.1 Identification of Uncertain Traffic Dynamics

M. Keller

We investigate how Gaussian Processes (GPs) can be used to represent and learn the traffic dynamics of a freeway. The proposed approach is data-driven and builds on the popular macroscopic Cell Transmission Model where we also model process noise and uncertainty. There are two main challenges: first, process noise in traffic dynamics seems to be poorly representable by stationary kernels and second, the abundance of data makes standard GP learning intractable. We present two approaches for the representation and estimation of non- stationary process noise in GPs: learning parametric functions by maximizing the data likelihood with a gradient-descent method, and learning process noise directly from data with an approximative EM algorithm. To keep computations tractable, we use a combination of the Hierarchical Mixture of Experts model (HGP) and reduce the data set size based on data density. We further introduce a simple approach for the interpolation of traffic data between sensors. Our work is the first step towards a robust control system for ramp metering based on reinforcement learning with Gaussian Processes.

Introduction

It is commonly observed on freeways that the maximum capacity is achieved at a specific vehicle density, the so-called critical density. If it is exceeded, speed (and with it, capacity) drops rapidly and congestion usually ensues. To mitigate this effect, ramp-metering systems that regulate the rate at which vehicles may enter a freeway have been installed with success around the world. They are usually implemented as simple traffic lights at onramps that let pass only one vehicle per green phase. Designing a ramp-metering system is a delicate task and it is desirable to have reliable models that predict the evolution of traffic over time such that different control strategies can be evaluated. A rough distinction can be made between microscopic and macroscopic traffic models where the former focus on the simulation of individual vehicles while the latter consider traffic systems as a whole and express state in abstract quantities such as densities and flows. We consider only macroscopic models. Macroscopic models have a long history with significant work dating back to fifties and have been improved on and successfully applied since. Typical macroscopic models are based on a small number of parameters such as the average speed of vehicles in light traffic, the maximum capacity, or the typical wave speed in traffic jams, quantities that can be easily estimated. While these parametric approaches are convenient and intuitive, they are also in a certain sense limited due to their simplicity. Nowadays, large amounts of traffic data can usually be obtained comparatively cheaply which suggests using machine learning techniques for developing more sophisticated data-based models. Gaussian Processes (GPs) can be interpreted as distributions over functions with a continuous domain and as such are well-suited for the purpose of learning traffic dynamics. They are nonparametric and therefore do not rely on strong prior assumptions which means that they can truly learn the structure that is present in the data. GPs are based on Bayesian statistics which allows us to include prior knowledge (and the uncertainty we have about it) in the our model. With GPs, one does not obtain one single model but a probability density distribution on models. As we will see, GPs can be extended in a natural way to represent the process noise that is inherent to traffic systems. In short, GPs can express modelling and process uncertainty which is crucial for designing robust control algorithms. The Probabilistic Inference for Learning Control framework (PILCO) Deisenroth and Rasmussen (2011); Deisenroth et al. (2015) provides us with a suite of algorithms for optimally controlling a system with a given performance function. Thus, there are control algorithms that take full advantage of the properties of GPs, in particular of the uncertainty information. The goal is to investigate how GPs can be used to represent the dynamics of a traffic system and how these can be learned from data. This report is structured as follows. Section 2 is dedicated to Gaussian Processes where we focus on the representation and estimation of non-stationary process noise. In Section 3, we give a brief introduction to macroscopic traffic modelling and discuss how we can represent traffic dynamics with Gaussian Processes. After that, in Section 4, we illustrate the traffic dynamics that are learned from data and highlight how we addressed some of the difficulties that we encountered. Finally, we recapitulate and discuss our findings in Section 5.

Gaussian Processes for Regression

Gaussian Processeces (GPs) are a generalization of multivariate Gaussian distributions (MVNs) to any finite number of stochastic variables. An MVN is fully specified by its mean vector $\boldsymbol{\mu}$ and its covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right). \tag{2.1}$$

Similarly, a GP is fully specified by its *mean function* $m(\boldsymbol{x})$ and its *kernel function* $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. It is denoted by

$$\boldsymbol{f} \sim \mathcal{GP}(m, k). \tag{2.2}$$

The GP above specifies the following MVN over collections of vectors $\boldsymbol{X} = \big(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\big)$ of any length n:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\begin{bmatrix} m(\boldsymbol{x}_1) \\ m(\boldsymbol{x}_2) \\ \vdots \\ m(\boldsymbol{x}_n) \end{bmatrix}}_{=:\boldsymbol{\mu}}, \underbrace{\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & k(\boldsymbol{x}_n, \boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}}_{=:\boldsymbol{K}}\right) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K}). \tag{2.3}$$

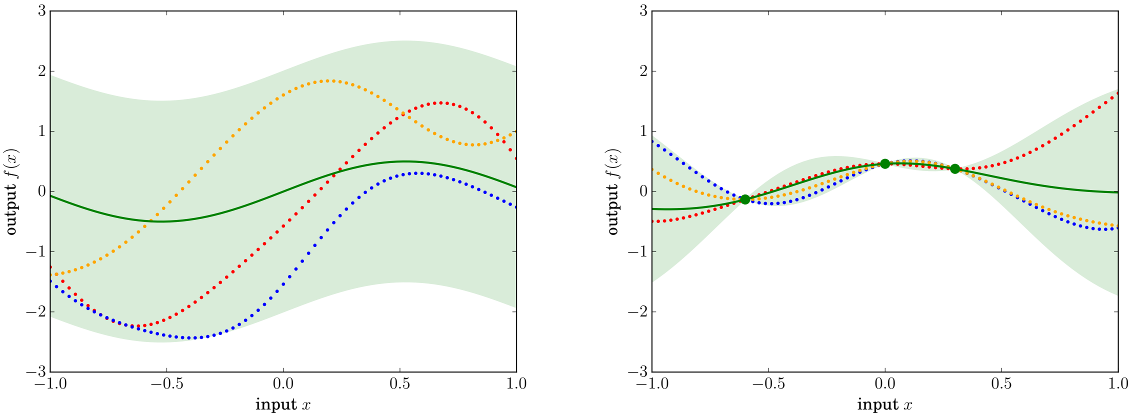
This leads us directly to the definition of a Gaussian Process:

Definition 2.1.1. *A* Gaussian Process (GP) *is a collection of stochastic variables, any finite number of which have a joint Gaussian distribution.*

GPs can be understood as distributions over functions $\mathbb{R}^D \rightarrow \mathbb{R}$. This is illustrated in Figure 2.4 (a) for $D = 1$. The green graph corresponds to the mean function $m(\boldsymbol{x})$ while the shadowed area represents the pointwise variance expressed by the kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The three colored dotted lines are sets of 75 points drawn at random from the distribution induced by the GP. Note that the sampled functions are all very smooth. The properties of the functions are strongly influenced by the choice of the kernel function as will be discussed later.

Figure 2.4 (b) illustrates how GPs can be used for regression. The three bold green dots indicate noise-free observations. Applied to the prior GP in Panel (a), they induce the posterior GP illustrated in Panel (b).

In the remainder of this section, we will look at GP regression in more detail and derive the posterior (for which an analytical formulation exists). We will also discuss some important kernel functions and have a look at automatic methods for learning the hyperparameters of a kernel.



(a): prior

(b): posterior

Figure 2.4: Illustration of regression with GPs. Panel (a) shows the prior GP, Panel (b) the posterior GP conditioned on three noise-free observations (the green dots). In both panels, the green line indicates the pointwise mean and the shaded area corresponds to the 95% confidence area (minus and plus two times the standard deviance). The dotted lines show three samples drawn at random from the prior and the posterior distributions.

Derivation of the Posterior This section closely follows Rasmussen (2004). We start from our prior GP:

$$\boldsymbol{f} \sim \mathcal{GP}(m, k). \tag{2.4}$$

Now suppose we are given a training set $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, $\boldsymbol{y} = (y_1, \ldots, y_N)$ where $y_i = f(\boldsymbol{x}_i)$ (i.e., we assume absence of noise for now). We want to compute the joint posterior distribution of the function values $y_i^* = f(\boldsymbol{x}_i^*)$ evaluated a test set $\boldsymbol{X}_* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_m^*)$. From the definition of a Gaussian Process, we know that the joint distribution of data and evaluation points is an MVN:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^\top & \boldsymbol{K}_{**} \end{bmatrix} \right) \tag{2.5}$$

where $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X})$ is $N \times N$, $\boldsymbol{K}_* = k(\boldsymbol{X}, \boldsymbol{X}_*)$ is $N \times N_*$, $\boldsymbol{K}_{**} = k(\boldsymbol{X}_*, \boldsymbol{X}_*)$ is $N_* \times N_*$, $\boldsymbol{\mu} = m(\boldsymbol{X})$ is an N-vector, and $\boldsymbol{\mu}_* = m(\boldsymbol{X}_*)$ is an N_*-vector. The formula for conditioning a joint Gaussian distribution is:

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right) \implies \boldsymbol{x}_1 \mid \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}) \tag{2.6}$$

If we apply this to (2.5), we immediately get analytical expressions for the posterior distribution of $\boldsymbol{y}_*$:

$$\mathrm{p}(\boldsymbol{y}_* \mid \boldsymbol{X}_*, \mathcal{D}) \sim \mathcal{N}(\boldsymbol{\mu}_* + \boldsymbol{K}_*^\top \boldsymbol{K}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}), \boldsymbol{K}_{**} - \boldsymbol{K}_*^\top \boldsymbol{K}^{-1}\boldsymbol{K}_*). \tag{2.7}$$

Handling Measurement and Process Noise

The posterior in (2.7) assumes absence of any kind of noise and therefore, the GP only assigns positive probability densities to functions that interpolate the data points exactly. Fortunately, it is straightforward to incorporate both measurement and process noise as long as they are normally distributed. We model process noise ϵ_p and measurement noise ϵ_m as independent additive quantities:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_m + \epsilon_p.$$

Both ϵ_m and ϵ_p are normally distributed with zero mean. Their variance may be non-stationary, that is, it may depend on $\boldsymbol{x}_i$. We denote by $s_m : \mathbb{R}^D \to \mathbb{R}$ and $s_p : \mathbb{R}^D \to \mathbb{R}$ the variance functions of ϵ_m and ϵ_p, respectively, i.e., $\epsilon_m(\boldsymbol{x}_i) \sim \mathcal{N}(0, s_m(\boldsymbol{x}_i))$ and $\epsilon_p(\boldsymbol{x}_i) \sim \mathcal{N}(0, s_p(\boldsymbol{x}_i))$. Thus, the total error caused by measurement and process noise is $\epsilon_m(\boldsymbol{x}_i) + \epsilon_p(\boldsymbol{x}_i) \sim \mathcal{N}(0, s_m(\boldsymbol{x}_i) + s_p(\boldsymbol{x}_i))$ and y_i is normally distributed:

$$y_i \sim \mathcal{N}(f(\boldsymbol{x}_i), s_m(\boldsymbol{x}_i) + s_p(\boldsymbol{x}_i)). \tag{2.8}$$

If we define the diagonal matrices

$$\boldsymbol{K}_p := \begin{bmatrix} s_p(\boldsymbol{x}_1) & & \\ & \ddots & \\ & & s_p(\boldsymbol{x}_N) \end{bmatrix}, \text{ and } \boldsymbol{K}_m := \begin{bmatrix} s_m(\boldsymbol{x}_1) & & \\ & \ddots & \\ & & s_m(\boldsymbol{x}_N) \end{bmatrix}, \tag{2.9}$$

we can update the Gram matrix with process and measurement noise: $\boldsymbol{C} := \boldsymbol{K} + \boldsymbol{K}_p + \boldsymbol{K}_m$. $\boldsymbol{K}_p$ and $\boldsymbol{K}_m$ are symmetric and positive definite, hence they are valid covariance functions. Since the sum of covariance functions is a covariance function, we can conclude that $\boldsymbol{C}$ is a valid covariance function. Therefore, we can rewrite (2.7) into our main formulation of the posterior. It is important to notice that the posterior variance only encompasses uncertainty about the function, that is, process noise has to be re-added explicitly [2]:

$$\begin{aligned} \mathrm{p}(\boldsymbol{y}_* \mid \boldsymbol{X}_*, \mathcal{D}) &\sim \mathcal{N}(\boldsymbol{\mu}_\mathcal{D}, \boldsymbol{K}_\mathcal{D}) \text{ where} \\ \boldsymbol{\mu}_\mathcal{D} &= \boldsymbol{\mu}_* + \boldsymbol{K}_*^\top \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}), \\ \boldsymbol{K}_\mathcal{D} &= \boldsymbol{K}_{**} + \boldsymbol{K}_{p,**} - \boldsymbol{K}_*^\top \boldsymbol{C}^{-1} \boldsymbol{K}_*. \end{aligned} \tag{2.10}$$

The difference between process and measurement noise is illustrated in Figure 2.5. Both images show a posterior GP conditioned on the data points. The prior GP in the left image assumes stationary measurement noise $s_m(\cdot) = 0.8$ and no process noise while the prior GP in the right image assumes a combination of measurement and process noise ($s_m(\cdot) = 0.2$ and $s_m(\cdot) = 0.6$). Note that both GPs assume the same amount of total noise $s_m(\cdot) + s_p(\cdot) = 0.8$. Consequently, both posterior GPs have the same mean but the variance is different due to the different noise models. Since the left GP assumes absence of process noise, its posterior variance only expresses uncertainty about the true mean and this uncertainty decreases with the number of data points which is well visible in the image. Differently, the right GP assumes process noise which is inherent to the function and does not disappear with more data. Note that it is not possible to distinguish between measurement and process noise just from the data.

Learning the Kernel Parameters We just discussed that we can update the prior GP with given data and compute the posterior GP. The choice of the prior GP is of crucial importance for the quality (and, truthfulness) of the posterior but unfortunately reliable knowledge about it is not usually available, or it is difficult to obtain. For the sake of an example, let us introduce a first kernel function:

[2] We interpret the posterior as a generative model that includes Gaussian process noise – for each input $\boldsymbol{x}_i$, it gives us a distribution over the output y_i. This is different from most formulations in the literature where GPs are used to learn functions that are not affected by process noise.

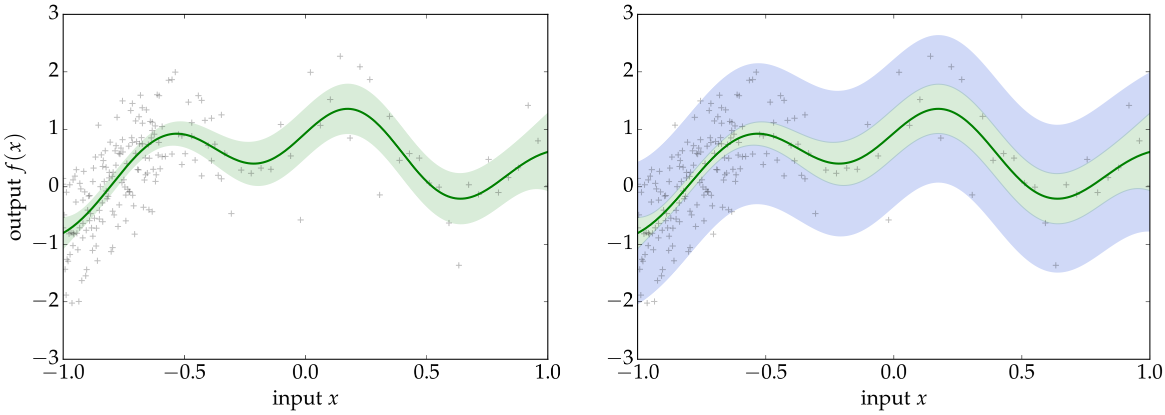
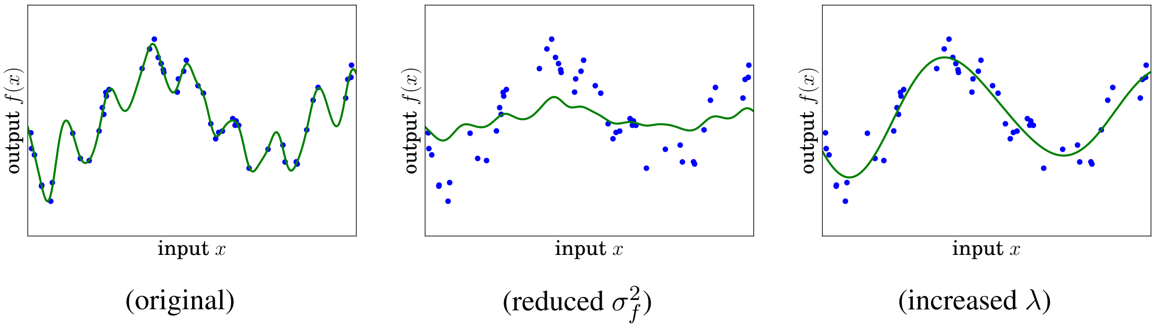


Figure 2.5: Fitting a GP on data with process noise. The data points in the plots were drawn from an artificial generative model. In the left plot, a GP is fitted assuming only measurement noise, in the right plot, both measurement and process noise. Both GPs use a squared exponential kernel. The posterior of the left GP therefore only quantifies the uncertainty we have about the mean function (green shade) while the right GP also quantifies the uncertainty stemming from process noise (blue shade).



(original) (reduced σ_f^2) (increased λ)

Figure 2.6: Posterior mean (green) obtained by the SE kernel for varying kernel parameters. The value of σ_f affects the amplitude of oscillations while λ has an influence on their frequency.

$$k(x_i, x_j) = \sigma_f^2 \exp\left[-\frac{1}{2}\frac{||x_i - x_j||^2}{\lambda^2}\right] \tag{2.11}$$

This is the *squared exponential kernel* that is defined only for one-dimensional inputs. All example GPs so far used this kernel. It expresses that data points that are close together in input space are highly correlated and that this correlation decreases with distance. There are two parameters that we have to choose here: λ and σ_f^2. The impact of changing the value of those two is illustrated in Figure 2.6. Generally speaking, increasing λ leads to slower oscillations of the predicted mean while increasing σ_f^2 increases the amplitude of those oscillations.

Just from the data, it is not possible to decide which of the predictions in Figure 2.6 is best. All three GPs above are valid models give that the prior is justifiable. If no or only little reliable prior information is available about the function to be learned, it is often a good idea to choose a prior GP that leads to a posterior that explains the data well. In the following, we will discuss two ways of automatically estimating the hyperparameters of the prior GP: the *maximum likelihood estimate (MLE)* and the *maximum a posteriori estimate (MAP)*. Automatically estimating the hyperparameters is referred to as *training the GP*.

Maximum Likelihood Estimate

The *maximum likelihood estimate (MLE)* of the hyperparameters of a GP is defined to be the set of hyperparamters that make the training data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ most likely:

$$\hat{\boldsymbol{\theta}}_{MLE} = \max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) =: \max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{D}|\boldsymbol{\theta}). \tag{2.12}$$

$\mathrm{p}(\mathcal{D}|\boldsymbol{\theta})$ is referred to as the *evidence* or *marginal likelihood*. Since probabilities are non-negative and the logarithm is a monotonous function, one usually maximizes over the logarithm of the evidence which avoids underflows in numerical computations:

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}) := \ln \mathrm{p}(\mathcal{D}|\boldsymbol{\theta}). \tag{2.13}$$

$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta})$ is called the *log-evidence* or *log-marginal likelihood*. From the definition of GPs, we know that the data points in $\mathcal{D}$ are joint Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{C}$ and can consequently write:

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{C}| - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}) - \frac{N}{2}\ln(2\pi). \tag{2.14}$$

Finding the MLE requires us to solve a highly non-convex optimization problem. In general, there is no guarantee that the optimal solution can be found, but there are gradient-based methods that efficiently find local minima.

Maximum A Posteriori Estimate

MLE provides us with an automated way of estimating the hyperparameters of a kernel. Its main disadvantage is that we ignore any knowledge we might have about those parameters. Say we do not know the exact value of measurement noise, but have good reason to assume that it is stationary and we roughly know it magnitude. In this case, we can define a *hyperprior* (a prior for a hyperparameter) for it.

Given hyperpriors, we can compute the posterior likelihood of the hyperparameters conditioned on our training data:

$$\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}, \mathcal{K}) = \frac{\mathrm{p}(\mathcal{D}|\boldsymbol{\theta}, \mathcal{K})\,\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{D}|\mathcal{K})} \propto \mathrm{p}(\mathcal{D}|\boldsymbol{\theta}, \mathcal{K})\,\mathrm{p}(\boldsymbol{\theta}) \tag{2.15}$$

The *maximum a posteriori estimate (MAP)* is defined to be the most likely set of hyperparameters conditioned on the data $\mathcal{D}$:

$$\hat{\boldsymbol{\theta}}_{MAP} = \max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{D}|\boldsymbol{\theta},\mathcal{K})\,\mathrm{p}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \ln\left(\mathrm{p}(\mathcal{D}|\boldsymbol{\theta},\mathcal{K})\,\mathrm{p}(\boldsymbol{\theta})\right) = \max_{\boldsymbol{\theta}} \left[\mathcal{L}(\mathcal{D}|\boldsymbol{\theta},\mathcal{K}) + \sum_k \ln \mathrm{p}(\theta_k)\right]. \tag{2.16}$$

Note that the MAP estimate with uniform priors over $\mathbb{R}$, i.e. $\mathrm{p}(\theta_k) = \mathcal{U}[-\infty,\infty]$, is equivalent to the MLE. In general, we can set $\mathrm{p}(\theta_k) = 1 = const.$ if we do not want to define a hyperprior for hyperparameter θ_k. It is not a valid probability density function but this does not matter for the MAP optimization problem. MAP allows us to incorporate the pre-knowledge we might have about our hyperparamters but still finds exact values for us. It is less prone to overfitting than MLE and is more justified from a Bayesian point of view; however, in practice it can be very difficult to specify hyperpriors with confidence because the interactions between hyperparameters can be highly nonintuitive and one should therefore proceed with care.

Solving the Optimization Problem

There is no general analytical solution to finding the optimal parameters in the sense of MLE or MAP. Fortunately, we can compute the derivative of the log-evidence w.r.t. to the hyperparameters and can therefore apply gradient-based optimization methods such as conjugate gradients or BFGS. Using the well known identities $\partial \ln|A|/\partial x = \mathrm{trace}\left(A^1 \cdot \partial A/\partial x\right)$ and $\partial A^{-1}/\partial x = -A^{-1} \cdot \partial A/\partial x \cdot A^{-1}$, we get

$$\frac{\partial \mathcal{L}}{\partial \theta_m}(\mathcal{D}|\boldsymbol{\theta},\mathcal{K}) = -(\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1}\frac{\partial \boldsymbol{\mu}}{\partial \theta_m}, \tag{2.17}$$

$$\frac{\partial \mathcal{L}}{\partial \theta_k}(\mathcal{D}|\boldsymbol{\theta},\mathcal{K}) = -\frac{1}{2}\,\mathrm{trace}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\theta_k}\right) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\theta_k}\boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}), \tag{2.18}$$

where θ_m and θ_k are hyperparameters of the mean and covariance functions, respectively. When computing the MAP, one additionally has to add the derivatives of the hyperpriors.

The MAP and MLE estimates of the kernel parameters are highly nonconvex optimization problems and it is in general not possible to determine whether the global optimum has been found. One can use random restarts or methods such as simulated annealing. However, optimizing the MLE or MAP is the limiting factor that quickly renders GPs intractable since the inverse of the Gram matrix has to be recomputed for each set of hyperparameters that is considered (which is in $O(N^3)$). We will later discuss two approaches to reduce the computation burden: the HGP model and density-driven subsampling of the data.

Matrn Class Kernels The choice of the kernel function for a GP is of crucial importance because it largely predetermines the properties of the functions that can be represented. In this thesis, we concentrate on one specific class of kernels, the *Matrn kernels*. It is one of the most popular classes of kernels in machine learning and is applicable to a large variety of problems. A general Matrn kernel has the following form:

$$k_\nu(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}r\right)^\nu B_\nu\left(\sqrt{2\nu}r\right) \tag{2.19}$$

where $\Gamma(\cdot)$ is the gamma function and $B_\nu(\cdot)$ is the modified Bessel function of the second kind and $r = ||\boldsymbol{x}_i - \boldsymbol{x}_j||$. We use the highly flexible Mahalanobis distance with a diagonal matrix $\boldsymbol{\Lambda}^{-1} = \mathrm{diag}[\lambda_1^{-2}, \ldots, \lambda_D^{-2}]$:

$$r^2 := (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{\Lambda}^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)\,. \tag{2.20}$$

We already encountered the parameters σ_f^2 and λ above when we introduced the squared exponential kernel in one dimension; it turns out that for $\nu \to \infty$, we obtain just the squared exponential kernel in D dimensions (where we have one distinct λ_d for each dimension d).

Besides $\nu \to \infty$, we will only consider the special cases of $\nu = 1/2$, $\nu = 3/2$, and $\nu = 5/2$ for which the Matrn kernel takes a particularly simple (and therefore computationally efficient) form. The explicit kernel functions are:

- Squared exponential kernel: $\nu \to \infty$:

$$k_{SE}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\left[-\frac{1}{2}r^2\right]. \tag{2.21}$$

- Exponential kernel: $\nu = 1/2$:

$$k_{E}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\left[-\frac{1}{2}r\right]. \tag{2.22}$$

- Matrn kernel with $\nu = 3/2$:

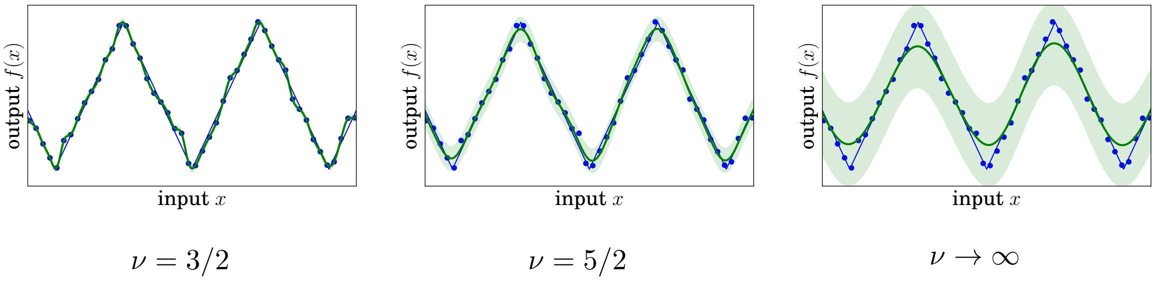
$$k_{\nu=3/2}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \left(1 + \sqrt{3}r\right) \exp\left(-\sqrt{3}r\right). \tag{2.23}$$

- Matrn kernel with $\nu = 5/2$:

$$k_{\nu=5/2}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right). \tag{2.24}$$

The parameter ν is called the *order* of the Matrn kernel. It can be interpreted as a measure for the smoothness of the function. The squared exponential kernel (i.e., $\nu \to \infty$) is infinitely mean-square differentiable and leads to the smoothest functions. It is very popular, but there are researchers who argue that it is a ill-suited for many practical problems because the smoothness assumption is too strong. This is the reason why we also consider the values of $\nu = 3/2$ and $\nu = 5/2$. For $\nu = 1/2$, the functions already become very rough as we will see; for values $\nu >= 7/2$, the functions are already so smooth that the difference to $\nu \to \infty$ is not palpable in many cases. Therefore, the three cases $\nu = 3/2$, $\nu = 5/2$, and $\nu \to \infty$cover all interesting cases.

Figure 2.7 illustrates the different behaviour of the three kernels on a very non-smooth function. The kernel parameters are learned using MLE. Like one would expect, the squared exponential kernel cannot represent the sharpness well and has to compensate it by assuming a lot of noise. Generally speaking, Matrn kernels with high values of ν are less affected by missing data or outliers, but they suffer from inability to express non-smooth behavior.



$\nu = 3/2$ $\nu = 5/2$ $\nu \to \infty$

Figure 2.7: Smoothness of Matrn kernels. The blue dots are drawn from a sawtooth function with a little noise. GPs with Matrn kernels and different values for ν are then fitted to the data where the hyperparameters are learned by MLE assuming stationary measurement noise and no process noise. It is well visible how the smoothness of the learned function (green) increases with ν. The inability to express the very non-smooth sawtooth function is compensated by increased noise.

The Mixture of Experts Model (HGP) Gaussian Processes are generally not directly applicable to large amounts of data because computation takes too long. One approach to tackle this issue is of course to aggregate or sample data points in order to reduce the size of the data set. We would however like to introduce another approach that was proposed by Ng and Deisenroth in Ng and Deisenroth (2014), the *Hierarchical Mixture of Experts Model for GP Regression (HGP)*. The idea is to partition the data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ into a set $\mathcal{S}$ of c subsets $\mathcal{S} = \{\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(c)}\}$ where $\mathcal{D}^{(i)} = (\boldsymbol{X}^{(i)}, \boldsymbol{y}^{(i)})$. Then, a GP is trained on each of these data sets as a *local expert*. The predictions of the local experts are then recombined into a global prediction. All local experts use the same kernel with the same set of hyperparameters.

This is computationally significantly less demanding. A GP that works on the full data set of size N has to compute the inverse of its Gram matrix which is in $O(N^3)$. If the data is evenly distributed across c local experts, the complexity is reduced to $O(\dot{c}(N/c)^3) = O(N^3/c^2)$, i.e., it decreases quadratically in the number of local experts. Prediction speed increases linearly. A possible approach to distributing the data among the local experts is to randomly assign it. This is fast and can work well. A more sophisticated approach proposed by Ng and Deisenroth is to use a KD-tree to recursively split the data into non-overlapping regions. The data is then distributed across the local experts in a manner that ensures that each of them has data from all regions, i.e., each local expert works on data that covers the whole input space.

Inference, Posterior, and Parameter Learning

The predictive distribution y_* of a single data point $\boldsymbol{x}_*$ is a Gaussian and is proportional to the product of the distributions of the local experts:

$$\mathrm{p}(y_* \,|\, \boldsymbol{x}_*, \mathcal{D}) = \mathcal{N}(y_* | \mu_*, \sigma_*^2) \propto \prod_{k=1}^{c} \mathrm{p}(y_* \,|\, \boldsymbol{x}_*, \mathcal{D}^{(k)}) \tag{2.25}$$

Let μ_k and σ_k be the mean and the standard deviation of the posterior of local expert k. The overall predictive mean and variance are[3]:

$$\mu_* = \sigma_*^2 \sum_{k=1}^{c} \frac{\mu_k(\boldsymbol{x}_*)}{\sigma_k^2(\boldsymbol{x}_*)}, \tag{2.26}$$

$$\sigma_*^2 = \left(\sum_{k=1}^{c} \frac{1}{\sigma_k^2(\boldsymbol{x}_*)} \right)^{-1}. \tag{2.27}$$

Note that this formulation only gives us pointwise predictions. With normal GPs, we got a closed-form expression for the joint posterior distribution at any number of evaluation points (Equation (2.10)). This is sufficient for our purposes but can in general be limiting because it gives us no information about the correlation of function values.

Training

All local experts use the same kernel. The log-marginal likelihood is approximated by

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}) \approx \sum_{k=1}^{c} \mathcal{L}(\mathcal{D}^{(k)}|\boldsymbol{\theta}) \tag{2.28}$$

Thanks to this simple structure, we immediately get an approximation for the derivative w.r.t. the kernel or mean parameters:

$$\frac{\partial \mathcal{L}}{\partial \theta}(\mathcal{D}|\boldsymbol{\theta}) \approx \sum_{k=1}^{c} \frac{\partial \mathcal{L}}{\partial \theta}(\mathcal{D}^{(k)}|\boldsymbol{\theta}) \tag{2.29}$$

Consequently, we can use exactly the same setup for training HGPs that we used for regular GPs.

Some Practical Issues The basic setup of GPs is based entirely on Bayesian statistics; we specify our prior belief and update it with data to obtain our posterior belief. Unfortunately, we cannot usually rely on a pure Bayesian approach because we lack sufficient information to specify a prior with confidence. This is why we use MLE or MAP to find explicit estimates of the “true” hyperparameters. This however means that we take an explicit decision, i.e., we claim that the learned hyperparameters are the “true” ones with a hundred percent certainty which will in reality never be the case. Since MLE and MAP choose the parameters by maximizing a function (a performance function from a machine learning point of view), this can lead to *overfitting* on the given data.

A typical indicator for overfitting are very large or very small hyperparameter values, in our case in particular σ_f and the λ_d. As an example, consider the GPs in Figure 2.6. The GP in the first plot has quite a high value of σ_f and low λ; it might be the case that it overfits by interpolating all data points; we cannot determine this with certainty without any knowledge of the ground truth. Standard machine-learning approaches such as crossvalidation can be used to counteract overfitting. Still, the general problem remains that it is difficult (or impossible) to say when a model overfits. In general, overfitting can only be assessed if some prior knowledge about the properties of the “true’ function is available. The remainder of this section will be dedicated to the study of an example where we highlight some of the issues that come with MLE. But before that, we introduce a measure for the dissimilarity of GPs that will allow us to assess the quality of the learned GPs

[3] Equations (2.28) and (2.29) assume that σ_k is the uncertainty of the GP, i.e., σ_k does *not* contain process noise like in our standard formulation of the GP posterior. If process noise is present, the computations must be done without it and it can be added at the end.

Comparing Gaussian Processes

The Bhattacharyya distance is a measure for the similarity of two continuous proabability density functions $p(\cdot)$ and $q(\cdot)$ and is defined by

$$d_B(p,q) = -\ln\left[\int \sqrt{p(x)q(x)}dx\right]. \tag{2.30}$$

It is symmetric and satisfies $0 \leq d_B \leq \infty$, but it does not obey the triangle inequality (and is thus not a metric). It takes a particularly simple form for two univariate normal distributions $p \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $p \sim \mathcal{N}(\mu_2, \sigma_2^2)$:

$$d_B(p,q) = \frac{1}{4}\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2} + \frac{1}{2}\ln\left[\frac{\sigma_1^2+\sigma_2^2}{2\sigma_1\sigma_2}\right]. \tag{2.31}$$

We can use the Bhattacharyya distance to define a measure for dissimilarity of GPs. Let $\mathcal{GP}_1$ and $\mathcal{GP}_2$ be two GPs that express distributions over functions $\mathbb{R}^D \to \mathbb{R}$. We define the distance between $\mathcal{GP}_1$ and $\mathcal{GP}_2$ on the closed input space $\mathcal{X} \subset \mathbb{R}^D$ as

$$\mathrm{D}_{\mathcal{X}}(\mathcal{GP}_1, \mathcal{GP}_2) := \sqrt{\frac{1}{V(\mathcal{X})}\int_{\mathcal{X}} d_B(\mathcal{GP}_1(\boldsymbol{x}), \mathcal{GP}_2(\boldsymbol{x}))^2 d\boldsymbol{x}} \tag{2.32}$$

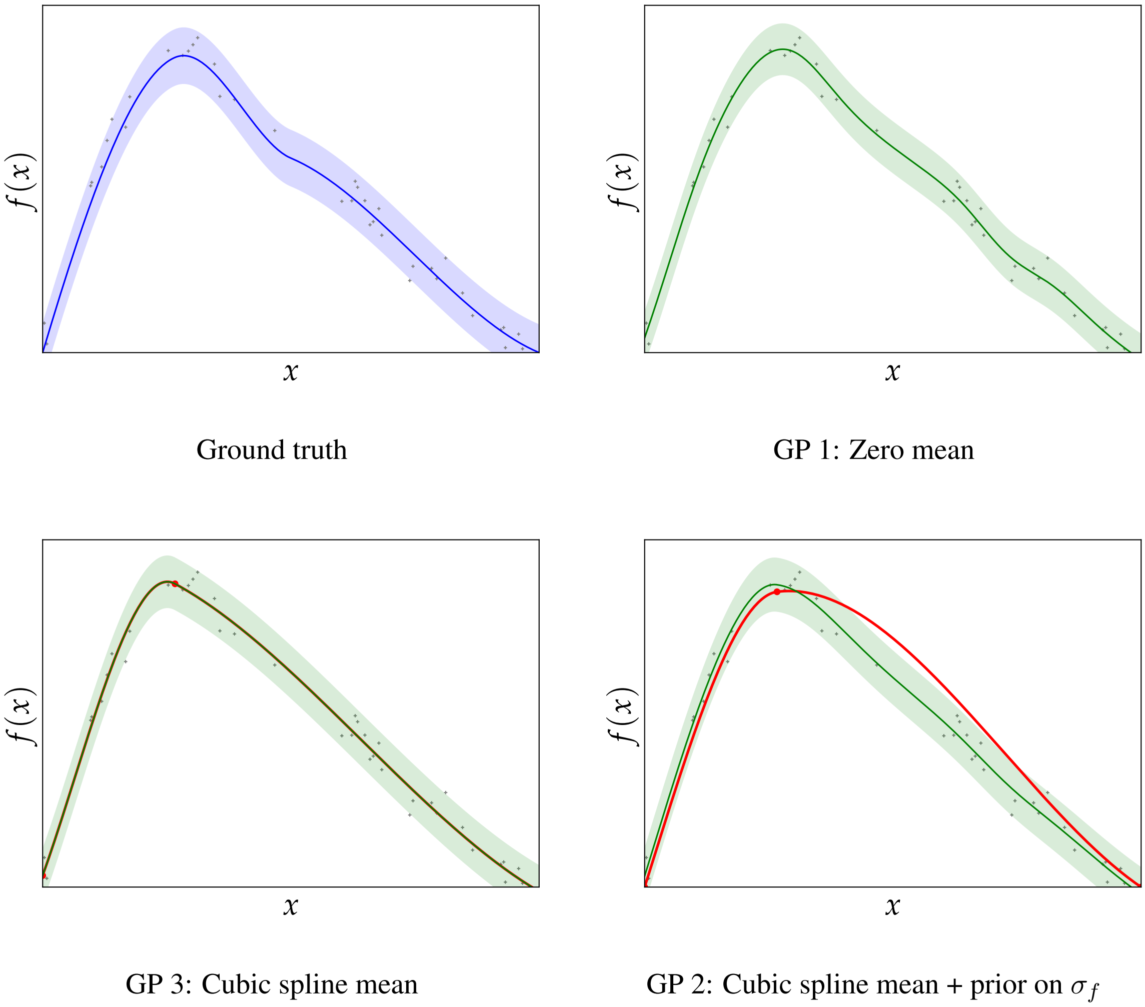
where $V(\mathcal{X})$ denotes the volume of subspace $\mathcal{X}$. The dissimilarity cannot be computed exactly but it can be estimated using MC methods. It is symmetric, satisfies $0 \leq D_{\mathcal{X}} \leq \infty$ but does not satisfy the triangle inequality.

Overfitting in GPs – an Example

We would like to give an example here which illustrates that maximizing the log-marginal likelihood can indeed lead to overfitting. To this end, we use the generative model that is illustrated in the first image in Figure 2.8. For each value of x, the corresponding function value is $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_p^2)$ is process noise (the shaded area in the image is $2\sigma_p$) and f is the function illustrated by the blue graph. We assume absence of any measurement noise. We want to learn the ground truth by fitting GPs to data points that are sampled from it (gray crosses); since we actually know the ground truth, we can estimate the dissimilarity of the learned GP from the true model.

The three other images in Figure 2.8 show GPs that were fit to the data. The hyperparameters were learned using MLE. All GPs use a Matern kernel with $\nu = 5/2$, stationary process noise and no measurement error (i.e., $s_m(x) = 0$ and $s_p(x) = \sigma_p^2 = const.$). GP 1 assumes a fixed zero mean function $m(\cdot) = 0$, while the mean $m(\cdot)$ of GPs 2 and 3 is represented as a two-segment cubic spline that is drawn in red in the images. The segment borders of the splines are marked with red dots. GP 1 therefore has three free hyperparameters while GPs 2 and 3 have a nine. Since the prior model of GP 2 is more expressive than the one of GP 1, we can expect its log-marginal likelihood $\mathcal{L}$ to be higher. This is indeed the case:

	GP 1	GP 2	GP3
Log-marginal likelihood $\mathcal{L}$	-285	-265	-280
Distance from ground-truth $\mathrm{D}_{[0,300]}$	0.059	0.16	0.087
Standard deviance of process variance σ_f	2283	0	2276
Characteristic length scale λ	101	(9454)	146
Standard deviance of process noise σ_p	182	185	191



Ground truth

GP 1: Zero mean

GP 3: Cubic spline mean

GP 2: Cubic spline mean + prior on σ_f

Figure 2.8: An example of overfitting caused by a complex prior model. The top left panel show the ground truth and the data sampled from it. The other images show GPs with different setups of the prior.

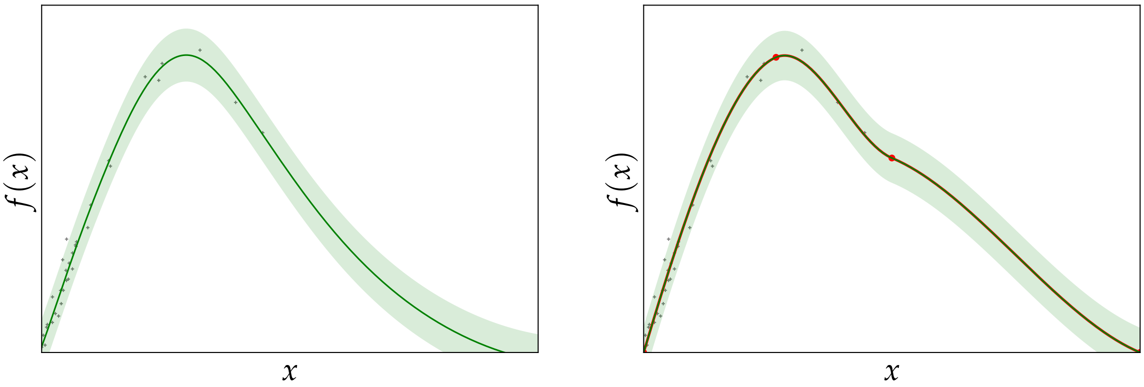


Figure 2.9: Imbalanced data and priors. The same ground truth as in Figure 2.8 applies but the data points are not evenly distributed. The GP in the left image uses a zero-mean prior, the one in the right image a three-segment cubic spline that is good approximation of the ground truth.

However, looking at the images it seems that GP 2, in spite of having achieved the highest value for $\mathcal{L}$, actually approximates the ground truth worst. This is backed by the dissimilarity values which show that the more powerful model of GP 2 is more dissimilar from the ground truth.

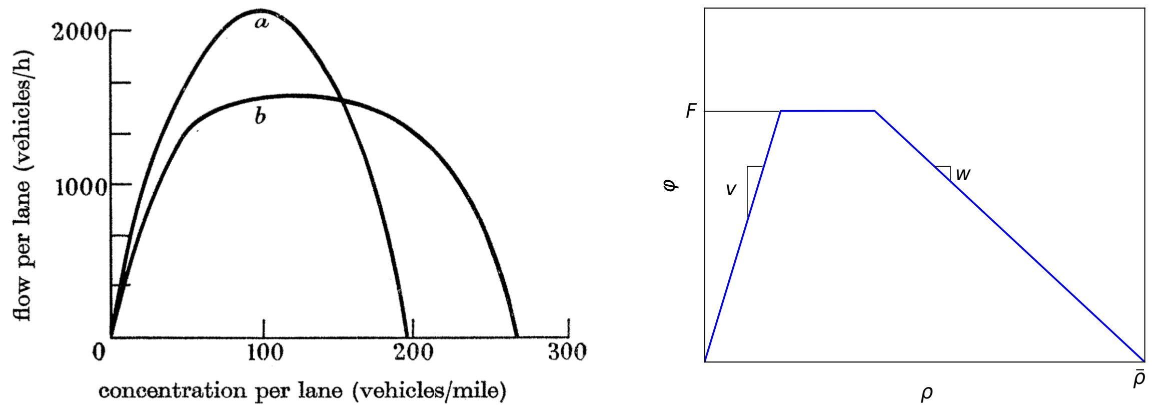
What is going on here? Looking at the parameter values, we see that GP 2 has learned a process variance σ_f of zero. From the definition of the Matrn kernel we get that $\sigma_f = 0$ implies $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ – in other words, all data points are learned to be completely uncorrelated. Let us take a closer look at the definition of the log-marginal likelihood $\mathcal{L}$ (Equation (2.14)) that is maximized in MLE to see why this is. The frist term is $-1/2 \ln |\boldsymbol{C}| = 1/2 \ln |\boldsymbol{K} + \boldsymbol{K}_m + \boldsymbol{K}_p|$, where $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. $\boldsymbol{C}$ is positive definite which means that $|\boldsymbol{C}|$ increases whenever its elements increase; therefore, for fixed noise covariance matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_m$, the first term is maximized by $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$. The second term of $\mathcal{L}$, $-1/2(\boldsymbol{y} - \boldsymbol{\mu})^\top \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$, on the other hand, is big whenever $\boldsymbol{C}$ explains the data distribution well. Hence, the first term of $\mathcal{L}$ can be interpreted as a model complexity penalty while the second term rewards data fit. There is a trade-off between the two that is inherently contained in MLE. From a machine-learning point of view, the first term can be interpreted as a regularizer that reduces the risk of overfitting. Note that increasing the number of data points has a lesser impact on the first term due to its logarithm than on the second term which implies that large amounts of data decrease the influence of the complexity penalty and increase the tendency towards a good data fit.

In our example, the mean function in the form of a cubic spline that is used in GP 2 can explain the shape of the ground truth function quite well. For this reason $\mathcal{L}$ is maximized by explaining all deviations of the data from it as pure process noise and assuming no correlation beyond the structure imposed by the prior mean function between nearby data points. Only large quantities of data would overrule this behavior. Learning the hyperparameters of prior GP 2 by MLE therefore can be understood as starting from our (learned) prior where MLE only "enhances" it by learning additional covariance between nearby data points if the prior mean alone cannot explain the data well. This behavior makes it difficult for us to reliably specify priors on the hyperparameter values. In GP 3, we used the same setup as for GP 2 but additionally imposed a prior on σ_f that favours values comparable to the ones in GP1. We can indeed see that the learned parameter values are similar to GP 1 – but in the image we see that this forces the shape of the prior mean function away from the true one. This nicely illustrates that it can be very difficult to specify hyperpriors because it the relationships between hyperparameters can be highly complex.

With the intuition about the behavior of mean functions obtained above, it becomes clear that complex parametrized mean functions are prone to overfitting. MLE inherently trades off data fit and model complexity *for the kernel* but there is *no such behavior for the prior mean function*. For this reason, it does not seem a good idea to learn parametrized mean functions unless we can specify concise priors on their expected parameter values. We've just seen that this is very difficult because there can be interactions between hyperparameters that are difficult to understand. For this reason, we have decided not to learn prior mean functions. However, specifying (but not learning) prior mean functions can make sense if we have good knowledge about its shape. Prior means are particularly useful if there are areas with no, very little, or very noisy data. This is illustrated in Figure 2.9. There, the prior mean in the right image is not learned from data, but it is specified entirely based on prior knowledge which we have in this case – it was taken directly from the ground truth. In practice such knowledge may be difficult to obtain. Obviously, the GP with mean prior is a better fit and it also achieves a better values for $\mathcal{L}$ and $D_{[0,300]}$:

	left GP	right GP 2
Log-marginal likelihood $\mathcal{L}$	-247	-230
Distance from ground-truth $D_{[0,300]}$	0.14	0.000022

In summary, we can conclude that prior mean functions, while justified from a statistical point of view, can cause problems in practice. They have a tendency of overruling the data and when their parameters are learned, they can overfit. They can be useful if only little data of good quality is available, but in most cases, they lead to behavior that is at first glance counter-intuitive. Since GPs work quite well for our purposes with zero-mean functions, we have decided not to use parametrized mean functions. This is a very common approach in the literature.



(a) (b)

Figure 2.10: Examples of fundamental diagrams. Panel (a) shows two two proposals for the density-flow relationship from the original Lighthill, Witham, and Richards paperLighthill and Whitham (1955). Panel (b) shows a fundamental diagram in the shape of an isosceles trapezoid like used by Daganzo in the cell transmission model.

Modeling Traffic

A basic observation that can be made on stretches of highways without any nearby entries or exits is that flow ϕ (vehicles per hour) and density ρ (vehicles per kilometer) seem to be closely related quantities. The local density-flow relationship is referred to as the *fundamental diagram* and is characteristic for a road. Examples of their typical shape are given in Figure 2.10. A fundamental diagram only expresses the direct relationship between density and flow at a fixed location and a fixed point in time, but it does by itself describe the dynamic behavior of traffic systems. Obviously, the evolution of the state of traffic over time depends on conditions both upstream and downstream. Further important properties that should be preserved by a model are the flow conservation law which expresses that the number of vehicles in the system is constant (except for vehicles leaving and entering they system at either end) and the requirement that vehicles only move in one direction. One of the most widespread approaches to representing this behavior at a macroscopic level is the *hydrodynamic model* that was proposed by Lighthill and Witham in 1955 and had originally been used for predicting flows in rivers. It treats traffic as a compressible fluid that obeys the continuity equation

$$\frac{\partial\phi(x,t)}{\partial x} = -\frac{\partial\rho(x,t)}{\partial t}, \tag{2.33}$$

where x denotes location and t time. Naturally, it observes the flow conservation law and also provides the time-space dynamics. Incorporating the fundamental diagram into the flow conservation law leads to a partial differential equation that cannot be solved analytically. One practical approach is Daganzo's cell transmission model that does not explicitly solve the equation but simulates a discrete-time model that exhibits similar properties as the hydrodynamic model, given some assumptions. In the remainder of this section, we first give an introduction to the cell transmission model and then explain how we can enhance it by incorporating Gaussian Processes.

Cell Transmission Model The cell transmission model (CTM) Daganzo (1994b, 1995) is a simple discrete-time model for traffic on highways without any intermediate entrances or exits. The highway segment under consideration is divided into I cells numbered consecutively from 1 to I in flow direction. The length of each cell is set equal to the distance that is traveled in light traffic by a typical vehicle in one clock tick of length Δt. If there is only light traffic, all vehicles in one cell will travel to the next cell in one time interval:

$$n_{i+1}(t+1) = n_i(t) \tag{2.34}$$

where $n_i(t)$ is the number of vehicles in cell i at time t. The CTM assumes that this equation holds in all situations unless traffic is slowed down by a downstream bottleneck. To incorporate such bottlenecks in the model, each cell i is associated with an *inflow capacity* $Q_i(t)$ that corresponds to the maximum number of vehicles that can flow into it during time interval t. Furthermore, each cell has a *jam density* $N_i(t)$ which corresponds to the maximum number of vehicles that can be present in it during time interval t – the maximum inflow into cell i is limited by the free space in it, $N_i(t) - n_i(t)$. In summary, the CTM expresses that the flow from cell $i-1$ to cell i between times t and $t+1$, denoted by $y_i(t)$, is the smallest of three quantities:

$n_{i-1}(t)$,		the number of vehicles in cell $i-1$ at time t,
$Q_i(t)$,		the inflow capacity of cell i during time interval t, and
$N_i(t)$	$-$	the amount of empty space in cell i at time step t.
$n_i(t)$,		

The change in number of vehicles in cell i equals inflow minus outflow:

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t) \tag{2.35}$$

where the flows are given by

$$y_i(t) = \min\left\{n_{i-1}(t), Q_i(t), N_i(t) - n_i(t)\right\}. \tag{2.36}$$

N_i, n_i, Q_i, and y_i are all dependent on the time step length. We now replace them by the following more general notions:

$\rho_i(t)$	[vehicles per km]	density of vehicles in cell i during time step t
$\phi_i(t)$	[vehicles per hour]	flow from cell $i-1$ to cell i during time step t
$\bar{\rho}_i(t)$	[vehicles per km]	jam density of cell i during time step t
$F_i(t)$	[vehicles per hour]	inflow capacity of cell i during time step t
l	[km]	length of the cells
v	[km per hour]	free-flow speed

Note that $v = l/\Delta t$ (Δt in hours). We have the following relationships:

$$n_i(t) = l \cdot \rho_i(t) = \Delta t \cdot v \cdot \rho_i(t), \tag{2.37}$$

$$y_i(t) = \Delta t \cdot \phi_i(t), \tag{2.38}$$

$$N_i(t) = l \cdot \bar{\rho}_i(t) = \Delta t \cdot v \cdot \bar{\rho}_i(t), \tag{2.39}$$

$$Q_i(t) = \Delta t \cdot F_i(t), \tag{2.40}$$

Hence, (2.36) can be rewritten as follows:

$$\phi_i(t) = \min\left\{v\rho_{i-1}(t), v(\bar{\rho}_i(t) - \rho_i(t)), F_i(t)\right\}. \tag{2.41}$$

Traffic jams generally dissolve in a backward wave manner with a characteristic *backward wave speed* w which is generally slower than the *free flow speed* v. The CTM as expressed in equation (2.41) however assumes $v = w$. Fortunately, a wave speed $w \leq v$ can be incorporated easily by adapting the outflow term:

$$\phi_i(t) = \min\left\{v\rho_{i-1}(t), w(\bar{\rho}_i(t) - \rho_i(t)), F_i(t)\right\}. \tag{2.42}$$

This equation is called the *transfer function*. Let us have a closer look at this relationship. If we assume that v and w as well as the cell parameters F_i and $\bar{\rho}_i$ are fixed, the flow ϕ_i is a function solely of ρ_{i-1} and ρ_i. The corresponding function is illustrated in Figure 2.11 (a) with realistic parameter values. We can see that the function is piecewise affine. The dotted blue line shows the flows at $\rho_{i-1} = \rho_i$, and it is re-plotted in Panel (b). It corresponds to the local fundamental diagram in the neighborhood of cells $i-1$ and i. Daganzo shows in Daganzo (1994b) that the CTM can indeed be understood as a discrete approximation of the hydrodynamic model when a fundamental diagram of the shape of an isosceles trapezoid like in Figure 2.10 (b) is used:

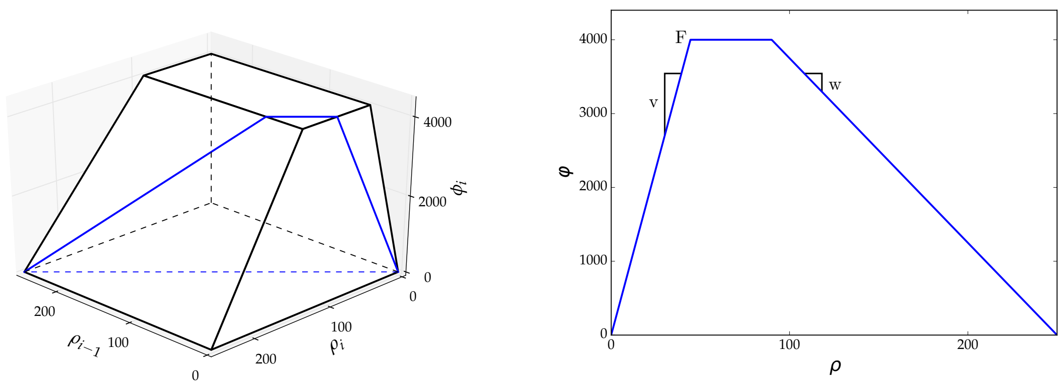
$$\phi = \min\left\{v\rho, w(\bar{\rho} - \rho), F\right\}. \tag{2.43}$$

In other words, the function in Equation (2.41) is an approximation to the hydrodynamic model that inherently contains both the fundamental diagram and the dynamics that follow from the continuity law (2.33). In its standard formulation, the CTM implicitly relies on the assumption that the fundamental diagram is given in the shape (2.43). However, the CTM can be easily adapted to different or more complex models by replacing the transfer function by any nonnegative function f_i:

$$\phi_i(t) = f_i(\rho_{i-1}(t), \rho_i(t)). \tag{2.44}$$

Of course, a general nonnegative function will not preserve the properties of the hydrodynamic model, in particular the continuity equation. However, if sufficient data is available, the transfer function can be found with machine-learning methods. It is not a problem if some of the properties of the hydrodynamic model are lost in this case because this behavior follows directly from data. From this perspective, the CTM is more flexible than the hydrodynamic approach because the original assumption that traffic behaves like a compressible fluid is lifted. In other words, the CTM is not just an approximation to the hydrodynamic model, but it is a much more general setup that is based only on the most basic properties of traffic and makes very few assumptions beyond this.

In the remainder of this report we discuss how the transfer function of a highway segment can be learned as a Gaussian Process. GPs are particularly well-suited because they allow us to take a non-parametric approach with very few assumptions about the transfer function. Most importantly, however, GPs do not just learn one fixed transfer function but instead provide us with a distribution over transfer functions. Also, GPs allow us to incorporate and learn process noise. All of this is very useful for example in the design of ramp metring algorithms that are robust to uncertainty in the model.



(a): flow between cells

(b): fundamental diagram

Figure 2.11: Illustration of the relationship between density and flow in the CTM. Panel (a) shows the flow between cells $i-1$ and i assuming a free-flow speed v of 90km/h, a backward-wave propagation speed w of 25km/h, a jam density $\bar{\rho}$ of 250 vehicles/km, and the capacity flow is F of 4000 vehicles/h. The blue line is the situation where $\rho_k = \rho_{k+1}$, i.e. where the density is the same in both cells. It is re-plotted in (b) – it corresponds to the fundamental diagram in the neighborhood of cells $i-1$ and i.

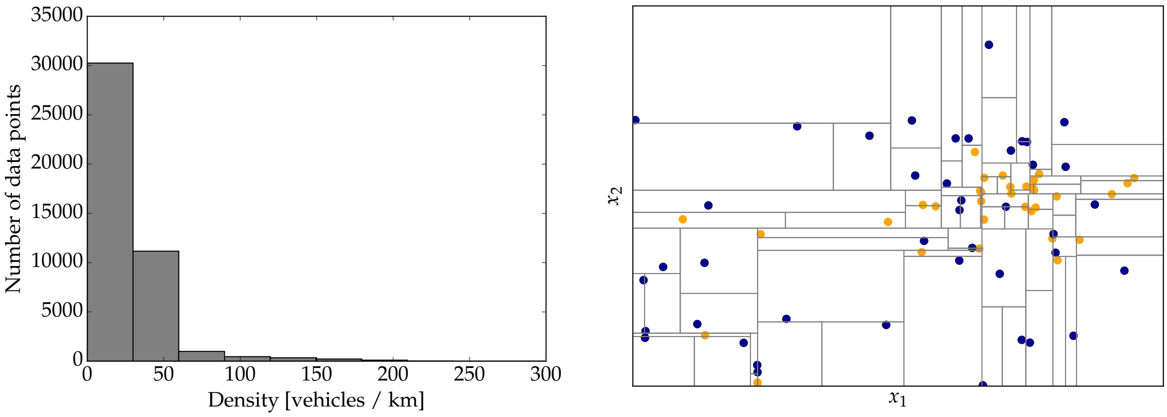


Figure 2.12: The left panel illustrates unbalanced data for the density at sensor location 7 in October 2014. The right panel illustrates density harmonization in 2D. The rectangles represent the cells obtained by building the KD-tree; the blue data points are kept, the orange ones are thrown out.

Learning a Traffic Model

In Sections 2.1.1 and 2.1.1, we introduced our setup of Gaussian Process Regression and the Cell Transmission Model. In this section, we discuss how we can learn the dynamics of a highway by combining GPs and insights from the CTM. Thanks to the Grenoble Traffic Lab (GTL) we had access to detailed traffic data from a 10.5 km stretch of a two-lane highway in Grenoble in France. A total of 130 sensors read density, flow and average speed at intervals of 15 seconds. Unfortunately, GPs are not able to cope with such an abundance of data directly due to their non-parameteric nature. Not only the learning of the hyperparameters takes long, but also predictions become obstructively slow when a GP is fit to tens of thousands of data points. This issue has received a lot of attention and many approaches to reducing the computational burden have been proposed, but the core problem remains. One such approach is the HGP model that we introduced in Section 2.1.1. We have found that in our case we obtain reasonable performance by just preprocessing and selecting the data points in the right manner. We will first discuss the preprocessing of the data and which issues we encoundered. Then, we illustrate how to learn the fundamental diagram and eventually continue to develop our full model.

Preprocessing and Selecting the Data Each sensor measures three different quantities at 15-second intervals. Firstly, they measure the percentage of time when a car is above the sensor. Sencondly, the number of cars passing by in each interval is registered. And finally, the average speed is estimated from the quantities above. We did not use the speed estimates.

For ease interpretability, we rescale the density ρ and flow ϕ such that they represent the number of vehicles per kilometer and the number of cars per vehicles, respecitvely; this is the same units that we already used in the CTM. The interval length of 15 seconds is rather short which can lead to inaccurate estimates of the density and the flow. For this reason, we aggregated the data from 4 consecutive measurement which gives us one measurement every minute.

Most of the day, there is only little or no traffic at all which means that we obtain a lot of data for similar conditions. Traffic jams on the other hand occur only infrequently and are often of short duration, consequently we have less data available for high densities. In spite of the high amount of data available there are areas in data space where the density is very low. The distribution of one sensor is given in Figure 2.12. A natural approach to reducing the data set sizeis therfore the throw out points in areas where the density is above some threshold d^* while points elsewhere are kept. To this end, we can apply the following algorithm that we call *density harmonization*:

1. Build a KD-tree until each leaf contains exactly one data point.
2. For each leaf l, compute the hypervolume V_l that it covers. We the density of that leaf to be $d_l := 1/V_l$.
3. For each leaf l and the data point p_l it contains:

   If $d_l < d^*$: keep data point p_l

   Else: keep data point p_l with probability d^*/d_l.

Note that in order to determine the hypervolume of each leaf, we need a bounding box for our data. Such a bounding box is very easy to specify for our traffic data. A two-dimensional example of density harmonization is given in Figure 2.12.

Learning the Fundamental Diagram Before turning to the dynamics, we take a look at the static density-flow relationship expressed by the fundamental diagram. Many of the issues that apply to the two-dimensional case occur in the one-dimensional case as well.

Figure 2.13 shows graphical representations of four different GPs that were fit to data obtained over one month at sensor location 3. All four GPs use a Matrn kernel with $\nu = 5/2$ (we will discuss this choice later) and the hyperparameters of the kernels were learned using MLE, that is, we did not assume any prior knowledge about them. Each plot shows the pointwise mean prediction (green line) and the 2σ-interval of the process noise variance (green shadow). Furthermore, the uncertainty of the GP is shown additionally to the process noise (in a lighter shade of green; it is only visible for high densitiy values where there are few data points).

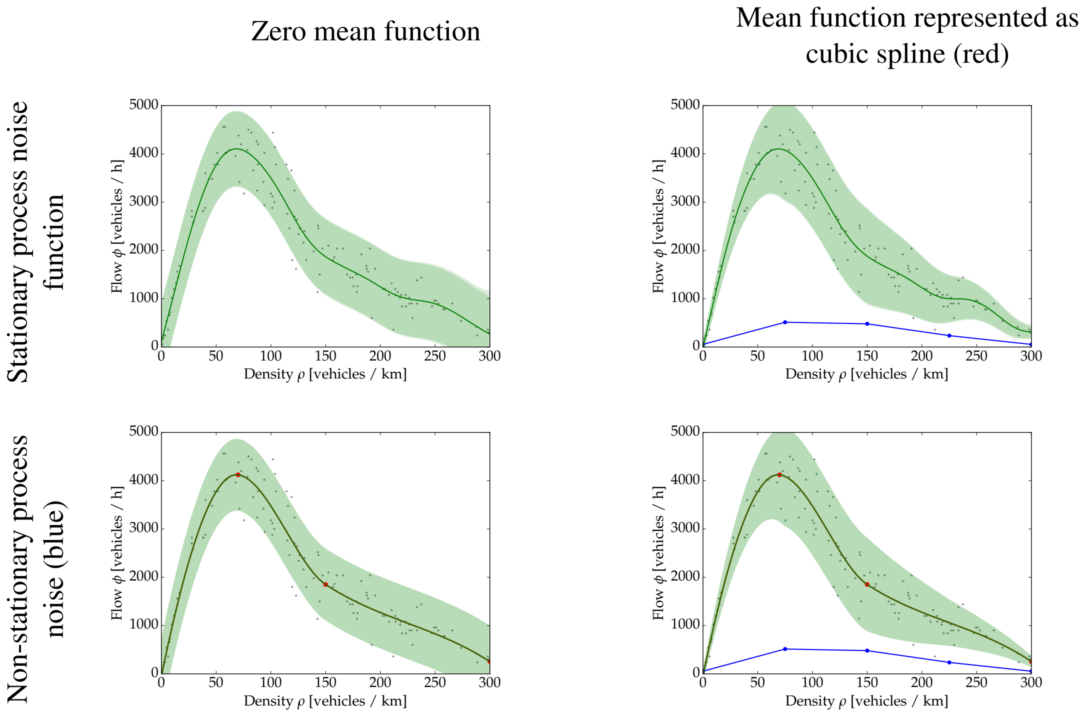
The plot in the top left corner assumes a fixed zero mean (i.e., $m(\boldsymbol{x}_i) = 0$) and stationary process noise (i.e., $s_p(\boldsymbol{x}_i) = \sigma_p^2$). Like we mentioned earlier, we assume absense of measurement noise (i.e., $s_p(\boldsymbol{x}_i) = 0$). Therefore, there are only three hyperparameters to be learned: the variance of process noise σ_p^2, the process variance σ_f^2, and the characteristic length scale λ. The GP seems to fit the data quite well and it also matches the shape that we would expect for a fundamental diagram.

One important aspect that our model cannot capture is that the process variance is probably not stationary. We would expect more variance and this seems to be the case indeed. Therefore, we assume non-stationary process variance in the second image in the top right corner. It is represented as a linear spline with regular spacings; the learned values are plotted in blue.

Finally, the two images at the bottom use cubic splines with three segments as mean functions. The same phenomenon that we already observed in Section 2.1.1 occurs; the data points are learned to be uncorrelated because the prior mean function explains the data well by itself. We will not consider these models further for the resons we discussed.

Learning process noise like in the top left GP seems reasonable but there seems to be an issue that the GP learns low variance in areas with few data points. For this reason it makes sense to put a prior on the process noise. In the following we will put a log-normal prior on it.

Learning the Transfer Function We discussed the Cell Transmission Model in Section 2.1.1. Under it, a highway segment of interest is split into cells where each cell i is associated with a traffic density $\rho_i(t)$. The dynamics of the system is expressed as a discrete-time process. During each time step t, there is an outflow $\phi_i(t)$ out of and an inflow $\phi_{i-1}(t)$ into cell i. The flow $\phi_i(t)$ between two cells $i-1$ and i is modeled by the transfer function f_i.



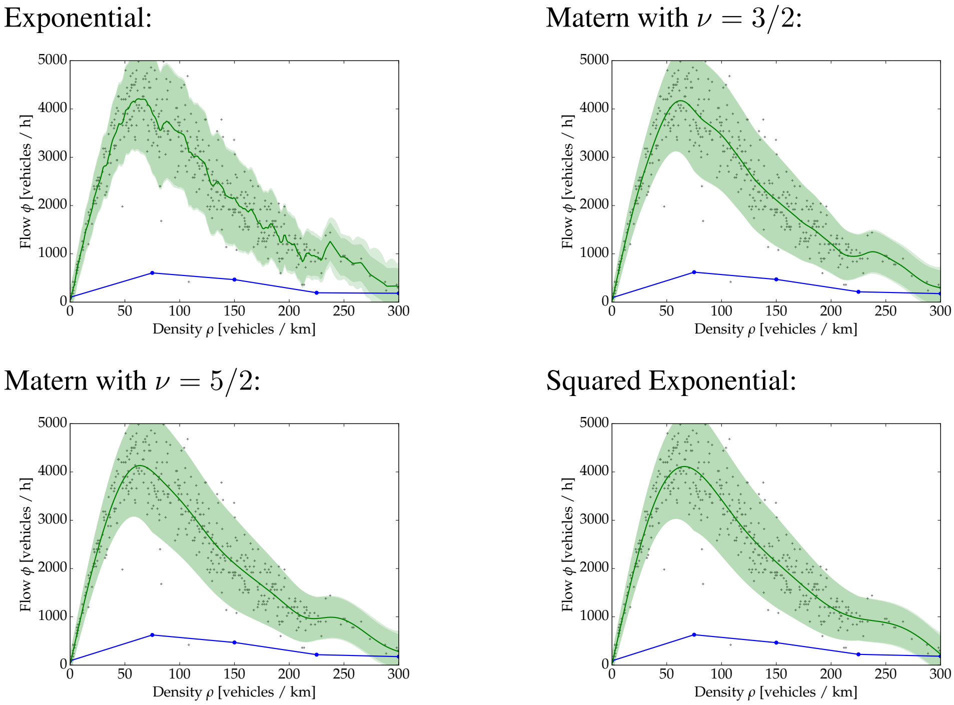
Zero mean function

Mean function represented as cubic spline (red)

Stationary process noise function

Non-stationary process noise (blue)

Figure 2.13: Learning GPs of different complexity. The green graph represents the pointwise mean posterior, the shaded area is twice the standard deviation of the process noise. All hyperparameters were learned by MLE.



Exponential:

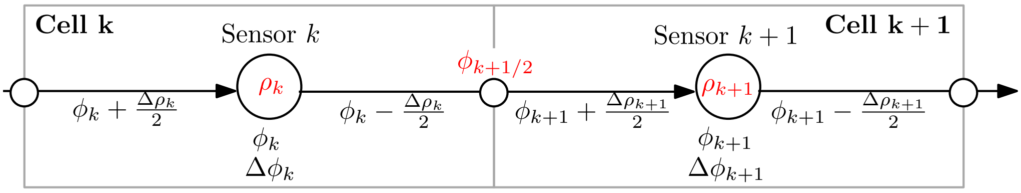
Matern with $\nu = 3/2$:

Matern with $\nu = 5/2$:

Squared Exponential:

Figure 2.14: Learning GPs of different complexity. The green graph represents the pointwise mean posterior, the shaded area is twice the standard deviation of the process noise. All hyperparameters were learned by MLE.

The Grenoble Traffic Lab data provides us with both flow and density at each sensor location. Unfortunately, this does not directly mirror the the structure that we need in the CTM: we want the flow *between* two sensor locations, not *at* the sensor locations. That is, we do not know $\phi_{k+1/2}$ in the following image (note that we've changed the use of indices for the flows such that flow and density at the sensors have the same index):


Cell k
Sensor k
ρ_k
$\phi_k + \frac{\Delta\rho_k}{2}$
$\phi_k - \frac{\Delta\rho_k}{2}$
ϕ_k
$\Delta\phi_k$
$\phi_{k+1/2}$
Sensor $k+1$
Cell $k+1$
ρ_{k+1}
$\phi_{k+1} + \frac{\Delta\rho_{k+1}}{2}$
$\phi_{k+1} - \frac{\Delta\rho_{k+1}}{2}$
ϕ_{k+1}
$\Delta\phi_{k+1}$

With the updated use of indices, the transfer function we want to learn has the shape

$$\phi_{k+1/2} = f_k(\rho_k, \rho_{k+1}). \tag{2.45}$$

We use a very simple approach to estimating $\phi_{i-1/2}$. For now, we consider just one single cell i and ignore its neighbor cells. We conceptually subdivide cell i into two subcells a and b with their border exactly on the sensor. Therefore, the sensor reading ϕ_i is just the flow rate between cells a and b. The inflow $\tilde{\phi}_{i,a}$ into cell a and the outflow $\tilde{\phi}_{i,b}$ out of cell b are unknown. However, we can calculate the net flow $\tilde{\phi}_{i,net} = \tilde{\phi}_{i,a} - \tilde{\phi}_{i,b}$ of cell i as a whole from the densities of two consecutive time steps:

$$\tilde{\phi}_{i,net}(t) = \frac{\rho_i(t+1) - \rho_i(t)}{l_i \Delta t}, \tag{2.46}$$

where l_i [km] is the length of the segment and Δt [h] is the length of the time interval between two sensor measurements. We assume that the flows $\phi_{i,a}$ and $\phi_{i,b}$ are roughly equal to ϕ_i and only deviate to explain the net flow which is caused partially by inflow into cell a and partially by outflow out of cell b:

$$\phi_{i,a} = \phi_i + \alpha_i \tilde{\phi}_{i,net}, \tag{2.47}$$

$$\phi_{i,b} = \phi_i - (1 - \alpha_i)\tilde{\phi}_{i,net}, \tag{2.48}$$

where $\alpha_i \in [0, 1]$. The value of α_i is unknown, we will discuss its choice in a moment.

So far, we've looked at individual cells in isolation, but we need to compute the flow between cells. We approximate the flow between cells $i-1$ and i by the average of $\tilde{\phi}_{i-1,b}$ and $\tilde{\phi}_{i,a}$:

$$\phi_{i-1/2} \approx \frac{\tilde{\phi}_{i-1,b} + \tilde{\phi}_{i,a}}{2} = \frac{\phi_{i-1} + \phi_i}{2} + \frac{(\alpha_{i-1} - 1)\tilde{\phi}_{i-1,net} + \alpha_i \tilde{\phi}_{i,net}}{2}. \tag{2.49}$$

What remains to be done is choose values for α_{i-1} and α_i. We base this on the assumption that flows show a roughly linear behavior across short distances. This is not truthful in general because bottlenecks tend to cause shocks where density and hence flow increase very quickly. Still, in most cases this approximation is not too bad. If shocks occur they will become apparent in sensor readings within a couple of time steps. Hence, we choose $\alpha_{i-1} = \alpha_i = \phi_{i-1}/(\phi_{i-1} + \phi_i)$.

Learning the Propagation Function

Now that we have estimates of the intersensor flow, we can actually fit GPs to the Grenoble Traffic Lab data. For an example, we proceed as follows to learn the propagation function of sensors 5 and 6:

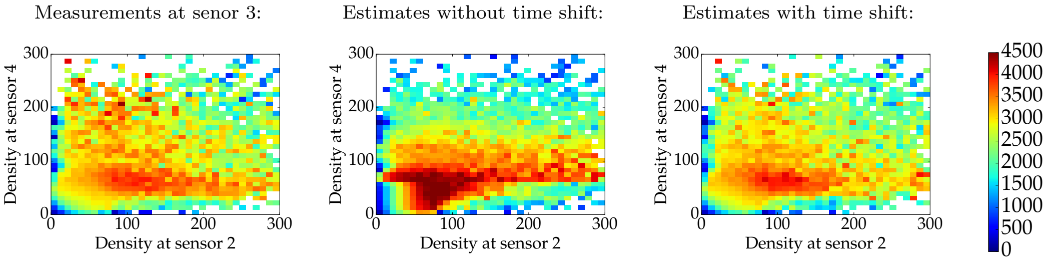


Figure 2.15: Estimating the intersensor flow. The left image depicts the actual measurements of the flow at sensor 3 (the "ground truth"), the middle image estimates the flow at sensor 3 without time shift and the right image estimates with time shift.

- Aggregation of data points: Four consecutive measurements of 15 seconds length are aggregated (that is, sampling time is reduced to 1 minute). The data is from October 2014; after aggregation, there is a total of 44603 data points.

- Intersensor flows: Intersensor flows are estimated using the method we explained (with time shift).

- Number of data points: 2500 data points are used. They are obtained via density sampling (after intersensor flow estimation).

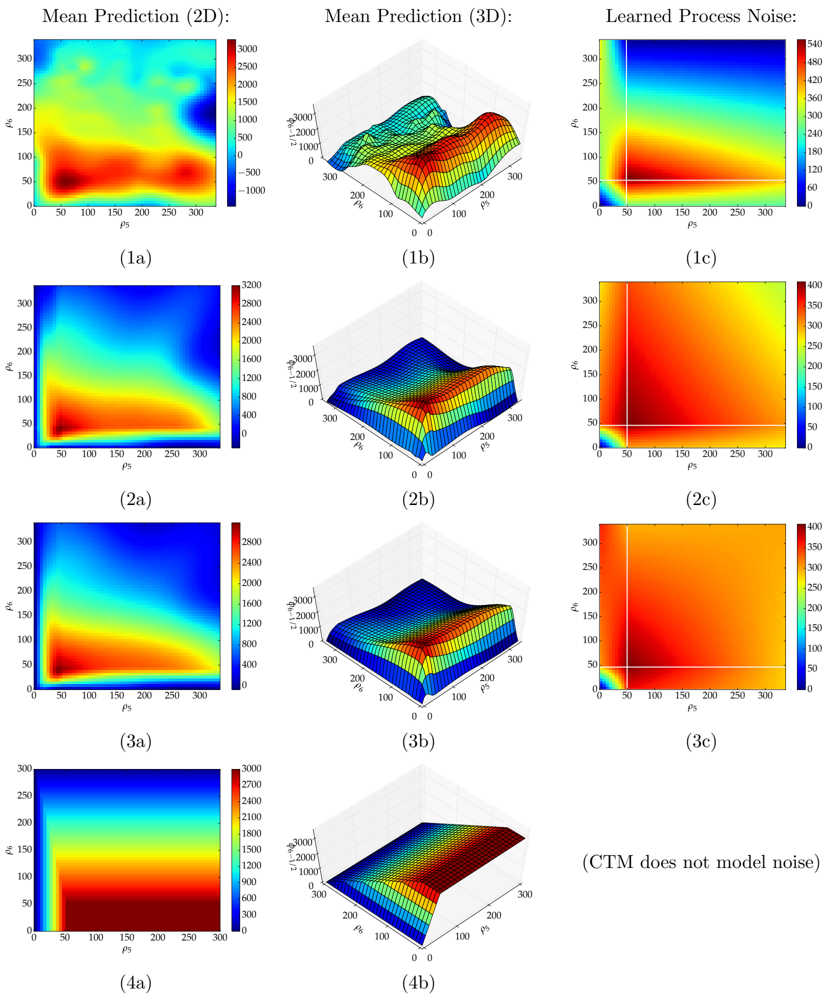
- Noise Model: We model process noise as a bilinear spline with a 3×3 grid of support points. The center point of the grid is located at the point $\rho_5^c, \rho_6^c)$. For simplicity, we assume absence of measurement noise. The critical densities ρ_5^c and ρ_6^c are estimated from the fundamental diagrams at sensors 5 and 6.

- Managing computational complexity: We use the HGP model with 10 nodes. Each node receives a subset of the data points of equal size, i.e., each node works on 250 data points.

- Kernel: A Matern kernel with order $\nu = 5/2$ is used.

- Training: The hyperparameters of the kernel and the process noise are learned using MLE. The noise at point $(0, 0)$ is fixed to zero. There are no priors on any of the other parameters.

The first row in Figure 15 depicts the GP that is learned from data as explained above. Panels (1a) and (1b) show the learned mean function, Panel (1c) the process noise where the white lines indicate the rectilinear grid of the bilinear spline. The learned GP is not quite what we expect: it is too rough. Looking at the values of the hyperparameters, we see that the characteristic length scales are particularly short and the process variance relatively big. Both of these contribute to the non-smooth behavior. The roughness allows the GP to fit the data quite closely which is expressed in low noise in areas where there are only few data points. We would prefer longer length scales and bigger process variance to obtain a smoother model. One approach to improving the GP representing the propagation function is to use non-stationary length scales and process variance in our kernel, but we would not like to do this because it makes the kernel even more complex and hence the optimization problem even more non-convex. Instead, we use the insights about the shape of the propagation function that we obtained when discussing the CTM (compare Figure 10). The propagation function has a steeper gradient at low densities than at higher densities. For this reason, we transform our densities to even out the length



(1a) (1b) (1c)

(2a) (2b) (2c)

(3a) (3b) (3c)

(4a) (4b)

Figure 2.16: Learning the propagation function with data from sensors 5 and 6. Row (1): Learning without any modifications. Row (2): With rescaling and added zeros along $\rho_5 = 0$. Row (4): Piecewise-affine model from the CTM. Learning the models takes only a few minutes on a standard laptop computer.

scales for the sake of learning as follows:

$$\tilde{\rho}_i := \begin{cases} \frac{250}{\rho_i^c} \cdot \rho_i & \text{if } \rho_i < \rho_i^c \\ \rho_i + 250 - \rho_i^c & \text{otherwise} \end{cases}$$

In words, the interval $[0, \rho_i^c]$ is stretched to $[0, 250]$ and the points beyond it are shifted accordingly. The GP is then fit to the transformed data (of course, the transformation has to be reversed when interpreting the GP). We expect the lengths scales of the kernel to be longer and the process variance to be lower because the steep gradients have been evened out. Indeed, the rescaling leads to a smoother GP as can be seen in the second row in Figure 15 (the densities have been transformed back).

One part of the propagation function that looks slightly different from what we would expect is the flows along the line $\rho_5 = 0$. The learned flows along it are slightly higher than zero which is also reflected by the data. Still, for reasons of stability (positive values at $\rho_5 = 0$ can cause negative densities in simulations because we have flow in spite of not registering any vehicles), we would like the values to

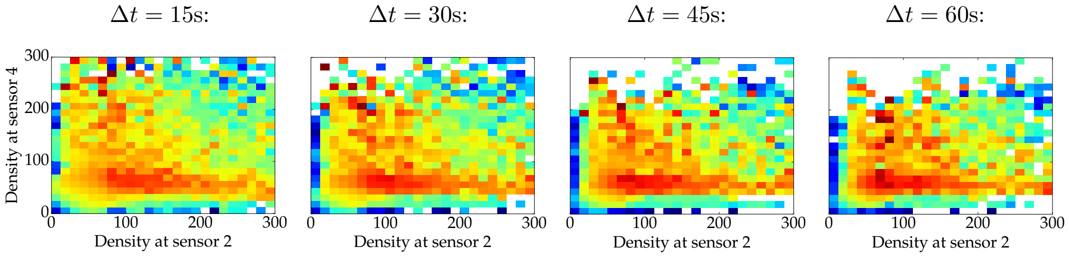


Figure 2.17: Actual measurements of the flow at sensor 3 for different aggregation times Δt for a long stretch of freeway between sensors 2 and 4. The distance between sensors 2 and 4 is roughly 1.37km, which means that vehicles in free-flow speed take about 55 seconds to pass it.

be exactly zero. To achieve this, we introduce artificial data points with value zero along the line $\rho_5 = 0$. The resulting GP is illustrated in the third row in Figure 15

Comparison with the CTM

The fourth row in Figure 15 depicts the piecewise-affine model of the propagation function that was used in the CTM by Daganzo. Just above it in the third row, we have the propagation function that is represented by a GP learned from data. It is obvious that there are certain similarities between the two models, but that there are also some major discrepancies. We would like to emphasize here that we expect to see some differences between the CTM and the learned propagation function, mostly because the CTM is based on some very particular assumptions that we cannot satisfy, but partly also because the piecewise-affine nature of the CTM is quite restricted. We have to be careful that we do not extend intuitions obtained from the CTM directly to our GP- based model. As an example, consider again the stretch between sensors 2 and 4. The two sensors are about 1.4 kilometers apart and consequently, vehicles at free-flow speed require just a little more than 25 seconds to reach sensor 3 that is roughly halfway between them. Backward waves in congested traffic are likely to take more than one minute to propagate from sensor 4 back to sensor 3. This is a major difference to the CTM where we assume that the flow is affected by the conditions in the neighboring cells immediately. Still, the effect of this behavior should not be too large in most situations because traffic conditions usually do not change too rapidly. We can expect the influence to be largest in cases where the densities at the two sensors are very different. Figure 16 shows again the measurements at sensor 3, but this time for different aggregation times. Like we anticipated, the differences are largest when ρ_4 or ρ_6 are very different. Consequently, we have to choose an aggregation time that is long enough such that the direct relationship between upstream/downstream sensor densites and the flow at the sensor in between is actually present in the data. On the other hand, it must not be too long because otherwise we “average away” short-time behavior such as shocks at the end of traffic jams. We cannot learn the behavior over a very short period of time, simply because traffic conditions require some time to propagate. This is unfortunate because our model will not adequately express rapid changes in flow as caused by shocks, but there is nothing we can do about this without making additional assumptions. Let us return to the results in Figure 15. One difference between the learned propagation function and the piecewise-affine version from the CTM that is particularly interesting is the flows along the line $\rho_6 = 0$. According to the CTM, the flow should increase along it as the upstream density increases until the capacity is reached, but this is not observed. Instead, flows close to zero are consistently obtained. A possible explanation is that this is caused by errors in the estimation of the intersensor flow, but this seems unlikely because the exact same phenomenon could be observed in the actual measurements in Figure 16 at sensor 3. Other possible explanations are erroneous measurements or simply that this behavior is actually present in the dynamics of the freeway.

Conclusions

We investigated how Gaussian Processes can be used to learn the traffic dynamics of a freeway from data. We first introduced Gaussian Processes in general and showed how both non- stationary process and measurement noise can be incorporated into them. We talked abut different kernels of the Matern class and their properties, discussed their suitability for learning traffic dynamics, and we showed how the kernel parameters can be learned using either MLE or MAP with a gradient- descent method. Furthermore, we proposed two approaches for learning the process noise, one based on maximizing the data likelihood via gradient descent, the other based on estimating the noise directly from data. To tackle the computational complexity of Gaussian Processes, we used the approximative Mixture of Experts Model (HGP). Moreover, we discussed some aspects of overfitting that occur with Gaussian Processes and explained why we decided to use zero-mean functions exclusively. We then continued to introduce the LWR model and a discrete approximation to it, the Cell Transmission Model (CTM). Based on ideas and insights from the CTM, we outlined how GPs can be used to learn traffic dynamics. In particular, we saw that learning the fundamental diagram alone is not sufficient, but that instead we have to learn propagation functions that link flow to traffic conditions both upstream and downstream. Thanks to the cell-structure of the CTM, the traffic conservation law is automatically observed. Our traffic model is an adaption of the CTM that is discrete in both time and state where the dynamics are modeled as Gaussian Processes. Therefore, it is suitable for use with the Probabilistic Inference for Learning Control (PILCO) framework. Thanks to the Grenoble Traffic Lab, we had access to data from the Rochade Sud freeway in Grenoble. We proposed density harmonization and density sampling as methods to reduce the data set size while preserving as much information as possible. A major issue that we encountered is that the sensor distribution does not mirror the structure that we need in our model, therefore we developed an estimation method for the traffic flow halfway between sensors. We compute these estimates based on insights from the CTM and Newell's principle of traffic propagation. We demonstrated that our estimates are reasonably accurate in a case study. Based on the data and the estimated intersensor flows, we then showed some practical examples of propagation functions that are represented as GPs where we highlighted some of the issues that we encountered and we provided practical solutions. In brief, our contributions are as follows. We proposed a data-based model that expresses the traffic dynamics of a freeway by means of Gaussian Processes with non-stationary process noise. We presented solutions to many of the challenges that we encountered such as the computational intractability, the non-stationarity of noise, missing data values, and the choice of the kernel. In short, we provide a box of tools for learning traffic dynamics with GPs. The models that we obtained look promising, but there are some major differences to the piecewise- affine cell transmission model that we did not expect. This is probably not an issue because there are some major differences between the CTM and our setup and we consequently expect to see some discrepancies. Still, our models should be evaluated in data-based simulations in a next step such that the predictions can be compared with actual data. The approach towards learning a model that we proposed works well in general, but it is still far from being fully automated. Ensuring that the quality of the learned model is satisfying in all circumstances will be the core challenge when designing a reinforcement learning control system based on Gaussian Processes.

2.1.2 Controller Learning for SGPs

M. Schmitt, based on a report by N. Hail

Gaussian Process (GP) Models have been proposed as a data-driven, parameter-free approach for system identification. Based on these type of models, algorithms like PILCO (Probabilistic Inference for Learning Control) have been derived, which optimize controller parameters for a given GP model and a given controller structure. Although no theoretical guarantees for convergence exist to this point, comparable or superior performance in comparison to traditional reinforcement learning techniques has been reported for certain applications. However, training GP models for high-dimensional systems is prohibitive so far, due to the large number of training data necessary to represent the dynamics in a high-dimensional space. We propose a new framework for structured GP models, primarily motivated by applications in traffic modeling and control. For traffic models, large parts of the systems dynamics are perfectly known, but certain parts of the dynamics, between small subsets of states, are affected by large uncertainty. In our framework, we allow to define systems as a combination of linear dynamics (assumed to be known a-priori) and unknown, preferably low-dimensional GPs. We derive and implement the equations necessary to train controller parameters for such models in a PILCO-like setting and demonstrate the efficiency of our approach in simulations.

Problem description

A novel reinforcement learning approach has been proposed Deisenroth et al. (2015), which uses Gaussian Processes to identify system dynamics in a model-free, data-driven way. The original approach considers nonlinear, discrete-time systems given as

$$x_{t+1} = f(x_t, u_t) + w, \quad w \sim \mathcal{N}(0, \sigma_w)$$

with continuous valued states $x_t \in \mathbb{R}^D$ and controls $u_t \in \mathbb{R}^F$. The objective is to determine controller parameters θ such that a parametrized controller $\pi_\theta : \mathbb{R}^D \to \mathbb{R}^F$, $u_t := \pi_\theta(x_t)$ minimizes the expected long-term cost

$$J^\pi(\theta) = \sum_{t=0}^{T} \mathbb{E}_{x_t}[c(x_t)], \quad x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$$

for some suitable stage-cost $c(x_t)$. As a model-based reinforcement learning technique, the proposed approach uses data gained by experiments to first identify a suitable model. For some mean and kernel as described in Section 2.1.1, a GP is used to approximate a stochastic, one-step prediction model

$$\begin{aligned}\mathbb{E}_{GP}[\Delta_t] &= m_{GP}(\tilde{x}_t) = k_*^\top (K + \sigma_w^2 \mathbb{I})^{-1} y \\ \mathrm{var}_{GP}[\Delta_t] &= k_{**} - k_*^\top (K + \sigma_w^2 \mathbb{I})^{-1} k_*\end{aligned}$$

with $\Delta_t := x_{t+1} - x_t, \;\; \tilde{x}_t := (x_t, u_t)$ and the training targets y and correlation terms k_{**}, k_* and K defined as in Section 2.1.1. The former section also provides detailed derivations for the posterior distribution and a discussion of suitable prior mean and kernel functions for GPs, with a particular focus on traffic models. These details are omitted here fore brevity.

In a second step, the identified GP model is used in order to approximate the expected long-term costs $J^\pi_{GP}(\theta) \approx J^\pi(\theta)$ and an optimization $\theta^* = \underset{\theta}{\operatorname{argmin}}\, J^\pi_{GP}(\theta)$ is performed. The main advantage of GP models is that they are nonparametric, stochastic system models while still allowing for a tractable approximation of the long-term cost such that the optimization over the controller parameters can be performed efficiently. The procedure consisting of data collection, model fitting and optimization of controller parameters is repeated until sufficient control performance is achieved. Algorithm 1, published as PILCO ("Probabilistic Inference for Learning Control") in Deisenroth et al. (2015); Deisenroth and Rasmussen (2011), summarizes the approach.

Algorithm 1 PILCO

```
init: Sample controller parameters $\boldsymbol{\theta} \sim \mathcal{N}(0, I)$. Apply random control signals and record data.
while task not learned do
    Learn probabilistic (GP) dynamics model, using all data
    while no convergence do
        Approximate inference for policy evaluation, get $J^{\pi}(\boldsymbol{\theta})$
        Gradient-based policy improvement, get $\mathrm{d}J^{\pi}(\boldsymbol{\theta})/\mathrm{d}\boldsymbol{\theta}$
        Update parameters $\boldsymbol{\theta}$
    end while
    $\pi^* \leftarrow \pi(\boldsymbol{\theta}^*)$
    Apply $\pi^*$ to system and record data
end while
```

Algorithm 1: PILCO algorithm

Methods

The PILCO algorithm is promising for applications in traffic control since it provides a data-driven, model-free approach for systems that may be subject to significant uncertainty. However, the algorithm cannot be applied in a straightforward manner because traffic control problems are usually high-dimensional, making global GP models impractical.

Therefore, we present an extension in which we combine low-dimensional GP models, which approximate the dynamics between a small subset of states, with linear dynamics, which couple all states. Let M, N and O be matrices of suitable dimensions, we define *Structured Gaussian Process Models* (SGP) as

$$x_{t+1} = M \cdot GP(x_t, u_t) + N \cdot x_t + O \cdot u_t + w_t, \quad w \sim \mathcal{N}(0, \Sigma_w)$$

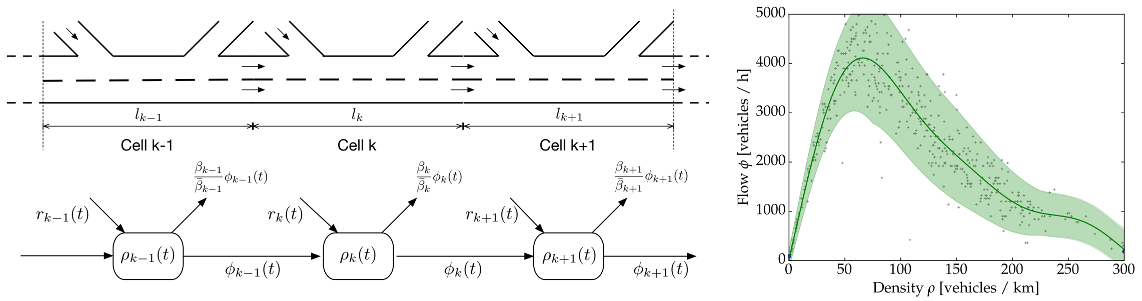
with

$$GP(x_t, u_t) = \begin{pmatrix} GP_1(x_t(\mathcal{I}_1^x), u_t(\mathcal{I}_1^u)) \\ GP_2(x_t(\mathcal{I}_2^x), u_t(\mathcal{I}_2^u)) \\ \vdots \\ GP_m(x_t(\mathcal{I}_m^x), u_t(\mathcal{I}_m^u)) \end{pmatrix}.$$

The vector valued function $GP(x_t, u_t)$ is comprised of scalar-valued, independent GPs $GP_k(x_t(\mathcal{I}_k), u_t(\mathcal{I}_k)$, which depend ideally only on a small subset of the states and inputs denoted bye the index sets $\mathcal{I}_k^x \subseteq \{1, \dots, D\}$ and $\mathcal{I}_k^u \subseteq \{1, \dots, F\}$. For ease of notation, we introduce the auxiliary variables $z_t := GP(x_t, u_t)$.

We use SGP models to obtain one-step predictions with mean

$$\mu_{x_{t+1}} = M \cdot \mu_{z_t} + N \cdot \mu_{x_t} + O \cdot \mu_{u_t}$$


l_{k-1}
l_k
l_{k+1}
Cell k-1
Cell k
Cell k+1
$\frac{\beta_{k-1}}{\bar{\beta}_{k-1}}\phi_{k-1}(t)$
$\frac{\beta_k}{\bar{\beta}_k}\phi_k(t)$
$\frac{\beta_{k+1}}{\bar{\beta}_{k+1}}\phi_{k+1}(t)$
$r_{k-1}(t)$
$r_k(t)$
$r_{k+1}(t)$
$\rho_{k-1}(t)$
$\rho_k(t)$
$\rho_{k+1}(t)$
$\phi_{k-1}(t)$
$\phi_k(t)$
$\phi_{k+1}(t)$
Flow ϕ [vehicles / h]
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
Density ρ [vehicles / km]

(a) Freeway topology with traffic flows

(b) Fundamental diagram

Figure 2.18: In the CTM, a freeway is modeled by a combination of conservation equations according to the freeway topology with on- and offramps, as depicted on the left and the fundamental diagram, which characterizes the flow as a function of the densities of upstream and downstream cells, as depicted on the right.

and variance

$$\begin{aligned}\Sigma_{x_{t+1}} &= N\Sigma_{x_t}N^\top + O\cdot\mathrm{cov}(x_t,u_t)^\top N^\top + M\cdot\mathrm{cov}(x_t,z_t)^\top N^\top + \\ &+ N\cdot\mathrm{cov}(x_t,u_t)O^\top + O\Sigma_{u_t}O^\top + M\cdot\mathrm{cov}(u_t,z_t)^\top O^\top \\ &+ N\cdot\mathrm{cov}(x_t,z_t)M^\top + O\cdot\mathrm{cov}(u_t,z_t)M^\top + M\Sigma_{z_t}M^\top .\end{aligned}$$

The posterior distribution of the GP variables z_t is obtained as usual

$$\begin{aligned}\mu_{z_t} &= m_{GP}(\tilde{x}_t) = k_*^\top(K+\sigma_w^2\mathbb{I})^{-1}y \\ \Sigma_{z_t} &= k_{**} - k_*^\top(K+\sigma_w^2\mathbb{I})^{-1}k_*\end{aligned}$$

and the covariances between x_t, u_t and z_t can readily be computed. Following the lines of Deisenroth et al. (2015), the long-term costs can now be approximated and the optimization of the controller parameters θ can be performed as in PILCO. We conclude this section with an example that serves to exemplify why the structure of SGPs naturally arises for traffic control problems.

Example 2.1.1. *Consider the problem of freeway ramp metering, for a single freeway with its dynamics given by the Cell Transmission Model (CTM) Daganzo (1994b); Gomes and Horowitz (2006b). In the CTM, the freeway is partitioned into cells of length l_k. The state of the mainline is described by the traffic density $\rho_k(t)$ of cell k at time t, i.e. the number of cars per length in each cell. The density evolves over time according to the conservation equation*

$$\rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k(t) - \frac{1}{\bar{\beta}_k}\phi_k(t)\right)$$

with the mainline flows are denoted as $\phi_k(t)$ and the metering rates as $r_k(t)$. The CTM models the flow $\phi_k(t)$ as a function of the traffic densities, which is represented by the so-called fundamental diagram, depicted in Figure 2.18b.

It is easy to see that the conservation law is linear in states $\rho(t)$, inputs $r(t)$ and flows $\phi(t)$. The flows, however, are nonlinear, stochastic functions of only two states each. In an SGP setting, low-dimensional GPs are used to model the fundamental diagram, using the techniques described in Section 2.1.1 and the data that are readily available in the SPEEDD project. By contrast, the conservation law is known without any uncertainty and can be implemented in an SGP setting without any approximations.

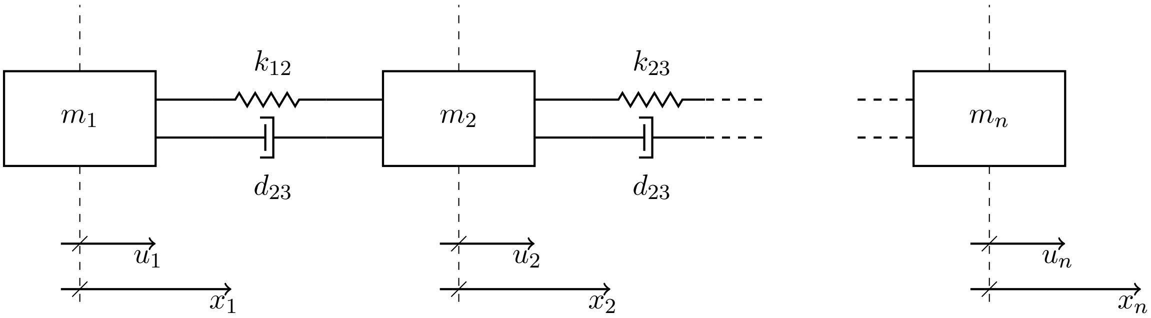


Figure 2.19: Coupled N-Mass Spring System with Damping.

Results

In order to asses the performance of the proposed, PILCO-like algorithm based on SGPs, we conduct a case study for a set of systems similar in structure to the CTM freeway. We consider a chain of spring-damper coupled masses as depicted in Figure 2.19. We seek to stabilize these chain of coupled masses via control inputs exerting a force on every second mass in the chain. We assume that the spring forces are nonlinear and a-priori unknown, but a data set is available which can be used to approximate the spring forces using GPs.

We first asses the computational complexity of the SG-PILCO algorithm. Therefore, we train a E GPs (one for every unknown functions f), using n simulated outputs $Y_{train} \in \mathbb{R}^{(D \times n)}$. Here, D is the dimensionality of the states and F is the dimensionality of the inputs. To determine the complexity of the hyperparameter training we make the assumption that the feature space of every GP consists of an equivalent number of data points $\hat{n}$ and has similar dimensionality $\hat{D}$. Given these assumptions, it takes $\mathcal{O}(\hat{n}^3)$ calculations per step to train the hyperparameter with iterative evidence maximization. In contrast to the PILCO algorithm, we only need to fit a GP for each of the E unknown functions instead of each state space dimension D. For low-dimensional GPs with $E \ll D$, we calculate the total computational load per evidence maximization step as:

$$\textbf{PILCO:} \quad \mathcal{C}_{Fit} \sim \mathcal{O}(n^3 D) \tag{2.50}$$

$$\textbf{SG-PILCO:} \; \mathcal{C}_{Fit} \sim \mathcal{O}(\hat{n}^3 E) \tag{2.51}$$

Once the dynamics model is fitted we can perform approximate inference steps to determine long term predictions. The computationally most expensive step requires us to calculate the variance $\mathbf{\Sigma}_w$. Calculating this variance has computational complexity $\mathcal{O}(\hat{n}^2\hat{D})$ as stated in Deisenroth et al. (2015). Again SG-PILCO profits from (possibly) lower feature space dimensions $\hat{D} \leq D$ and $\hat{n} \leq n$. The total complexity of an inference step using linearization is thus:

$$\textbf{PILCO:} \quad \mathcal{C}_{InfLin} \sim \mathcal{O}(n^2 D^2) \tag{2.52}$$

$$\textbf{SG-PILCO:} \; \mathcal{C}_{InfLin} \sim \mathcal{O}(\hat{n}^2 \hat{D} E) \tag{2.53}$$

The most expensive operation, when calculating the analytical gradients is the calculation of derivatives with respect to the state variance $\partial\mathbf{\Sigma}_{x_{t+1}}/\mathbf{\Sigma}_{x_t}$. They result in a $E \times E \times D \times D$ tensor. In one time step and for fixed training points, they thus require at least a computational complexity of:

$$\mathcal{C}_{Grad.} \in \omega(D^2 E^2) \tag{2.54}$$

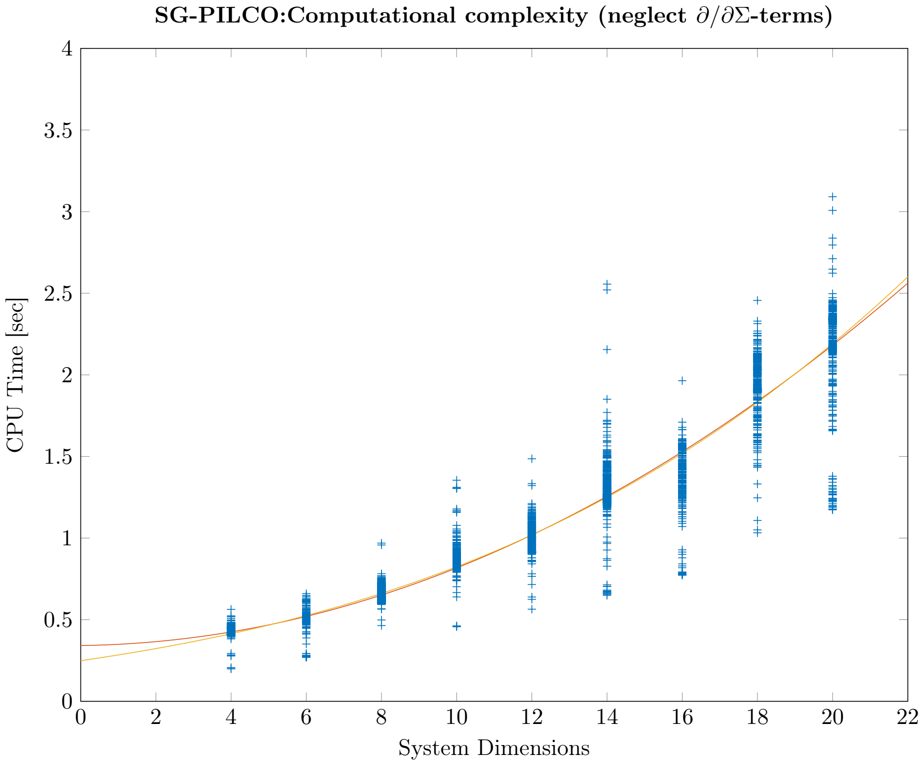


Figure 2.20: This plot illustrates a 2nd and 3rd order polynomial fit of the CPU-time required for the gradient calculation in one time step. Both the 2nd and the 3rd order polynomial have significant parameters in every entry: $p_3(x) = 0.0001 \cdot x^3 + 0.0016 \cdot x^2 + 0.0336 \cdot x + 0.2488$, $p_2(x) = 0.0044 \cdot x^2 + 0.0031 \cdot x + 0.3416$., whereas the 3rd order fit shows a smaller residual.

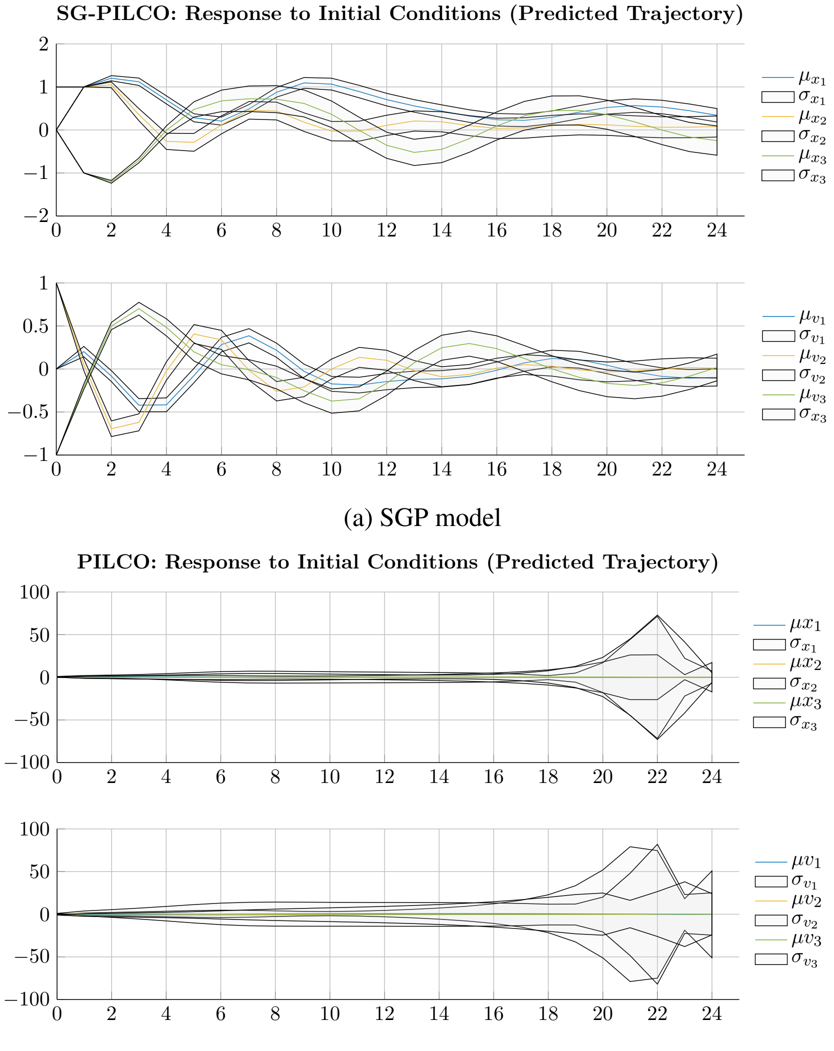
Note, $g \in \omega(f)$ is the Bachman-Landau notation for f dominates g asymptotically. To prevent the computational complexity from growing to fast for large dimensional systems, we therefore propose to only consider covariances of states that affect over the course of the next two to three time steps. Covariance terms hat have no immediate effect on each other are neglected and set to zero. Such an approximation would allow us to reduce the computational complexity of the gradient calculation for one time step and for fixed training points to:

$$\mathcal{C}_{Grad.} \in \omega(DE^2) \tag{2.55}$$

It remains to asses the performance of the proposed algorithms in terms of the control objective. We demonstrate that the inclusion of the additional prior knowledge of the linear part of the dynamics leads to a significant improvement in the predictive performance of the estimated models as is can be concluded from Figure 2.21. This effect is not surprising at all, however, the proposed SGP approach allows us to use this prior knowledge for the first time.

Conclusions

We have demonstrated that reinforcement learning techniques based on GPs can be extended to high-dimensional systems such as traffic networks if the systems exhibit a distributed structure and can be decomposed into small-dimensional, uncertain subsystems coupled by linear and known dynamics.



(a) SGP model

(b) Full GP model

Figure 2.21: Comparison of long-term predictions for an SGP system model using information about the linear part of the dynamics and a full GP model without any prior information, using the same data set. It can be seen that while the SGP model provides meaningful and accurate predictions, the variance of the full GP model grows over the horizon. Clearly, the full GP model is not suitable for large-dimensional systems, when the amount of data available is insufficient to represent the dynamics in a high-dimensional space.

Knowledge of the linear coupling allows to train GPs only in small subspaces of the large state-space, which keeps GP training computationally feasible and the amount of required data remains small. The proposed SGP models further allow to use the known, linear parts of the dynamics directly in the identified system model while still using to the controller parameter learning methods developed for full GP models. The developed SGP approach is especially suitable for the control of traffic network, in which large parts of the systems dynamics are perfectly known, but certain parts of the dynamics, between small subsets of states, are affected by heavy uncertainty.

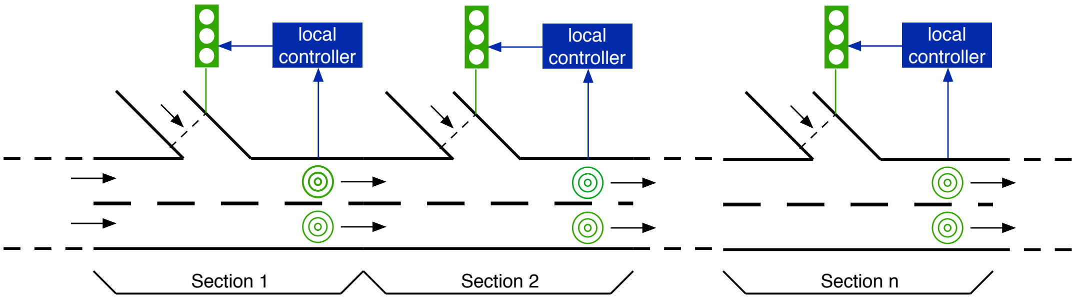


Figure 2.22: Sketch of a freeway with ramp metering controlled by local, decentralized controllers acting on local measurements.

2.2 Low-Level Control

The low-level control algorithms act in a decentralized manner. This means that local control actions, such as traffic light phases or ramp metering rates, are chosen only based on measurements from local, nearby sensors, as depicted in Figure 2.22. The algorithms can be stated in terms of explicit feedback laws as in the approach described in Section 2.2.1 or implicitly in terms of (small-scale) optimization problems as in Section 2.2.3. The main objective of these algorithms is either optimization of some local quantity, e.g. maximization of local flows, or stabilization of some quantity, e.g. the traffic density, at a desired value, or both. Interaction of local controllers with others only happens indirectly via the coordination algorithms described in Section 2.3, which may change the control targets of the local controllers. Similarly, local controllers do not react directly to detected or forecasted events, but they react indirectly via potential change of the control targets set by the coordination algorithms. The main advantage of local, low-level control algorithms is their abliity to react quickly to changes in the local state of the traffic network, estimates of which are provided by State Estimation.

The algorithms described in the following sections address both the freeway and the inner-city traffic control scenario. We first (Section 2.2.1) consider the freeway ramp metering problem and compare the performance of distributed, non- predictive feedback strategies to the optimal solution. We propose a simple feedback law and derive sufficient conditions under which the distributed controller achieves globally optimal performance. Next (Section 2.2.2), we take a more theoretical stance and consider the class of convex, monotonic systems. We show that such systems in general, and monotone freeways in particular, are optimally operated in steady-state, which is an important presupposition for efficient decentralized control. Finally (Section 2.2.3), we consider the inner-city traffic optimal control problem, which is difficult to solve exactly because of the presence of binary variables in the problem, rendering the optimization problem nonconvex. An approximate decision-making scheme is presented, that only relies on linear optimization and is therefore very efficient and scalable.

2.2.1 Optimal, Distributed Freeway Ramp Metering

M. Schmitt, C. Ramesh, J. Lygeros

We consider the freeway ramp metering problem, based on the Cell Transmission Model. Assuming perfect model knowledge and perfect traffic demand prediction, the ramp metering problem can be cast as a finite horizon optimal control problem with the objective of minimizing the Total Time Spent, i.e., the sum of the travel times of all drivers. For this reason, the application of Model Predictive Control (MPC) to the ramp metering problem has been proposed. However, practical tests on freeways show that MPC rarely outperforms simple, decentralized feedback strategies. Until now, a theoretical justification for this empirical observation is lacking. This work compares the performance of distributed, non-predictive feedback strategies to the optimal solution. To do so, we propose a distributed, non-predictive controller, and show that this controller preserves monotonicity of the closed-loop system. We exploit this result to derive sufficient conditions under which the distributed controller achieves globally optimal performance. In a small case study based on real-world traffic data, we also demonstrate that these optimality conditions are rarely violated on a typical day. In addition, even in cases when they are violated, the performance deterioration is shown to be negligibly small.

Introduction

Ramp metering refers to the active control of the inflow of cars on a freeway via the onramps, by means of installing and controlling a traffic light at every onramp. In this work, we consider the freeway ramp metering problem over a finite horizon, e.g. one day or one rush-hour period. Freeway ramp metering has been established as an effective and practically useful tool to improve traffic flows on congestion-prone freeways (Papageorgiou et al., 2003; Papageorgiou and Kotsialos, 2000). We study the problem by adopting the Cell Transmission Model (CTM) for freeways, as introduced in the seminal work by Daganzo (Daganzo, 1994b, 1995), which can be interpreted as a first-order Godunov approximation of the continuous Lighthill-Whitham-Richards-model (LWR) (Lighthill and Whitham, 1955; Richards, 1956). The popularity of the CTM for model-based control stems from the simplicity of the model equations, allowing for computationally efficient solutions methods for optimal control problems (Gomes and Horowitz, 2006b; Ziliaskopoulos, 2000) and the fact that the CTM might be regarded as more realistic than some competing models, since it never predicts negative velocities for freeway traffic. Modifications and generalizations of this model have since been introduced, and we consider a slight generalization of the original, piece-wise affine CTM to a more general, concave fundamental diagram as defined and analyzed in (Lovisari et al., 2014). Furthermore, we employ a simplified onramp model originally introduced as the “asymmetric” CTM (Gomes and Horowitz, 2006b; Gomes et al., 2008), which simplifies the model of onramp-mainline merges by distinguishing mainline- and onramp-traffic demand.

While tractable model-based, optimal-control strategies have been proposed (Gomes and Horowitz, 2006b; Pisarski and Canudas-de Wit, 2012b; Alamir, 2014; Burger et al., 2013), a conceptually different approach uses distributed or even completely decentralized feedback. In such ramp metering strategies, local controllers only receive measurements from sensors in close vicinity to a particular onramp location and only exchange limited amounts of information, preferably with adjacent controllers if at all (Papageorgiou et al., 1991; Stephanedes, 1994; Zhang and Ritchie, 1997). Those control policies aim at maximizing bottleneck flows locally, but have been shown to come close to the performance of optimal ramp metering strategies in real-world evaluations (Smaragdis et al., 2004; Wang et al., 2014; Papamichail et al., 2010b). While it is apparent that such local feedback controllers are far easier to implement than model-based optimal control strategies, it is not obvious why and when the performance of distributed, non-predictive ramp metering strategies comes close to the centralized optimal control solution. A special case for which the optimal control strategy can be explicitly constructed is analyzed in (Zhang and Levinson, 2004). It is stated that the structure of the explicit solution "explains why some local metering algorithms [...] are successful – they are really close to the most-efficient logic". However, no proof of optimality is provided.

This work addresses the question of how distributed, non-predictive ramp metering policies compare to optimal, centralized and predictive control strategies for freeways modeled by the monotonic CTM. If perfect model knowledge and perfect traffic demand prediction is assumed, the ramp metering problem can be cast as a finite horizon optimal control problem with the objective of minimizing the Total Time Spent. It is known that this problem can be reformulated as a convex optimization problem for monotonic demand and supply functions in the fundamental diagram (Lovisari et al., 2014) and as a Linear Program in the case of a piecewise-affine fundamental diagram (Gomes and Horowitz, 2006b). Since perfect model knowledge and in particular perfect traffic demand prediction seem to be unrealistic (see e.g. (Burger et al., 2013; Morbidi et al., 2014; Ojeda et al., 2013) for ongoing work to mitigate the challenges involved), we introduce a local feedback controller for freeway ramp metering instead. This controller performs a distributed, one step ahead maximization of local traffic flows. Exploiting monotonicity of the CTM, we derive sufficient optimality conditions for the problem of minimizing the Total Time Spent (TTS), which is defined as the sum of the travel times of all drivers.

We analyze performance in a case study, based on a real-world freeway described in (de Wit et al., 2015). The study is conducted using the monotonic CTM, with freeway parameters and demand patterns estimated from real measurements. Even though the optimality conditions are occasionally violated, the control performance of the centralized optimal controller and the proposed ideal non-predictive, distributed controller differ only by fractions of percents in simulations based on the CTM. Finally, analyzing differences between the CTM and reality allows us to identify key issues that have to be addressed by control strategies that aim at outperforming existing, distributed feedback controllers for ramp metering in the real world.

To sum up, the main contributions of this paper are:

- We derive sufficient optimality conditions for minimal TTS ramp metering in the CTM, providing a thorough theoretical assessment of why (and when) performance of distributed controllers and optimal control strategies may coincide.
- We demonstrate the usefulness of these conditions by defining a suitable distributed, non-predictive feedback controller, which achieves nearly optimal control performance for simulations of the CTM, using real-world parameters and traffic demands.

Section 2.3.1 defines the CTM and the main problem of minimizing TTS. In Section 2.2.1, we introduce a non-predictive, distributed controller for ramp metering. We also show that this controller is not always optimal, by stating two counterexamples. In Section 2.2.1, we derive sufficient optimality conditions that can be used to check optimality a posteriori. Section 2.3.1 introduces a case study based on real-world freeway parameters, and we demonstrate that the performance deterioration of distributed, best-effort control is negligibly small. Finally, in Section 4, we comment on the implications of the results, and highlight immediate consequences for the development of real-world, predictive, coordinated ramp-metering strategies.

We adopt the following notation: The k-th component of a vector x is denoted x_k. The index k will always be used to denote the index of a cell in the CTM and hence $k \in \{1, \dots, n\}$ with n the number of cells in the model, unless a different range is explicitly specified. If a variable x evolves over

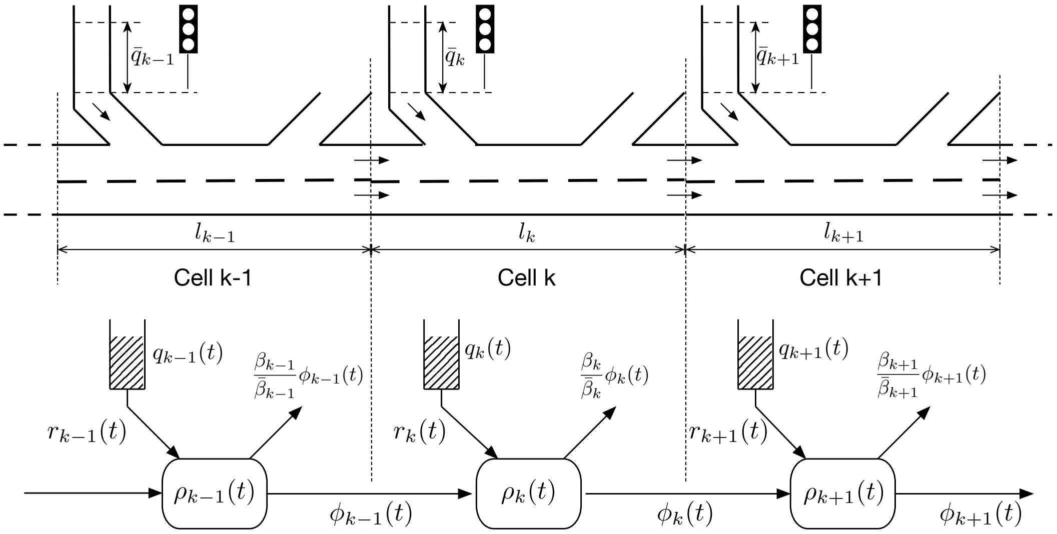


Figure 2.23: Sketch of the CTM of a freeway segment controlled by ramp metering

time, the value at time t is denoted by $x(t)$. We consider discrete time models over a horizon T, i.e., $t \in \{0, \dots, T\}$. The operators $\leq, \geq, <, >$ used with vectors denote elementwise inequalities. Also, we write $[x]_l^u := \min\{u, \max\{x, l\}\}$ for a saturation at an upper bound u and lower bound l.

Problem formulation

The original CTM was introduced in (Daganzo, 1994b, 1995). We consider the (asymmetric) CTM as introduced in (Gomes and Horowitz, 2006b; Gomes et al., 2008), which employs a simplified onramp model, as visualized in Figure 2.37. In the CTM, the freeway is partitioned into *cells* of length l_k. The state of the mainline is described by the traffic *density* $\rho_k(t)$, i.e., the number of cars per length in each cell. The metered onramps are modeled as integrators, and their state is given by the queue length $q_k(t)$, i.e., the number of cars waiting in the queue. The evolution of the system is described by flows of cars during discrete time intervals of duration Δt. The mainline *flow* between cells k and $k+1$ in time interval t is denoted by $\phi_k(t)$. We make the assumption of constant split ratios, which means that the outflows from the offramps are modeled as percentages β_k of the mainline flows, where β_k are designated as the *split ratios* of traffic at the offramp in cell k. For notational convenience, we also use $\bar{\beta}_k := 1 - \beta_k$. For metered onramps, the inflows to the freeway are given by the ramp *metering rates* $r_k(t)$. The *external traffic demands* $w_k(t)$, i.e., the number of cars per unit time seeking to enter the freeway from either a ramp $k \in \{1, \dots, n\}$ or from the upstream mainline, act as external disturbances on the system. The evolution of the state of the CTM is described by the conservation laws

$$\rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k(t) - \frac{1}{\bar{\beta}_k}\phi_k(t)\right), \tag{2.56a}$$

$$q_k(t+1) = q_k(t) + \Delta t \cdot (w_k(t) - r_k(t)). \tag{2.56b}$$

The metering rates $r_k(t)$ serve as control inputs to the system. The onramp model relies on the implicit assumption that congestion does not spill back onto the onramps. In particular, the model assumes that all cars assigned to enter the freeway in a sampling interval can indeed do so. While this assumption might not be satisfied for an uncontrolled freeway, it was shown to be satisfied for a freeway controlled by ramp metering in a case study (Gomes and Horowitz, 2006b). In this study, the upper bounds on the metering rates as determined by the mainline congestion exceeded the actual metering rates by a large margin. Naturally, the ramp metering rates are nonnegative and also subject to a constant upper bound

$$0 \le r_k(t) \le \bar{r}_k, \tag{2.57}$$

which characterizes the finite maximal onramp flow. We will assume that the external demand is bounded $0 \le w_k(t) \le \bar{r}_k$ for all k and for all times t. In addition, the limited space on the onramp $0 \le q_k(t) \le \bar{q}_k(t)$ potentially limits the allowed ramp metering rates further and using (2.56b), we obtain bounds

$$\frac{1}{\Delta t}\left(q_k(t) - \bar{q}_k\right) + w_k(t) \le r_k(t) \le \frac{1}{\Delta t} q_k(t) + w_k(t) \tag{2.58}$$

in terms of the metering rates. The upper bound simply ensures that only cars seeking admission in time interval t can be admitted to enter the freeway, while the lower bound mandates that the ramp is operated such that the queue length on the onramp never exceeds the length of the ramp, to avoid queue spill back[4]. We do not impose similar bounds on the outflows via the offramps: Effects like a spill back of congestion from the adjacent aterials are beyond the scope of this model and a constant, upper bound on the outflow from an offramp is redundant under the assumption of constant split ratios.

The CTM is a first order model and therefore, the mainline traffic flows $\phi(t)$ are not states but functions of the densities. The mainline flow $\phi_k(t)$ depends on the traffic *demand* $d_k(\rho_k(t))$ proportional to the number of cars that seek to travel downstream from cell k at time t, and the *supply* $s_{k+1}(\rho_{k+1}(t))$ of free space available downstream in cell $k+1$ at time t. Demand- and supply functions are often identified as the *fundamental diagram* of a cell, as depicted in Figure 2.24. In the original work of (Daganzo, 1994b, 1995), a piecewise-affine (PWA) fundamental diagram was assumed, as it is obtained from the Godunov discretization of the LWR-model. In practice, one might want to consider more general shapes of the fundamental diagram and hence of the demand- and supply functions, in order to better approximate real world data (see e.g. (Karafyllis et al., 2016; Lovisari et al., 2014) for recent examples). In the remainder of this work, we will assume that:

Assumption 2.2.1. *For every cell k, define a maximal density $\bar{\rho}_k$, often called the jam density, and the critical density $0 < \rho_k^c < \bar{\rho}_k$. The demand $d_k(x)$, $d_k : [0, \bar{\rho}] \to \mathbb{R}_+$ is a Lipschitz-continuous, nondecreasing function with $d_k(0) = 0$ and $d_k(\rho_k^c) = d_k(\bar{\rho}_k)$. Conversely, the supply $s_k(x)$, $s_k : [0, \bar{\rho}] \to \mathbb{R}_+$ is a Lipschitz-continuous, nonincreasing function with $s_k(\bar{\rho}) = 0$ and $s_k(\rho_k^c) = s_k(0)$. Furthermore, the sampling time Δt is chosen such that it satisfies the bounds*

$$\Delta t \cdot c_k^d \le l_k \bar{\beta}_k \quad \forall k, \tag{2.59a}$$

$$\Delta t \cdot c_k^s \le l_k \quad \forall k, \tag{2.59b}$$

with respect to the Lipschitz constants of the demand c_k^d and of the supply c_k^s.

The flow

$$\phi_0(t) = w_0(t), \tag{2.60a}$$

$$\phi_k(t) = \min\{d_k(\rho_k(t)), s_{k+1}(\rho_{k+1}(t))\} \quad \forall k \in \{1, \dots, n-1\}, \tag{2.60b}$$

$$\phi_n(t) = d_n(\rho_n(t)), \tag{2.60c}$$

[4] The effects of congestion spill back from the onramps to the arterials is usually considered to have much worse effects than a congestion on the freeway mainline. Thus, avoiding such a spill back takes priority over avoiding a congestion on the mainline.

is now given as the minimum of upstream demand and downstream supply. Note that the assumption of monotonicity of the demand functions implies that $d_k(\rho_k) = d_k(\rho_k^c)\ \forall\ \rho_k : \rho_k^c \leq \rho_k \leq \bar{\rho}_k$, i.e., the traffic demand is constant for densities larger than the critical density. Similarly, the assumption on monotonicity of the supply function implies that the supply is constant for all densities smaller than the critical density, i.e., $s_k(\rho_k) = s_k(\rho_k^c)\ \forall\ \rho_k : 0 \leq \rho_k \leq \rho_k^c$. It is sometimes helpful to introduce the maximal flow $F_k := \min\{d_k(\rho_k^c), s_{k+1}(\rho_{k+1}^c)\}$ as an additional parameter in the flow equations. The flow equation (2.60b) can then equivalently be written as

$$\phi_k(t) = \min\{d_k(\rho_k(t)), F_k, s_{k+1}(\rho_{k+1}(t))\}.$$

We will use F_k in the flow equations only when it simplifies the presentation of a proof or of an example. The flow equations are slightly modified for the first and the last cell. For the last cell, we assume that the outflow from the freeway is unobstructed, i.e., no congestion exists downstream of the (part of the) freeway that is modeled. Similarly, the inflow in the first cell is given by the external traffic demand $w_0(t)$, which is *not* restricted by the supply in the first cell. The reason for this modification is that the flow into the first cell arises from external traffic demand. The equations therefore ensure that all external demand eventually will be served. By contrast, limiting the inflow according to the supply of free space, as it is done for any internal flow, would amount to discarding the surplus external demand. It is worth emphasizing that local traffic flows (2.60) are maximized at the critical density. In particular, the flow ϕ_k into some cell $k+1$ with density equal to or smaller than the critical density $\rho_{k+1}(t) \leq \rho_{k+1}^c$ is not constrained by the supply of free space and hence $\phi_k(t) = \min\{d_k(\rho_k(t)), F_k\}$. Similarly, the flow out of a cell k with density equal to or greater than the critical density $\rho_k(t) \geq \rho_k^c$ is not constrained by the traffic demand and hence $\phi_k(t) = \min\{F_k, s_{k+1}(\rho_{k+1}(t))\}$. A CTM freeway model that satisfies Assumption 2.2.1 will be called a *monotonic CTM*. The classical, piecewise affine fundamental diagram satisfies this assumption: Here, we have that $d_k(\rho_k) = \min\left\{\bar{\beta}_k v_k \rho_k, \frac{w_k}{\beta_k v_k + w_k}\bar{\rho}_k\right\}$ and $s_k(\rho_k) = \min\left\{\frac{w_k}{\beta_k v_k + w_k}\bar{\rho}_k, (\bar{\rho}_k - \rho_k)w_k\right\}$ with v_k the free-flow speed and w_k the congestion wave speed. Monotonicity assumptions are trivially satisfied for affine demand and supply functions and the condition on Δt can be recognized as the stability condition $v_k \cdot \Delta t \leq l_k,\ \forall k$ that arises if the CTM is derived as a discretization of the wave PDE using the Godunov scheme. Note that in practice, the congestion-wave speed w_k is significantly lower than the free-flow speed, thus the upper bound in inequality 2.59b in Assumption 2.2.1 is not restrictive. Different shapes of the fundamental diagram encountered in practice are depicted in Figure 2.24 for illustration.

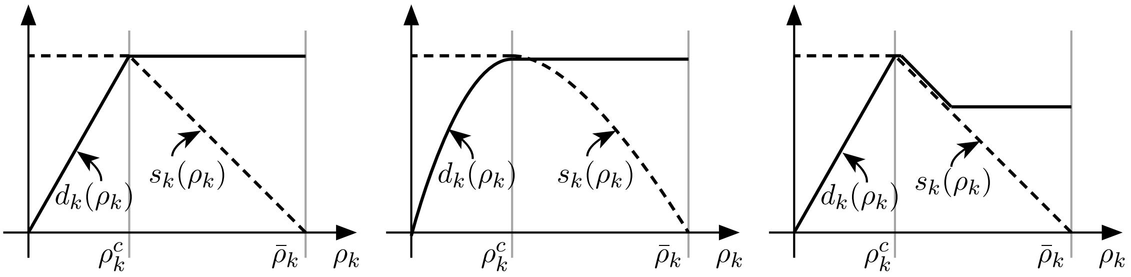
We are now ready to formulate the main problem. Assume a freeway modeled by the CTM, subject to a demand profile $d_k(t)$ for $k \in \{0, \ldots, n\}$ over a horizon $t \in \{0, \ldots, T-1\}$. By controlling the metering rates, we seek to minimize the total time spent (TTS), given as the sum of the total travel time (TTT),

$$TTT := \Delta t \cdot \sum_{t=0}^{T} \sum_{k=0}^{n} l_k \rho_k(t),$$

which is the sum of the travel times of all drivers on the freeway and the total waiting time (TWT),

$$TWT = \Delta t \cdot \sum_{t=0}^{T} \sum_{k=0}^{n} q_k(t),$$

the sum of the waiting time of all drivers on the metered onramps. The main problem is therefore given



(a) PWA FD (b) Concave FD (c) Nonmonotonic FD

Figure 2.24: Different shapes of the fundamental diagram (FD) may be desirable in order to approximate real-world data. Figure (a) depicts the traditional PWA version. Figure (b) shows a version with monotonic and concave demand and supply functions, satisfying Assumption 2.2.1 for suitable Δt. Clearly, the class of monotonic, concave fundamental diagrams is a generalization of the PWA fundamental diagram. Figure (c) shows a fundamental diagram with non-monotonic demand function, which does *not* satisfy Assumption 2.2.1 for any Δt.

as:

$$\begin{aligned}
\text{minimize}\quad & TTS := TTT + TWT = \Delta t \cdot \sum_{t=0}^{T}\sum_{k=0}^{n}(l_k\rho_k(t) + q_k(t)) && \text{(2.61a)}\\
\text{subject to}\quad & \rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k(t) - \frac{1}{\bar{\beta}_k}\phi_k(t)\right) && \text{(2.61b)}\\
& q_k(t+1) = q_k(t) + \Delta t \cdot (w_k(t) - r_k(t)) && \text{(2.61c)}\\
& \phi_0(t) = w_0(t) && \text{(2.61d)}\\
& \phi_k(t) = \min\{d_k(\rho_k(t)), s_{k+1}(\rho_{k+1}(t))\} && \text{(2.61e)}\\
& \phi_n(t) = d_n(\rho_n(t)) && \text{(2.61f)}\\
& 0 \le r_k(t) \le \bar{r}_k && \text{(2.61g)}\\
& 0 \le q_k(t) \le \bar{q}_k && \text{(2.61h)}\\
& \rho_k(0), q_k(0), w_k(t) \text{ given.} && \text{(2.61i)}
\end{aligned}$$

Note that the metering bounds (2.58) for time t are equivalent to (2.61h) at time $t+1$. Problem (2.61) is a standard problem in traffic control, which has been studied extensively. It is non-convex, because of the nonlinear fundamental diagram (2.61e). In (Gomes and Horowitz, 2006b), it is shown that the special case of a piecewise-affine fundamental diagram admits an LP reformulation and hence can be solved efficiently. The case of a concave, but not necessarily PWA fundamental diagrams is briefly discussed in (Lovisari et al., 2014), although in a continuous-time setting. Similarly to the PWA case, a convex reformulation is possible.

Remark 2.2.1. *In the statement of the main problem* (2.61)*, we introduce parameters* $\bar{r}_k$*,* $\bar{q}_k$ *and* β_k *for every cell. In practice, not every cell will be equipped with both an onramp and an offramp and many cells might have neither. The equations* (2.61) *are general enough to capture all these situations and one can simply select* $\bar{r}_k = \bar{q}_k = w_k(t) = 0,\ \forall t$*, if no onramp is present in cell* k*. Similarly, choosing* $\beta_k = 0$*, which in turn implies that* $\bar{\beta}_k = 1$*, means no offramp is present in cell* k*.*

A distributed controller

The problem of minimizing TTS represents the global perspective, in which we optimize the whole system over the complete horizon. By contrast, one might also seek to optimize the here-and-now performance. This idea can be formalized by introducing the Total Travel Distance (TTD), defined as

$$TTD(t) := \Delta t \cdot \sum_{k=1}^{n} l_k \phi_k(t),$$

which is simply the total distance traveled by all drivers on the mainline within time interval t. In this section, we formulate an one-step look-ahead controller, which maximizes TTD for the next time step in a greedy fashion:

$$\begin{aligned}
\text{maximize} \quad & TTD(t+1) = \Delta t \cdot \sum_{k=1}^{n} l_k \phi_k(t+1) && \text{(2.62a)}\\
\text{subject to} \quad & \rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k(t) - \frac{1}{\bar\beta_k}\phi_k(t)\right) && \text{(2.62b)}\\
& q_k(t+1) = q_k(t) + \Delta t \cdot (w_k(t) - r_k(t)) && \text{(2.62c)}\\
& \phi_0(t+1) = w_0(t+1) && \text{(2.62d)}\\
& \phi_k(t+1) = \min\{d_k(\rho_k(t+1)), s_{k+1}(\rho_{k+1}(t+1))\} && \text{(2.62e)}\\
& \phi_n(t+1) = d_n(\rho_n(t+1)) && \text{(2.62f)}\\
& 0 \le r_k(t) \le \bar r_k && \text{(2.62g)}\\
& 0 \le q_k(t+1) \le \bar q_k && \text{(2.62h)}\\
& \rho_k(t), q_k(t), w_k(t) \text{ given.} && \text{(2.62i)}
\end{aligned}$$

It is straightforward to show that the following distributed feedback controller provides a (non-unique) explicit solution to this optimization problem. The explicit, feedback solution requires perfect model knowledge, but it does not rely on any online-optimization or traffic demand prediction:

Lemma 2.2.1. *An explicit solution to the one-step-ahead optimal control problem* (2.62) *is given by the following feedback policy:*

$$r_k^*(t) := \left[\frac{l_k}{\Delta t}\left(\rho_k^c - \rho_k(t)\right) + \frac{\phi_k(t)}{\bar\beta_k} - \phi_{k-1}(t)\right]_{\max\left\{0, \frac{1}{\Delta t}(q_k(t) - \bar q_k) + w_k(t)\right\}}^{\min\left\{\bar r_k, \frac{1}{\Delta t} q_k(t) + w_k(t)\right\}} \quad \forall k \in \{1, \dots, n\}. \quad (2.63)$$

Proof. Consider an arbitrary flow $\phi_k(t+1)$, $k \in \{1, \dots, n-1\}$. It is easy to see that it only depends on the ramp metering rates in the adjacent cells k and $k+1$ at time t, since we consider a one-step-ahead control problem. More precisely, it depends on the traffic demand $d_k(\rho_k(t))$ and the supply of free space $s_{k+1}(\rho_{k+1}(t))$. Consider the problem of maximizing this flow

$$\begin{aligned}
\max_{r(t)} \; \phi_k(t+1) &= \max_{r(t)} \; \min\{d_k(\rho_k(t+1)), s_{k+1}(\rho_{k+1}(t+1))\}\\
&\le \min\Big\{ \underbrace{\max_{r(t)} \; d_k(\rho_k(t+1)),}_{(1)} \; \underbrace{\max_{r(t)} \; s_{k+1}(\rho_{k+1}(t+1))}_{(2)} \Big\}
\end{aligned}$$

alone. It turns out that the policy (2.63) maximizes both terms in the previous equation. In order to show this, we resort to a case distinction:

(i) If $r_k^*(t)$ does not saturate at the upper bound, it follows that $r_k^*(t) \geq \frac{l_k}{\Delta t}\left(\rho_k^c - \rho_k(t)\right) + \frac{\phi_k(t)}{\beta_k} - \phi_{k-1}(t)$ and hence $\rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k^*(t) - \frac{1}{\beta_k}\phi_k(t)\right) \geq \rho_k^c$. Therefore, $d_k(\rho_k(t+1)) = d_k(\rho_k^c)$ and the first term is maximized.

(ii) Otherwise, recall that $d_k(\cdot)$ is nondecreasing. Since $r_k^*(t)$ saturates at the upper bound w.r.t (2.62g) and (2.62h) of the feasible range, the first term is maximized.

Similarly, we can analyze the effects of the metering rate at cell $k+1$:

(i) If $r_{k+1}^*(t)$ does not saturate at the lower bound, it follows that $r_{k+1}^*(t) \leq \frac{l_{k+1}}{\Delta t}\left(\rho_{k+1}^c - \rho_{k+1}(t)\right) + \frac{\phi_{k+1}(t)}{\beta_{k+1}} - \phi_k(t)$ and hence $\rho_{k+1}(t+1) = \rho_{k+1}(t) + \frac{\Delta t}{l_{k+1}}\left(\phi_k(t) + r_{k+1}^*(t) - \frac{1}{\beta_{k+1}}\phi_{k+1}(t)\right) \leq \rho_{k+1}^c$. Therefore, $s_{k+1}(\rho_{k+1}(t+1)) = s_{k+1}(\rho_{k+1}^c)$ and the second term is maximized.

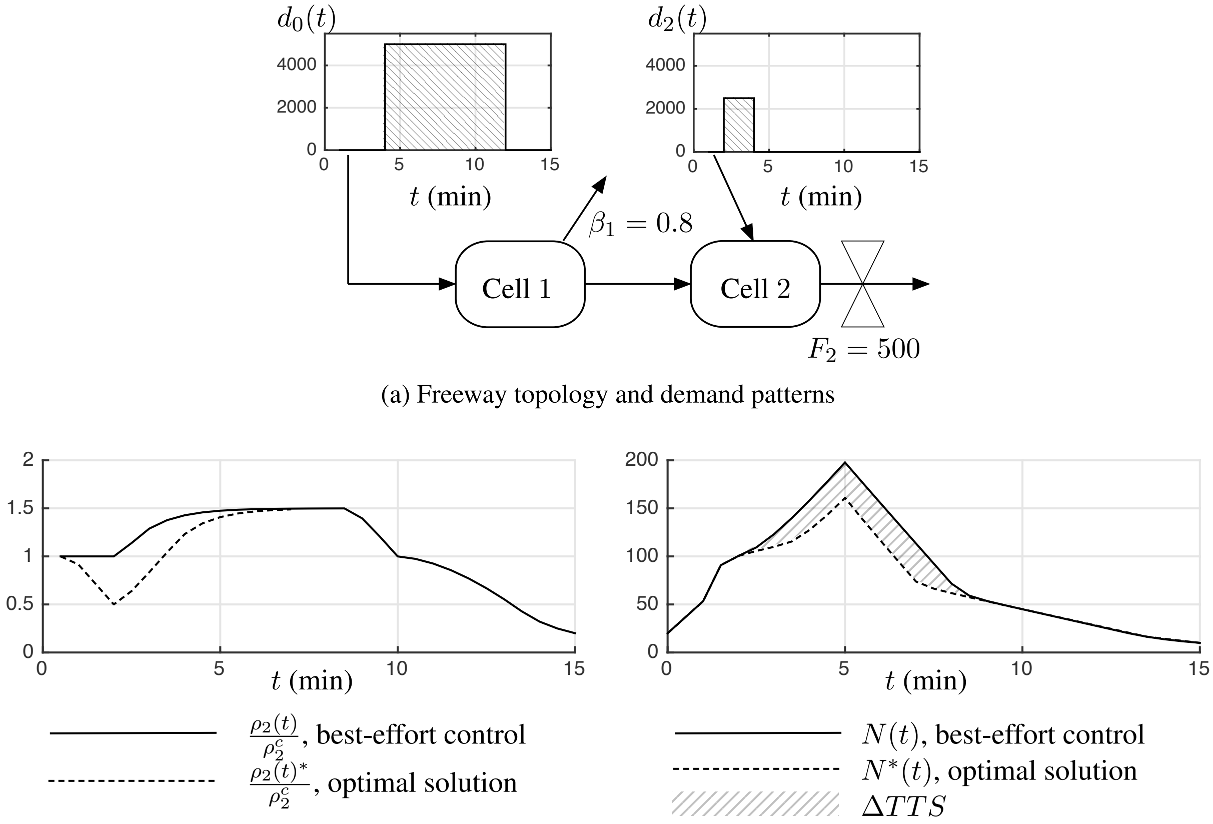
(ii) Otherwise, recall that $s_{k+1}(\cdot)$ is nonincreasing. Since $r_{k+1}^*(t)$ does saturate at the lower bound of the feasible range, the second term is maximized.

Since both the first and the second term are maximized, we conclude that the proposed feedback law maximizes $\phi_k(t+1)$, for all $k \in \{1, \dots, n-1\}$. The case of $\phi_n(t+1)$ is similar, except that the second term is not present. Also $\phi_0(t+1)$ does not depend on the metering rates. All flows are jointly maximized and therefore, TTD is maximized as well. □

This controller aims at moving the local density to the critical density as fast as possible and in that sense, it is very similar in spirit to many existing and practically successful ramp metering strategies, most notably ALINEA (Papageorgiou et al., 1991). In the following, we will be referring to this one-step-ahead optimal controller as the (distributed) *best-effort controller*. There has been speculation in (Zhang and Levinson, 2004) on whether a control policy similar to the feedback law (2.63) minimizes TTS globally and over the complete horizon. We will present two counterexamples showing that this is not necessarily the case in all scenarios.

Example 2.2.1 (Lower bound saturation)**.** *Consider a two-cell freeway as depicted in Figure 2.25, with a metered onramp at the second cell and an offramp with a large split ratio at the first cell, in this example $\beta_1 = 0.8$. Some traffic demand is present at the onramp in the beginning, but a spike in demand is arriving on the mainline with some time delay. The situation is sketched in Figure 2.25a. Because of the large split ratio at the offramp, a backlog of congestion into the first cell severely reduces the total discharge flows and hence increases the TTT. Once the spike in demand arrives, the constraint $0 \leq r_k(t)$ becomes active. However, cars already admitted to the freeway from the onramp can not be retracted to the onramp queue. For suitably chosen parameters, the optimal policy is to preemptively reduce the density in the second cell below the critical density, as depicted in Figure 2.25b. The resulting reduction in outflow from the mainline is more than compensated by the increased discharge from the offramp. The demand is piecewise constant: $d_0(t) = 5000$ for $4 \leq t \leq 12min$, $d_0(t) = 0$ otherwise and $d_2(t) = 2200$ for $3 \leq t \leq 4min$, $d_2(t) = 0$ otherwise. The parameters are chosen as $l_1 = l_2 = 1km$, $v_1 = v_2 = w_1 = w_2 = 100$km/h, $\rho_1^c = 100$(cars)/km, $\rho_2^c = 10$(cars)/km, $\bar{\rho}_1 = 200$(cars)/km, $\bar{\rho}_2 = 20$(cars)/km, $F_1 = 1000$(cars) and $F_2 = 500$(cars). Note that the parameter values have been chosen to amplify the suboptimality of the policy for illustration purposes, without considering whether such parameters are realistic.*

Conversely, we can also construct an example in which the upper bound on the ramp metering rate prevents optimality of the local feedback controller:



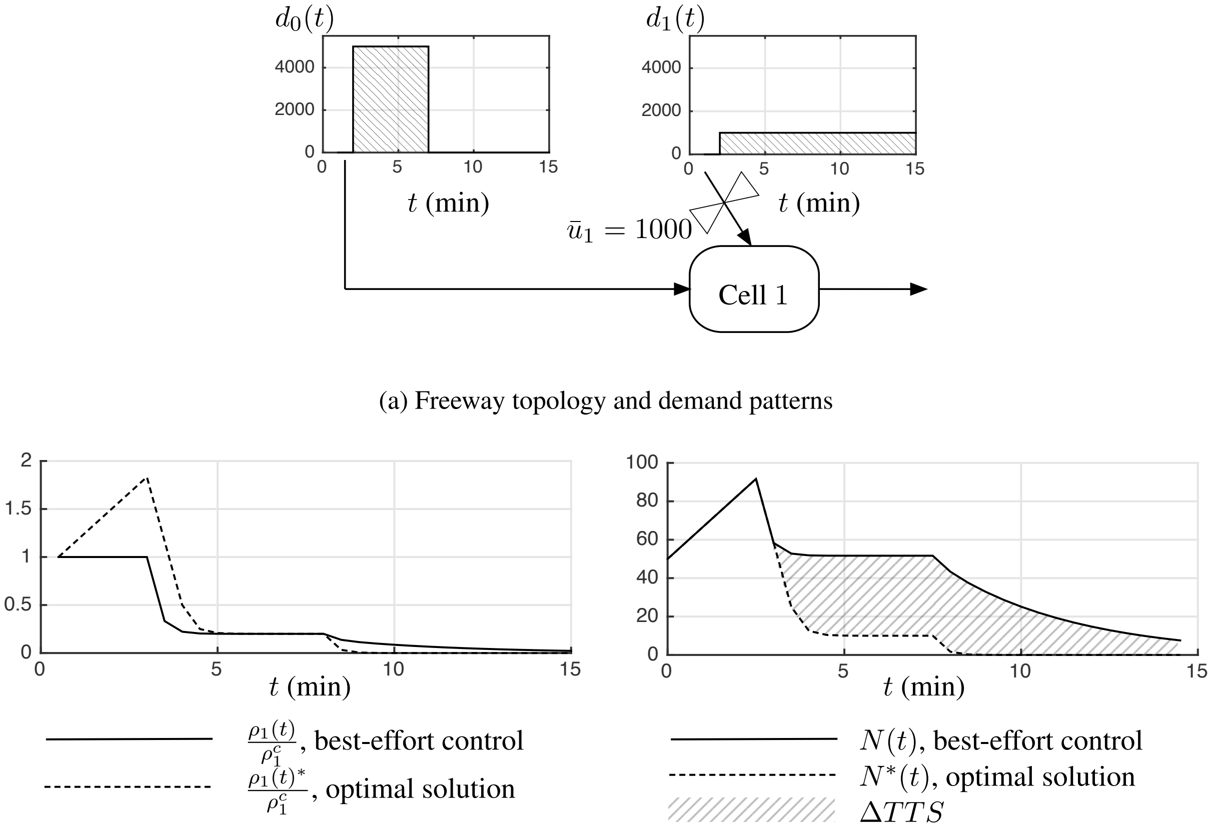
(a) Freeway topology and demand patterns

(b) Traffic evolution

Figure 2.25: In case of a spike in demand, it might be advisable to preemptively reduce the local density at a bottleneck below the critical density, in oder to accomodate (parts of) the traffic wave in the free space and avoid spill back of congestion. The variables $N(t)$ and $N^*(t)$ represent the total number of cars on the freeway and at the onramps, so $N(t) := l_1\rho_1(t)+l_2\rho_2(t)+q_2(t)$ and likewise for $N^*(t)$. Therefore, the savings, i.e., the difference in TTS between the two controllers, are proportional to the highlighted region.

Example 2.2.2 (Upper bound saturation)**.** *Consider a one cell freeway with a metered onramp, as sketched in Figure 2.26a. Some traffic demand is present at the onramp. A spike in mainline demand arrives at the beginning of the considered time interval, but afterwards, the mainline demand decays to zero. In case of local feedback, the initial spike in mainline demand causes a congestion and hence local feedback will use ramp metering to hold cars back on the queue. But when the mainline density decays again, ramp metering will not be able to release cars sufficiently fast to keep the density at the critical density due to inherent limits on the flow from the onramp, which require* $r_k(t) \leq \bar{r}_k$*. The optimal strategy does not use ramp metering at all and allows for a larger congestion in the beginning, as depicted in Figure 2.26b. Since there is no offramp, spill back of congestion is not an issue in this scenario. The demand is piecewise constant:* $d_0(t) = 5000$ *for* $0 \leq t \leq 3min$*,* $d_0(t) = 0$ *otherwise and* $d_1(t) = 1000$ *for* $0 \leq t \leq 8min$*,* $d_1(t) = 0$ *otherwise. The parameter values are chosen as* $l_1 = 1$*km,* $v_1 = 100$*km/h,* $w_1 = 25$*km/h,* $\rho_1^c = 50$*(cars)/km,* $\bar{\rho}_1 = 250$*(cars)/km and* $F_1 = 5000$*(cars). Again, the parameter values have been chosen such as to amplify the suboptimality of the policy for illustration purposes.*

Closer inspection reveals that whether the ramp metering bounds (2.57) are active or not is essential for deriving a certificate of optimality. Indeed, in Section 2.2.1, we derive explicit conditions relating the activation of these constraints to the optimality of the best effort controller.



(a) Freeway topology and demand patterns

(b) Traffic evolution

Figure 2.26: If no spill back of the congestion is possible, it might be advisable (in the monotonic CTM) to allow a congestion to form immediately upstream of a bottleneck (highlighted in the upper plot of Figure 2.26b). The variables $N(t)$ and $N^*(t)$ represent the total number of cars on the freeway and at the onramps, so $N(t) := l_1\rho_1(t) + q_1(t)$ and likewise for $N^*(t)$. Therefore, the savings, i.e., the difference in TTS between the two controllers, are proportional to the highlighted region.

Sufficient optimality conditions

It turns out that even in the presence of controller saturation, we can find sufficient conditions that can be used to verify optimality of a system trajectory, in a minimal-TTS sense. To derive these, we first apply a linear state transformation to the CTM to obtain a system description better suitable for our analysis and show that the closed-loop system comprised of the CTM in the new coordinates and the best-effort controller is monotonic. Then, we introduce conditions that characterize the state of onramps at individual time steps and show how these conditions, together with monotonicity of the closed loop, ensure global minimization of TTS.

Let us introduce the *throughput* $\Phi_k(t)$, defined as

$$\Phi_k(t) =: \Phi_k(0) + \Delta t \cdot \sum_{\tau=0}^{t-1} \phi_k(\tau).$$

with $\Phi_k(0) := \sum_{j=k+1}^{n} \frac{1}{\bar{\beta}_{(k+1,j-1)}} (l_j \rho_j(0))$ and $\bar{\beta}_{(k,j)} := \prod_{i=k}^{j} \bar{\beta}_i$ if $k \leq j$ and $\bar{\beta}_{(k+1,k)} := 1$. This constant offset accounts for the initial density of the freeway, that is cars that have entered the freeway and travelled to cell k or further before the considered time horizon. This offset is required to express densities in terms of throughput; it does not affect our optimality arguments.. Note that $\Phi_k(t)$ is dimensionless and can be interpreted as the number of cars that have passed cell k. Using this insight, we can we express the throughput directly in terms of the original system states: We first extend the model to include an additional cell with index $n+1$ at the end of the freeway. This cell is defined to have infinite storage capacity and collects all of the flow that leaves the freeway via the mainline. An infinite storage capacity can be formalized by choosing $s_{n+1}(\rho_{n+1}) \equiv +\infty$ such that all of the upstream demand is admitted and $d_{n+1}(\rho_{n+1}) \equiv 0$ such that no car leaves the cell. The purpose of this cell is to assist in bookkeeping of all the cars that have travelled through the freeway and it will therefore *not* be considered in the computation of the Total Travel Time. Note that some cars also leave the freeway via the offramps, but because of the assumption of constant split ratios, we can reconstruct the outflows from the densities in cells $k \in \{1, \ldots, n+1\}$. Let us also define the total inflow up to and including time t as $R_k(t) := \Delta t \cdot \sum_{\tau=0}^{t-1} r_k(\tau)$ and the total, external demands as $W_k(t) := q_k(0) + \Delta t \cdot \sum_{\tau=0}^{t-1} w_k(\tau)$. Both quantities are dimensionless and can be interpreted as the number of cars that have entered the freeway or arrived at the onramp, respectively[5]. We can now express the throughputs

$$\Phi_k(t) = \sum_{j=k+1}^{n+1} \frac{1}{\bar{\beta}_{(k+1,j-1)}} \left(l_j \rho_j(t) - R_j(t)\right). \tag{2.64}$$

as linear functions of the original states and inputs of the CTM. The whole CTM can be expressed equivalently in terms of the aggregated quantities:

$$\left.\begin{array}{rrl} \Phi_k(t+1) & = f_k(\Phi(t), R(t)) & := \Phi_k(t) + \Delta t\, \phi_k(t), \\ \text{with:} & \phi_0(t) & = w_0(t), \\ & \phi_k(t+1) & = \min\{d_k(\rho_k(t+1)), s_{k+1}(\rho_{k+1}(t+1))\}, \\ & \phi_n(t) & = d_n(\rho_n(t)) \\ \text{and:} & \rho_k(t) & = \frac{1}{l_k}\left(\Phi_{k-1}(t) - \frac{1}{\beta_k}\Phi_k(t) + R_k(t)\right), \end{array}\right\} \forall k \in \{1, \ldots, n\} \tag{2.65}$$

[5] Note however, that the CTM is an averaged model, so neither $R_k(t)$ nor $W_k(t)$ are restricted to integer values.

in which the throughputs $\Phi_k(t)$ are the states and the inflows $R_k(t)$ are the inputs. Note that we introduced the shorthand notation $f_k(\Phi(t), R(t))$ for the systems equations. We will also use $f(\cdot,\cdot) := \left[f_1(\cdot,\cdot)\dots f_n(\cdot,\cdot)\right]^\top$. We also need to translate the input constraints (2.57) and (2.58), which now read

$$R_k(t-1) \leq R_k(t) \leq R_k(t-1) + \bar{r}_k, \tag{2.66a}$$

$$W_k(t) - \bar{q}_k \leq R_k(t) \leq W_k(t). \tag{2.66b}$$

We will refer to the system described by equations (2.65) and (2.66) as the *aggregated CTM*. Monotonicity properties will facilitate the analysis of optimal control problems of the aggregated CTM.

Lemma 2.2.2. *The systems equations $f_k(\Phi(t), R(t))$ for all $0 \leq k \leq n$ in the aggregated CTM are nondecreasing in the aggregated flows $\Phi(t)$.*

Proof. Using the definition of the aggregated CTM (2.65), we can write equivalently

$$\Phi_k(t+1) = \min\left\{\Phi_k(t) + \Delta t\; d_k(\rho_k(t)), \Phi_k(t) + \Delta t\; s_{k+1}(\rho_{k+1}(t))\right\}$$

for $k \in \{1, \dots, n-1\}$. The minimum of monotonic functions is monotonic. We can therefore verify monotonicity of the CTM by checking that both of the functions

$$f_k^{(-)}(\Phi_{k-1}(t), \Phi_k(t)) := \Phi_k(t) + \Delta t\; d_k\left(\frac{1}{l_k}\left(\Phi_{k-1}(t) - \frac{1}{\bar{\beta}_k}\Phi_k(t) + R_k(t)\right)\right),$$

$$f_k^{(+)}(\Phi_k(t), \Phi_{k+1}(t)) := \Phi_k(t) + \Delta t\; s_{k+1}\left(\frac{1}{l_{k+1}}\left(\Phi_k(t) - \frac{1}{\bar{\beta}_{k+1}}\Phi_{k+1}(t) + R_{k+1}(t)\right)\right),$$

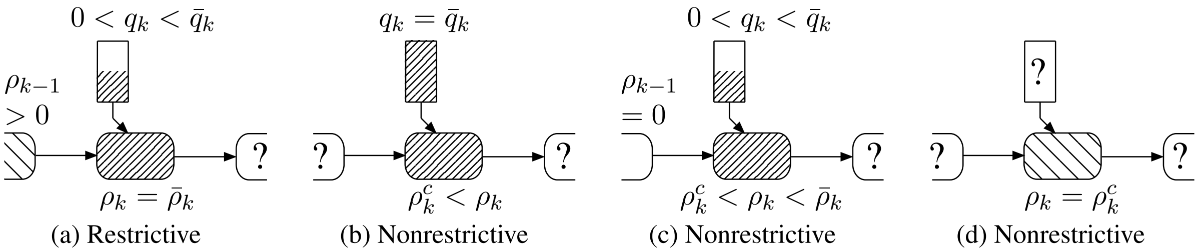
are nondecreasing in $\Phi_{k-1}(t), \Phi_k(t)$ and $\Phi_{k+1}(t)$.

(i) To verify monotonicity in $\Phi_{k-1}(t)$, first note that $f_k^{(+)}$ does not depend on $\Phi_{k-1}(t)$, so it is trivially nondecreasing. Furthermore, $\rho_k(t)$ is nondecreasing in $\Phi_{k-1}(t)$, since by assumption $l_k > 0$. Also, $d_k(\cdot)$ is nondecreasing according to Assumption 2.2.1. Recalling that $\Delta t > 0$, we conclude that $f_k^{(-)}$, which is a composition of the previously analyzed functions, is nondecreasing in $\Phi_{k-1}(t)$.

(ii) To verify monotonicity in $\Phi_{k+1}(t)$, first note that $f_k^{(-)}$ does not depend on $\Phi_{k+1}(t)$, so it is trivially nondecreasing. Furthermore, $\rho_k(t)$ is nondecreasing in $\Phi_{k+1}(t)$, since by assumption $l_k > 0$ and $\bar{\beta}_{k+1} > 0$. Also, $s_{k+1}(\cdot)$ is nonincreasing according to Assumption 2.2.1. Recalling that $\Delta t > 0$, we conclude that $f_k^{(+)}$, which is a composition of the previously analyzed functions, is nondecreasing in $\Phi_{k+1}(t)$.

(iii) To verify monotonicity of $f_k^{(-)}$ in $\Phi_k(t)$, recall that according to Assumption 2.2.1, $d_k(\cdot)$ is Lipschitz continuous, so $|d_k(x + \Delta x) - d_k(x)| \leq c_k^d |\Delta x|$. Therefore:

$$\begin{aligned}
&f_k^{(-)}(\Phi_{k-1}(t), \Phi_k(t) + \Delta\Phi) - f_k^{(-)}(\Phi_{k-1}(t), \Phi_k(t)) \\
&\quad = \Delta\Phi + \Delta t \cdot \left[d_k\left(\frac{1}{l_k}\left(\Phi_{k-1}(t) - \frac{1}{\bar{\beta}_k}(\Phi_k(t) + \Delta\Phi) + R_k(t)\right)\right)\right. \\
&\qquad\qquad \left. - d_k\left(\frac{1}{l_k}\left(\Phi_{k-1}(t) - \frac{1}{\bar{\beta}_k}\Phi_k(t) + R_k(t)\right)\right)\right] \\
&\quad \geq \Delta\Phi - \Delta t \cdot c_k^d \cdot \left|\frac{1}{l_k\bar{\beta}_k}\Delta\Phi\right| \\
&\quad \geq \Delta\Phi \cdot \left(1 - \Delta t \cdot \frac{l_k\bar{\beta}_k}{\Delta t} \cdot \frac{1}{l_k\bar{\beta}_k}\right) = 0.
\end{aligned}$$



(a) Restrictive (b) Nonrestrictive (c) Nonrestrictive (d) Nonrestrictive

Figure 2.27: Example situations in which cell k is restrictive or nonrestrictive. A question mark indicates that the corresponding density or ramp occupancy is not relevant for determining if cell k is restrictive or not.

In the first inequality, we use Lipschitz continuity and in the last inequality, we replace the Lipschitz constant for the demand function with its upper bound from Assumption 2.2.1. Monotonicity of $f_k^{(+)}$ in $\Phi_k(t)$ can be shown in a similar way.

Also, the cases $k = 0$ and $k = n$ follow along the same lines. □

Similarly, the dependency of the states on the inputs is characterized by the following relationship:

Lemma 2.2.3. *The systems equation $f_k(\Phi(t), R(t))$ for k fixed ($1 \leq k \leq n-1$) is nondecreasing in input $R_k(t)$ and nonincreasing in input $R_{k+1}(t)$. Also, $f_0(\Phi(t), R(t))$ is nonincreasing in $R_1(t)$ and $f_n(\Phi(t), R(t))$ is nondecreasing in $R_n(t)$.*

The proof of Lemma 2.2.3 follows the same ideas as the proof of Lemma 2.2.2. Next, we will define conditions that will be useful in deriving sufficient optimality conditions for best-effort control.

Definition 2.2.1. *A cell k with onramp queue is called* restrictive *at time t if*

(i) the onramp is not completely full and the upstream flow into the cell is limited by non-maximal supply of free space, i.e., $q_k(t) < \bar{q}_k$ and $s_k(\rho_k(t)) \leq d_{k-1}(\rho_{k-1}(t)) \ \wedge \ s_k(\rho_k(t)) < F_{k-1}$

or

(ii) the onramp is not empty and the downstream flow out of the cell is limited by non-maximal demand, i.e., $q_k(t) > 0$ and $d_k(\rho_k(t)) \leq s_{k+1}(\rho_{k+1}(t)) \ \wedge \ d_k(\rho_k(t)) < F_k$

Otherwise, we call the cell nonrestrictive.

This definition formalizes the intuitive idea to operate each onramp so as to maximize local mainline flow in the subsequent time step $t+1$: In case (i), the existence of free space on the onramp indicates that a potential congestion on the mainline, which is equivalent to the flow being restricted by the traffic supply, might have been prevented or at least reduced by keeping additional cars on the onramp. Conversely, the existence of free-flow conditions on the mainline below the maximal flow, as indicated by the restriction of the flow by the traffic demand, raises the question as to why any cars should be held back on the onramp queue. The idea will be illustrated in the following example.

Example 2.2.3. *Consider the situations sketched in Figure 2.27, which exemplify when a cell is restrictive. In all these examples, we assume that $s_k(\rho_k(t)) > 0$ if $\rho_k(t) < \bar{\rho}_k$ and $d_k(\rho_k(t)) > 0$ if $\rho_k(t) > 0$.*

(i) *A completely congested cell implies $s_k(\rho_k(t)) = s_k(\bar{\rho}_k) = 0$. Since the upstream cell is nonempty, it follows that $s_k(\rho_k(t)) = 0 < d_{k-1}(\rho_{k-1}(t))$. But even though cell k is completely congested, the onramp is not completely filled $q_k(t) < \bar{q}_k$. Therefore, cell k is restrictive.*

(ii) *Cell k is congested $\rho_k(t) > \rho_k^c$, but in contrast to case (i), the maximal number of cars is stored on the onramp $q_k(t) = \bar{q}_k$. Therefore, this cell is nonrestrictive.*

(iii) *Cell k is congested $\rho_k^c < \rho_k(t) < \bar{\rho}_k$ and the onramp is not completely filled $q_k(t) < \bar{q}_k$. However, in contrast to case (i), the upstream cell $k+1$ is empty, therefore, the congestion in cell k does not obstruct the mainline flow $s_k(\rho_k(t)) > 0 = d_{k-1}(0)$ and cell k is nonrestrictive*

(iv) *Cell k is operating exactly at the critical density. Therefore, $s_k(\rho_k(t)) = s_k(\rho_k^c) \geq F_{k-1}$ and $d_k(\rho_k(t)) = d_k(\rho_k^c) \geq F_k$ and the cell is nonrestrictive, for arbitrary upstream demand, downstream supply and local ramp occupancy.*

Remark 2.2.2. *Definition 2.2.1 characterizes cells with onramps. Proofs in the remainder of this section will analyze pairs of adjacent cells, to determine the behavior of the flow between these cells. To avoid unnecessary case distinctions with regard to which of these cells are equipped with onramps and which are not, note that cells without an onramp can still be expressed in the ACTM framework by $\bar{r}_k = \bar{q}_k = W_k(t) = 0,\ \forall t,$ for all cells k without an onramp, exactly in the same way as described in Remark 2.2.1 for the CTM. Then constraints (2.66) imply $R_k(t) = 0,\ \forall t$, as desired. For such cells k, it always holds that $0 = q_k(t) = \bar{q}_k(t)$ and therefore, these cells are classified as* nonrestrictive *at all times.*

It is important to note that a cell operating exactly at the critical density ρ_c is always nonrestrictive in the sense of Definition 2.2.1: According to Lemma 2.2.1, both upstream and downstream flow are jointly maximized at this density. In fact, the best-effort controller defined in said Lemma, which tracks the critical density locally, is able to keep cells nonrestrictive under certain conditions, as explained by the following lemma:

Lemma 2.2.4. *Assume that for a freeway controlled by the best effort controller, the onramp at cell k at time t satisfies*

$$0 \leq \left[\frac{l_k}{\Delta t} \left(\rho_k^c - \rho_k(t)\right) + \frac{\phi_k(t)}{\beta_k} - \phi_{k-1}(t) \right]_{\frac{1}{\Delta t}(q_k(t) - \bar{q}_k) + w_k(t)}^{\frac{1}{\Delta t} q_k(t) + w_k(t)} \leq \bar{r}_k,$$

i.e., the best-effort policy (2.63) is not affected by the constraints (2.57). Then, cell k is nonrestrictive at time $t+1$.

Sketch of proof. Note that this condition is sufficient, but not necessary. Sufficiency can be readily verified by plugging the feedback policy in the CTM equations and making a case distinction for each affine piece of the control law. □

We will now use monotonicity of the aggregated CTM to show that throughput is maximized over several time periods for nonrestrictive freeways. For the upcoming analysis, it will be useful to introduce the following notation for the maximal achievable throughput

$$\begin{array}{llll}
\Phi_k^*(t) := & \max & \Phi_k(t) & \\
& \text{subject to} & \Phi(\tau+1) = f(\Phi(\tau), R(\tau)) & \forall \tau \in \{0, \ldots, t-1\} \\
& & R_k(\tau) \leq R_k(\tau+1) \leq R_k(\tau) + \bar{r}_k & \forall \tau \in \{0, \ldots, t-1\} \\
& & W_k(\tau) - \bar{q}_k \leq R_k(\tau) \leq W_k(\tau) & \forall \tau \in \{0, \ldots, t-1\} \\
& & \text{Initial state } \Phi(0) \text{ and } R(0) = \mathbf{0} \text{ given.} & ,
\end{array}$$

at a particular time t and cell k.

Lemma 2.2.5. *Assume that $\Phi_j(t) = \Phi_j^*(t)$ for $j \in \{k-1, k, k+1\}$ and that the cells k and $k+1$ are nonrestrictive at time t. Then $\Phi_k(t+1) = \Phi_k^*(t+1)$.*

Proof. For notational convenience, let us introduce a relaxation of the set of feasible metering rates

$$\mathcal{R}_k(t) := \{R_k : W_k(t) - \bar{q}_k \leq R_k \leq W_k(t)\} \tag{2.67}$$

at time t. Note that according to this definition, the metering rates satisfy the constraints arising from the limited space on the onramps (2.58), but not (necessarily) the additional, constant bounds (2.57). Also note that the feasible sets for the metering rates $\mathcal{R}_k(t)$ are decoupled in time in the aggregated formulation, i.e., they do not depend on the system state or previous actions. In the following, we will also use the shorthand notation $\mathcal{R}(t) := \otimes_{k=1}^{n} \mathcal{R}_k(t)$. Now, we can write the optimal throughputs as

$$\begin{array}{rlll} \Phi_k^*(t+1) \leq & \max & f_k(\Phi(t), R(t)) & \\ & \text{subject to} & \Phi(\tau+1) = f(\Phi(\tau), R(\tau)) & \forall \tau \in \{0, \ldots, t-1\} \\ & & W_k(\tau) - \bar{q}_k \leq R_k(\tau) \leq W_k(\tau) & \forall \tau \in \{0, \ldots, t-1\} \\ & & \text{Initial state } \Phi(0) \text{ and } R(0) = \mathbf{0} \text{ given.} & \\ & & & \\ = & \max_{R \in \mathcal{R}(t)} & f_k(\Phi^*(t), R). & \end{array}$$

Note that we only have an inequality in the first step, since we operate with a relaxation of the feasible set of inputs. This ensures that the constraints become decoupled in time. Then, we use monotonicity of $\Phi_k(t+1) = f_k(\Phi(t), R(t))$ in the throughputs $\Phi(t)$ to obtain the final equality. Using this preliminary result and the fact that cells k and $k+1$ are nonrestrictive at time t, we will now show that also $\Phi_k(t+1) \geq \Phi_k^*(t+1)$ and hence $\Phi_k(t+1) = \Phi_k^*(t+1)$.

Cell k is nonrestrictive by assumption and therefore, it satisfies either

(A) $d_k(\rho_k(t)) > s_{k+1}(\rho_{k+1}(t)) \;\vee\; d_k(\rho_k(t)) \geq F_k$ or

(B) $q_k(t) = 0$

or both, according to Definition 2.2.1. Similarly, cell $k+1$ is also nonrestrictive and satisfies either

(C) $s_{k+1}(\rho_{k+1}(t)) > d_k(\rho_k(t)) \;\vee\; s_{k+1}(\rho_{k+1}(t)) \geq F_k$ or

(D) $q_{k+1}(t) = \bar{q}_{k+1}$

or both. We proceed with a case distinction, depending on which of these conditions are satisfied for cell k and cell $k+1$, respectively:

(i) Assume (A) and (C) hold, that is, the flow $\phi_k(t)$ is neither limited by submaximal supply or submaximal demand. It follows that $\phi_k(t) = \min\{d_k(\rho_k), F_k, s_{k+1}(\rho_{k+1})\} = F_k$ and hence

$$\begin{aligned} \Phi_k(t+1) &= \Phi_k^*(t) + \Delta t \; \cdot F_k \\ &\geq \max_{R \in \mathcal{R}(t)} f_k(\Phi^*(t), R) \\ &= \Phi_k^*(t+1). \end{aligned}$$

(ii) Assume (B) and (D) hold, that is, the onramp in cell k is empty and the onramp in cell $k+1$ is full. The first condition implies that $R_k(t) = W_k(t)$, according to equation (2.66b). Conversely, the second condition implies that $R_{k+1}(t) = W_{k+1}(t) - \bar{q}_{k+1}$. It follows that

$$\begin{aligned}
\Phi_k(t+1) &= \Phi_k^*(t) + \Delta t \ \cdot \min \left\{d_k(\rho_k), s_{k+1}\left(\rho_{k+1}\right)\right\} \\
&= \Phi_k^*(t) + \Delta t \ \cdot \min \left\{d_k\Big(\rho_k\big(\Phi^*(t), W_k(t)\big)\Big), s_{k+1}\Big(\rho_{k+1}\big(\Phi^*(t), W_{k+1}(t) - \bar{q}_{k+1}\big)\Big)\right\} \\
&= \Phi_k^*(t) + \Delta t \ \cdot \min \left\{d_k\Big(\rho_k\big(\Phi^*(t), \max_{R\in\mathcal{R}_k(t)} R\big)\Big), s_{k+1}\Big(\rho_{k+1}\big(\Phi^*(t), \min_{R\in\mathcal{R}_{k+1}(t)} R\big)\Big)\right\} \\
&\geq \max_{R\in\mathcal{R}(t)} f_k(\Phi^*(t), R) \\
&= \Phi_k^*(t+1).
\end{aligned}$$

Here, we use the lower and upper bounds on the aggregated ramp metering rates $R_k(t)$ according to the definition of $\mathcal{R}_k(t)$ in equation (2.67). Finally, the inequality holds because the throughput $\Phi_k(t+1)$ function is nondecreasing in $R_k(t)$ and nonincreasing in $R_{k+1}(t)$, according to Lemma 2.2.3.

(iii,iv) In the third case, assume that (A) and (D) hold, whereas in the fourth case, assume that (B) and (C) hold. The derivations to show that $\Phi_k(t+1) \geq \Phi_k^*(t+1)$ can easily be constructed by combining parts from cases (i) and (ii).

All cases compatible with Definition 2.2.1 have been verified and the proof has thus been completed. □

A maximization of TTD leads to a minimization of the total discharge flows at every time instant, which in turn corresponds to a minimization of TTS over the whole horizon, as we will show next.

Theorem 2.2.1. *Assume that every cell $k \in \{1, \ldots, n\}$ of a freeway is nonrestrictive, for the entire horizon $t \in \{1, \ldots, T-1\}$. Then TTS is minimized.*

Proof. The initial conditions are assumed to be fixed, so $\Phi_k^*(0) = \Phi_k(0)$, for all k. Because Definition 2.2.1 holds for every cell in every time step, we can apply Lemma 2.2.5 for every cell and $t = 1$, yielding $\Phi_k(1) = \Phi_k^*(1)$ for all k. Employing induction, we can proceed in the same manner to show that $\Phi_k(t) = \Phi_k^*(t)$ over the complete control horizon. Thus, throughput is maximized jointly for every cell and every step.

It remains to be shown that joint maximization of all throughputs implies minimization of TTS. To do so,

$$TTT(t) = \Delta t \cdot \sum_{k=1}^{n} l_k \rho_k(t) = \Delta t \cdot \left(\Phi_0(t) - \Phi_n(t) + \sum_{k=1}^{n}\left(R_k(t) - \frac{\beta_k}{\bar{\beta}_k}\Phi_k(t)\right)\right)$$

and

$$TWT(t) = \Delta t \cdot \sum_{k=1}^{n} q_k(t) = \Delta t \cdot \sum_{k=1}^{n} \left(W_k(t) - R_k(t)\right)$$

have to be expressed in terms of the aggregated variables. If we sum according to the definition of TTS, we find

$$\begin{aligned} TTS &= \sum_{t=0}^{T} TTT(t) + TWT(t) \\ &= \Delta t \cdot \sum_{t=0}^{T} \left(\Phi_0(t) - \Phi_n(t) + \sum_{k=1}^{n} \left(W_k(t) - \frac{\beta_k}{\bar{\beta}_k} \Phi_k(t) \right) \right). \end{aligned}$$

Analyzing the sign of the coefficients yields the desired result

$$\begin{aligned} \text{minimize}\, TTS &= \Delta t \ \cdot \sum_{t=0}^{T} \left(\Phi_0(t) - \max\left\{\Phi_n(t)\right\} + \sum_{k=1}^{n} \left(W_k - \underbrace{\frac{\beta_k}{\bar{\beta}_k}}_{\geq 0} \max\left\{\Phi_k(t)\right\} \right) \right) \\ &= \Delta t \ \cdot \sum_{t=0}^{T} \left(W_0(t) - \Phi_n^*(t) + \sum_{k=1}^{n} \left(W_k(t) - \frac{\beta_k}{\bar{\beta}_k} \Phi_k^*(t) \right) \right). \end{aligned}$$

All minimizations and maximizations in the previous equations are to be understood as an optimization over the set of feasible ramp metering patterns, with respect to the fixed initial condition and traffic demands and the CTM system model. Note that $\bar{\beta}_k \in (0,1]$ for all k by definition of the split ratios. The last equality confirms that TTS is indeed minimized by maximal $\Phi_k^*(t)$ due to an entirely nonrestrictive freeway and thus concludes the proof. □

It is important to keep in mind that Definition 2.2.1 provides only *sufficient* optimality conditions. In particular, it is possible that, depending on the freeway parameters and the demand profile, there does not exist a policy which satisfies the given optimality conditions. It also becomes clear that the difficulties in solving the main problem (2.61) hinge mainly on the ramp metering bounds (2.57), as the following result states:

Corollary 2.2.1. *Assume local ramp metering rates in cell k can be chosen as*

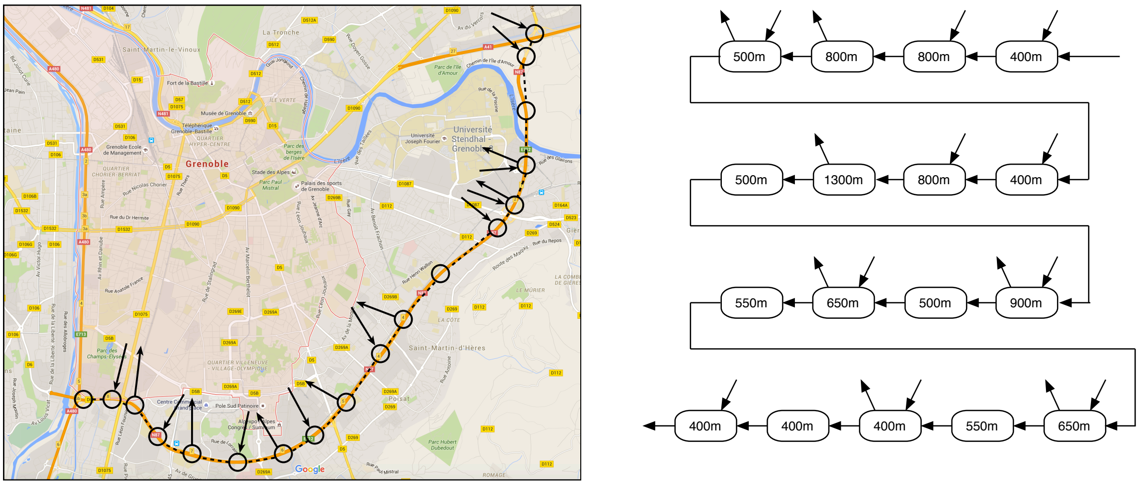
$$r_k(t) = \left[\frac{l_k}{\Delta t} \left(\rho_k^c - \rho_k(t) \right) + \frac{\phi_k(t)}{\bar{\beta}_k} - \phi_{k-1}(t) \right]_{\frac{1}{\Delta t} q_k(t) + d_k(t)}^{\frac{1}{\Delta t}(q_k(t) - \bar{q}_k) + d_k(t)}$$

for all $k \in \{1, \ldots, n\}$ and all $t \in \{0, \ldots, T\}$, i.e., the ramp metering bounds (2.57) are not support constraints. Then, the local feedback policy minimizes TTS over the whole horizon.

This follows immediately from Theorem 2.2.1 and Lemma 2.2.4.

Application

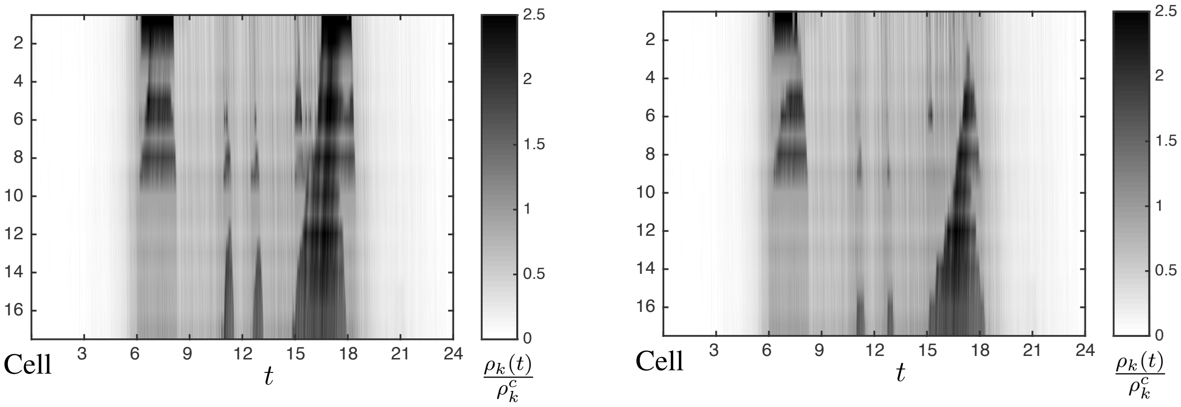
We consider a congestion prone freeway in the vicinity of Grenoble, France (de Wit et al., 2015). The total length of this freeway stretch, the so-called Rocade Sud, is 11.8km, out of which we consider 10.45km (exluding short sections in the beginning and in the end). The freeway has 7 offramps in total and 10 onramps, which will be subject to ramp metering in the future. The topology is depicted in Figure 2.28. The freeway has been equipped with loop detectors. Traffic data are reported to a control center every 15 seconds. In this case study, the freeway is modeled by the CTM, with one cell per mainline sensor location.



(a) Annotated map of the *Rocade Sud*, Grenoble. Circles represent the approximate location of the onramps, offramps and sensors. Map data ©2015 Google

(b) Topology of the CTM modeling the the *Rocade Sud*, with onramps and offramps. Boxes represent the cells of the CTM.

Figure 2.28: Topology of the *Rocade Sud*, Grenoble



(a) No ramp metering

(b) Optimal ramp metering

Figure 2.29: Partial simulation results of the Grenoble freeway and traffic demands of April, 14th, 2014. Depicted are the mainline densities without ramp metering, and with optimal ramp metering. Because of the limited space on the onramps, it is not possible to prevent congestion altogether, but the improvement is clearly visible.

Since ramp metering has not yet been installed, we can obtain the traffic demand profiles simply from the onramp- and mainline inflows to the system. For the following case study, we consider data from 5 weeks in March, April and June 2014, corresponding to 35 days in total[6]. For each day, we simulate the evolution of traffic using the CTM and best-effort control and report the TTS. For comparison, we also pose the finite horizon optimal control problem over every day, assuming perfect prediction of demand. Employing the results of (Gomes and Horowitz, 2006b), which derives an equivalent LP reformulation of the optimal control problem, we can compute the optimal trajectories. We are interested in two main questions:

- How often are the (local) optimality conditions violated, i.e., how often are cells restrictive in the sense of Definition 2.2.1, over the course of each day?
- How do the travel times under best-effort control compare to the theoretical best-case under predictive optimal control?

Performance To establish a baseline for the maximal potential improvement possible, we first compare the open-loop performance, i.e., TTS without ramp metering, to the globally optimal solution. We then analyze the trajectories obtained for best effort control, with particular emphasis on how often cells behave restrictive and therefore, not optimal, and the differences in TTS between best-effort control and the globally optimal solution.

Congestion patterns for a typical day (14.04.2014) are presented in Figure 2.29, for the uncontrolled freeway and for the freeway under ramp metering. The complete results are summarized in Figure 2.31. It can immediately be seen that (optimal) ramp metering indeed reduces TTS in most instances. This is achieved by a reduction in congestion and the spill back thereof, thus increasing the outflows from the offramps. On certain days, no improvement can be achieved by ramp metering. These days are characterized by a low traffic demand, as it is typically encountered on weekends. From real traffic data, one can verify that even the uncontrolled freeway typically does not become congested on these days, so obviously no ramp metering is the best policy.

To quantify the benefits of ramp metering, we relate the savings in TTS to the TTS itself, over all days and obtain average savings of

$$\frac{TTS_{noControl} - TTS_{optimal}}{TTS_{noControl}} \approx 5.38\%.$$

A large part of total TTS is caused by vehicles traveling at free-flow velocity, for which ramp metering does not provide any benefits. Therefore, we chose a second metric and compare the savings in TTS to the total time wasted in congestion and in onramp queues. To this end, we define the TFT (Total free-flow time) as the TTS achieved for a hypothetical freeway in which all cars instantly enter the mainline after arriving on an onramp and always travel at free-flow speed on the freeway itself. The relative savings in terms of time wasted in congestion and on onramp queues, over all days, amount to

$$\frac{TTS_{noControl} - TTS_{optimal}}{TTS_{noControl} - TFT} \approx 17.71\%.$$

We note that these numbers are similar to the ones reported in (Gomes and Horowitz, 2006b).

We now simulate the system using the best-effort controller. In a first step, we verify for every cell with an adjacent onramp and every time step if the cell is nonrestrictive according to Definition 2.2.1. It

[6] Data for the full three months are available, however, data from many days are incomplete due to sensor failures. To ensure representative performance evaluation, only weeks for which data are complete for all sensors at days of the respective week are chosen for the case-study.

turns out that most cells are nonrestrictive for most of the timesteps. More precisely, the day on which violations most often occur is April 09th on which some cells are restrictive at less than 0.4% of all sampling time instances. We use April 14th, 2014 in order to visualize typical traffic patterns for the freeway controlled by ramp metering. In particular, we highlight several time intervals in which an increase in traffic demand occurs and outline if the optimality conditions for the best-effort controller are satisfied or violated (Figure 2.30).

The example depicts the evolution of the traffic density in cell 4 and the evolution of the queue length at the adjacent onramp. The morning and the evening rush hour periods are easily visible from the data. Outside of these rush hour periods, cell 4 operates in free-flow and there is no onramp queue, which means that the cell is nonrestrictive. The figure also provides a detailed depiction of the rush hour periods. The intervals in which cell 4 is nonrestrictive are highlighted (A,B,C). However, there exists a time interval D during which the mainline becomes congested, while the onramp queue is not completely filled yet. Note that this example was specifically chosen to depict a situation during which a cell becomes restrictive. According to Figure 2.30a the traffic demand at the onramp at cell 4 originates from the same roundabout as the traffic demand at the next downstream onramp (cell 5). Therefore, the traffic demand is low at both onramps (in comparison to other onramps on the freeway). Thus it takes longer to fill up the onramp queues and cells 4 and 5 are more likely to become restrictive than other cells in this example.

The performance achieved by best-effort control is also depicted in Figure 2.31. We can see that the performance in terms of the savings in TWT achieved by the best-effort controller is indistinguishable from the optimal one in a plot scaled according to the absolute values of the savings. A closer look comparing upper- and lower bounds to the optimal solution reveals that the performance deterioration of the best-effort controller, which amounts to

$$\max_{days} \left\{ \frac{TTS_{bestEffort} - TTS_{optimal}}{TTS_{optimal}} \right\} \approx 0.178\%$$

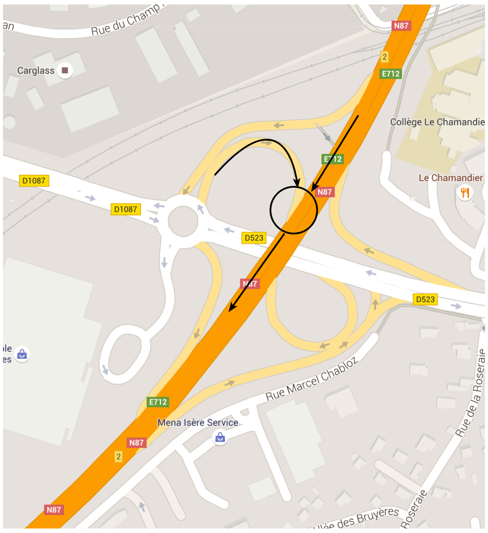
in terms of the time wasted in congestion, is worst on May 13th, 2014. Since the optimality of the best-effort controller depends on the ability of the controller to keep the freeway nonrestrictive, bounds on suboptimality in case of failure to do so are of interest. For a given initial state and demand profile, one can use Corollary 2.2.1 to compute a lower bound on the TTS. Recall that the range of admissible ramp metering rates $r_k(t)$, depending on the demand $w_k(t)$ and the queue length $q_k(t)$ is given as

$$\min \left\{ 0, \frac{1}{\Delta t} \left(\bar{q}_k - q_k(t) \right) + w_k(t) \right\} \leq r_k(t) \leq \min \left\{ \bar{r}_k, \frac{1}{\Delta t} q_k(t) + w_k(t) \right\},$$

i.e., the ramp metering rate is constrained by constraints on the queue length and fix bounds on the metering rate. For the purpose of deriving a lower bound, one can formulate a relaxed minimal TTS problem starting from the main problem (2.61), in which the constant bounds on the ramp metering rates are removed. Employing Corollary 2.2.1, we find that the solution to this problem can be obtained efficiently by simulating the system for the given demand pattern, using the best-effort controller.

Conversely, an upper bound on the optimal TTS can be computed efficiently by simulating the best-effort controller and respecting all bounds on the metering rates by saturating the control actions accordingly.

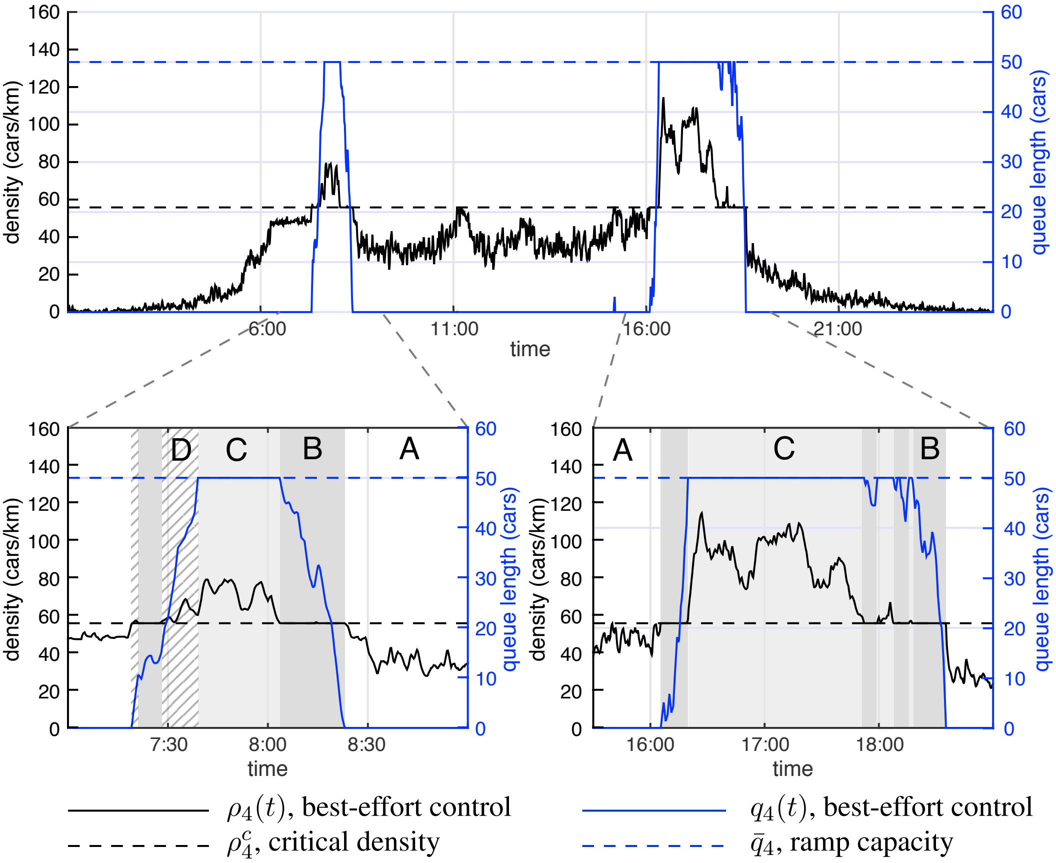
It turns out that by using only these inexpensive simulations to compute upper and lower bounds, we can verify a posteriori that the suboptimality is smaller than 0.2%, without the need to compute the optimal solution exactly, i.e., solve a (potentially expensive) optimization problem.



(a) Map of the onramp at cell 4 of the *Rocade Sud*, Grenoble. Note that the onramp is very close to a second onramp, leading to cell 5. Map data ©2015 Google

Symbol in Figure 2.30c	Density range	Queue length range	Cell k non-restrictive?
A	$\rho_k \leq \rho_k^c$	$q_k = 0$	Yes
B	$\rho_k = \rho_k^c$	$0 \leq q_k \leq \bar{q}_k$	Yes
C	$\rho_k^c \leq \rho_k$	$q_k = \bar{q}_k$	Yes
D	$\rho_k^c < \rho_k$	$0 < q_k$	Frequently not

(b) Legend to identify time intervals in which the optimality conditions are satisfied (A,B,C) and most likely violated (D). Note that the conditions for classifying a period in group D, i.e., $\rho_k^c < \rho_k$ and $0 < q_k$, are not sufficient for a cell being restrictive, in particular if the critical density is only slightly exceeded, as illustrated in Example 2.2.3.



(c) Evolution of traffic density in cell 4 and the queue lengths of the adjacent onramp on April 14th, 2014.

Figure 2.30: Partial simulation results of the *Rocade Sud*, using the traffic demand of April 14th, 2014.

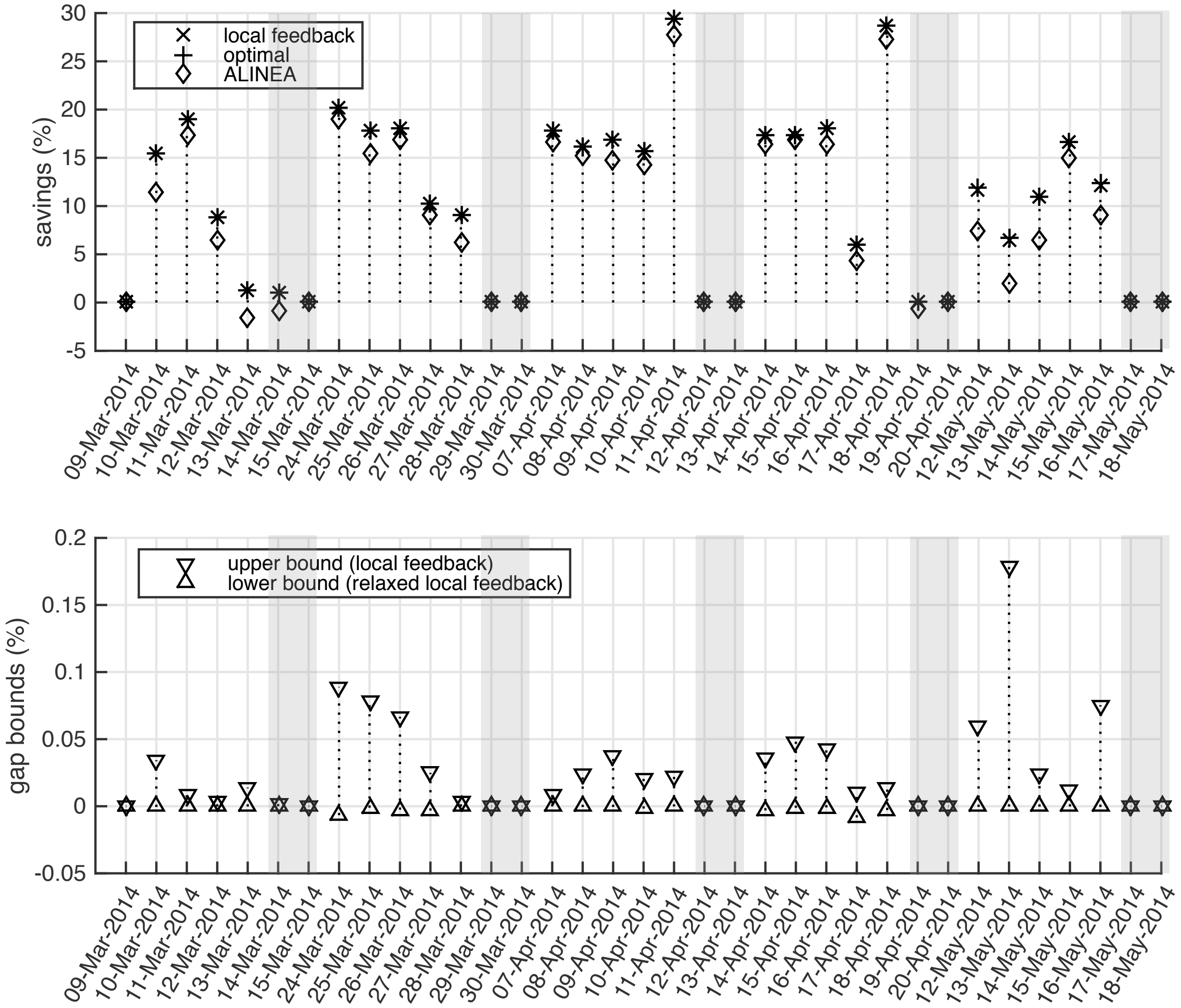


Figure 2.31: Summary of the TTS achieved by the various controllers, for every day of our case study. In the upper plot, the savings in TTS in relation to the total time in congestion for the uncontrolled freeway (second metric) are depicted. For all days, the performance achieved by local feedback is indistinguishable from the performance of the optimal ramp metering strategy. ALIENA comes close most of the times. On weekends, highlighted in grey, ramp metering does not provide any benefits. In the second plot, we chose a much smaller scale to depict the suboptimality of local feedback, serving as an upper bound, to the optimal TTS (normalized to zero). The performance deterioration is consistently in the range of fractions of percent. We also plot the lower bound on TTS, computed by relaxing the ramp metering constraints, which is even closer to the optimum.

Comparision with ALINEA To demonstrate that the theoretical conclusions drawn from the analysis of the best-effort controller indeed extend to practically relevant ramp metering policies, we compare it numerically to ALINEA (Papageorgiou et al., 1991), a popular decentralized ramp metering policy. ALINEA in its basic form consists of local, anti-windup integral feedback controllers. The metering rates are first computed as integral feedback

$$\tilde{r}_k(t) := r_k(t-1) + K_I \cdot (\rho_k^c - \rho_k(t))$$

in which $K_I := 70/(\rho_k^c \cdot \Delta t)$ is the integral gain recommended in (Papageorgiou et al., 1991). Then, they are saturated

$$r_k(t) = [\tilde{r}_k(t)]_{\max\left\{0, \frac{1}{\Delta t}(q_k(t) - \bar{q}_k) + d_k(t)\right\}}^{\min\left\{\bar{r}_k, \frac{1}{\Delta t} q_k(t) + d_k(t)\right\}} \tag{2.68}$$

in the same way as for the local feedback controller. There exists a variety of extensions to this basic controller (Papamichail and Papageorgiou, 2008; Smaragdis et al., 2004; Wang et al., 2014), that introduce coordination between ramps or permit use of different sensor configurations. The standard ALINEA controller requires only the critical density as a model parameter. Instead of using model knowledge to estimate and predict traffic demand and supply, this controller uses integral feedback to converge asymptotically to the optimal metering rates in the equilibrium, for constant boundary conditions (Schmitt et al., 2015).

Comparing the TTS achieved by the best-effort controller to the freeway controlled by the basic version of ALINEA, we find that the performance deteriorates only slightly, with an average loss of

$$\frac{TTS_{alinea} - TTS_{optimal}}{TTS_{optimal}} = 0.43\%$$

in terms of TTS. A detailed comparison between the TTS achieved by the best-effort controller and ALINEA is also provided in Figure 2.31. Indeed, for most cases both controllers not only show comparable performance but also show very similar trajectories (see Figure 2.32). Note that in the top-most figure which depicts the entire rush hour period, differences between the evolution of the densities are hardly visible. They can best be recognized at time intervals during which the controllers do not saturate, i.e., when the density is stabilized at (or close to) the critical density. Here, the lack of perfect model information in ALINEA becomes evident. Changes in upstream- and downstream mainline flow act as disturbances and induce small oscillations in the ALINEA controlled system. By contrast, the best-effort controller law is assumed to have perfect model knowledge and keeps the density exactly at the critical density.

The similarity of the trajectories for both controllers suggests that we can indeed draw conclusions about the behavior of practical ramp metering strategies from our theoretical analysis. In this sense, one should view the slight performance deterioration of ALINEA in comparison to best-effort control as the price one has to pay for not exactly knowing the fundamental diagram in reality.

Conclusions

In this work, we have derived sufficient optimality conditions for minimal TTS ramp metering for the monotonic CTM. Based on the analysis, we have defined a suitable distributed, non-predictive feedback controller motivated as a one-step-ahead maximization of local traffic flows. In simulations using real-world parameters and traffic demands, we demonstrated that the freeway is nonrestrictive most of the time and the controller achieves almost optimal performance. For the sake of keeping the theoretical analysis tractable, we assumed perfect model knowledge which was subsequently used to counteract the

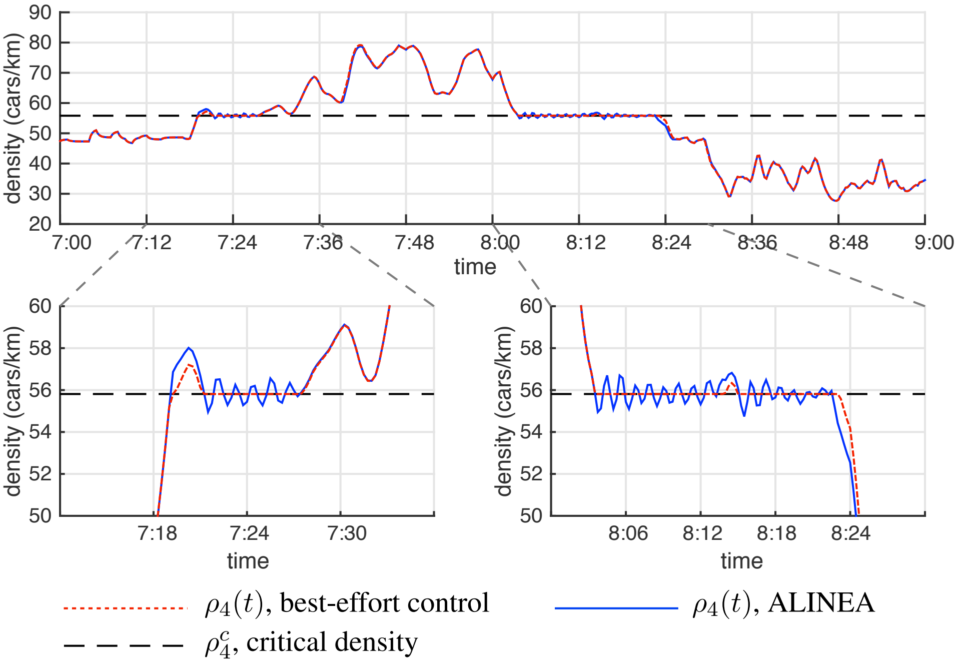


Figure 2.32: Comparison between the closed-loop trajectories of the local feedback controller and ALINEA for cell 4 during the morning rush hour period of April 14th, 2014. This is the same time interval also depicted in Figure 2.30, although we focus on the density in this graph.

model dynamics in the best-effort controller. This assumption is certainly not satisfied in practice and traffic models are typically subject to large uncertainties. However, we demonstrated that the behavior of our proposed controller is very close to that of ALINEA which uses feedback to mitigate the lack of exact model knowledge. Therefore we have verified that our analysis indeed extends to practically successful ramp metering schemes.

We highlight again, that as long as the cells are nonrestrictive, the best-effort controller is optimal and in the CTM framework used in this work, no additional benefit can be realized either by coordination between the ramps or prediction of the traffic demand. The fact that space on the onramps is limited according to constraint (2.58) does not make coordination between ramps desirable, though it does make ramp metering less effective, of course. In addition, the numerical studies in Section 2.3.1 reveal that the conditions for cells being nonrestrictive are satisfied most of the time for a case study using realistic freeway parameters and traffic demands. We achieve results comparable to the global optimal solution with very limited information exchange (only densities of adjacent cells) and without any coordination between the ramps at all. These considerations provide a theoretical explanation for part of the conclusions drawn from practical experience and heuristic considerations in (Papamichail et al., 2010a).

However, they seem to contradict other practical experience reported in (Papamichail et al., 2010a), which that suggests that“limited ramp storage space and the requirement of equity . . . are the main reasons for coordinated ramp metering”. We can resolve this seeming contradiction by recalling that the results on global optimality apply to the monotonic CTM. In practice, there is empirical evidence of a capacity drop at a congested bottleneck. This effect is usually represented by higher-order model such as MetaNet (Messner and Papageorgiou, 1990), but can also be approximated in a modified CTM (Karafyllis et al., 2016) by allowing for non-monotonic demand functions, which (slightly) decrease for densities larger than the critical density as depicted in Figure 2.24c. A congestion in such a model will reduce the bottleneck flow and subsequently decrease densities further downstream, which shows that the dynamic system is no longer monotonic. In a non-monotonic setting, an incentive exists to prevent congestion of a bottleneck, even if there is no danger of spill back of the congestion queue. An important conclusion about the potential benefits of coordinated ramp metering can be drawn from this analysis: coordinated ramp metering may target inefficiencies that result from limited space on the onramps *in conjunction with the nonmonotonic behavior of a congested bottleneck*. Any model-based, coordinated ramp metering strategy therefore must employ a model that is able to reproduce this effect, in order to recognize and avoid it. For example, model-predictive control based on the monotonic CTM and a maximize-TTD or minimize-TTS objective is unlikely to provide any improvements over the presented distributed strategy at all. A heuristic targeting the nonmonotonic effects, such as the addition of a penalty term for the congested region, may serve the same purpose.

Based on these results, two main future research directions seem promising: First, an a-priori quantification of the suboptimality incurred because of infrequent violation of the optimality conditions or because of uncertainty in the knowledge of the fundamental diagram would be helpful. Theoretical bounds on the suboptimality in the presence of a general, not necessarily monotonic CTM would be very helpful, but since no convex reformulation or other efficient, global solution methods for optimal control problems involving the nonmonotonic CTM exist, it is questionable, whether tight bounds can be derived. Instead, the known results on the impact of nonmonotonic behavior can now be used to design control laws for nonmonotonic, realistic freeway models in a more systematic manner by starting from the optimal solution for the monotonic case and specifically targeting the nonmonotonicities with modifications to this baseline control strategy.

2.2.2 Optimal Steady-State Operation

M.Schmitt, C. Ramesh, P. Goulart, J. Lygeros

We consider a special class of monotone systems for which the system equations are also convex in both the state and the input. For such systems, we study optimal infinite horizon operation, with respect to an objective function that is also monotone and convex. The main results states that, under some technical assumptions, these systems are optimally operated at steady state, i.e. there does not exist a trajectory over an infinite horizon that outperforms stabilizing the system in the optimal equilibrium. We draw a connection to recent results on dissipative systems in the context of Economic Model Predictive Control, where systems that are optimally operated at steady state have already been studied. Finally, we apply the main result to a problem in traffic control, where we are able to disprove the existence of improving periodic trajectories involving the alternation of congestion and free flow for freeway ramp metering.

Introduction

In this work, we consider a special class of nonlinear, monotone systems. Monotone systems are systems for which trajectories preserve a partial ordering on the states Angeli and Sontag (2003); Hirsch and Smith (2005); Hirsch et al. (2005), a fact that is favorable for controller design and has recently been exploited in a series of papers Rantzer et al. (2013); Ito et al. (2014); Dirr et al. (2015). Even stronger results can be obtained for positive systems, that is, systems that are not just monotone, but also linear Rantzer (2015); Colombino and Smith (2014, 2015). Here, we consider the more general class of nonlinear, state-monotone systems, whose system equations are jointly convex in state and input. For so-called convex-monotone systems Rantzer and Bernhardsson (2014), certain control tasks which are difficult for general, nonlinear systems are in fact tractable. Most significantly, it has been shown that trajectories of convex, monotone systems are convex in the control inputs, which makes certain optimization problems over trajectories tractable Rantzer and Bernhardsson (2014).

We study the problem of optimal average, infinite horizon control of convex, state-monotone systems with respect to a convex, state-monotone cost, in the absence of any noise or disturbances and assuming perfect knowledge of state and system dynamics. For controllable, linear systems and radially unbounded, convex objectives, such a trajectory planning problem is not very challenging since there exists an optimal (minimal cost) equilibrium and any controller that stabilizes the system to this equilibrium achieves optimal average, infinite horizon cost. However, this question is not as simple for general, nonlinear systems. In particular, the optimal operation of systems for which the costs represent actual, economic performance criteria like operation costs, production output or throughput of some commodity through a network, is studied in the context of Ecomonic Model Predictive Control (MPC) Ellis et al. (2014); Angeli et al. (2012). Depending on the system, the optimal operation might very well involve the system repeating a cycle Grüne and Zanon (2014); Müller et al. (2015a). Therefore, the question of which systems are optimally operated at steady state, that is, systems for which there does not exist a periodic (or a nonperiodic) trajectory improving average infinite horizon cost over the cost incurred in the optimal equilibrium, is of importance. It has been shown that, under some technical assumptions, the class of systems that are optimally operated at steady-state is exactly the class of systems that are dissipative with respect to a particular storage function Müller et al. (2015b). However, finding a certificate that a particular system is dissipative is difficult. The main result of this work is to show that convex, state-monotone systems are optimally operated at steady state.

We apply this theoretical result to the problem of freeway ramp metering. Ramp metering refers to the active control of the inflow of cars on a freeway via the onramps, by means of traffic lights Papageorgiou et al. (2003); Papageorgiou and Kotsialos (2000). We consider a (quasi-) steady-state situation as in Schmitt et al. (2015), in which the traffic demand cannot be completely served, a common occurence during rush-hour periods. Using our theoretical results, we can prove that for the commonly used monotone Cell Transmission Model (CTM) Daganzo (1994b); Gomes and Horowitz (2006b), there does not exist a control policy that improves average demand satisfaction over the one achieved in the optimal equilibrium. This answers a question posed in Gomes et al. (2008) regarding the possibility to improve throughput by using time varying trajectories involving the alternation of congestion and free flow periods.

The problem description is formalized next in Section 2.2.2. The theoretical results are derived in Section 2.2.2. In Section 2.3.1, we apply the results to the ramp metering case and we provide concluding remarks in Section 4.

We will use the following notation: All inequalities that involve vectors are to be understood component-wise. We use subscripts to index components of a vector or elements of a scalar-valued sequence, that is, $x_k \in \mathbb{R}$ can denote the k^{th} component of a vector or the k^{th} element of a sequence. We exclusively consider systems in discrete time $t \in \mathbb{Z}_0^+$ and for quantities that evolve over time, like the state of a system, we write $x(t)$. For a given discrete-time system and the input sequence $u = (u(0), u(1), \ldots, u(T-1))$ of finite or infinite length, we denote the system trajectory for the input u starting at $x(0)$ as $x_u(t, x(0))$, if we want to emphasize that it is valid trajectory. If we are interested in the value at a particular time t, we also use shorthand notation $x(t) = x_u(t, x(0))$. The average cost over any finite trajectory will be denoted $c\left(x_u(t, x(0))\right) := \frac{1}{T}\sum_{t=0}^{T-1} c\left(x(t), u(t)\right)$.

Problem statement

Consider a nonlinear, discrete time dynamical system with state $x(t) \in \mathbb{X} \subseteq \mathbb{R}^n$ and input $u(t) \in \mathbb{U} \subseteq \mathbb{R}^m$ given as

$$x(t+1) = f(x(t), u(t)). \tag{2.69}$$

with $f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$. In this work, we study systems with convex and monotone dynamics:

Definition 2.2.2. *A system is* state-monotone *if the system equations $f : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$ are component-wise nondecreasing in the state, i.e. in its first n arguments. A system is* convex *if the system equations $f(x, u)$ are jointly continuous and jointly convex in x and u and the sets $\mathbb{X}$ and $\mathbb{U}$ are closed and convex. We write convex, state-monotone (CSM) system for systems that satisfy both properties.*

The standard definition of a monotone system as used in Angeli and Sontag (2003); Hirsch and Smith (2005); Hirsch et al. (2005) is equivalent to the system equation f being component-wise nondecreasing in all its arguments, whereas we only require monotonicity in the states. In that sense, our problem setup is a generalization of the notion of *convex-monotone* systems, as introduced recently in Rantzer and Bernhardsson (2014).

We will analyze the performance of controlling CSM systems with respect to the minimization of a lower-semicontinuous CSM cost $c : \mathbb{X} \times \mathbb{U} \to \mathbb{R} \cup \{+\infty\}$. The definition of a CSM cost is analogous to Definition 2.2.2, except that the cost may become infinite and therefore, we only require lower semicontinuity instead of continuity.

Assumption 2.2.2. *We assume that sets $\mathbb{X}$ and $\mathbb{U}$ and the cost function c satisfy the following conditions:*

(i) The set $\mathbb{X}$ is contained in a translation of the positive cone: $\mathbb{X} \subseteq \mathbb{R}^n_+ - \{\underline{x}\}$ for some finite $\underline{x}$.

(ii) The set $\mathbb{X}$ is positively control invariant for all $u \in \mathbb{U}$.

(iii) The set $\mathbb{U}$ *is compact.*

(iv) The cost is unbounded in positive directions, that is, for every $c \geq 0$*, there exists* $\bar{x}$ *such that for every* $x : x \geq \bar{x}$*, it follows that* $c(x,u) \geq c$ $(\forall u \in \mathbb{U})$.

(v) The minimum $c^* = \min_{x\in\mathbb{X},\ u\in\mathbb{U}} c(x,u)$ *s.t.* $x = f(x,u)$ *is attained.*

Under assumption (v), there exists a (potentially non-unique) minimizer $(x^*, u^*) = \operatorname*{argmin}_{x\in\mathbb{X},\ u\in\mathbb{U}} c(x,u)$ s.t. $x = f(x,u)$ that we will call the optimal equilibrium. Conditions (i,ii,iii,iv) imply that that there do not exist cost-optimal trajectories for which certain states diverge. In practice, divergence of the states is hardly ever a desired outcome[7]. These conditions can be relaxed in certain cases, for example, control invariance is not necessary if the system is exponentially stable and the cost on actuation is “sufficiently large”. Also, conditions (i,iv) can be omitted if $\mathbb{X}$ is compact.

In Section 2.2.2, we show that the systems of interest are in fact dissipative with respect to a specific supply function and that this is a crucial property to explain optimal operation of the systems in infinite horizon. The literature on dissipative systems, e.g. Müller et al. (2015a); Angeli et al. (2012), provides different sets of criteria for determining when dissipativity of a system is equivalent to the system being optimally operated at steady state and parts of these criteria are interchangeable with the ones presented here.

Theory

We seek to compare the long-term performance achievable by stabilizing the system at the optimal equilibrium with the minimal infinite horizon average cost of any feasible trajectory. To this end, we first need to show an intermediate result that will be helpful in constructing cost-efficient equilibria.

Lemma 2.2.6. *Given a CSM system with CSM cost satisfying Assumption 2.2.2, assume there exists* $x \in \mathbb{X}$ *and* $u \in \mathbb{U}$ *such that* $x \geq f(x,u)$*. Then, there exists an equilibrium* $x' = f(x', u')$*,* $x' \in \mathbb{X}$*,* $u' \in \mathbb{U}$ *with* $c(x', u') \leq c(x,u)$*.*

Proof. Consider the fixed point iteration

$$x(t+1) = f(x(t), u)\ , \quad x(0) := x.$$

From the inequality above, we know that $x(1) = f(x,u) \leq x(0) = x$. Because of monotonicity of the system, we can conclude that $x(t+1) \leq x(t)$ inductively. The iterates $x(t)$ can be interpreted as the system trajectory $x_u(t, x(0))$ for constant control inputs $u \in \mathbb{U}$. By Assumption 2.2.2, the set $\mathbb{X}$ is positively control invariant and any $x(t) \in \mathbb{X}$ is bounded below by $\underline{x} \leq x(t)$. Boundedness and monotonicity of the sequence imply that the fixed point iteration converges to some $x_\infty = f(x_\infty, u)$. Note that $x_\infty \in \mathbb{X}$ because $\mathbb{X}$ is closed and furthermore, $x_\infty \leq x$ which implies, by monotonicity of the objective in the states, that $c(x_\infty, u) \leq c(x,u)$. □

Using this intermediate result, we first restrict our attention to periodic trajectories. If a finite trajectory returns to the initial state, i.e. $x_u(T, x(0)) = x(0)$, we call it a cycle. The following result shows how to construct improving equilibria from cycles.

[7] Note that the assumption of radially unbounded state costs, which is often invoked to exclude such a behavior, is not suitable in this case, because of the assumption of monotonicity of the cost in the states.

Theorem 2.2.2. *Let f be a CSM system with c a CSM cost satisfying Assumption 2.2.2. The average cost of any cycle $x_u(t, x(0))$ of finite, but arbitrary length T is lower bounded by the cost in the optimal equilibrium.*

Proof. For notational convenience, let us define the average state $\bar{x} := \frac{1}{T}\sum_{t=0}^{T-1} x_u(t, x(0))$ and average input $\bar{u} := \frac{1}{T}\sum_{t=0}^{T-1} u(t)$ over the cycle. Then

$$\begin{aligned}\bar{x} &= \frac{1}{T}\sum_{t=0}^{T-1} x_u(t, x(0)) = \frac{1}{T}\sum_{t=1}^{T} x_u(t, x(0)) \\ &= \sum_{t=0}^{T-1} \frac{1}{T} f\left(x_u(t, x(0)), u(t)\right) \geq f\left(\bar{x}, \bar{u}\right)\end{aligned}$$

by convexity of f. Moreover, by convexity of c

$$\bar{c} := c\left(x_u(t, x(0))\right) \geq c(\bar{x}, \bar{u}).$$

Now, Lemma 2.2.6 implies the existence of a feasible equilibrium x', u' improving over the cost of the average state and input $c(x', u') \leq c(\bar{x}, \bar{u})$. Hence $c(x', u') \leq c(\bar{x}, \bar{u}) \leq \bar{c}$, which is the desired result. □

Note that this result does not require Assumption 2.2.2 (iv). In a next step, we show that the system is optimally operated at steady-state, that is, that the average infinite horizon cost of any feasible trajectory is lower bounded by the cost in the optimal equilibrium. To do so, let us first introduce the set of all feasible state-action pairs that lead to an (incremental) cost that is less than or equal to the cost incurred in the optimal equilibrium:

$$\mathbb{Y}_{c^*} := \{(x, u) \in \mathbb{X} \times \mathbb{U} : c(x, u) \leq c^*\}$$

Also, the following technical lemmas will be useful:

Lemma 2.2.7. *The set $\mathbb{Y}_{c^*}$ is compact.*

Lemma 2.2.8. *For any infinite sequence c with*

$$\bar{c} := \liminf_{T\to\infty} \frac{1}{T} \cdot \sum_{t=0}^{T-1} c(t) = c^* - \delta < c^*$$

and $\delta > 0$, there exists a subsequence with indices $T := (t_1, t_2, t_3, \dots)$ such that for all $k \in \mathbb{Z}^+$

(i) $c(t_k) \leq c^*$ *and*

(ii) $\frac{1}{t_{k+1}-t_k} \cdot \sum_{\tau=t_k}^{t_{k+1}-1} c(\tau) \leq c^* - \frac{\delta}{2}$.

The proofs are provided in Schmitt et al. (2016a). We are now ready to state the main result:

Theorem 2.2.3. *CSM systems with CSM objectives satisfying Assumption 2.2.2 are* optimally operated at steady-state*, that is, the minimum average infinite horizon cost achievable is equal to the cost in any optimal equilibrium*

$$c^*_\infty := \inf_{x(0),u} \liminf_{T\to\infty} c\left(x_u(t, x(0))\right) = c^*$$

Proof. By definition, $c^*_\infty \leq c^*$. Assume for the sake of contradiction that there exists an initial state $x(0) \in \mathbb{X}$ and a sequence $u(t) \in \mathbb{U}$, such that

$$\inf_{x(0),u} \liminf_{T\to\infty} c\left(x_u(t, x(0))\right) = c^* - \delta < c^*$$

for some $\delta > 0$. The proof relies on partitioning the trajectory into suitable segments, such that the system converges towards a limit point at the end of each of these intervals. Then, a procedure to construct an improving equilibrium from the trajectory during these intervals similar to the proof of Theorem 2.2.2 is described.

Consider the sequence $c = (c(x(0), u(0)), c(x(1), u(1), \dots)$ and let $T := (t_1, t_2, t_3, \dots)$ be a sequence of indices in accordance with Lemma 2.2.8. Since $\mathbb{Y}_{c^*}$ is compact, the sequence $(x(t_k), u(t_k))$ has a limit point $(x_\infty, u_\infty) \in \mathbb{Y}_{c^*}$. Therefore, one can define new time indices $\mathbb{S} := \{s_1, s_2, s_3, \dots\} \subseteq T$ such that $x(s_k) \overset{k\to\infty}{\longrightarrow} x_\infty$. These new indices induce a partition of the time into segments $\{s_k, \dots, s_{k+1} - 1\}$ and we will study the average state and input over each of these segments:

$$\bar{x}(k) := \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} x(\tau),$$

$$\bar{u}(k) := \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} u(\tau).$$

Because of convexity of the cost function c and Lemma 2.2.8, it follows that

$$c(\bar{x}(k), \bar{u}(k)) \leq \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} c(x(\tau), u(\tau)) \leq c^* - \frac{\delta}{2} \tag{2.70}$$

and therefore, $(\bar{x}(k), \bar{u}(k)) \in \mathbb{Y}_{c^*}$. Summing over each interval $\{s_k, \dots, s_{k+1} - 1\}$, we find

$$\begin{aligned}
\bar{x}(k) &= \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} x(\tau) \\
&= \frac{x(s_k) - x(s_{k+1})}{s_{k+1} - s_k} + \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k+1}^{s_{k+1}} x(\tau) \\
&= \frac{x(s_k) - x(s_{k+1})}{s_{k+1} - s_k} + \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} f\left(x(\tau), u(\tau)\right).
\end{aligned}$$

Using $\epsilon_k := (x(s_k) - x(s_{k+1}))/(s_{k+1} - s_k)$ and convexity of the system equations,

$$\begin{aligned}
\bar{x}(k) &= \epsilon_k + \frac{1}{s_{k+1} - s_k} \cdot \sum_{\tau=s_k}^{s_{k+1}-1} f\left(x(\tau), u(\tau)\right) \\
&\geq \epsilon_k + f\left(\bar{x}(k), \bar{u}(k)\right)
\end{aligned} \tag{2.71}$$

Again, since $\mathbb{Y}_{c^*}$ is compact and $(\bar{x}(k), \bar{u}(k)) \in \mathbb{Y}_{c^*}$, it follows that the sequence $(\bar{x}(k), \bar{u}(k))$ has a limit point $(\bar{x}_\infty, \bar{u}_\infty) \in \mathbb{Y}_{c^*}$. Therefore, one can yet again define new indices $K := \{k_1, k_2, k_3, \dots\} \subseteq \mathbb{Z}_0^+$ such that $\bar{x}(k_i) \overset{i\to\infty}{\longrightarrow} \bar{x}_\infty$.

Taking the limit as all indices go to infinity we note that $x(s_k) \overset{k\to\infty}{\longrightarrow} x_\infty$, $\bar{x}(k_i) \overset{i\to\infty}{\longrightarrow} \bar{x}_\infty$ and $\epsilon_k \overset{k\to\infty}{\longrightarrow} 0$. It follows that $(\bar{x}(k_i), \bar{u}(k_i)) \overset{i\to\infty}{\longrightarrow} (\bar{x}_\infty, \bar{u}_\infty) \in \mathbb{Y}_{c^*}$. In the limit, equation (2.71) turns into

$$\bar{x}_\infty \geq f\left(\bar{x}_\infty, \bar{u}_\infty\right)$$

Now, Lemma 2.2.6 implies the existence of an equilibrium x', u' improving over the cost $c(\bar{x}_\infty, \bar{u}_\infty)$, i.e. $c(x', u') \leq c\left(\bar{x}_\infty, \bar{u}_\infty\right)$. Using the bound derived in (2.70), it follows that

$$c(x', u') \leq \lim_{k\to\infty} c\left(\bar{x}(k), \bar{u}(k)\right) \leq c^* - \frac{\delta}{2}$$

which contradicts the initial assumption that c^* is the cost incurred in the optimal equilibrium and therefore completes the proof. □

In the context of economic MPC Ellis et al. (2014); Angeli et al. (2012); Grüne and Zanon (2014); Müller et al. (2015a), systems that are optimally operated at steady-state have been of particular interest and a connection between these systems and dissipative systems has been established: A system (2.69) is *dissipative* on a set $\mathbb{X} \times \mathbb{U} \subset \mathbb{R}^{n+m}$ with respect to the *supply rate* $s : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ if there exists a *storage function* $\lambda : \mathbb{X} \to \mathbb{R}_0^+$ such that

$$\lambda(f(x,u)) - \lambda(x) \leq s(x,u) \; \forall (x,u) \in \mathbb{X} \times \mathbb{U}.$$

In fact, by combining results from Angeli et al. (2012); Müller et al. (2015b,a) it has been shown that under some technical assumptions (closely related to Assumption 2.2.2), dissipativity with respect to the supply rate $s(x,u) = c(x,u) - c^*$ is a necessary and sufficient condition for optimal steady state operation. From Theorem 2.2.3, it therefore immediately follows:

Corollary 2.2.2. *Let system (2.69) be a CSM system and $c(x,u)$ a CSM cost, satisfying Assumption 2.2.2. Then, system (2.69) is dissipative on $\mathbb{X} \times \mathbb{U}$ with respect to the supply rate $s(x,u) = c(x,u) - c^*$.*

Before proceeding to an application of our main result, it is worth studying the applicability and limitations of the results. Recall, that in contrast to the general definition of a monotone system, we do not require monotonicity of the system dynamics in the control inputs. On the other hand, the standard definition of monotone systems is only concerned with the system dynamics, but not the objective, whereas we restrict ourselves to control objectives that are monotone in the state. In fact, quadratic objectives of the form $||x(t) - x^*||_Q^2$ with $Q \succeq 0$ and x^* being the desired steady state are a common choice in control. But if the state is not constrained to $x(t) \geq x^*$, such an objective is not monotone in the state and the presented results do not apply as demonstrated by the following counterexample.

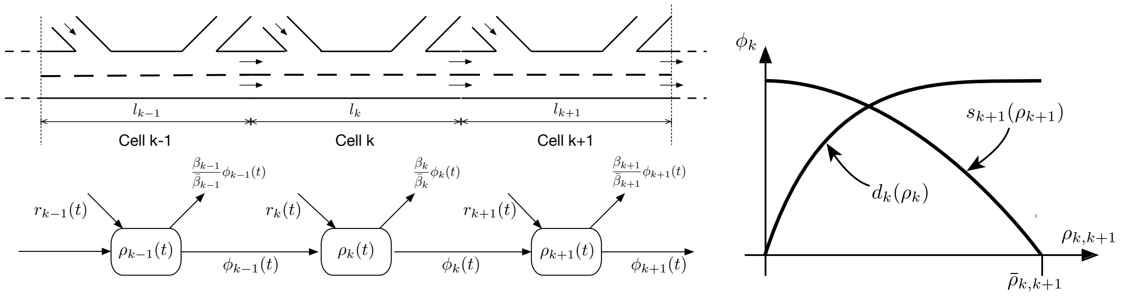
Example 2.2.4. *Consider the system with state $x \in \mathbb{R}_+^2$ and input $u \in [0,\ 2]$ with system dynamics*

$$x(t+1) = \begin{pmatrix} u \\ x_1(t)^2 \end{pmatrix}$$

and cost

$$c(x(t), u(t)) := \left\| x(t) - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|_2$$

It is easy to verify that the system dynamics are CSM (in $\mathbb{X} \times \mathbb{U}$) and that the cost is convex, but not *monotone in the state. We can parametrize all equilibria as $x_1 = u$, $x_2 = u^2$ with steady state costs*



(a) Freeway topology with traffic flows

(b) Fundamental diagram

Figure 2.33: In the CTM, a freeway is modeled by a combination of conservation equations according to the freeway topology with on- and offramps, as depicted on the left and the fundamental diagram, which characterizes the flow as a function of the densities of upstream and downstream cells, as depicted on the right.

of $\sqrt{u^2+(1-u^2)^2} \geq \frac{\sqrt{3}}{2}$. *But the cycle* $x_u(t,x(0))$ *with* $x(0) = (1\ 0)^\top$, $u(0) = 0$ *and* $u(1) = 1$ *achieves a better average cost of*

$$c\left(x_u(t,x(0))\right) = \frac{1}{2}\left(\sqrt{1+1}+0\right) = \frac{\sqrt{2}}{2} < \frac{\sqrt{3}}{2}$$

Therefore, this system is optimally operated off steady state.

A second counterexample illustrates the role of the state constraints in potentially causing a system to not be optimally operated in steady state.

Example 2.2.5. *Consider the system with state* $x \in \mathbb{R}^2_+$ *and input* $u \in [0,\ 4]^2$ *with system dynamics*

$$x(t+1) = x(t) + \begin{pmatrix} -2+u_1(t)^2 \\ -2+u_2(t)^2 \end{pmatrix},$$

and cost

$$c(x(t),u(t)) := \|u(t)\|_1 .$$

It is easy to verify that this is a CSM system with a CSM cost. However, note that the set $\mathbb{X}$ *is* not *control invariant and therefore violates Assumption 2.2.2. Any equilibrium requires* $u_1 = u_2 = \sqrt{2}$ *with steady-state cost* $c(x,u) = 2\cdot\sqrt{2}$. *But the cycle* $x_u(t,x(0))$ *with* $x(0) = (4\ 0)^\top$, $u(0) = (0,\ 2)^\top$ *and* $u(1) = (2,\ 0)^\top$ *achieves the average cost of*

$$c\left(x_u\left(t,x(0)\right)\right) = \frac{2+2}{2} = 2$$

Therefore, this system is optimally operated off steady-state. Without state constraints, the cost optimal trajectory makes the system diverge.

Application

We consider the problem of freeway ramp metering Papageorgiou et al. (2003); Papageorgiou and Kotsialos (2000), for a single freeway with its dynamics given by the asymmetric, monotone Cell Transmission Model (CTM) Daganzo (1994b); Gomes and Horowitz (2006b). In the CTM, the freeway is partitioned into cells of length l_k. The state of the mainline is described by the traffic density $\rho_k(t)$ of cell k

at time t, i.e. the number of cars per length in each cell. The density is constrained to $0 \leq \rho_k(t) \leq \bar{\rho}_k$, with the jam density $\bar{\rho}_k$. The density evolves over time according to the conservation equation

$$\rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + r_k(t) - \frac{1}{\bar{\beta}_k}\phi_k(t)\right).$$

Here, the mainline flows are denoted as $\phi_k(t)$ and the metering rates as $r_k(t)$. The metering rates are equivalent to the inflow to the freeway from the onramps.

The CTM models the flow $\phi_k(t)$ as a function of the traffic densities, which is represented by the so-called fundamental diagram, depicted in Figure 2.33b. The fundamental diagram for each cell can be decomposed in the concave, monotone nondecreasing traffic demand $d_k(\rho_k(t))$, i.e. the number of cars that seek to travel downstream, and the concave, monotone nonincreasing supply of free space $s_k(\rho_k(t))$ in a cell. The flow $\phi_k(t)$ is then given as the minimum of upstream demand and downstream supply $\phi_k = \min\{d_k(\rho_k(t)), s_{k+1}(\rho_{k+1}(t))\}$. The equations are slightly adapted for the first $\phi_0(t) = \min\{d_0, s_1(\rho_1(t))\}$ and last cell $\phi_n(t) = d_n(\rho_n(t))$. Here, d_0 is a constant, external traffic demand. We make the assumption of constant split ratios, which means that the outflows from the offramps are modeled as percentages β_k (often called the split ratios) of the mainline flows at the offramp in cell k. For notational convenience, we also use $\bar{\beta}_k := 1 - \beta_k$.

The metering rates $r_k(t)$ are obviously limited by the traffic demand $\bar{r}_k$ at the respective onramp $r_k(t) \leq \bar{r}_k$. The external demands $\bar{r}_k$ are assumed to be constant, since we are ultimately interested in a (quasi-) steady state setting. We also allow for a lower bound $\underline{r}_k \leq r_k(t)$ modeling a minimal flow that has to be admitted at all times.

We seek to maximize the (average) demand served, given as a nonnegative, linear combination of the cars admitted to enter the freeway $c^\top r(t)$, $c \geq 0$. This is equivalent to minimizing $c(\rho(t), r(t)) := -c^\top r(t)$. We note that the Total Travel Distance (TTD), commonly used as a short-term objective in ramp metering, is in fact equivalent to using the demand served as the objective in any equilibrium of the CTM. This is because the equilibrium flows can be expressed in terms of the metering rates as $\phi_k = \bar{\beta}_{1..k} d_0 + \sum_{i=1}^{k} \bar{\beta}_{i..k} r_k$, with $\bar{\beta}_{i..k} := \prod_{j=i}^{k} \bar{\beta}_i$, because of the conservation equations and vice versa. The same holds if the infinite horizon average case is considered, because of the finite storage capacity of the freeway.

It has been shown that for concave demand functions $d_k(\cdot)$ and concave supply functions $s_k(\cdot)$, finite horizon optimal control problems can be cast as convex optimization problems Gomes and Horowitz (2006b). It has also been shown that monotonicity of the systems equations plays a role in this Gomes et al. (2008). However, this observation also raised the possibility of the existence of a "free-lunch opportunity" in ramp metering for constant boundary conditions, wherein throughput is improved by repeating a congestion-decongestion cycle.

The theory for CSM systems now allows us to refute such a result:

Corollary 2.2.3. *There does not exist a periodic or aperiodic trajectory that improves infinite horizon average demand satisfaction in the monotone CTM over the optimal equilibrium, i.e.*

$$\inf_{\rho(0), r} \mathop{\mathit{lim\,inf}}_{T\to\infty} c(\rho_r(t, \rho(0))) = \left\{ \begin{array}{rl} \min & c(r, \rho) \\ \textit{s.t.} & \rho = f(\rho, r) \\ & \underline{r} \leq r \leq \bar{r} \end{array} \right.$$

Proof. Although the CTM in the presented form is monotone, it is easy to verify that it is *not* convex. Therefore, the results cannot be applied immediately. Instead, we study the following relaxation of the CTM.

We define a system with state $x_k(t)$ and controlled inputs $r_k(t)$ and $\varphi_k(t)$, given by the CSM systems equation

$$x_k(t+1) = x_k(t) + \frac{\Delta t}{l_k} \cdot \\ \left(\varphi_{k-1}(t) + r_k(t) + \frac{1}{\bar{\beta}_k} \max\left\{ -d_k(x_k(t)), -\varphi_k(t) \right\} \right)$$

with CSM objective

$$\tilde{c}(x(t), r(t), \varphi(t)) = -c^\top r(t) + \chi_{x_k(t) \leq \bar{\rho}_k(t)} + \chi_{\varphi_k(t) \leq s_{k+1}(x_{k+1}(t))}.$$

Here, χ_A denotes the characteristic function that is zero, if $x \in A$ and (positive) infinity otherwise. Checking convexity and state monotonicity of the objective is straightforward. For the system equations, note that the demand functions $d_k(\cdot)$ is concave, therefore, $-d_k(\cdot)$ is convex. Also, the $\max\{\cdot\}$ operator preserves convexity of its arguments. The feasible state set is chosen as $\mathbb{X} := \mathbb{R}^n_+$ in accordance with Assumption 2.2.2 and the compact, convex input set as $\mathbb{U} := \{u(t) = (r(t), \varphi(t)) : \underline{r}_k \leq r_k(t) \leq \bar{r}_k, 0 \leq \varphi_k(t) \leq \max\{s_{k+1}(0), d_k(\bar{\rho}_k)\}$. The bounds on $r_k(t)$ are taken from the original model, the bounds on $\varphi_k(t)$ are bounds on the range of $\phi_k(t)$ in the original model and are needed to make the input set $\mathbb{U}$ bounded.

In the following, let c^* denote the cost in an optimal equilibrium of the CTM and c^*_∞ the minimal infinite horizon average cost. We use $\tilde{c}^*$ to denote the cost in an optimal equilibrium of the relaxation and $\tilde{c}^*_\infty$ as the minimal infinite horizon average cost in the relaxation. From Theorem 2.2.3, it follows immediately that the relaxed system is optimally operated in steady state. In comparison to the CTM, the controlled inputs φ_k replace the flows ϕ_k of the original CTM and provide us with additional degrees of freedom, but any trajectory of the original CTM can be replicated in the relaxed system with the same costs. Thus, we have that

$$c^*_\infty \geq \tilde{c}^*_\infty = \tilde{c}^*. \tag{2.72}$$

It remains to be shown that the cost of the optimal steady state of the relaxed system can also be achieved by the CTM. To do so, we construct a suitable equilibrium of the CTM starting from an optimal equilibrium of the relaxation (x^*, r^*, φ^*). Note that for every equilibrium $\min\{\varphi^*_k, d_k(x^*_k)\} = \bar{\beta}_k \cdot (\varphi^*_{k-1} + r^*_k)$. We keep the optimal inflows r^*_k and propagate the flows through the freeway

$$\phi^*_0 := \varphi^*_0 \;, \; \phi^*_{k+1} := \bar{\beta}_{k+1} \left(\phi^*_k + r^*_{k+1} \right) \leq \varphi^*_{k+1}$$

By construction, the flows ϕ^* and metering rates r^* satisfy the conservation equations. We then proceed by computing the equilibrium densities as

$$\rho^*_k := d^{-1}_k(\phi^*_k)$$

with the pseudo-inverse of the demand function defined as $d^{-1}_k(\phi) := \min\left\{\rho : d_k(\rho) = \phi\right\}$. It remains to be checked that $\phi^*_k \leq s_k(\rho^*_k)$. Since $\phi^*_k = d_k(\rho^*_k) \leq d_k(x^*_k)$ and by monotonicity of the demand function, it follows that $\rho^*_k \leq x^*_k$ and by monotonicity of the supply function $\phi^*_k \leq s_{k+1}(x^*_{k+1}) \leq s_{k+1}(\rho^*_{k+1})$, as desired. Therefore, (ρ^*, r^*) is indeed an equilibrium of the original CTM, achieving the same cost as the equilibrium of the relaxation, since the metering rates have not been modified. This implies

$$\tilde{c}^* = c^* \tag{2.73}$$

By definition $c^* \leq c^*_\infty$, and by combining this inequality with equations 2.72 and 2.73, we conclude that

$$c^*_\infty = \tilde{c}^*_\infty = \tilde{c}^* = c^*.$$

which completes the proof. □

It shall be noted that the construction of a feasible equilibrium starting from the metering rates follows the lines of Gomes et al. (2008) which provides a detailed characterization of all equilibria of the piecewise-affine CTM. The generalization to the monotone CTM studied here is straightforward.

Conclusions

In this work, we have shown that convex, state monotone systems are optimally operated at steady-state. The question about which systems are optimally operated at steady state has so far mostly arisen in the context of economic MPC, in which costs are motivated by the actual operating costs/benefits instead of merely being a tuning tool for the control designer. Identification of classes of (nonlinear) systems which are optimally operated at steady state helps to facilitate controller design for the systems that fall into these classes. In particular, any stabilizing controller that makes a convex, state-monotone system converge to an optimal equilibrium is optimal with respect to the average, infinite horizon costs. Such a controller may be much easier to design and to certify than an economic MPC that stabilizes (optimal) cycles.

We applied our theoretical results to a problem in freeway ramp metering, for freeways modeled by the monotone CTM. By identifying a suitable, convex and state monotone relaxation of the systems dynamics, we were able to disprove a conjecture about the existence of a "free-lunch opportunity" by operating the system off steady-state. This result serves to reinforce trust in the CTM as a suitable tool for modeling freeway traffic since practical experience seems to contradict the existence of improving cycles for freeway ramp metering.

Based on these results, two main future research directions seem promising: On one hand, efforts should focus on identifying further classes of systems that are optimally operated at steady state. One the other hand, even though the importance of exploiting monotonicity and in particular, the combination of monotonicity and convexity in controller design has been stressed Rantzer (2015), it seems that results in literature are not fully developed yet.

2.2.3 One-Step-Ahead Optimal Urban Traffic Control

P. Grandinetti, F. Garin, C. Canudas de Wit

Our contribution to the Real-time event-based decision-making under uncertainty work package consists in the design of a novel algorithm for optimization of large–scale urban traffic network Grandinetti et al. (2015b).

The steadily increasing traffic demands have given rise to the need for efficient network operations. In this sense, traffic lights assume a fundamental role, since they are the major control measure in urban scenario.

Urban traffic control strategies are classified as *fixed–time* techniques and *model–based* algorithms. The main drawback of the former ones is that their settings are based on historical rather than real–time data, while the latter ones basic problem is the presence of discrete variables that require exponential complexity algorithms for a global optimization.

We have instead developed a decision–making scheme that can be implemented via linear optimization, and it is therefore computationally very efficient and scalable. This technique is based on a dynamical representation of traffic flow inside the newwork.

Traffic flow model

Traffic evolution in time and space is a complex system which strongly affects security and pollution; hence, effective and easy–to–handle models are needed to represent and control its behaviour.

The scientific community relies on macroscopic models of time–space evolution of the traffic. Such models describe traffic as a fluid, and are based on a mass conservation law . With respect to microscopic models, macroscopic ones are preferred due to their simplicity and accuracy in characterizing vehicles' flows and densities. The Cell Transmission Model (CTM) is an example, widely used, of this kind of representation.

We have build a model for large signalized traffic network based on the CTM, where flows at intersection of roads are regulated by traffic lights, and we have introduced the concept of averaged CTM, a more effective characterization of the system by means of control purpose. The averaged CTM evolution, tested in software simulation within a network with 40 roads, results to also have a good precision in terms of reliability with respect to the actual CTM network, as figure 2.34 shows.

Traffic performance

Traffic behaviour needs to be evaluated and assessed with respect to performance indices properly defined. There exist several metrics in literature to address performance evaluation; we focused on the following two features:

Service of Demand An urban traffic network is an highly dynamical environment that continously receives demand from outside. This demand cannot be ignored just to favour the inner quality of the system, because the external request will end up growing with several undesired effects, due to the bigger and bigger queues arising outside.

For this reason we define as service of demand the number of vehicles (users) served:

$$\mathrm{SoD}(t) = \int_0^t \varphi_r^{\text{in}}(\tau)d\tau$$

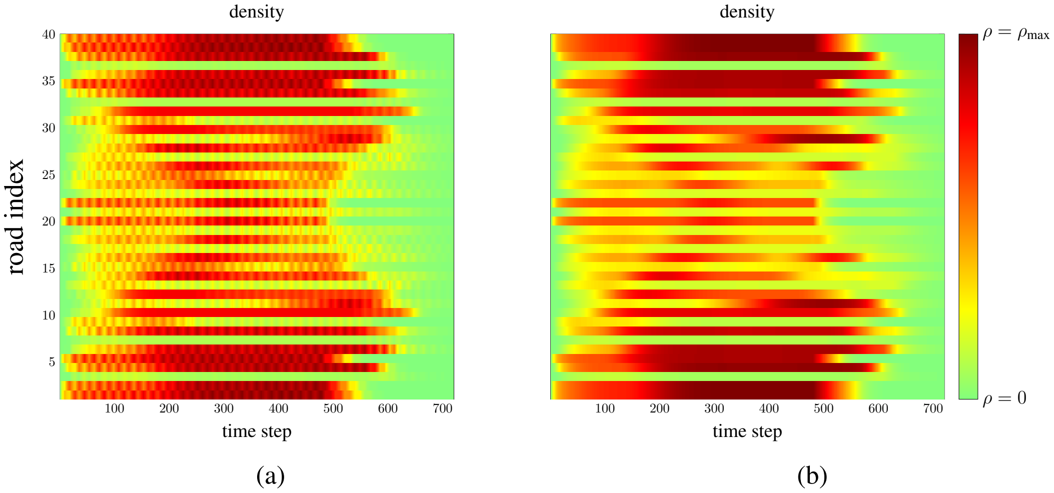


Figure 2.34: Precision of the averaged CTM approved via software simulation. In 2.34a (regarding the actual system) and in 2.34b (regarding the averaged system) each row shows the density of a road evolving in time.

where φ_r^{in} is the flow at the network boundary which enters in road r. The Service of Demand is a quantity that we would like to maximize.

Optimization of the infrastructures In urban networks some roads are preferred than others by the users. The infrastructure holder would like to set traffic lights as to diminish this usage disparity, to guarantee a more equilibrate diffusion of vehicles, thus reducing hard congestions in main streets as well as the possibility of accidents.

A standard metric that takes into account this behaviour is the Total Travel Distance, a cumulative index here defined as:

$$\text{TTD}(t) = \int_0^t \Big(\sum_{r \in \text{network}} \varphi_r(\tau) \Big) d\tau$$

where φ_r is the flow inside the road r.

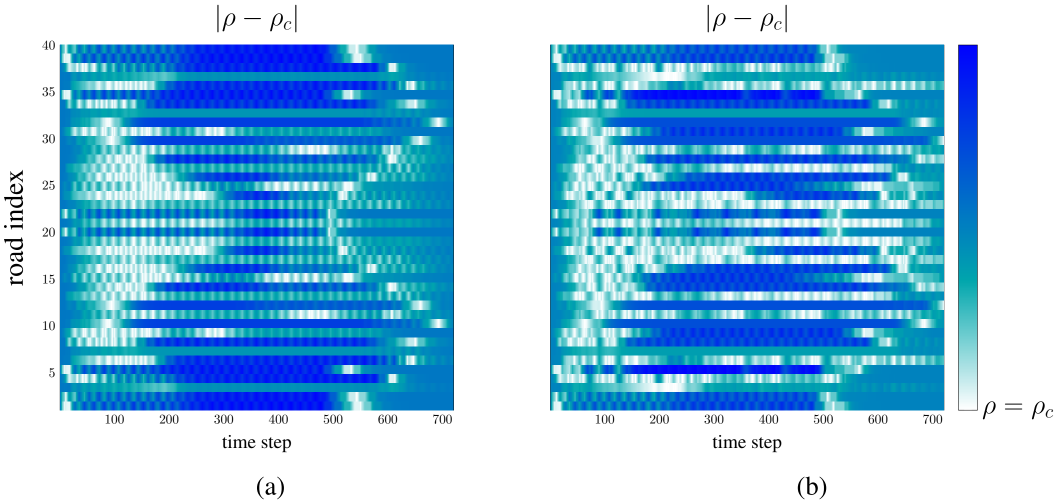
Decision–making strategy

We developed a strategy for deciding the duty cycle to be assigned to the traffic lights using a one–step–ahead control. The optimization problem we stated can be solved by means of linear programming, and it is therefore very suitable for practical purpose. The size of the Linear Program scales linearly ($\mathcal{O}(n)$) with the number of cells in the CTM of the road network. Thanks to this efficient optimization we can set duty cycles periodically, achieving good performance improvements for the two metrics previously defined. The effectiveness of our algorithm was tested in software simulation and compared to a fixed–decision strategy.

Representative results are showed in Figure 2.35 and in Table 2.2. Note that:

- Our algorithm achieves good performance regarding the optimization of the infrastructures. Figure 2.35 shows how far each road's density is from its best working point ρ_c, where lighter color means better performance;

- Table 2.2 gives the quantitative measures of the improvement, which is positive for both the choosen indices.

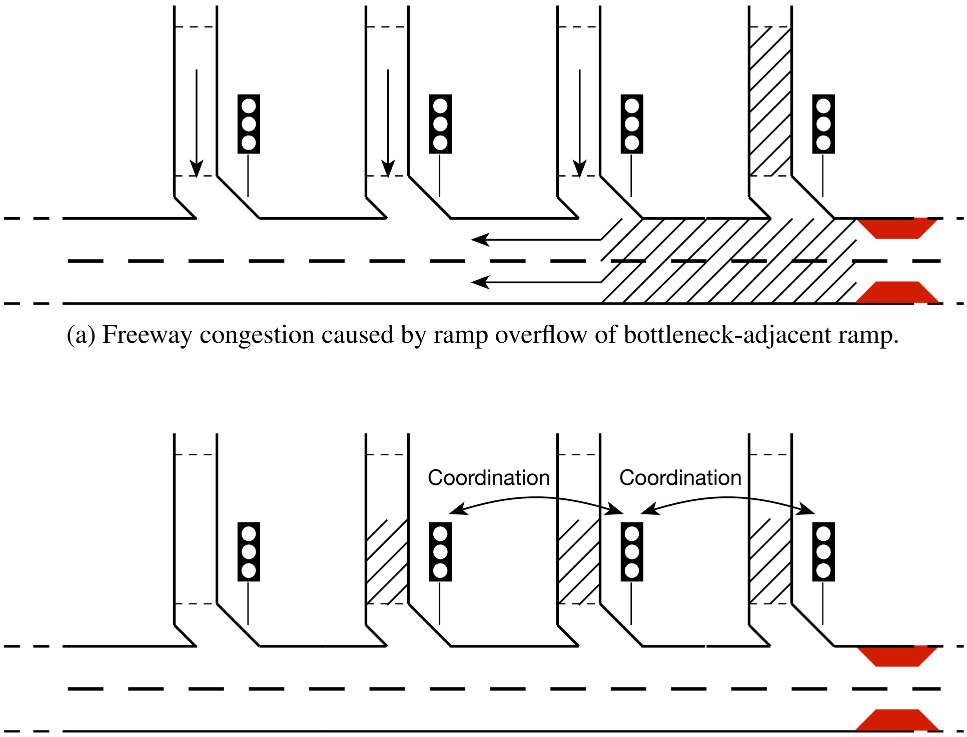


(a) (b)

Figure 2.35: Application of the proposed decision–making strategy. Figures 2.35a and 2.35b show the distance from the best working point (called *critical density*, ρ_c) for the system with fixed strategy and with our algorithm, respectively. Lighter color means better performance.

Table 2.2: Improvement in the network with the proposed control strategy.

Index	Improvement (%)
SoD	5.5 (per entering road)
TTD	14.6



(a) Freeway congestion caused by ramp overflow of bottleneck-adjacent ramp.

(b) Prevention of ramp overflow and hence freeway congestion by ramp coordination.

Figure 2.36: Benefits of coordination exemplified on the example of freeway ramp metering.

2.3 High-Level Coordination

The main objective of the high-level coordination algorithms is to ensure the optimization of a system-wide performance criterium or control objective. In traffic control, this objective is usually either flow maximization, maximization of demand satisfaction or travel time minimization. The necessity for global coordination can be exemplified via an intuitive example from freeway ramp metering. Consider a freeway segment as depicted in Figure 2.36. Assume furthermore that, during a limited period of time, the traffic demand exceeds the bottleneck (in red) capacity such that the demand cannot be served completely. In case of entirely uncoordinated local ramp metering as described in previous sections, the onramp immediately upstream of the bottleneck will solely attempt to prevent a congestion forming at the bottleneck. In practice, this situation will usually lead to ramp overflow quickly and a congestion spreading upstream from the bottleneck is the result as depicted in Figure 2.36a. By contrast, coordination between the ramps allows to distributed the control burden onto multiple ramps thereby preventing ramp overflow without causing a congestion on the mainline (Figure 2.36b. A similar case for the need of coordination can be made in the case of inner-city traffic control, where local maximization of traffic flows might cause more severe congestion downstream.

The algorithms described in the following sections address both the freeway and the inner-city traffic control scenario. We first (Section 2.3.1) present a case in freeway ramp metering, in which explicit coordination is *not* required in order to maximize flows and characterize the local feedback laws that achieve this. However, this case depends on monotonicity of the freeway traffic dynamics and next (Section 2.3.2), we present a generic algorithm for model-free learning of coordination patterns in the presence of disturbances. This algorithms is applied to the case of freeway ramp metering in the presence of *non-monotonic* effects and improvements resulting from the coordination are reported. Finally (Section 2.3.3), we consider the inner-city traffic optimal control problem. We pose a global optimization problem that is then decomposed into local problems similar to the ones described in previous sections. Then, an iterative coordination scheme is described that allows to approximate the solution to the global problem by iterating over solving the local problems and resolving conflicts in an averaging step.

2.3.1 Convergence to Flow-Optimal Equilibria

M. Schmitt, P. Goulart, A. Georghiou, J. Lygeros

We consider the freeway ramp metering problem, based on the Cell Transmission Model. This work addresses the question of how well distributed control strategies, e.g. local feedback controllers at every onramp, can maximize the traffic flow asymptotically under time-invariant boundary conditions. We extend previous results on the structure of steady-state solutions of the Cell Transmission Model and use them to optimize over the set of equilibria. By using duality arguments, we derive optimality conditions and show that closed-loop equilibria of certain distributed feedback controllers, in particular the practically successful "ALINEA" method, are in fact globally optimal.

Introduction

Active traffic control schemes have been established as an effective and practically useful tool to improve traffic flows on congestion-prone road networks Papageorgiou et al. (2003). This work concentrates on the freeway ramp metering problem, where we can actively control the number of cars that enter the freeway using a specific onramp. A survey of ramp metering strategies can be found in Papageorgiou and Kotsialos (2000). To model the freeway traffic dynamics, we make use of the Cell Transmission Model (CTM), which was originally derived as a first-order Godunov approximation of the kinematic wave partial differential equation Daganzo (1994b, 1995). More precisely, we adopt the "asymmetric" CTM Gomes and Horowitz (2006b); Gomes et al. (2008), which simplifies the model of onramp-mainline merges in comparison to the originally proposed formulation. Its popularity for model-based control stems from the simplicity of the model equations, allowing for computationally efficient solutions methods for optimal control problems Gomes and Horowitz (2006b); Ziliaskopoulos (2000).

A variety of local feedback strategies, i.e., ramp metering controllers that only receive measurements from sensors in close vicinity of the onramp location, have been described in literature, e.g. Papageorgiou et al. (1991); Stephanedes (1994); Zhang and Ritchie (1997). These strategies have been shown to come close to the performance of optimal control strategies in practical applications, even though they only aim to maximize bottleneck flows locally Smaragdis et al. (2004); Wang et al. (2014). While it is apparent that such local feedback controllers are far easier to implement and to configure than centralised, model-based optimal control strategies, it is not obvious why the former tend to come close in performance to the latter in practice Papamichail et al. (2010b). An explanation is given in Zhang and Levinson (2004), which explicitly constructs the optimal control strategy for a special case – it is assumed for example that there are no internal freeway queues – and states that the structure of the explicit solution "explains why some local metering algorithms [...] are successful – they are really close to the most-efficient logic."

In this work, we address the question of how distributed control strategies, such as local feedback controllers at every onramp, compare to optimal control strategies asymptotically under time-invariant boundary conditions. The idea to focus on traffic controllers which are only required to achieve convergence to an optimal equilibrium (instead of solving the far more challenging problem of optimizing the transient behavior) has recently received attention in a series of papers on traffic density balancing Pisarski and Canudas-de Wit (2012a, 2013, 2012b). It is well known that the problem of maximizing the flow over the set of equilibria of the CTM can be posed as a linear program (LP). By using duality arguments, we derive simple optimality conditions for flow-maximizing equilibria. We show that all closed-loop equilibria of certain local feedback controllers, in particular the practically successful ALINEA method Papageorgiou et al. (1991), are in fact globally optimal in the idealized CTM.

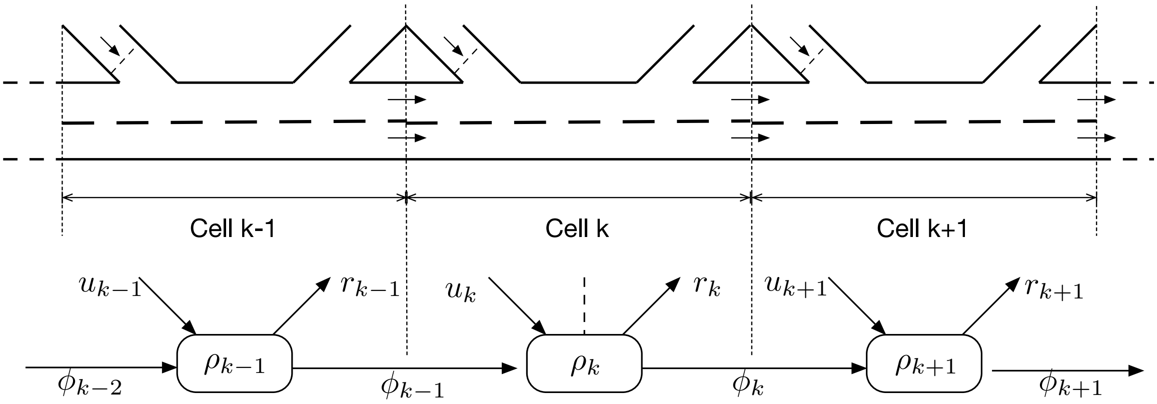


Figure 2.37: Sketch of the cell transmission model

The paper is organized as follows: Section 2.3.1 introduces the CTM. In Section 2.3.1, we extend previous results on the structure of equilibria of the CTM. In Section 2.3.1, we optimize over the set of equilibria of the CTM and derive optimality conditions. Section 2.3.1 demonstrates the results on a simple example and Section 4 comments on the main results and discusses practical implications.

Problem formulation

In this section, we motivate and introduce the (asymmetric) Cell Transmission Model (CTM) Gomes and Horowitz (2006b); Gomes et al. (2008). Throughout this work, we consider a freeway section as depicted in Figure 2.37. The CTM admits the following intuitive explanation: The freeway is partitioned into n sections or *cells* of length l_k. The state of the highway is described by the traffic *density* $\rho_k(t)$ in each cell k at sampling time t. Since the CTM is a first order model, the velocity is not part of the state. The evolution of the traffic is described by the traffic *flows* $\phi_k(t)$, i.e., the number of cars that move from cell k to cell $k+1$ in one time interval Δt. We model the *off-ramp flows* $r_k(t) = \frac{\beta_k}{\bar{\beta}_k}\phi_k(t)$ as a proportion of the mainline flow $\phi_k(t)$ with the (constant) *split ratios* β_k and $\bar{\beta}_k := 1 - \beta_k$. The flow entering the freeway via an onramp at cell k is denoted u_k. The flow entering the considered freeway section on the mainline is denoted u_0. We can thus formulate the conservation law for each cell as:

$$\rho_k(t+1) = \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + u_k(t) - \frac{\phi_k(t)}{\bar{\beta}_k}\right).$$

The road conditions are described by the so-called *free-flow velocity* v_k, the *congestion-wave velocity* w_k and the *jam density* $\bar{\rho}_k$. The flows $\phi_k(t)$ are limited by the number of cars in the origin cell that want to travel downstream $(\bar{\beta}_k v_k \rho_k(t))$, the *capacity* of the highway F_k and the available free space $((\bar{\rho}_{k+1} - \rho_{k+1}(t))w_{k+1})$ in the receiving cell:

$$\phi_k(t) = \min\left\{\bar{\beta}_k v_k \rho_k(t), F_k, (\bar{\rho}_{k+1} - \rho_{k+1}(t))w_{k+1}\right\}.$$

This relationship can be visualized with the so-called fundamental diagram as depicted in Figure 2.38. The *critical density* $\rho_k^c = \frac{w_k}{v_k+w_k}\bar{\rho}_k$ is the density value at which the fundamental diagram is maximized. The peak value of both the sending as well as the receiving cell determine the values of the capacities F_k, which are defined as $F_k := \min\left\{\bar{\beta}_k v_k \rho_k^c, (\bar{\rho}_{k+1} - \rho_{k+1}^c)w_{k+1}\right\}$ [8] . This equation is slightly adapted

[8] Note that some authors allow for capacities $0 < F_k \le \min\left\{\bar{\beta}_k v_k \rho_k^c, (\bar{\rho}_{k+1} - \rho_{k+1}^c)w_{k+1}\right\}$ which leads to a fundamental diagram of trapezoidal shape.

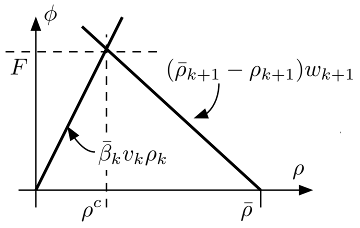


Figure 2.38: Sketch of the fundamental diagram for uniform freeway conditions, i.e., parameters and densities that do not differ between consecutive cells.

	Symbol	Name/Definition	Unit
Variables	ϕ_k	flow	1/h
	ρ_k	density	1/mile
	u_k	onramp flow	1/h
Parameters	β_k	split ratio	1
	$\bar\beta_k$	$1 - \beta_k$	1
	l_k	cell length	mile
	v_k	free-flow velocity	mile/h
	w_k	congestion-wave velocity	mile/h
	$\bar\rho_k$	jam density	1/mile
	Δ_t	sampling time interval	h

Table 2.3: Summary of symbols

for the first and the last cell, where we define $F_0 = (\bar\rho_1 - \rho_1^c)w_1$ and $F_n = \min\left\{\bar\beta_n v_n \rho_n^c, \bar\phi_n\right\}$. Here, $u_0 \geq 0$ models the *mainline traffic demand* and $\bar\phi_n \geq 0$ is some arbitrary, constant bound on the outflow from the highway. All parameters and variables of the CTM are summarized in Table **??**. Note that all parameters of the CTM are positive and all states are nonnegative. Furthermore, the split ratios are limited to the interval $0 \leq \beta_k < 1$ and the sampling time Δ_t is restricted to $\Delta_t \leq \frac{l_k}{v_k}$ to ensure convergence.

For a given initial state $\rho_k(0)$ and ramp metering rates $u_k(t)$ for $1 \leq k \leq n$ and $t \in \mathbb{N}$, the evolution of the highway is described by the following equations:

$$\begin{aligned}
\rho_k(t+1) &= \rho_k(t) + \frac{\Delta t}{l_k}\left(\phi_{k-1}(t) + u_k(t) - \frac{\phi_k(t)}{\bar\beta_k}\right), \\
\phi_0(t) &= \min\left\{u_0, F_0, (\bar\rho_1 - \rho_1(t))w_1\right\}, \\
\phi_k(t) &= \min\left\{\bar\beta_k v_k \rho_k(t), F_k, (\bar\rho_{k+1} - \rho_{k+1}(t))w_{k+1}\right\}, \\
\phi_n(t) &= \min\left\{\bar\beta_n v_n \rho_n(t), F_n, \bar\phi_n\right\}.
\end{aligned}$$

Note that this model includes the implicit assumption that congestion does not spill back onto the on-ramps. While this assumption might not be satisfied for an uncontrolled highway, it was shown to be satisfied by a large margin for a highway controlled by ramp metering in a field study Gomes and Horowitz (2006b), since such a controller will limit inflows and mainline congestion by design.

Equilibria of the CTM

In this work, we are mainly interested in optimal steady-state solutions to the CTM equations. We start by deriving some general properties in the form of three lemmas that characterize equilibria of the CTM, which will be important tools in the analysis of optimality of such equilibria in the following sections. In the following, if the time index is omitted for some variable, e.g. ϕ_k instead of $\phi_k(t)$, the variable is to be understood as a steady-state value.

The equations describing steady-state solutions or equilibria of the CTM can be derived by imposing $\rho_k(t+1) = \rho_k(t) =: \rho_k$ (and removing the time index t from all variables), which yields:

$$\begin{aligned}
\phi_k &= (\phi_{k-1} + u_k)\,\bar{\beta}_k, && 1 \le k \le n,\\
\phi_0 &= \min\left\{u_0, F_0, (\bar{\rho}_1 - \rho_1)w_1\right\}, &&\\
\phi_k &= \min\left(\bar{\beta}_k v_k \rho_k, F_k, (\bar{\rho}_{k+1} - \rho_{k+1})w_{k+1}\right), && 1 \le k < n,\\
\phi_n &= \min\left\{\bar{\beta}_n v_n \rho_n, F_n, \bar{\phi}\right\}.
\end{aligned} \tag{2.74}$$

We call u the (traffic) *demand*, which consists of the mainline demand u_0 and the onramp inflows u_k, $1 \le k \le n$ (as introduced earlier). Note that the onramp flows can be changed by ramp metering and thus serve as control inputs, whereas the mainline demand is fixed. For ease of notation, we define the *equilibrium set* $\mathcal{E}(u)$ as

$$\mathcal{E}(u) = \{(\phi, \rho) : \text{For fixed demand } u,\ (\phi, \rho) \text{ satisfy } (2.74)\}.$$

We call a section k a *bottleneck* if $\phi_k = F_k$. The locations of the bottlenecks are important in the analysis of the CTM equilibria Gomes et al. (2008). Assuming that there are $m-1$ bottlenecks $b_1, b_2, \ldots, b_{m-1}$, these bottlenecks partition the highway into m *segments* $S_1 = \{0, \ldots, b_1\}$, $S_2 = \{b_1+1, \ldots, b_2\}, \ldots,$ $S_m = \{b_{m-1}+1, \ldots, n\}$, as depicted in Figure 2.39. Note that the first and the last segment may be empty if ϕ_0 or ϕ_n are bottleneck flows. For a constant demand u, define the *induced flows* φ as:

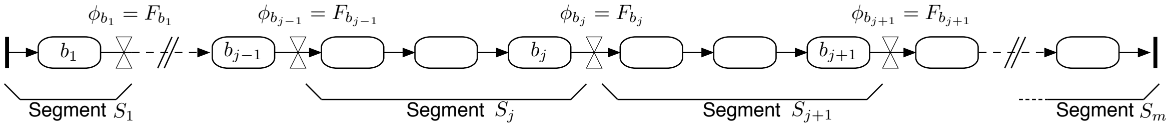


Figure 2.39: Segments, defined according to the bottleneck locations, in a CTM representation of a freeway.

$$\begin{aligned}
\varphi &:= u_0, &&\\
\varphi_k &:= (\varphi_{k-1} + u_k)\bar{\beta}_k, && 1 \le k \le n,
\end{aligned}$$

i.e., the induced flows are the flows that result in steady-state if the complete traffic demand can be accommodated by the freeway.

We can categorize traffic demands for a particular freeway according to whether or not they can be served: The traffic demand u is called *feasible* if the induced flows are equal or smaller than the local capacities in every section $\varphi_k \le F_k\ \forall k$. It is called *strictly feasible* if the induced flows are strictly smaller than the capacity in every section $\varphi_k < F_k\ \forall k$ and it is called *marginally feasible* if it is feasible, but not strictly feasible. Using these definitions, we are now ready to characterize equilibria of the CTM.

Lemma 2.3.1. *For a feasible traffic demand u, the unique equilibrium flows of the CTM are equal to the induced flows $\phi_k = \varphi_k$. Equilibrium densities are as follows:*

(i) For strictly feasible demands, the unique equilibrium is the uncongested equilibrium with densities $\rho_k = \frac{\phi_k}{\beta_k v_k} < \rho_k^c$.

(ii) For marginally feasible demands, the equilibrium densities are no longer unique. One particular equilibrium is the uncongested equilibrium with densities $\rho_k = \frac{\phi_k}{\beta_k v_k} \leq \rho_k^c$.

Proof. This result follows immediately from Theorem 4.1 in Gomes et al. (2008). □

Note that Theorem 4.1 in Gomes et al. (2008) also states the complete set of equilibrium densities for case (ii) explicitly. For our purposes, it is sufficient to know that a non-empty set of equilibrium densities exists for every feasible demand. We also need to consider situations in which the mainline demand cannot be completely served and a congestion on the mainline spills back outside of the considered part of the freeway. If the demand $u = \{u_0, u_1, \dots, u_n\}$ is infeasible, but the demand $\tilde{u} = \{0, u_1, \dots, u_n\}$ is feasible, we call the demand u *onramp-feasible*. Onramp-feasible demands exceed the capacity of the freeway, but the capacity is sufficient to accommodate the onramp flows alone, assuming zero mainline-flow.

Lemma 2.3.2. *For onramp-feasible demands, the unique equilibrium flow is given as* $\varphi_0 = \max\{x \geq 0 : (x, u_1, \dots, u_n)$ *is feasible*$\})$ *and* $\varphi_k := (\varphi_{k-1} + u_k)\bar{\beta}_k$ *for* $1 \leq k \leq n$. *The equilibrium densities are not unique in general. One particular equilibrium is given as a congested first segment* $\rho_k = \bar{\rho}_k - \frac{\phi_k}{w_k} \geq \rho_k^c,\ \forall k \in S_1$ *and the remaining freeway operating in free-flow* $\rho_k = \frac{\phi_k}{\beta_k v_k} \leq \rho_k^c,\ \forall k \notin S_1$.

Proof. See Appendix. □

In an equilibrium, some cells in a segment might be congested while other cells operate in free-flow. The order of congestion and free-flow conditions is not arbitrary, but follows a specific pattern instead. Let cell q_j be the last cell within segment S_j which is not congested, i.e. $q_j := \text{argmax}_k\{k \in S_j\ :\ \rho_k \leq \rho_k^c\}$. If all cells within a segment j are congested, define $q_j := b_j + 1$.

Lemma 2.3.3. *All cells downstream of cell* q_j *within a segment are congested:* $\rho_k > \rho_k^c,\ k \in S_j, k > q_j$. *All cells upstream of cell* q_j *within a segment operate in free-flow:* $\rho_k < \rho_k^c,\ k \in S_j, k < q_j$.

Proof. The first assertion holds by definition of q_j. The second assertion can be proven by contradiction: Assume some cell(s) upstream of cell q_j, but within the same segment are congested. Let $n_j := \text{argmax}_k\{k \in S_j, k < q_j : \rho_k \geq \rho_k^c\}$ be the most downstream one of those cells. By definition, cell n_j is congested, but cell n_j+1 is not. It follows that $\phi_{n_j} = \min\left\{\bar{\beta}_{n_j} v_{n_j} \rho_{n_j}, F_{n_j}, (\bar{\rho}_{n_j+1} - \rho_{n_j+1}) w_{n_j+1}\right\} \geq \min\left\{\bar{\beta}_{n_j} v_{n_j} \rho_{n_j}^c, F_{n_j}, (\bar{\rho}_{n_j} - \rho_{n_j}^c) w_{n_j}\right\} = F_{n_j}$, i.e. cell $n_j + 1$ is a bottleneck. This is a contradiction to the assumption that there are no bottlenecks within a segment. □

Intuitively, this means that there exists only one congestion queue per segment[9], which starts in cell q_j. Furthermore, there can at most be one cell within a segment (cell q_j) in which the density actually equals the critical density. The resulting congestion/ free-flow pattern is visualized in Figure 2.40.

Remark 2.3.1. *Note that the equilibrium flows as a function of the traffic demand* $\phi(u)$ *can be written as*

$$\begin{aligned} &\phi_0 = \max\{\varphi : \varphi \leq u_0,\ (\varphi, u_1, \dots, u_n)\}, && \\ &\phi_k = (\phi_{k-1} + u_k)\bar{\beta}_k, && 1 \leq k \leq n, \end{aligned}$$

for both feasible demands, yielding simply $\phi_0 = u_0$, *and onramp-feasible demands, by definition.*

[9] A similar result is stated in Lemma 4.3 in Gomes et al. (2008). The cases $\rho_k < \rho_k^c$ and $\rho_k = \rho_k^c$ are not distinguished, however, but this distinction will be crucial in our further analysis.

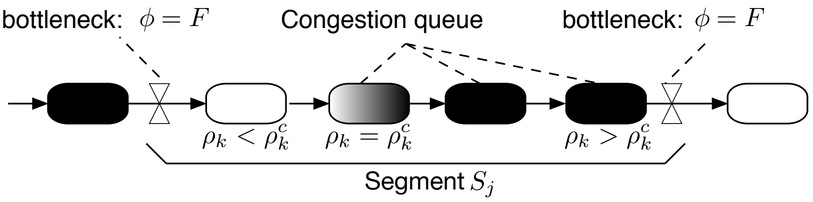


Figure 2.40: Pattern of the highway state within one segment. Note that there is not necessarily a cell in which the critical density is exactly achieved ($\rho_k = \rho_k^c$). In this case, cell q_i is the last cell in free flow: $\rho_k < \rho_k^c$.

Having established various properties of the equilibrium flows and densities for fixed onramp inflows u, we are now ready to address the problem of finding the optimal ramp metering rates u to maximize traffic flows.

Optimal Equilibria

In this section, we will address the problem of optimizing over steady-state equilibria of the CTM. To this end, we will first state the (nonconvex) main problem, show the equivalence of this problem with a suitable linear relaxation and then use duality arguments to derive optimality conditions.

Consider a highway modeled by the CTM and controlled by ramp metering. We want to find the equilibrium that maximizes a positive combination of all flows $c^\top \phi$ ($c \in \mathbb{R}_+^{n+1}$) by choosing appropriate (steady-state) ramp metering rates u_k. The ramp metering rates are assumed to be constrained by box constraints $\underline{u}_k \leq u_k \leq \bar{u}_k$. In the simplest case, the lower limit might be equal to zero to prevent negative flows and the upper bounds will reflect the maximal number of cars that want to enter the freeway at a certain onramp per time period. We assume that the lower bounds $(u_0, \underline{u}_1, \dots, \underline{u}_n)$ on the inflows are onramp-feasible, guaranteeing the existence of a feasible solution. Thus we consider the following optimization problem over equilibria of the CTM:

$$\begin{aligned} \underset{u,\phi,\rho}{\text{maximize}} \quad & c^\top \phi \\ \text{subject to} \quad & \underline{u}_k \leq u_k \leq \bar{u}_k, \quad 1 \leq k \leq n, \\ & (\phi, \rho) \in \mathcal{E}(u). \end{aligned} \tag{2.75}$$

Note that Problem (2.75) includes the maximization of the Total Travel Distance TTD $:= \sum_{k=0}^{n} \frac{\phi_k}{\beta_k}$ or the total discharge flows $r_{tot} := \phi_n + \sum_{k=1}^{n} \frac{\beta_k}{\beta_k} \phi_k$ as special cases. Also note that Problem (2.75) is nonconvex, due to the nonlinear flow constraints that make the set of equilibria $\mathcal{E}(u)$ a nonconvex set.

Even though the flow-constraints are nonconvex, it is well-known that there exists an uncongested, flow-maximizing equilibrium for freeways under fairly general conditions Wattleworth (1965); Chen et al. (1974). One can find this solution by solving a suitable Linear Program, for example by introducing a condition that restricts the highway to free-flow conditions (which does not introduce any conservativeness in terms of the objective value, as long as the problem remains feasible). We want to avoid this explicit restriction and consider the following relaxation:

$$\begin{aligned} \underset{u,\phi}{\text{maximize}} \quad & c^\top \phi \\ \text{subject to} \quad & \underline{u}_k \leq u_k \leq \bar{u}_k, && 1 \leq k \leq n, \\ & \phi_k = (\phi_{k-1} + u_k) \cdot \bar{\beta}_k, && 1 \leq k \leq n, \\ & \phi_k \leq F_k, && 1 \leq k \leq n, \\ & \phi_0 \leq \tilde{F}_0 := \min\{F_0, u_0\}, \end{aligned} \tag{2.76}$$

in which the nonconvex flow-constraints have been relaxed and the constraints involving the densities have been removed altogether.

Proposition 2.3.1. *Let $c \in \mathbb{R}_+^{n+1}$. Then problem (2.75) is equivalent to the relaxation (2.76), in the sense that the objective values are equal. Furthermore, given a maximizer (u^*, ϕ^*) of the relaxed problem (2.76), we can compute densities ρ^* such that (u^*, ϕ^*, ρ^*) is a maximizer of (2.75).*

Proof. Since (2.76) is a relaxation of (2.75), it is sufficient to show that for any optimizer ϕ^*, u^* to (2.76), we can find ρ^* such that ϕ^*, u^*, ρ^* are feasible for (2.75). Assume u^*, ϕ^* are solutions of the relaxed problem (2.76). Then ϕ^* is also the solution of $\phi^* = \underset{\phi}{\text{argmax}}\{c^\top \phi : \phi \in \mathcal{F}\}$ with:

$$\mathcal{F} := \left\{ \begin{array}{ll} \phi_k = (\phi_{k-1} + u_k^*) \cdot \bar{\beta}_k & 1 \leq k \leq n \\ \phi_k \leq F_k & 0 \leq k \leq n \end{array} \right. .$$

For fixed onramp flows, all flows ϕ_k can be expressed as affine functions of ϕ_0, thus we can rewrite:

$$\begin{aligned} \phi^* &= \underset{\phi}{\text{argmax}}\{c^\top \phi : \phi \in \mathcal{F}\} && = \underset{\phi}{\text{argmax}}\{a\phi_0 + b : \phi \in \mathcal{F}\} \\ &= \underset{\phi}{\text{argmax}}\{\phi_0 : \phi \in \mathcal{F}\} && = \phi(u^*) \end{aligned}$$

for suitable $a, b \in \mathbb{R}_+$. We see that the flows ϕ^* equal the equilibrium flows for the fixed traffic demands u^* as stated in Remark 2.3.1. We have previously established that the set of equilibrium densities is nonempty in every case, according to Lemmas 2.3.1 and 2.3.2. Therefore, we can always find feasible equilibrium densities ρ^* and equivalence of the optimization problems holds. □

Note that in general, the optimal solution set for the densities includes equilibria in which some cells are congested. While optimizing over free-flow conditions will yield a global optimum, the CTM also allows for (partly) congested equilibria, which achieve the same objective value.

We will now derive sufficient optimality conditions for the maximal flow problem (2.75). Theses optimality conditions will be tailored for the analysis of distributed control approaches and are based on the following consideration: Assume a distributed controller with the objective of moving the local density at every controlled onramp (as close as possible) to the critical density. It is easy to check that each individual flow will be maximal if $\rho_k = \rho_k^c$ is achieved in every cell, which is the main idea behind this control approach. However, this will in general not be the case due to saturation of the available control inputs (the ramp metering rates), in particular if multiple local controllers interact while controlling a freeway. Therefore, the performance of such closed-loop equilibria in terms of flow maximization is not clear a priori. The following theorem presents sufficient optimality conditions, by imposing only local constraints on the ramp metering rates, dependent only on the local densities:

Theorem 2.3.1. *Let u, ϕ, ρ be an equilibrium of the CTM and assume the equilibrium ramp metering u rates satisfy:*

$$\begin{aligned}
&\text{“free-flow”:} && u_k = \bar{u}_k && \forall k \,:\, \rho_k < \rho_k^c, \\
&\text{“critical density”:} && \underline{u}_k \le u_k \le \bar{u}_k && \forall k \,:\, \rho_k = \rho_k^c, \\
&\text{“congestion”:} && u_k = \underline{u}_k && \forall k \,:\, \rho_k > \rho_k^c.
\end{aligned} \tag{2.77}$$

Then u, ϕ, ρ solve the maximal flow problem (2.75).

Proof. Feasibility of the primal problem (2.75) holds by assumption. To verify optimality, it is sufficient to show that the equilibrium flows solve the LP-relaxation (2.76), according to Proposition 2.3.1. Consider the dual of problem (2.76) which is given as:

$$\begin{aligned}
\underset{\mu.\nu,\xi,\lambda}{\text{minimize}} \quad & \nu^\top \underline{u} - \xi^\top \bar{u} - \lambda^\top F \\
\text{subject to} \quad & \mu_k + \nu_k - \xi_k = 0 && 1 \le k \le n \\
& -c_k + \tfrac{\mu_k}{\beta_k} - \mu_{k+1} + \lambda_k = 0 && 0 \le k \le n \\
& \mu_0 = \mu_{n+1} = 0 \\
& \mu \in \mathbb{R}^{n+2}, \nu \in \mathbb{R}_+^n, \xi \in \mathbb{R}_+^n, \quad \lambda \in \mathbb{R}_+^{n+1}
\end{aligned} \tag{2.78}$$

To simplify notation, we have introduced $\mu_0 := 0$ and $\mu_{n+1} := 0$. We will show that for any solution to the primal problem which satisfies (2.77), we can construct a complementary dual solution, thus proving that the primal solution is indeed optimal. The complementarity conditions are given as

$$\begin{aligned}
0 \le (\underline{u} - u) \perp \nu \ge 0, \\
0 \le (u - \bar{u}) \perp \xi \ge 0, \\
0 \le (\phi - F) \perp \lambda \ge 0.
\end{aligned}$$

For ease of notation, we define the following index sets: The set of all bottlenecks $B := \{k : F_k = \phi_k\}$, the set of all congested cells $C := \{k : \rho_k > \rho_k^c\}$, the set of all cells in free flow $F := \{k : \rho_k < \rho_k^c\}$ and the set of all cells in which the critical density is achieved $D := \{k : \rho_k = \rho_k^c\}$. By combining the constraints for dual feasibility and the complementarity constraints, we end up with the following optimality conditions:

$$\phi, \rho \in \mathcal{E}(u), \quad u, \rho \text{ satisfy (2.77)}, \tag{2.79a}$$

$$-c_k + \frac{\mu_k}{\beta_k} - \mu_{k+1} + \lambda_k = 0, \tag{2.79b}$$

$$\mu_k = 0 \text{ if } k \in D, \quad \lambda_k = 0 \text{ if } k \notin B, \tag{2.79c}$$

$$\mu_k \ge 0 \text{ if } k \in F, \quad \mu_k \le 0 \text{ if } k \in C, \tag{2.79d}$$

$$\lambda_k \ge 0 \text{ if } k \in B. \tag{2.79e}$$

We will now find an explicit solution to the dual problem, by constructing a solution to every segment S_j. Define

$$\mu_{q_j} := 0.$$

Then feasible multipliers μ_k for all other cells in segment S_j can be computed by iterating in upstream/downstream direction, starting from the cell $b_j + n_F$:

$$\begin{aligned}
\mu_k &:= \beta_k \left(\mu_{k+1} + c_k\right), && b_{j-1} < k < q_j, \\
\mu_k &:= \frac{\mu_{k-1}}{\beta_{k-1}} - c_{k-1}, && q_j < k \le b_j,
\end{aligned}$$

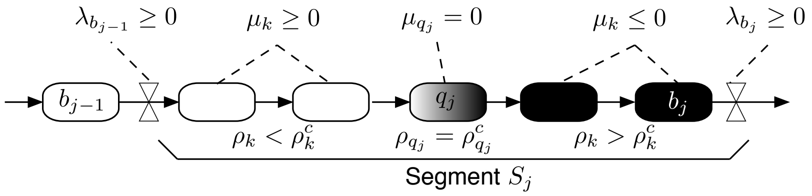


Figure 2.41: Dual variables corresponding to the congestion/ free-flow pattern of the freeway within one segment.

and the multipliers λ are computed as

$$\begin{aligned} \lambda_k &:= c_k - \frac{\mu_k}{\bar{\beta}_k} + \mu_{k+1}, && k \in B, \\ \lambda_k &:= 0, && k \notin B. \end{aligned}$$

The resulting pattern within one segment is depicted in Figure 2.41.

We will now verify that the proposed solution indeed solves the optimality conditions (2.79). Note that condition (2.79a) is satisfied by assumption. Conditions (2.79b) and (2.79c) are satisfied by construction. Conditions (2.79d) and (2.79e) remain to be checked.

We will first show that conditions (2.79d) hold for a generic segment S_j. For this we need some intermediate steps:

- Recall that the parameters $\bar{\beta}_k$ and c_k are nonnegative. We claim that the multipliers of all cells upstream of cell q_j, but within segment S_j, are nonnegative. This can be verified by induction: $\mu_{q_j} = 0$ by definition. Assume now that $\mu_k \geq 0$. Then $\mu_{k-1} = \bar{\beta}_{k-1}(\mu_k + c_{k-1}) \geq 0$ for all cells with indices $b_{j-1} < k < q_j$.

- Conversely, the multipliers of all cells downstream of cell q_j, but within segment S_j, are nonpositive. This can again be verified by induction: $\mu_{q_j} = 0$ by definition. Assume now that $\mu_k \leq 0$. Then $\mu_{k+1} = \frac{\mu_k}{\bar{\beta}_k} - c_k \geq 0$ for all cells with indices $q_j < k \leq b_j$.

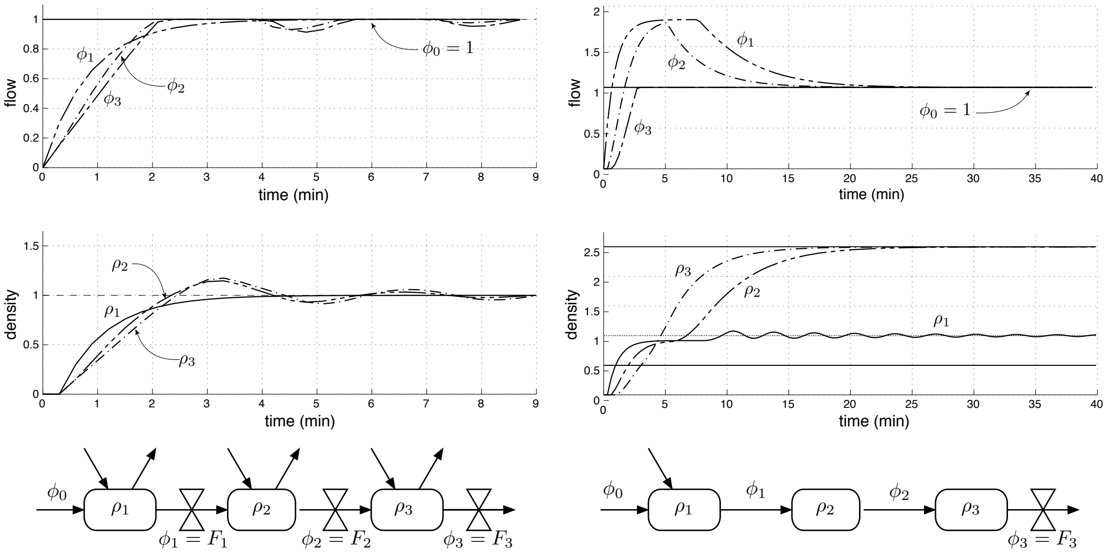
Recall now that according to Lemma 2.3.3, the congestion pattern within a segment is ordered. In particular, all cells $b_{j-1} < k < q_j$ operate in free-flow and all cells $q_j < k \leq b_j$ are congested. Satisfaction of conditions (2.79d) immediately follows.

We can also conclude from the previous analysis that for every bottleneck b_j, it holds that $\mu_{b_j} \leq 0$ (since cell b_j is located downstream of cell q_j) and also $\mu_{b_j+1} \geq 0$ (since cell $b_j + 1$ is located upstream of cell $q_j + 1$). Therefore $\lambda_{b_j} := c_{b_j} - \frac{\mu_{b_j}}{\beta_{b_j}} + \mu_{b_j+1} \geq 0$ and conditions (2.79e) follow. □

It is interesting to note that the result does not depend on the choice of the weights for the individual mainline flows c (as long as all are non-negative), so all such objectives will be jointly maximized by an equilibrium satisfying the optimality condition (2.77). This is a consequence of the assumption of constant split rations β_k, under which it is impossible to trade off the mainline flow at some part of the freeway with the mainline flow at another place.

Application

To demonstrate the practical relevance of the previously derived optimality conditions, we use them to analyze the well-known distributed ramp-metering strategy ALINEA Papageorgiou et al. (1991). ALINEA is in essence an integral controller that aims at stabilizing the local densities at the critical



Example 1.

Example 2.

Figure 2.42: Simulation results using the CTM. Note that the flows are normalized such that a flow of "1" means that the optimal steady-state flow (computed separately) is achieved. Similarly, the densities are normalized by the respective critical densities. Note the different timescales.

density. It is easy to verify that every closed-loop equilibrium of a freeway controlled by local ALINEA controllers satisfies the optimality conditions (2.77).

We implement the standard ALINEA anti-windup integral controller

$$u_k(t) := \tilde{u}_k(t-1) + K_I \cdot (\rho_k^c - \rho_k(t))$$

in which $\tilde{u}_k(t-1)$ is the saturated inflow $\tilde{u}_k(t) := \max(\underline{u}_k, \min(u_k(t), \bar{u}_k))$ and $K_I := 70/\rho_k^c$ is the integral gain chosen as recommended in Papageorgiou et al. (1991). Closed-loop simulation results of this controller operating on simple freeways represented by the CTM are depicted in Figure 2.42. In both examples, the closed-loop system converges to the optimal closed-loop equilibrium, as predicted. In the first example, the controller converges to the unique uncongested equilibrium. In fact, all densities converge to the respective critical densities. In the second example, however, the controlled onramp and the bottleneck are two sections apart. We observe convergence to a partly congested equilibrium, also the convergence is slow and heavy oscillations occur. These effects have already been described Wang et al. (2014), nevertheless, the optimal flows are achieved in the limit as long as we use the CTM for simulations.

Conclusions

This work analyzes flow-maximizing equilibria of the CTM. We find simple optimality conditions that are tailored to distributed control approaches, since only constraints between local ramp metering rates and the densities in the adjacent cells are imposed. In particular, the optimality conditions are satisfied for all closed-loop equilibria of a freeway controlled by independent local ALINEA controllers, thus proving that all such equilibria are globally optimal (w.r.t. flow maximization).

It is important to keep in mind that this result holds only under the assumptions made in the CTM. In particular, the equilibrium may contain congested bottlenecks, which lead in practice to an empirically observed reduction of the bottleneck capacity. Such effects are not described by the simplistic CTM. In this sense, the simulations described in the previous section are not meant to be a realistic assessment of the real-world traffic behavior, but rather a way to demonstrate the properties of the idealized CTM. Note however, that the equilibrium densities are not unique and techniques described in Gomes et al. (2008) or Wang et al. (2014) can be employed to steer the system to the preferred equilibrium (usually the unique, least congested equilibrium) in order to avoid the aforementioned capacity drop.

2.3.2 Model-Free Coordination Strategy

C. Ramesh, M. Schmitt, J. Lygeros

We consider a problem where multiple agents must learn an action profile that maximises the sum of their utilities in a distributed manner. The agents are assumed to have no knowledge of either the utility functions or the actions and payoffs of other agents. These assumptions arise when modelling the interactions in a complex system and communicating between various components of the system are both difficult. In Marden et al. (2014), a distributed algorithm was proposed, which learnt Pareto-efficient solutions in this problem setting. However, the approach assumes that all agents can choose their actions, which precludes disturbances. In this paper, we show that a modified version of this distributed learning algorithm can learn Pareto-efficient solutions, even in the presence of disturbances from a finite set. We apply our approach to the problem of ramp coordination in traffic control for different demand profiles.

Introduction

In complex systems, modelling the interactions between various components and their relationship to the system performance is not an easy task. This poses a challenge while designing controllers for such systems, as most design methods require a model of the system. Even when considerable effort has been expended in identifying suitable models for such systems, utilising these models to design online controllers is not always easy. This is because collecting measurements of a complex system, computing control signals using complex algorithms and applying these controls to actuators across the system is communication intensive and computationally demanding. The resulting delays are not well suited to the control of real-time complex systems.

An example is the real-time control of freeway traffic, where often traffic models are highly nonlinear and methods to design controllers using these models do not scale well Allsop (2008). Furthermore, to use these models, the traffic flow from every segment of the freeway must be measured and collected, and the control signals must be delivered to the ramps on the freeway. To reduce the communication and computation burden, distributed controllers that act on mostly local information are required.

One approach is to use a distributed randomised algorithm to explore the policy space and learn the optimal actions. Recently, a distributed learning algorithm has been proposed in Marden et al. (2014) where agents learn action profiles that maximise the system welfare. This algorithm is payoff-based, and the agents require no prior knowledge of either the utility functions or the actions and payoffs of other agents. An implicit assumption in this approach is that every agent that influences the utility can choose its actions. In reality, there might always be disturbances which cannot be chosen in a desired manner. In this paper, we extend this approach to include the effects of disturbances.

Our main contribution is a modification to the algorithm in Marden et al. (2014) to deal with disturbances. We show that agents learn Pareto-efficient solutions in a distributed manner using our algorithm, even in the presence of disturbances from a finite set. We verify the theoretical results on a small example. In this case, all assumptions can be verified and strong convergence guarantees can be given. To demonstrate versatility of the approach, we also apply the results to a realistic coordination problem motivated by freeway traffic control. We use our newly developed algorithm to learn a high-level coordination strategy for a ramp metering problem with promising results, using simulation parameters and traffic demand data from a real-world use case.

The learning rule used in this paper is related to the trial and error learning procedure from Young (2009) and its cognates Pradelski and Young (2012); Marden et al. (2014). These papers proposed algorithms that learnt Nash equilibria Young (2009), Pareto efficient equilibria Pradelski and Young (2012) and Pareto-efficient action profiles Marden et al. (2014), respectively. Convergence guarantees for the latter were presented in Menon and Baras (2013a). Restrictions on the payoff structure, which are required for the result in Marden et al. (2014) to hold, were eliminated through the use of explicit communication in Menon and Baras (2013b). We also draw on the analysis of deliberate experimentation using the theory of regular perturbed Markov processes from Young (1993).

This paper is organised as follows: We describe the algorithm in Section 3.1.1, and present known results in Section 2.3.2. Our main result is presented in Section 2.3.2 and illustrated on a few examples in Section 2.3.2. The conclusion is in Section 2.3.2.

Problem Formulation

We consider a set of agents $N := \{1, \dots, n\}$, each with a finite action set $\mathcal{A}_i$ for $i \in N$. The disturbance is modelled as an independent and identically distributed (i.i.d.) process w_k, which takes values from a finite set $\mathbb{W}$ according to a probability distribution $\mathbf{P}_w$ that is fully supported on $\mathbb{W}$. Given an action profile $a \in \mathcal{A}$, where $\mathcal{A} := \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$, and a disturbance $w \in \mathbb{W}$, the payoff for each agent is $u_i(a, w)$. The payoffs are generated by utility functions $\mathcal{U}_i : \mathcal{A} \times \mathbb{W} \to [0, 1)$ whose functional forms are unknown to the agents. The welfare of the network of agents is $\mathcal{W}(a, w) = \sum_{i \in N} \mathcal{U}_i(a, w)$.

The agents play a repeated game; in the k^{th} iteration, each agent chooses its action $a_{i,k}$ with probability $p_{i,k} \in \Delta(\mathcal{A}_i)$, where $\Delta(\mathcal{A}_i)$ is the simplex of distributions over $\mathcal{A}_i$. The strategy $p_{i,k}$ is completely uncoupled or pay-off based, i.e., $p_{i,k} = \psi_i(\{a_{i,\tau}, u_{i,\tau}(a_\tau, w_\tau)\}_{\tau=0}^{k-1})$. In other words, an agent does not know the actions or payoffs of any other agent in the network.

Each agent maintains an internal state $z_{i,k} := [\bar{a}_{i,k}, \bar{u}_{i,k}, m_{i,k}]$ in the k^{th} iteration, where $\bar{a}_{i,k} \in \mathcal{A}_i$ is the baseline action, $\bar{u}_{i,k}$ is the corresponding baseline utility that lies in the range of $\mathcal{U}_i$ and $m_{i,k} \in \{\mathcal{C}, \mathcal{D}\}$ is the mood variable that connotes whether the agent is content or discontent. The state $z_k := \{z_{1,k}, \dots, z_{n,k}\}$ lies in the finite state space $\mathbb{Z}$.

The algorithm is initialised with all agents setting their moods to discontent, i.e., $m_{i,0} = \mathcal{D}$ for $i \in N$. An experimentation rate $0 < \varepsilon < 1$ is fixed and a constant $c > n$ is selected. Then, each agent selects an action $a_{i,k}$ according to its mood and the corresponding probabilistic rule:

$$\begin{aligned} m_{i,k} = \mathcal{C} : \; p_i(a_{i,k}) &= \begin{cases} \frac{\varepsilon^c}{|\mathcal{A}_i|-1} & a_{i,k} \neq \bar{a}_{i,k} \\ 1 - \varepsilon^c & a_{i,k} = \bar{a}_{i,k} \end{cases} \\ m_{i,k} = \mathcal{D} : \; p_i(a_{i,k}) &= \frac{1}{|\mathcal{A}_i|} \quad \forall \quad a_{i,k} \in \mathcal{A}_i \end{aligned} \tag{2.80}$$

The agents choose their strategies based on their moods. A content agent selects its baseline action with high probability and experiments by choosing other actions with low probability. A discontent agent selects an action with uniform probability.

Each agent plays the action it has selected and receives a payoff $u_{i,k}(a_k, w_k)$, which it uses to update its state as

$$z_{i,k+1} = \begin{cases} z_{i,k} & \begin{array}{l} m_{i,k} = \mathcal{C}, a_{i,k} = \bar{a}_{i,k}, \\ |u_{i,k} - \bar{u}_{i,k}| \leq \rho \end{array} \\ \left.\begin{array}{l} z_{\mathcal{C}} \text{ w.p. } p_{\mathcal{C}} \\ z_{\mathcal{D}} \text{ w.p. } 1 - p_{\mathcal{C}} \end{array}\right\} & \text{otherwise} \end{cases} \tag{2.81}$$

where $z_{\mathcal{C}} = [a_{i,k}, u_{i,k}, \mathcal{C}]$, $z_{\mathcal{D}} = [a_{i,k}, u_{i,k}, \mathcal{D}]$, $p_{\mathcal{C}} = \varepsilon^{1-u_{i,k}}$ and ρ is the maximum deviation in the payoffs due to the disturbance process w, as defined in (2.82). The state update also depends on the mood of the agent. A content agent that chose to play its baseline action and received a payoff within the interval $u_{i,k} \in [\bar{u}_{i,k} - \rho, \bar{u}_{i,k} + \rho]$ retains its state. In all other cases, the state is updated to the played action and received payoff, and the mood is set to content or discontent with high probabilities for high or low payoffs, respectively. Thus, a content agent must receive a payoff outside an interval $\pm\rho$ of its baseline payoff to reevaluate its mood or change its state. This interval rule renders an agent insensitive to small changes in the payoff.

The variable ρ is defined as the maximum deviation in the payoffs received by any agent $i \in N$ for every action profile $a \in \mathcal{A}$ and every pair of disturbances $w_1, w_2 \in \mathbb{W}$, i.e.,

$$\begin{aligned} \rho := \arg\min_{r\in\mathbb{R}} \{ |u_i(a, w_1) - u_i(a, w_2)| \leq r \} \\ \forall\, i \in N,\ \forall\, a \in \mathcal{A} \ \text{ and } \ \forall\, w_1, w_2 \in \mathbb{W}\ . \end{aligned} \tag{2.82}$$

We are interested in identifying the set of states the above algorithm converges to. A necessary condition for this algorithm to function as desired is the interdependence property, stated below.

Definition 2.3.1. *An n-person game is* interdependent *if for every action profile $a \in \mathcal{A}$, every disturbance $w \in \mathbb{W}$ and every proper subset of agents $J \subset N$, there exists an agent $i \subset N \setminus J$, a choice of actions $a'_J \in \prod_{j\in J} \mathcal{A}_j$ and a disturbance $w' \in \mathbb{W}$, such that*

$$\big|\mathcal{U}_i(a'_J, a_{-J}, w') - \mathcal{U}_i(a_J, a_{-J}, w)\big| > \rho \tag{2.83}$$

This property ensures that the set of agents cannot be divided into two mutually non-interacting groups, and that a discontent agent always has recourse to actions that influence the utilities of other content agents despite the algorithm's insensitivity to the interval $[\bar{u}_i - \rho, \bar{u}_i + \rho]$ in (2.81).

Remark 2.3.2 (A remark on the state space $\mathbb{Z}$:)**.** *The state z_k is an aggregation of the states $z_{i,k} := [\bar{a}_{i,k}, \bar{u}_{i,k}, m_{i,k}]$ of each of the agents. Thus, one would expect the cardinality of the state space to be $|\mathbb{Z}| = 2^N |\mathcal{A}||\mathbb{W}|$, because the payoffs obtained are completely determined by the choice of the actions and the disturbance. However, the interval rule in* (2.81) *results in more states becoming reachable and $|\mathbb{Z}| \leq 2^N |\mathcal{A}|^2 |\mathbb{W}|$. The exact number of states depends on the payoffs. For the proofs presented in this paper, we define the state space in terms of the states reachable from the initial point of our algorithm, i.e., $\mathbb{Z} := \{z : \exists \tau > 0 \text{ s.t. } \mathbf{P}(z_\tau = z|z_0) > 0\}$, where z_0 is any state where all agents are discontent.*

Preliminaries

We briefly outline Young's result on regular perturbed Markov processes Young (1993). Consider the Markov processes on a state space $\mathbb{X}$ with transition matrices $\mathbf{P}^0$ and $\mathbf{P}^\epsilon$, where a finite-valued $\epsilon > 0$ measures the noise level. The Markov chain induced by $\mathbf{P}^0$ describes some basic evolutionary process such as best response dynamics, while the chain induced by $\mathbf{P}^\epsilon$ represents the perturbed process obtained by introducing mistakes or experiments. This notion is formalised as follows.

Definition 2.3.2. *A family of Markov processes $\mathbf{P}^\epsilon$ is called a regular perturbation of a Markov chain with transition matrix $\mathbf{P}^0$ if it satisfies the following conditions:*

1. *$\mathbf{P}^\epsilon$ is aperiodic and irreducible for all finite $\epsilon > 0$.*

2. *$\lim_{\epsilon\to 0} \mathbf{P}^\epsilon_{xy} = \mathbf{P}^0_{xy}$, $\forall x, y \in \mathbb{X}$.*

3. *If $\mathbf{P}^\epsilon_{xy} > 0$ for some ϵ, then $\exists\, r(x, y) \geq 0$, called the* resistance *of the transition $x \to y$, such that*

$$0 < \lim_{\epsilon\to 0} \epsilon^{-r(x,y)}\, \mathbf{P}^\epsilon_{xy} < \infty \tag{2.84}$$

Property i ensures that there is a unique stationary distribution for all finite $\epsilon > 0$. Property ii ensures that the perturbed process converges to the unperturbed process in the limit as $\epsilon \to 0$. Property iii states that a transition $x \to y$ is either impossible under $\mathbf{P}^\epsilon$ or it occurs with a probability $\mathbf{P}^\epsilon_{xy}$ of order $\epsilon^{r(x,y)}$ for some unique, real $r(x,y)$ in the limit as $\epsilon \to 0$. Note that $r(x,y) = 0$ if and only if $\mathbf{P}^0_{xy} > 0$. Thus, the transitions of resistance zero are the same as the transitions that are feasible under $\mathbf{P}^0$.

Definition 2.3.3. *A state $x \in \mathbb{X}$ is said to be stochastically stable if $\mu^0_x > 0$, where μ^0 is a stationary distribution of* $\mathbf{P}^0$.

We are interested in characterizing the limiting distribution μ^0 of $\mathbf{P}^0$ through its support, or the set of stochastically stable states. To do this, we define two directed graphs. The first graph $G := (\mathbb{X}, \mathbb{E}_G)$ has as vertex set the set of states $\mathbb{X}$, and as directed edge set $\mathbb{E}_G := \{x \to y \mid \mathbf{P}^\epsilon_{xy} > 0,\ x, y \in \mathbb{X}\}$. Thus, a directed edge exists in G only if a single transition under $\mathbf{P}^\epsilon$ gets us from state x to y, for all values of $\epsilon \geq 0$. Finally, $r(x,y)$ in (2.84) defines the weight or resistance of this directed edge in G.

To define the second graph, we first enumerate the recurrence classes of $\mathbf{P}^0$ as $X_1, \ldots, X_L$. Then, we can define the resistance between two classes as the minimum resistance between any two states belonging to these classes, i.e.,

$$r_{\ell_1 \ell_2} := \min_{x \in X_{\ell_1}, y \in X_{\ell_2}} r(x,y), \text{ for } \ell_1, \ell_2 \in \{1, \ldots, L\} . \tag{2.85}$$

Note that there is at least one path from every class to every other because $\mathbf{P}^\epsilon$ is irreducible. We now define the second graph as $\mathcal{G} := (\{1, \ldots, L\}, \mathbb{E}_{\mathcal{G}})$. This graph has as vertex set the set of indices of the recurrence classes of $\mathbf{P}^0$, and as edge set the set of directed edges between members of the recurrence classes. Also, $r_{\ell_1 \ell_2}$ defines the resistance or weight of this directed edge.

Definition 2.3.4. *Let an ℓ-tree in $\mathcal{G}$ be a spanning sub-tree of $\mathcal{G}$, such that for every vertex $\ell' \neq \ell$, there exists exactly one directed path from ℓ' to ℓ. Then, the stochastic potential γ_ℓ of the recurrence class X_ℓ is defined as*

$$\gamma_\ell := \min_{T \in \mathcal{T}_\ell} \sum_{(a,b) \in T} r_{ab} \tag{2.86}$$

where $\mathcal{T}_\ell$ is the set of all ℓ-trees in $\mathcal{G}$.

We can now state Young's result for perturbed Markov processes Young (1993).

Theorem 2.3.2 (Theorem 4 from Young (1993))**.** *Let $\mathbf{P}^0$ be a time-homogenous Markov process on the finite state space $\mathbb{X}$ with recurrence classes $X_1, \ldots, X_L$. Let $\mathbf{P}^\epsilon$ be a regular perturbation of $\mathbf{P}^0$, and let μ^ϵ be its unique stationary distribution for every small positive ϵ. Then,*

1. *as $\epsilon \to 0$, μ^ϵ converges to a stationary distribution μ^0 of $\mathbf{P}^0$,*
2. *the recurrence class $X_{\ell *}$, with stochastic potential $\gamma_{\ell *} := \min_{\ell \in \{1, \ldots, L\}} \gamma_\ell$, contains the stochastically stable states $\{x \in \mathbb{X} : \mu^0_x > 0\}$.*

Learning Pareto-efficient solutions

We begin by establishing that Young's result applies to our system, resulting in a distributed algorithm for Pareto-efficient learning in the presence of disturbances. To prove this result, we enumerate the recurrence classes of $\mathbf{P}^0$ and the resistances between the classes. We use these values to identify the structure of the tree with minimum stochastic potential.

Main Result For $\varepsilon = 0$, the transition matrix $\mathbf{P}^0$ corresponds to an unperturbed Markov process, and we begin by showing that $\mathbf{P}^\varepsilon$ is a regular perturbation on $\mathbf{P}^0$.

Lemma 2.3.4. *The Markov process with transition matrix $\mathbf{P}^\varepsilon$ is a regular perturbation on $\mathbf{P}^0$.*

The proof is presented in Ramesh et al. (2015). Next, we use Young's result from Theorem 2.3.2 to obtain the distributed learning outcome stated below.

Theorem 2.3.3. *Let G be an interdependent n-person game on a finite joint action space $\mathcal{A}$, subject to i.i.d. disturbances from a finite set $\mathbb{W}$. Under the dynamics defined in (2.80)–(2.81), a state $z = [\bar{a}, \bar{u}, m] \in \mathbb{Z}$ is stochastically stable if and only if the following conditions are satisfied:*

1. *The action profile $\bar{a}$ maximises the network welfare, i.e.,*

$$(\bar{a}, \bar{w}) \in \underset{a\in\mathcal{A}, w\in\mathbb{W}}{\arg\max}\, \mathcal{W} = \underset{a\in\mathcal{A}, w\in\mathbb{W}}{\arg\max} \sum_{i\in N} \mathcal{U}_i(a, w) \tag{2.87}$$

2. *The benchmark actions and payoffs are aligned for the maximising disturbance, i.e., $\bar{u}_i = \mathcal{U}_i(\bar{a}, \bar{w})$.*

3. *The mood of each agent is content.*

We present the proof for this theorem in the next section.

Recurrence Classes The states $z \in \mathbb{Z}$ can be classified into three categories: states where all agents are content or discontent and states where some agents are content and others discontent. By inspecting the algorithm in (2.80)–(2.81), it is easy to see that as $\varepsilon \to 0$ the former states can be recurrent, but not the latter. We formalise this notion below, by defining the recurrence classes $\mathbb{D}$ and $\mathbb{C}^m$ for $0 \le m < n$ and showing that there are no other recurrence classes.

Discontent Class $\mathbb{D}$: The states in this recurrence class correspond to those where all agents are discontent.

$$\mathbb{D} := \left\{ z \in \mathbb{Z} \quad \middle| \quad m_i(z) = \mathcal{D},\ \forall i \in N \right\} \tag{2.88}$$

Note that the payoffs and action profiles are aligned, i.e., $\bar{u}_i(z) = \mathcal{U}_i(\bar{a}(z), w)$, $\forall z \in \mathbb{D}$, $\forall i \in N$ and for some $w \in \mathbb{W}$. Also, corresponding to each action profile and disturbance pair $(a, w) \in \mathcal{A} \times \mathbb{W}$, there is a discontent state in this recurrence class.

States containing only content agents can be categorised further into n classes, $\mathbb{C}^m$ for $0 \le m < n$, as follows.

0th-Content Class $\mathbb{C}^0$: This recurrence class contains singleton states where all agents are content, and where the payoffs of all agents are aligned with the action profile for some value of the disturbance $w \in \mathbb{W}$, while satisfying the interval rule in (2.81) for all other values of the disturbance. Let B_i denote the set of states that satisfy these conditions on the payoffs of the i^{th} agent:

$$\begin{aligned} B_i := \Big\{ z \in \mathbb{Z} \big| \bar{u}_i(z) = \mathcal{U}_i(\bar{a}(z), w), \text{for some } w \in \mathbb{W}, \\ |\bar{u}_i(z) - \mathcal{U}_i(\bar{a}(z), \tilde{w})| \le \rho,\ \ \forall\, \tilde{w} \in \mathbb{W} \Big\}. \end{aligned} \tag{2.89}$$

Then, the recurrence class $\mathbb{C}^0$ is defined as

$$\mathbb{C}^0 := \left\{ z \in \mathbb{Z} \quad \middle| \quad m_i(z) = \mathcal{C}, z \in B_i, \forall i \in N \right\}. \tag{2.90}$$

From the definition of ρ in (2.82), we know that corresponding to each action profile and disturbance pair $(a, w) \in \mathcal{A} \times \mathbb{W}$, there is a state in this recurrence class with payoffs satisfying (2.89).

There might also be states where the payoffs of all agents are not aligned with the action profile for any single value of the disturbance $w \in \mathbb{W}$. Some of these states can be recurrent, and belong to the classes defined below.

1^{st}-Content Class $\mathbb{C}^1$: Suppose that a proper subset of agents $J_1 \subset N$ from a state $z' \in \mathbb{C}^0$ experiment with different actions despite being content, and become content with their new utilities. If the rest of the agents $j_0 \in J_0 = N \setminus J_1$ do not notice this change, because their new utilities lie within the interval $[\bar{u}_{j_0}(z') - \rho, \bar{u}_{j_0}(z') + \rho]$ for all values of the disturbance, then the agents find themselves in a state z in a recurrence class $\mathbb{C}^1$.

$$
\begin{aligned}
\mathbb{C}^1 := \Big\{ z \in \mathbb{Z} \;\Big|\; & m_i(z) = \mathcal{C}, \forall i \in N, \\
& \exists (J_0, J_1) \text{ s.t. } J_0 \uplus J_1 = N, \\
& z \in B_{j_1}, \quad \forall\, j_1 \in J_1, \\
& \exists z' \in \mathbb{C}^0 \text{ s.t. } z_{j_0} = z'_{j_0}, \\
& |\bar{u}_{j_0}(z') - \mathcal{U}_{j_0}(\bar{a}(z), \tilde{w})| \leq \rho, \\
& \qquad \forall\, \tilde{w} \in \mathbb{W}, \forall\, j_0 \in J_0 \Big\},
\end{aligned}
\tag{2.91}
$$

where the symbol $\uplus$ denotes a disjoint union of the subsets.

A subset of agents from a state in $\mathbb{C}^1$ could experiment and find themselves in a state in a recurrence class $\mathbb{C}^2$. In general, states in the recurrence class $\mathbb{C}^m$ can be reached from a state in $\mathbb{C}^{m-1}$, following a similar procedure. The recurrence class $\mathbb{C}^m$ is defined below.

m^{th}-Content Class $\mathbb{C}^m$: These recurrence classes contain singleton states where all agents are content, and where the agents can be divided into $m+1$ mutually disjoint subsets $J_0, \ldots, J_m$, such that the utilities of the agents within each subset are aligned with an action profile for some value of the disturbance.

$$
\begin{aligned}
\mathbb{C}^m := \Big\{ z \in \mathbb{Z} \;\Big|\; & m_i(z) = \mathcal{C}, \forall i \in N, \\
& \exists (J_0, \ldots, J_m) \text{ s.t. } \uplus_{l=0}^{m} J_l = N, \\
& z \in B_{j_m}, \quad \forall\, j_m \in J_m, \\
& \exists z' \in \mathbb{C}^{m-1} \text{ s.t. } z_{j_\ell} = z'_{j_\ell}, \\
& |\bar{u}_{j_\ell}(z') - \mathcal{U}_{j_\ell}(\bar{a}(z), \tilde{w})| \leq \rho, \\
& \qquad \forall\, \tilde{w} \in \mathbb{W}, \forall\, j_\ell \in N \setminus J_m \Big\}.
\end{aligned}
\tag{2.92}
$$

There can be at most n disjoint subsets from a set of n agents, and hence $m < n$. Clearly, there might be many states in $\mathbb{Z}$, where the baseline payoffs and actions satisfy some, but not all, of the above conditions for classes $\mathbb{C}^m$, $1 \leq m < n$. These states are not recurrent, as we show below.

Lemma 2.3.5. *The recurrence classes corresponding to the n-person interdependent game described by (2.80)–(2.81) are $\mathbb{D}$, and the singletons in $\mathbb{C}^0$ and $\mathbb{C}^m$, for $0 < m < n$, as defined in (2.88)–(2.92), respectively.*

The proof of this result is presented in Ramesh et al. (2015).

Resistances and Trees Transitions can occur between all three recurrence class types, namely $\mathbb{D} \to \mathbb{C}^0$ and vice versa, $\mathbb{D} \to \mathbb{C}^m$ and vice versa, and $\mathbb{C}^l \to \mathbb{C}^m$ for $0 \leq l, m < n$ and $l \neq m$. In addition,

Table 2.4: Resistances Between Recurrence Classes

No.	Path	Resistance Relationship
1	$\mathbb{D} \to \mathbb{C}^0$	$r_{dz^0} = \sum_{i\in N} 1 - \bar{u}_i(z^0)$
2	$\mathbb{D} \to \mathbb{C}^m$	$r_{dz^m} =$ $\min_{z^0 \in \mathbb{C}^0} r_{dz^0} + r_{z^0 z^m}$
3	$\mathbb{C}^0 \to \mathbb{D}$	$r_{z^0 d} = c$
4	$\mathbb{C}^m \to \mathbb{D}$	$r_{z^m d} = c$
5	$\mathbb{C}^0 \to \mathbb{C}^0$	$c \le r_{z_1^0 z_2^0} \le 2c$
6	$\mathbb{C}^m \to \mathbb{C}^m$	$c \le r_{z_1^m z_2^m} \le \bar{r}_m$
7	$\mathbb{C}^l \to \mathbb{C}^m$, $0 \le l, m < n, m \ne l$	$\lvert m - l \rvert c \le r_{z_1^m z_2^m} \le \bar{r}_m$

the singleton states in $\mathbb{C}^0$ and $\mathbb{C}^m$ can transition to other singleton states within the same classes. All these transitions are enumerated along with the corresponding resistances in Table 2.4. In this table, we use $d \in \mathbb{D}$, $z^0_\cdot \in \mathbb{C}^0$ and $z^m_\cdot \in \mathbb{C}^m$ to denote states in the respective recurrence classes. Some of the entries contain the term $\bar{r}_m$, which is given by

$$\bar{r}_m = mc^2 + \frac{4 + m - m^2}{2}c - \frac{m(m+1)}{2}, \; 0 < m < n \; . \tag{2.93}$$

The calculations for the entries in Table 2.4 are presented in Ramesh et al. (2015). We can now compute the stochastic potential of a state in $\mathbb{C}^0$ and show that a minimum potential tree is rooted at a singleton in $\mathbb{C}^0$.

Lemma 2.3.6. *The stochastic potential of a state $z^0 \in \mathbb{C}^0$ is*

$$\gamma(z^0) = c\left(\sum_{m=0}^{n-1} |\mathbb{C}^m| - 1\right) + \sum_{i\in N}\left(1 - \bar{u}_i(z^0)\right) \; . \tag{2.94}$$

Lemma 2.3.7. *The states in the recurrence class $\mathbb{D}$ and the singletons $\mathbb{C}^m$, for $0 < m < n$, are not stochastically stable.*

The proofs for both Lemmas are presented in Ramesh et al. (2015).

Proof of the Main Result We now present the proof of Theorem 2.3.3.

Proof. The stochastically stable states are contained in the recurrence class of $\mathbf{P}^0$ with minimum stochastic potential (from Theorem 2.3.2). From Lemma 2.3.7, we also know that the recurrence class with minimum stochastic potential is rooted at a singleton in $\mathbb{C}^0$.

Lemma 2.3.6 gives us the minimum stochastic potential as

$$\gamma(z^{0,*}) = \min_{z^0 \in \mathbb{C}^0} c(\sum_{m=0}^{n-1} |\mathbb{C}^m| - 1) + \sum_{i\in N}(1 - \bar{u}_i(z^0))$$

Thus, the action profile corresponding to the state $z^{0,*}$ must satisfy $\bar{a}(z^{0,*}) \in \arg\max_{a\in\mathcal{A}, w\in\mathbb{W}} \sum_{i\in N} \mathcal{U}_i(a, w)$.

From the definition of the recurrence class $\mathbb{C}^0$ in (2.90), we obtain statements ii and iii of the theorem. □

Table 2.5: Payoffs in Example 2.3.1

$\{a_1, a_2, w\}$	u_1	u_2	$\{a_1, a_2, w\}$	u_1	u_2
$\{0, 0, 0\}$	0.30	0.40	$\{1, 1, 0\}$	0.80	0.90
$\{0, 0, 1\}$	0.40	0.30	$\{1, 1, 1\}$	0.90	0.80
$\{0, 1, 0\}$	0.20	0.10	$\{2, 0, 0\}$	0.65	0.55
$\{0, 1, 1\}$	0.10	0.20	$\{2, 0, 1\}$	0.55	0.65
$\{1, 0, 0\}$	0.60	0.50	$\{2, 1, 0\}$	0.75	0.85
$\{1, 0, 1\}$	0.50	0.60	$\{2, 1, 1\}$	0.85	0.75

Examples

We present a simple example of a two-agent interdependent game to illustrate the results of Theorem 2.3.3, and then apply this method to the ramp coordination problem.

Example 2.3.1. *Consider a simple game G_2 with $n = 2$ agents. The action sets, disturbance set and payoffs are given in Table 2.5. The disturbance process is uniformly distributed on $\{0, 1\}$. It is easy to verify that $\rho = 0.1$ (from (2.82)), and that the interdependence property (from Definition 2.3.1) is satisfied, for G_2.*

We simulated 10^6 iterations of the algorithm (2.80)–(2.81) in Matlab, with a time-varying ε-sequence, and $c = 2$. The experimentation rate was modified by setting $\varepsilon_{k+1} = 0.99995\varepsilon_k$, with an initial value of $\varepsilon_1 = 0.1$. In a typical sample run of our simulation (presented in Ramesh et al. (2015)), the state corresponding to the action profile $\{1, 1\}$ with payoff 1.7 was visited 95.32% of the time, thus validating the results of Theorem 2.3.3. The average welfare over all the iterations was 1.6937.

The average welfare obtained from playing the optimal action profile(s) will, in general, be different from $\mathcal{W}^*$, the maximum welfare in (2.87). This is because the optimal action profile maximises the welfare for the most favourable value of the disturbance as per Theorem 2.3.3. When averaged over all possible values of the disturbance, the welfare will lie in the interval $[\mathcal{W}^* - n\rho, \mathcal{W}^* + n\rho]$, depending on $\mathbf{P}_w$. For the above example, the average welfare equals $\mathcal{W}^*$, which may not always be the case as we see in the next example.

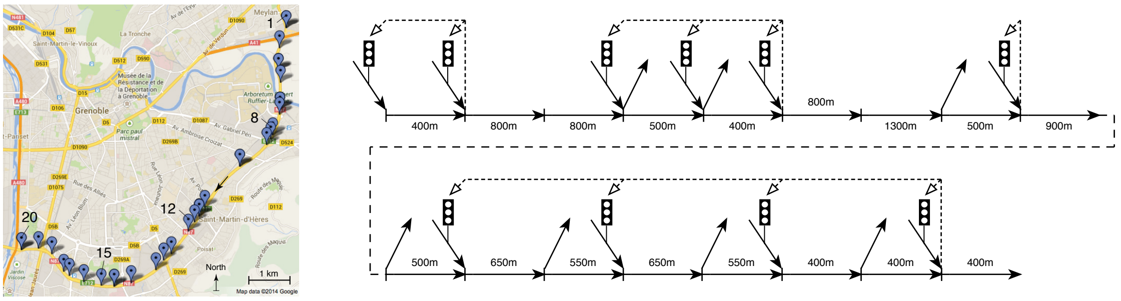


Figure 2.43: Map of the Grenoble South Link as depicted in de Wit et al. (2015) and the corresponding freeway topology. Also shown is a particular coordination pattern between onramps, corresponding to the action profile $[\mathrm{COR}, \mathrm{LOC}, \mathrm{COR}, \mathrm{COR}, \mathrm{LOC}, \mathrm{LOC}, \mathrm{COR}, \mathrm{COR}, \mathrm{COR}, \mathrm{LOC}]^\top$.

Example 2.3.2. *In freeway traffic control, one seeks to estimate the occupancy of a freeway, usually via loop detectors Gibbens and Saatci (2008), and subsequently adjust speed limits Hegyi et al. (2005) or traffic lights on the onramps Papageorgiou and Kotsialos (2000), a technique known as ramp metering,* to improve traffic flow. However, popular freeway traffic models such as the cell transmission model Daganzo (1994a, 1995) or the Metanet model Messner and Papageorgiou (1990) are highly nonlinear and methods to design controllers for traffic networks often do not scale well. To reduce the communication and computation burden, distributed controllers that act on mostly local information are required. Model-based approaches for decomposition exist Dunbar and Murray (2006); de Oliveira and Camponogara (2010), but in fact, local feedback Papageorgiou et al. (1991) and the combination of local feedback and heuristic, high-level coordination Papamichail et al. (2010b) are among the most popular and practically successful strategies. To demonstrate its versatility, we will evaluate the efficacy of our method in learning a ramp coordination pattern, for given local controllers.

Consider a freeway with a number of onramps. The idea of ramp metering is to control the traffic inflow from the ramps via traffic lights so as to avoid congestion on the mainline. Both theoretical Gomes and Horowitz (2006b) and practical Papageorgiou et al. (2003) studies have demonstrated that this approach can potentially avoid traffic breakdown in congestion and reduce the sum of travel times of all drivers (TTS, Total Time Spent). An effective metering strategy is to control the inflow such that the local traffic density does not exceed the threshold to congestion, the so-called critical density Papageorgiou et al. (1991). However, there are limits to this strategy. Multiple ramps, each controlling the local traffic densities, are coupled by the mainline flow as traffic travels downstream and congestion queues can spill back upstream. If no control action of a single ramp is sufficient to prevent congestion of an adjacent bottleneck, then coordination between ramps may hold the answer Papamichail et al. (2010b).

In this example, we aim to learn a coordination pattern, while the low-level metering policy remains fixed. We consider ten ramps on a freeway located in Grenoble, as presented in de Wit et al. (2015) and depicted in Figure 2.43. We allow for ramps either to control only the local traffic density (LOC) or to coordinate with downstream ramps (COR) and control the ramp occupancy, i.e. the queue length divided by the ramp length, according to the occupancy of the next downstream ramp. Therefore, the action set for every agent, i.e., every ramp i is $\mathcal{A}_i = \{LOC, COR\}$. The utility is computed by simulations of the freeway using the modified cell-transmission model as described in Karafyllis et al. (2014), which uses a non-monotonic demand function to model the capacity drop empirically observed in a congested freeway. The local utility for agent i is computed as the sum of the total travel time of all cars in the adjacent section of the freeway and the total waiting time in the onramp queue, which is then mapped to the interval $[0, 1]$ via a linear transformation. The utilities do not only depend on the action profile but also on the traffic demand, which acts as an external disturbance. We consider real traffic demands during peak hours of the weekdays May 11th - May 15th, and hence, the disturbance set is $\mathbb{W} = \{Mon, Tue, Wed, Thu, Fri\}$.

We do not try to identify ρ as per (2.82) or verify the interdependence property in this example. Instead, we simply choose ρ to be sufficiently large to ensure convergence of the algorithm within a reasonable number of iterations. We ensure that interdependence holds by complimenting the interaction graph with a communication graph, as suggested in Menon and Baras (2013b). Each agent broadcasts the mood it computes in (2.81). It then receives all the other agents' moods and performs the following update step to finalize its own mood, as per

$$m_{i,k+1} = \begin{cases} \mathcal{D} & \tilde{m}_{i,k+1} = \mathcal{D} \\ \mathcal{C} & \tilde{m}_{j,k+1} = \mathcal{C},\ \forall j \in N \\ \left.\begin{matrix} \mathcal{C} \text{ w.p. } \varepsilon^{\beta} \\ \mathcal{D} \text{ w.p. } 1 - \varepsilon^{\beta} \end{matrix}\right\} & \text{otherwise} \end{cases} \tag{2.95}$$

where $\tilde{m}_{i,k+1}$ is the mood of the i^{th} agent updated locally as per (2.81). The above update compliments the interaction between the agents by coupling the moods, and is controlled by the parameter β. If each *agent broadcasts its mood to all other agents, this update alone will suffice to ensure the interdependence property, irrespective of the utility functions. Thus, all the results in this paper, including Theorem 2.3.3, can be shown to hold for this modified algorithm, for ρ chosen as per (2.82). However, in a real-world setting it might be difficult to compute a suitable bound on ρ beforehand. Instead, we chose ρ empirically to facilitate quick convergence, sacrificing the guarantees that come with Theorem 2.3.3. The performance is then checked a posteriori.*

We simulated 1000 iterations of the algorithm (2.80)–(2.81), (2.95), in Matlab, with $\varepsilon = 0.0001$, $c = 10$, $\beta = 0.00005$ and $\rho = 0.6$. The algorithm explored 36 different action profiles before settling on the ramp coordination schedule $[COR, LOC, LOC, COR, COR, COR, LOC, LOC, COR, LOC]^\top$. The corresponding baseline utility was 9.6 and the algorithm spent 890 out of 1000 iterations in the above state. The average utility obtained over the entire simulation run was 8.72, in comparison to an average utility of 6.30 for the uncontrolled case. In terms of travel times, this corresponds to savings of 31% over the uncoordinated case. Note that we compute the savings just for the rush-hour period and therefore this value might exceed the savings typically reported for ramp metering field trials, which are usually computed for the entire day Papageorgiou and Kotsialos (2000).

Conclusions

We presented a distributed learning algorithm, based on the algorithm in Marden et al. (2014), that can be used to learn Pareto-efficient solutions in the presence of disturbances. Our algorithm learns efficient action profiles corresponding to the most favourable disturbance, and specifies a range for the average welfare. In general, the approach outlined in this paper is particularly well suited to problems where the disturbances can be modelled as a finite set of small perturbations from a nominal model. Our examples validated the main result in our paper, and also illustrated the potential of this randomised approach. In many applications, the average welfare is an important performance metric. In future work, we wish to explore randomized approaches that optimize the average welfare obtained.

2.3.3 Towards Optimal, Scalable Traffic Control

P. Grandinetti, F. Garin, C. Canudas de Wit

The problem of traffic congestion has always been a crucial aspect for the design of efficient infrastructures, but it is particularly in the second half of the last century that this phenomenon has become predominant, due to the quickly increasing traffic demand and the more frequent congestions. Congestions appear when too many vehicles try to use a common transportation route, which, due to physical reasons, has limited capacity. If it happens, they lead to queueing phenomena or, even worse, to a severe degradation of the available infrastructure's usage. Hence, congestions result in reduced safety, increased pollution and excessive delays. Economic implication of such events are recently widely discussed (He and Zhang, 2015; Sweet, 2015).

Traffic in urban scenarios is mainly regulated by the traffic lights installed at intersections of roads. Even though these devices were first conceived to guarantee avoidance of collision, with steadily increasing traffic demands it was soon realized that they may lead to more or less efficient network operation. Therefore, the idea to increase the efficiency of the infrastructure using smart traffic lights policies has constantly been employed in academic and industrial research.

Moreover, a crucial aspect of the today traffic management policies is the huge amount of data available to the decision maker (e.g., sensors installed at every urban intersection) and the heavy number of decision it has to take (e.g., one for each urban intersection). Modern traffic control policies, therefore, have to take into account such vital aspects. Overloading the decision maker can lead to the most negative consequences, like failure and crushes. As such, control strategies have to come with some key features, among which capability to handle big data and scalability to any network size. These goals are, in some way, in contrast with the optimality objective. In general scenarios, it is very hard to produce scalable algorithms that also ensure optimality of the result. The work presented in this deliverable addresses these topics, developing a decentralized decision making scheme, that has low computational prize and achieves good performances with respect to a global optimum that would practically be too expensive to calculate in real time. Also, the strategy we elaborated is based on the receding horizon philosophy; this choice is due to the fact that in traffic scenarios demands are continuously changing and, therefore, proactive decisions are needed. Our control scheme is able to accomplish such requirement, i.e., it may improve the quality of the system in the upcoming future taking into account external demands.

Some existing strategies, called fixed–time techniques (Robertson, 1969), have limitations due to their settings, which are based on historical, rather than real–time, data. A survey about the existing techniques can be found in Papageorgiou et al. (2003). More advanced schemes have been presented recently and they refer to different models for the network and for the chosen control actions, such as max pressure control (Varaiya, 2013) and cooperative green lights policies (Savla et al., 2014). Concerning the control action, we will instead contribute focusing on a more realistic representation of traffic lights, whose mathematical abstraction often produces problems difficult to solve, due to the binary (green-red) nature of these signals. Because of that, scientific works often consider optimization that takes into account either bandwith maximization via phase shift selection (Little, 1966; Gartner et al., 1991) or duty cycle design (Grandinetti et al., 2015a). The control of both parameters is a very challenging problem still open in transportation optimization environments. In this work, we indeed decided to deal with the most general and difficult problem: the control of traffic lights, in urban networks, parametrized by two degrees of freedom: such representation (i.e., their description with two variable time instants) includes the most generic scenario, being able to embody phase shift as well as duty cycle.

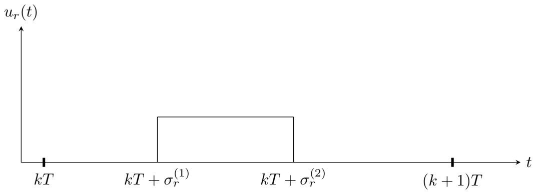


Figure 2.44: Traffic lights described as a time trajectory with two degrees of freedom, represented by the timed-controlled variables $\sigma_r^{(1)}, \sigma_r^{(2)}$.

Traffic network description In this section we first motivate the mathematical representation of traffic lights and, then, describe the network's model and our chosen measures of traffic performance.

Traffic lights model

Traffic lights are electronic devices installed at the end point (w.r.t. the flow direction) of roads for the purpose to avoid any collision between vehicles. A traffic light can either allow of forbid vehicles to continue their route, therefore it can be mathematically described as a map from the set of time instants to the set $\{0, 1\}$.

In a typical urban environment a fixed cycle length T is assigned for the traffic lights. This cycle is a slot of time during which every traffic light can switch from green to red (or viceversa) at most once. Several physical reasons justify such a behavior: faster switching may, in fact, give unfavourable consequences regarding pollution generation and drivers comfort, since in this case vehicles will have to stop–and–go with higher frequency. Typical values for T are from 90 seconds up to 2 minutes.

To fully capture the above mentioned dynamics we decide to describe the time trajectory of traffic lights with a two-degree-of-freedom trajectory (see Fig. 2.44), given by the vector of two natural numbers $\sigma_r = \left[\sigma_r^{(1)} \; \sigma_r^{(2)}\right] \in \{T_s, 2T_s, \dots, T\}^2$, where T_s is the sampling time used to discretize the dynamics, such that $T/T_s \in \mathbb{N}$. Hence, σ_r represents the raising and falling time instants of the signal u_r:

$$u_r(t, \sigma_r) = \begin{cases} 1 & \text{if } \sigma_r^{(1)} \le t \le \sigma_r^{(2)} \\ 0 & \text{otherwise,} \end{cases} \tag{2.96}$$

where t has to be intended modulo T.

Traffic dynamics on signalized networks

To describe traffic's time evolution we use the description given by a mass conservation law (Lighthill and Whitham, 1955; Richards, 1956) and by its discrete-time representation widely known as Cell Transmission Model (CTM) (Daganzo, 1994a). In particular, we consider an extension of the CTM for networks with FIFO policy at intersections, similar to the one we already introduced in Grandinetti et al. (2015a). A urban network is a collection of roads entering it ($\mathcal{R}^{\text{in}}$), internal roads ($\mathcal{R}$), and exiting roads ($\mathcal{R}^{\text{out}}$). We use the word *intersections* to identify locations where two or more roads merge. Such intersections have no capacity storage and they are signalized, in the sense that the traffic flow exiting each road $r \in \mathcal{R}^{\text{in}} \cup \mathcal{R}$ is regulated by a traffic light whose value at time instant t will be denoted by $u_r(t, \sigma_r) \in \{0, 1\}$. Whether two roads r and q are connected to the same intersection in such a way that flow can exit r and enter q we use the notation $r \multimap q$ (q is downstream w.r.t. r). Furthermore, to every road q is associated a value $\beta_q \in (0, 1)$ (split ratio) that expresses the percentage of the flow upstream q which actually wants to turn in q.

In urban scenarios we consider every road r as a cell of the CTM, hence the value of vehicles' density in it, indicated as ρ_r (which we consider as normalized w.r.t. the length L_r of the roads itself) depends on the flows f_r^{in} and f_r^{out}, entering and exiting r, respectively.

Inflow and outflow values are given according to the demand and supply paradigm (Lebacque, 1996). Given a road r its demand D_r is the flow of vehicles that want to exit r; its supply S_r is the flow that r can receive according to its storage capacity.

Every road r is characterized by given parameters: the maximum speed in free flow v_r, the speed in congestion phase w_r, the maximum density allowed ρ_r^{max} and the maximum flow φ_r^{max}. Whether the system is sampled with step size T_s, to ensure stability it must hold $T_s v_r / L_r < 1$, for every road.

Finally, external traffic demand for every road r entering the network is indicated by D_r^{in}, and external supply for every road q exiting the network is indicated by S_q^{out}.

The model is given by the following set of equations:

$$\rho_r(t + T_s) = \rho_r(t) + T_s\Big(f_r^{\text{in}}(t) - u_r(t, \sigma_r) f_r^{\text{out}}(t)\Big) \tag{2.97a}$$

$$f_r^{\text{in}}(t) = \begin{cases} \min\{D_r^{\text{in}}(t), S_r(t)\},\ r \in \mathcal{R}^{\text{in}} \\ \beta_r \sum_{q:q \rightarrowtriangle r} u_q(t, \sigma_q) f_q^{\text{out}},\ \text{oth.} \end{cases} \tag{2.97b}$$

$$f_r^{\text{out}}(t) = \begin{cases} \min\{D_r(t), S_r^{\text{out}}(t)\},\ r \in \mathcal{R}^{\text{out}} \\ \min\left\{ D_r(t), \left\{ \frac{S_q(t)}{\beta_q} \right\}_{q:r \rightarrowtriangle q} \right\},\ \text{oth.} \end{cases} \tag{2.97c}$$

$$D_r(t) = \min\{v_r \rho_r(t), \varphi_r^{\text{max}}\} \tag{2.97d}$$

$$S_r(t) = \min\{\varphi_r^{\text{max}}, w_r(\rho_r^{\text{max}} - \rho_r(t))\} \tag{2.97e}$$

$$u_r(t, \sigma_r) = \begin{cases} 1 & \text{if } \sigma_r^{(1)} \leq t \leq \sigma_r^{(2)} \\ 0 & \text{otherwise.} \end{cases} \quad \mod T \tag{2.97f}$$

Urban traffic performance metrics

Traffic behavior needs to be evaluated and assessed with respect to properly defined performance indices. There exist several metrics in literature to address traffic performance evaluation; in this paper we focus mainly on two features of the urban network.

Service of demand (SoD) A urban traffic network is an highly dynamical environment that continously receives demand from outside. This demand cannot be ignored just to favour the inner quality of the system, because the external request will end up growing with several undesired effects, due to the bigger and bigger queues arising outside.

For this reason we consider as quality of the service for a road $r \in \mathcal{R}^{\text{in}}$ the number of vehicles (users) served by that road:

$$\begin{aligned} \text{SoD}_r(t) &= f_r^{\text{in}}(t)|_{r \in \mathcal{R}^{\text{in}}} = \\ &= \min\left\{D_r^{\text{in}}(t), \varphi_r^{\text{max}}, w_r(\rho_r^{\text{max}} - \rho_r(t))\right\}, \end{aligned} \tag{2.98}$$

To improve performance we would like to maximize the sum of (2.98) over all roads in $\mathcal{R}^{\text{in}}$.

Optimization of the infrastructures usage In urban networks there are roads preferred by the users. The civil authority would like to set traffic lights as to diminuish this usage disparity, to guarantee a more equilibrate diffusion of vehicles, thus reducing hard congestions in main streets as well as the possibility of accidents.

A standard metric that takes into account this behavior is the Total Travel Distance (Gomes and Horowitz, 2006a), defined as follows:

$$\mathrm{TTD}_r(t) = \min\left\{v_r\rho_r(t), w_r(\rho_r^{\max} - \rho_r(t))\right\}. \tag{2.99}$$

We want to maximize the sum of (2.99) over all roads in $\mathcal{R}\cup\mathcal{R}^{\text{out}}$ (because boundary flows are considered by SoD).

Centralized optimal control The control strategy we designed consists in solving an optimization problem at the beginning of every cycle (i.e., every $t_0 = kT$, $k \in \mathbb{N}$), in order to decide the optimal traffic lights in the upcoming cycle. Such procedure optimizes the traffic behavior using a receding horizon philosophy that predicts densities $\rho(t_0 + nT_s), n = 1, \dots, T/T_s$. Predictions are carried out assuming that a measure of densities $\rho(t_0)$ is available for the controller.

The optimal values σ^*, i.e., the optimal activation time instants for the traffic lights, are given by:

$$\begin{aligned}\sigma^* = \arg\max_{\sigma} \sum_{n=1}^{T/T_s} \Big(a_1 \sum_{r\in\mathcal{R}\cup\mathcal{R}^{\text{out}}} & \mathrm{TTD}_r(t_0+nT_s) + \\ & + a_2 \sum_{r\in\mathcal{R}^{\text{in}}} \mathrm{SoD}_r(t_0+nT_s)\Big) \qquad (2.100) \\ & \text{under network dynamics (2.97)} \\ & \forall\, t = t_0 + nT_s,\ n = 1,\dots,T/T_s,\end{aligned}$$

where $a_1, a_2 \in \mathbb{R}$ are weights for the two involved objectives.

In the rest of this section we show that equations (2.97) and the objective function can be reformulated using logical constraints and variables, obtaining a mixed integer linear problem (MILP). To this aim, we use the reasoning outlined in Bemporad and Morari (1999). At the end of the reformulation, the problem's constraints, instead of (2.97), will be given by (2.102)–(2.106),(2.109),(2.111)–(2.113).

Such final formulation of the problem is more convenient because MILPs are extensively studied, and there exist good numerical solvers to deal with them, e.g. Gurobi Optimization (2015).

Traffic lights constrained trajectory

Introducing the binary variables $\delta_r^{(1)}(t)$, $\delta_r^{(2)}(t)$, the constraint expressed by (2.96), for a given time instant t, is equivalent to the following:

$$\left[\delta_r^{(1)}(t) = 1\right] \longleftrightarrow \left[\sigma_r^{(1)} \le t\right] \tag{2.101a}$$

$$\left[\delta_r^{(2)}(t) = 1\right] \longleftrightarrow \left[t \le \sigma_r^{(2)}\right] \tag{2.101b}$$

$$\left[u_r(t) = 1\right] \longleftrightarrow \left[\delta_r^{(1)}(t) = 1 \wedge \delta_r^{(2)}(t) = 1\right]. \tag{2.101c}$$

Let $M^{(1)}$ and $m^{(1)}$ upper and lower bounds[10] such that $m^{(1)} < \sigma_r^{(1)} - t < M^{(1)}$, for every $\sigma_r^{(1)}$; then (2.101a) is equivalent to the following constraints:

$$\sigma_r^{(1)} - t \leq M^{(1)}(1 - \delta_r^{(1)}(t)) \tag{2.102a}$$

$$\sigma_r^{(1)} - t > m^{(1)}\delta_r^{(1)}(t). \tag{2.102b}$$

Similarly, let $M^{(2)}$ and $m^{(2)}$ such that $m^{(2)} < \sigma_r^{(2)} - t < M^{(2)}$, for every $\sigma_r^{(2)}$; then (2.101b) is equivalent to the following constraints:

$$\sigma_r^{(2)} - t \leq M^{(2)}(1 - \delta_r^{(2)}(t)) \tag{2.103a}$$

$$\sigma_r^{(2)} - t > m^{(2)}\delta_r^{(2)}(t). \tag{2.103b}$$

Finally, the constraint (2.101c) is equivalent to the following:

$$u_r(t) - \delta_r^{(1)}(t) \leq 0 \tag{2.104a}$$

$$u_r(t) - \delta_r^{(2)}(t) \leq 0 \tag{2.104b}$$

$$\delta_r^{(1)}(t) + \delta_r^{(2)}(t) - u_r(t) \leq 1 \tag{2.104c}$$

$$\delta_r^{(1)}(t), \delta_r^{(2)}(t) \in \{0, 1\}.$$

Depending on the physics of system it may be useful to impose another constraint: we can require that the rising instant $\sigma_r^{(1)}$ and the falling one $\sigma_r^{(2)}$ are sufficiently separated in time, so ensuring that too short *green slots* are avoided. This can be obtained with the following constraints:

$$\sigma_r^{(1)} + \sigma_r^{\min} \leq \sigma_r^{(2)} \tag{2.105a}$$

$$T_s \leq \sigma_r^{(1)}, \sigma_r^{(2)} \leq T, \tag{2.105b}$$

where $\sigma_r^{\min}$ are consistently assigned to every traffic light.

Collision avoidance constraints

To guarantee the safe crossing we impose an hard constraint such that at every time instant only one among roads entering the same intersection has the right of way. This condition is expressed by the following linear constraint:

$$\sum_{r : r \rightarrowtriangle q} u_r(t, \sigma_r) \leq 1, \tag{2.106}$$

for every road q. Notice that constraint (2.106) let the further freedom to assign *red* to all traffic lights at the same intersection, if it is required.

State constraints

The dynamics defined in (2.97) is non linear due to the $\min$ operator and to the product between flows and traffic lights' values. Our aim is to show how such a dynamics may be reformulated with mixed integer linear constraints.

[10] Here and in what follows the numerical values for these bounds are omitted. Notice that flows in the network are always in $[0, \varphi^{\max}]$, while time instants indicated by σ are in $[T_s, T]$. Bounds can be manually set even if some software, like Yalmip (Lofberg, 2004), is able to determine them while parsing the problem formulation.

Let σ be a vector containing σ_r's for every road r, and let $\bar{f}_r^{\text{in}}(t,\sigma)$ and $\bar{f}_r^{\text{out}}(t,\sigma)$ be the following modified flows:

$$\bar{f}_r^{\text{in}}(t,\sigma) = \begin{cases} \beta_r \sum_{q:q \rightarrowtriangle r} \bar{f}_q^{\text{out}}(t,\sigma), & r \in \mathcal{R} \cup \mathcal{R}^{\text{out}} \\ f_r^{\text{in}}(t), & r \in \mathcal{R}^{\text{in}} \end{cases} \tag{2.107a}$$

$$\bar{f}_r^{\text{out}}(t,\sigma) = \begin{cases} u_r(t,\sigma_r) f_r^{\text{out}}(t), & r \in \mathcal{R}^{\text{in}} \cup \mathcal{R} \\ f_r^{\text{out}}(t), & r \in \mathcal{R}^{\text{out}}. \end{cases} \tag{2.107b}$$

We now show a scheme to rewrite (2.107b), i.e., $\min$ of several functions multiplied by a binary variable.

For every road $r \in \mathcal{R}^{\text{in}} \cup \mathcal{R}$ let $d^{(\text{r})}(t) \in \{0,1\}^h$ be a vector of binary variables, where $h = 1 + \#\{q : r \rightarrowtriangle q\}$. Then the definition of outflow (2.97c) is equivalent to the following constraints:

$$\left[d_0^{(\text{r})}(t) = 1\right] \longrightarrow \left[f_r^{\text{out}}(t) = D_r(t)\right] \tag{2.108a}$$

$$\left[d_q^{(\text{r})}(t) = 1\right] \longrightarrow \left[f_r^{\text{out}}(t) = \frac{S_q(t)}{\beta_q} \, \forall\, q : r \rightarrowtriangle q\right] \tag{2.108b}$$

$$\left[d_0^{(\text{r})}(t) = 1\right] \longrightarrow \left[D_r(t) \leq \frac{S_q(t)}{\beta_q} \, \forall\, q : r \rightarrowtriangle q\right] \tag{2.108c}$$

$$\left[d_i^{(\text{r})}(t) = 1\right] \longrightarrow \left[\frac{S_q(t)}{\beta_q} \leq \frac{S_i(t)}{\beta_i} \, \forall\, i \neq q\right] \tag{2.108d}$$

$$\sum_{i=1}^{h} d_i^{(\text{r})}(t) = 1. \tag{2.108e}$$

Given the following upper and lower bounds: $l_0 < f_r^{\text{out}}(t) - D_r(t) < L_0$, $l_q < f_r^{\text{out}}(t) - S_q(t)/\beta_q < L_q$, $\psi_0 < D_r(t) < \Psi_0$, $\psi_q < S_q(t)/\beta_q < \Psi_q$, logical constraints (2.108a)–(2.108d) are equivalent to the following linear ones:

$$f_r^{\text{out}}(t) - D_r(t) \geq l_0(1 - d_0^{(\text{r})}(t)) \tag{2.109a}$$

$$f_r^{\text{out}}(t) - \frac{S_q(t)}{\beta_q} \geq l_q(1 - d_q^{(\text{r})}(t)) \tag{2.109b}$$

$$f_r^{\text{out}}(t) - D_r(t) \leq L_0(1 - d_0^{(\text{r})}(t)) \tag{2.109c}$$

$$f_r^{\text{out}}(t) - \frac{S_q(t)}{\beta_q} \leq L_q(1 - d_q^{(\text{r})}(t)) \tag{2.109d}$$

$$D_r(t) \leq \frac{S_q(t)}{\beta_q} + (\Psi_0 - \psi_q)(1 - d_q^{(\text{r})}(t)) \tag{2.109e}$$

$$\frac{S_q(t)}{\beta_q} \leq \frac{S_i(t)}{\beta_i} + (\Psi_q - \psi_i)(1 - d_q^{(\text{r})}(t)) \tag{2.109f}$$

$$\forall\, q : r \rightarrowtriangle q,\ \forall\, i \neq q$$

$$d^{(\text{r})}(t) \in \{0,1\}^h,$$

for every road i and $j \neq i$.

Finally, given $u_r(t,\sigma) \in \{0,1\}$ setting

$$\bar{f}_r^{\text{out}}(t,\sigma) = u_r(t,\sigma) f_r^{\text{out}}(t) = \begin{cases} f_r^{\text{out}}(t) & \text{if } u_r(t,\sigma) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.110}$$

is equivalent to

$$g(1 - u_r(t,\sigma)) + \bar{f}_r^{\text{out}}(t,\sigma) \leq f_r^{\text{out}}(t) \tag{2.111a}$$

$$-G(1 - u_r(t,\sigma)) - \bar{f}_r^{\text{out}}(t,\sigma) \leq -f_r^{\text{out}}(t) \tag{2.111b}$$

$$-Gu_r(t,\sigma) + \bar{f}_r^{\text{out}}(t,\sigma) \leq 0 \tag{2.111c}$$

$$gu_r(t,\sigma) - \bar{f}_r^{\text{out}}(t,\sigma) \leq 0, \tag{2.111d}$$

where $g < f_r^{\text{out}}(t) < G$.

Objective functions

The metrics illustrated in Section 2.4.2 are nonlinear functions of roads' density. Therefore, we use the same technique previously employed, which allows us to transform their expression as linear constraints.

SoD Using the scheme illustrated by (2.108)–(2.111), expression (2.98) is equivalent to:

$$\pi_1(1 - b_1(t)) \leq \text{SoD}_r(t) - D_r^{\text{in}}(t) \leq \Pi_1(1 - b_1(t)) \tag{2.112a}$$

$$\pi_2(1 - b_2(t)) \leq \text{SoD}_r(t) - \varphi_r^{\text{max}} \leq \Pi_2(1 - b_2(t)) \tag{2.112b}$$

$$\pi_3(1 - b_3(t)) \leq \text{SoD}_r(t) - w_r(\rho_r^{\text{max}} - \rho_r(t)) \tag{2.112c}$$

$$\text{SoD}_r(t) - w_r(\rho_r^{\text{max}} - \rho_r(t)) \leq \Pi_3(1 - b_3(t)) \tag{2.112d}$$

$$D_r^{\text{in}}(t) \leq \varphi_r^{\text{max}} + (P_1 - p_2)(1 - b_1(t)) \tag{2.112e}$$

$$D_r^{\text{in}}(t) \leq w_r(\rho_r^{\text{max}} - \rho_r(t)) + (P_1 - p_3)(1 - b_1(t)) \tag{2.112f}$$

$$\varphi_r^{\text{max}} \leq D_r^{\text{in}}(t) + (P_2 - p_1)(1 - b_2(t)) \tag{2.112g}$$

$$\varphi_r^{\text{max}} \leq w_r(\rho_r^{\text{max}} - \rho_r(t)) + (P_2 - p_3)(1 - b_2(t)) \tag{2.112h}$$

$$w_r(\rho_r^{\text{max}} - \rho_r(t)) \leq D_r^{\text{in}}(t) + (P_3 - p_1)(1 - b_3(t)) \tag{2.112i}$$

$$w_r(\rho_r^{\text{max}} - \rho_r(t)) \leq \varphi_r^{\text{max}} + (P_3 - p_2)(1 - b_3(t)) \tag{2.112j}$$

$$b(t) \in \{0,1\}^3,$$

where Π, P (π, p) are upper (lower) bounds consistently chosen.

TTD Similarly, expression (2.99) is equivalent to the following:

$$\gamma_1(1 - c_1(t)) \leq \text{TTD}_r(t) - v_r\rho_r(t) \tag{2.113a}$$

$$\text{TTD}_r(t) - v_r\rho_r(t) \leq \Gamma_1(1 - c_1(t)) \tag{2.113b}$$

$$\gamma_2(1 - c_2(t)) \leq \text{TTD}_r(t) - w_r(\rho_r^{\text{max}} - \rho_r(t)) \tag{2.113c}$$

$$\text{TTD}_r(t) - w_r(\rho_r^{\text{max}} - \rho_r(t)) \leq \Gamma_2(1 - c_2(t)) \tag{2.113d}$$

$$v_r\rho_r(t) \leq w_r(\rho_r^{\text{max}} - \rho_r(t)) + (\Theta_1 - \theta_2)(1 - c_1(t)) \tag{2.113e}$$

$$w_r(\rho_r^{\text{max}} - \rho_r(t)) \leq v_r\rho_r(t) + (\Theta_2 - \theta_1)(1 - c_2(t)) \tag{2.113f}$$

$$c(t) \in \{0,1\}^2,$$

where Γ, Θ (γ, θ) are upper (lower) bounds consistently chosen.

Decentralized suboptimal control The control strategy illustrated in the previous section, while providing the optimal values for traffic lights, requires to solve an intractable (NP–hard) problem. In this section we therefore propose a decentralized realization of such strategy, which reduces the computational burden substantially. It is based on two main ingredients: suboptimal solutions for local problems and agreement policy between local solutions.

Local problems

The optimization procedure is decentralized among intersections, each of which solves a local MILP. For every local problem, i.e., over intersection A, a receding horizon approach is used and the predicted densities belong to the set

$$\mathcal{R}_{\mathrm{A}} = \big\{ r \in \mathcal{R}^{\mathrm{in}} \cup \mathcal{R} \cup \mathcal{R}^{\mathrm{out}} \; : r \text{ entering or exiting A} \big\}, \tag{2.114}$$

while the optimization variables are all traffic lights in the following set:

$$\Omega_{\mathrm{A}} = \big\{ \sigma_r \; : r \in \mathcal{R}_{\mathrm{A}} \big\} \cup \big\{ \sigma_q \; : q \rightarrowtriangle r, r \text{ entering A} \big\}. \tag{2.115}$$

Furthermore, densities for roads in $\mathcal{R}^{\mathrm{in}} \cup \mathcal{R} \cup \mathcal{R}^{\mathrm{out}} \setminus \mathcal{R}_{\mathrm{A}}$ are considered constant (equal to the last measured values). This is the main reason of suboptimality, since the objective function maximized in every subproblem is computed over the set $\mathcal{R}_{\mathrm{A}}$, i.e.,

$$\begin{aligned} J_{\mathrm{A}} = \sum_{n=1}^{T/T_s} \Bigg(a_1 \sum_{r \in \mathcal{R}_{\mathrm{A}} \setminus \mathcal{R}^{\mathrm{in}}} & \mathrm{TTD}_r(t_0 + nT_s) + \\ & + a_2 \sum_{r \in \mathcal{R}_{\mathrm{A}} \cap \mathcal{R}^{\mathrm{in}}} \mathrm{SoD}_r(t_0 + nT_s) \Bigg). \end{aligned} \tag{2.116}$$

As illustrative example look at Fig. 2.45, where we consider the optimization solved by intersection A. In such a scenario density predictions are carried out only for roads 1–4, and the problem is solved with respect to the variables $\sigma_1, \dots, \sigma_8$ (notice that $\sigma_5, \dots, \sigma_8$ are needed to compute inflows for roads 1 and 2).

It is worth to note that, among all σ's considered by an intersection, only a subset of them fulfills hard constraints for collision avoidance within the local optimization: referring again to Fig. 2.45, intersection A guarantees such fulfillment only for σ_1 and σ_2. For the other variables considered by A (marked in red in the figure) there might be some hard constraint which is not included within the local problem (in Fig. 2.45 this happens for σ_3 and σ_4); Therefore, the values computed by A for these variables can be interpreted only as suggestions that A would like to advice its neighbor intersections, to optimize its own objective. How such suggestions are considered is explained in the next section.

Agreement policy

To let local problems taking care of advices provided by neighbors we save the result of every local optmization. For instance, when A solves its subproblem, we save the variables $\sigma_{r,\mathrm{A}}$, for every r involved in such problem. Once these informations are available we can identify, for every road r, the set $\mathcal{S}_r$ of all intersections whose local problem involves σ_r. Then the average suggestion given by intersections about σ_r is computed as:

$$\hat{\sigma}_r = \frac{1}{|\mathcal{S}_r|} \sum_{\mathrm{I} \in \mathcal{S}_r} \sigma_{r,\mathrm{I}}. \tag{2.117}$$

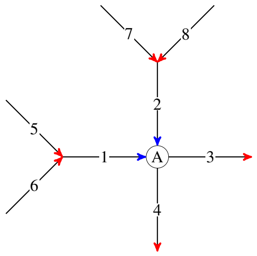


Figure 2.45: Illustrative example of local subproblems.

We now use the values $\hat{\sigma}_r$ to modify the cost function of every local problem, as the following:

$$\hat{J}_{\mathrm{A}} = J_{\mathrm{A}} - a_3 \sum_{\sigma_r \in \Omega_{\mathrm{A}}} \|\sigma_r - \hat{\sigma}_r\|_1 \,, \tag{2.118}$$

where a_3 is a real number used to weight neighbors' suggestions w.r.t. A's own objective. Notice also that the 1–norm can be turned in a mixed integer linear formulation adding two binary variables for every $\sigma_r \in \Omega_{\mathrm{A}}$. This is the computational prize to pay in order to take into account suggestions by neighbors, rather then ignoring them; the problem, however, is still significantly numerically more efficient than its centralized version. This idea can be iterated N_{it} times, in order to let suggestions spread among intersections. The scheme we implemented is the following:

0. Let N_{it} be assigned;
1. For every intersection A solve the local MILP with cost function J_{A} and save the resulting $\sigma_{r,\mathrm{A}}$;
2. If $N_{\mathrm{it}} = 0$ then stop;
3. For every signalized road r compute the average suggestions $\hat{\sigma}_r$;
4. For every intersection A solve the local MILP with cost function $\hat{J}_{\mathrm{A}}$, and use the result to update $\sigma_{r,\mathrm{A}}$;
5. If for every r and for every A there was no update in $\sigma_{r,\mathrm{A}}$ then stop;
6. Decrement N_{it} by 1, goto 2.

Whenever the algorithm stops, the value assigned to every traffic light is

$$\sigma_r^* = \sigma_{r,\mathrm{A}}, \text{ where } r \text{ enters A}, \tag{2.119}$$

since every intersection guarantees the fulfillment of collision avoidance (hard) constraints only over roads entering it.

Notice that if $N_{\mathrm{it}} = 0$ is given, this means that suggestions from neighbors are ignored. An important feature of our scheme is that the subproblems' complexity does *not* depend on the network size; therefore the algorithm complexity is linear in the number of intersections and in the chosen number of iteration, which is a tunable parameter. Our numerical results, presented in the next section, show that the algorithm stops already after three iterations as no changes appear in the solution.

Another benefit of this decentralized strategy is that step 1 (the same applies to step 4), requires to solve a set of optimization problems that are completely indipendent of each other. Therefore the procedure can be implemented even more efficiently in a parallel architecture, where there is a controller at every intersection that exchanges σ's values with the others.

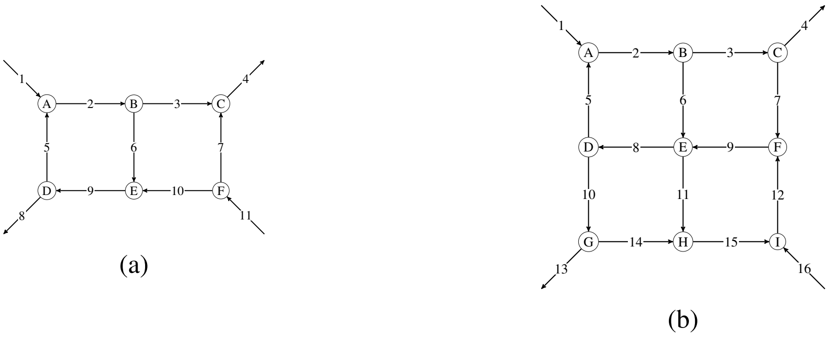


Figure 2.46: Networks used for simulations and comparisons.

Simulations and comparisons The strategies illustrated in the previous sections have been tested via software simulations in MatLab environment, using software (Gurobi Optimization, 2015; Lofberg, 2004) to solve the mixed integer linear programs. The simulated scenario is the following: all roads in the network have same physical properties (v,w,$\varphi^{\max}$,$\rho^{\max}$). Each simulation is run for a virtual time of 45 minutes, when traffic lights' cycle is 90 seconds and sampling time is 10 seconds. As explained in Section 2.3.3, the optimization problem is solved at the beginning of each cycle, therefore in this set-up the control values are computed 30 times. The constant $\sigma^{\min}$ is set equal to 2 sample steps, so the minimum green slot is 20 seconds. Outside the network, time-varying demands and supplies are randomly uniformly generated in the interval $[0.5, 1]\varphi^{\max}$ for the entire simulated time; by doing so, the network's state changes during the simulation (from overall free to overall congested, and viceversa), and the controllers are then tested in different circumstances. The numerical results here presented are obtained as mean values over the entire simulation time.

We would like to stress the fact that the centralized strategy we propose is able to solve a very general problem, despite the computational inefficiency. The proposed traffic lights' representation, along with the numerical optimization, guarantees optimal behavior by means of the chosen objective index, as it may cleverly choose phase shifts between lights as well as green time for each of them. Therefore, the results obtained applying this techinique are considered as benchmark (upper bounds) to evaluate performances of the decentralized strategy, from traffic and computational performance point of view. We generate the afore mentioned scenario for two sample networks, shown in Fig. 2.46.

Representative results of the simulations are reported in Table 2.6. Our numerical results are encouraging:

- The decentralized strategy obtains performance around 80-85% of the ones obtained by the centralized optimization;
- Computational time is dramatically reduced, especially with growing network's size (as expected from the NP–hardness of the problem). Notice also that in these simulation we did not make use of possible multithread implementation.

Conclusions and future works We presented an optimal control scheme for urban signalized traffic networks that schedules traffic lights taking into account upcoming external demands. This strategy, on one hand, ensures optimal performances, on the other, requires to solve a mixed integer linear program. To overcome the hardness of this problem, we proposed a decentralized realization of the same decision making scheme, that allows parallel computation and scalabiility to any network size. This approach guarantees effective handling of large amount of data coming from traffic sensors and scalability to any

Table 2.6: Normalized performance comparison between the centralized strategy (MILP), considered as benchmark, and the decentralized one (Dec–MILP).

	Network 2.46a		Network 2.46b	
	MILP	Dec–MILP	MILP	Dec–MILP
TTD	1	0.82	1	0.83
SoD	1	0.86	1	0.85
cpu time	1	0.6	1	0.03

network size. Our software simulations have shown that the computational load is extremely reduced by the decentralized scheme, which moreover achieves quite good performances from traffic point of view.

Future work will aim to investigate the scalability properties of the depicted techniques, combined with improvements on the gap with the upper bound provided by the centralized algorithm.

Event	Relative frequency
PredictedCongestion	10%
Congestion	5%
ClearCongestion	10%
setMeteringRateLimits	5%
CoordinateRamps	5%
PredictedRampOverflow	5%
ClearRampOverflow	5%
AverageOnRampValuesOverInterval	20%
AverageDensityAndSpeedPerLocation	35%

Table 2.7: Relative frequency of events in the test case used to obtain the timings in Figure 2.47.

2.4 Performance Evaluation

In this section, we provide an overview of the component evaluation of the DM algorithms. We perform the evaluation with respect to two main criteria: In the next section 2.4.1, we analyze computation times of the algorithms, implemented in an event-driven framework as outlined in the corresponding deliverable (D 6.1). This serves to demonstrate **real time capabilities** of the algorithms. In the following section 2.4.2, we compare the performance of the algorithms in terms of traffic metrics to quantify the **improvement over the state of the art**.

2.4.1 Real-time Decision Making

A detailed evaluation of the time needed to process individual events by DM is summarized in Figure 2.47. To ensure perfect control over the type of events processed as well as the attribute values, this test is run outside of the SPEEDD cluster architecture on a local computer (MacBook Pro with OSX El Capitan, 2.3 GHz Intel Core i7). The events used in this experiment have been chosen such as to overrepresent "difficult" cases, i.e. attribute values are more often close to the margins than one might expect in reality, congestion events are reported more often and user commands are sent frequently. In addition, up to 20% (depending on the event type) of the events are sent with invalid attributes, to test code stability and exception handling. The individual events are chosen with relative frequencies as reported in Table 2.7. In total, 10^6 events are processed in 362s. Note that this time and all the other times reported in this section, only account for the computation time of DM **once the event has been received by DM**, as we only intend to show **real time capability of the DM algorithms**. For example, the overhead incurred to send, transmit and emit events is not included. A summary of the computation times is displayed in Figure 2.47. It should be highlighted that Figure 2.47 shows a multimodal distribution of computation times. Many events are processed in less than 3μs (e.g. almost 90% of AverageDensityAndSpeedPerLocation events), but certain events take between 10μs and 0.1ms to be processed, (e.g. the remaining AverageDensityAndSpeedPerLocation events). The most computationally expensive events on the other hand take up to 10ms to be processed.

This multimodal distribution with three distinct modes results from the type of action required to process each event. For certain events, no immediate action by DM is necessary or advisable. The processing of such events can be as simple as updating certain variables, and is often accomplished in less than 1μs. In contrast, other events require immediate action. An example is a reevaluation of

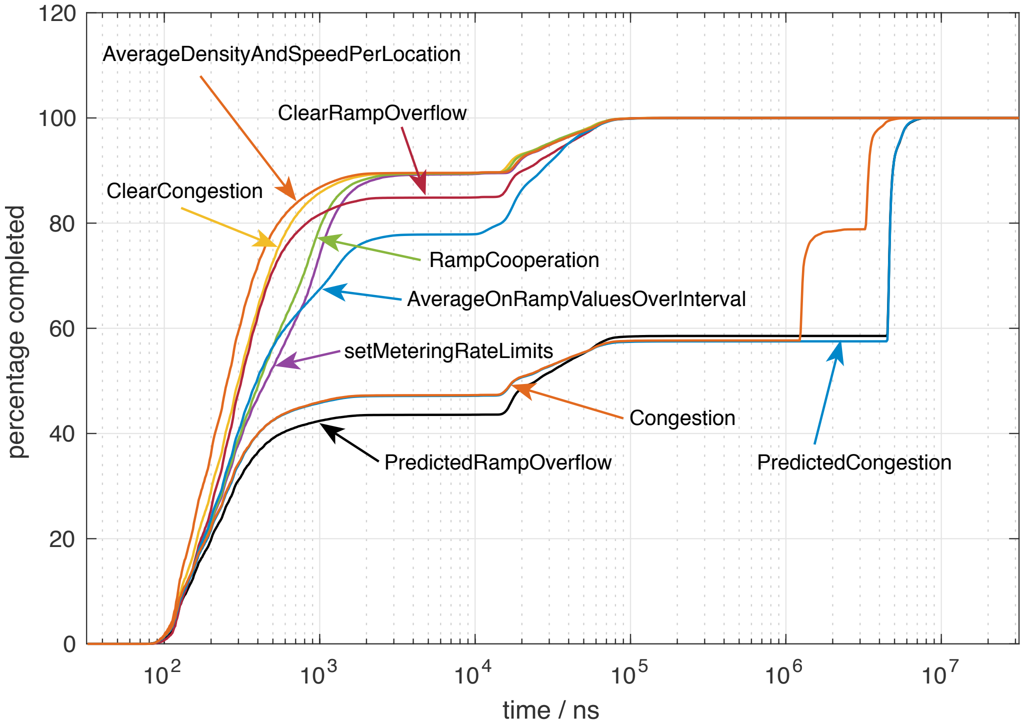


Figure 2.47: The percentage of events of a certain type processed by the DM module for the freeway use case as a function of the execution time.

the ramp metering policy, which requires at least a recomputation of the metering rates. Most effort is required for predicted events, for which often a re-estimation of the GP system model (Section 2.1.1) is necessary. We provide a comprehensive explanation for the observed results for different types of events next.

(i) **AverageDensityAndSpeedPerLocation, AverageOnRampValuesOverInterval:** These events contain aggregate information about the system state. For most such events, processing is as simple as performing one (linear) update of the internal state of the DM. Therefore, almost 90% of these events are processed within 3μs. Only a small fraction of these events of about 10% (more precisely: measurements from a sensor on a metered onramp) is also used as a trigger for recomputing ramp metering rates. The additional time required to perform these computations increases the time to process the events to 10μs and 0.1ms.

(ii) **ClearCongestion, ClearRampOverflow:** It can be seen from Figure 2.47 that these types of events are processed comparatively fast. This is not surprising since these events always exhibit 100% certainty. They mainly act as triggering mechanisms to turn off ramp metering control after a critical traffic situation has been resolved. Therefore, processing them is often as simple as updating the internal state of DM and potentially resetting the state of the high-level coordination algorithms.

(iii) **setMeteringRateLimits:** This event corresponds to a user command and is always sent with 100% certainty. It always requires an (inexpensive) update of the corresponding variables. In about 22% of the instances in this test case, this change in the metering rate limits requires an

Event	Mean	50%	95%	99.9%
PredictedCongestion	$2060\mu s$	$18\mu s$	$5305\mu s$	$7345\mu s$
Congestion	$1030\mu s$	$18\mu s$	$3535\mu s$	$5363\mu s$
ClearCongestion	$3.9\mu s$	$0.3\mu s$	$29\mu s$	$94\mu s$
setMeteringRateLimits	$4.4\mu s$	$0.5\mu s$	$31\mu s$	$103\mu s$
CoordinateRamps	$41\mu s$	$0.4\mu s$	$29\mu s$	$99\mu s$
PredictedRampOverflow	$2014\mu s$	$25\mu s$	$5324\mu s$	$7398\mu s$
ClearRampOverflow	$5.5\mu s$	$0.3\mu s$	$39\mu s$	$100\mu s$
AverageOnRampValuesOverInterval	$7.4\mu s$	$0.4\mu s$	$39\mu s$	$108\mu s$
AverageDensityAndSpeedPerLocation	$4.1\mu s$	$0.3\mu s$	$31\mu s$	$99\mu s$

Table 2.8: Statistics of computation times for individual events.

immediate change in the metering policy, since the bounds become active. The effect, that only sometimes an immediate change in the metering policy is required, is likely responsible for the bimodal distribution of computation times for this event.

(iv) **RampCoordination:** This event is very similar in nature to the **setMeteringRateLimits** event, since one can interpret the call for ramp coordination just as a request to restrict the metering rates depending on the event attributes. The only difference to the metering rate limits defined by the user via the UI is that for ramp coordination, the rates are restricted based on the requirements of downstream ramps for which ramp overflow is imminent.

(v) **Congestion:** Note that some congestion events, close to 50% in this experiment according to Figure 2.47, do not require further processing. This happens if the DM is already aware of a particular congestion, e.g. because it was perfectly predicted or congestion events are sent multiple time. In all other instances, the main computational effort that arises for a **Congestion** event comes from the necessity to evaluate the internal GP model. The evaluation of the GP is always executed if the congestion was not detected earlier and serves to determine the maximal bottleneck flow, which in turns influences the metering rates. However, since a congestion event always exhibits 100% certainty, no re-estimation of the GP is necessary, which makes processing congestion events somewhat cheaper than predicted events.

(vi) **PredictedCongestion, PredictedRampOverflow:** These events are in general uncertain. Therefore, a re-estimation of the internal GP model is required, in order to related the reported uncertainty to the uncertainty present in the internal model and ultimately to determine the appropriate reaction. Note that similar to congestion events, some events are essentially duplicates and do not require further processing.

For completeness, we also report the computation times in Table 2.8 numerically. To account for the multimodal nature of the distribution, we first report mean computation time, which might be more or less meaningful in practice, since it of course heavily depends on the percentage of events that do not require immediate processing. We then also report the 50%, 95% and 99.9% quantile, which are more accurately able to capture the tail of the distribution of computation times.

It shall be highlighted that the distribution of computation times might vary depending on the composition of the event stream. For example, less predicted events (**PredictedCongestion**, **PredictedRampOverflow**), which routinely require a re-estimation of the internal GP model, might lead to increased

	Benchmark		Optimum	
	TTS	TWT	TTS	TWT
Sec. 2.2.1 *Local feedback*, benchmark: no control	−5.33%	−17.53%	+0.054	+0.178%
Sec. 2.2.1 *Local feedback*, benchmark: ALINEA	−0.43%	−1.42%	+0.054	+0.178%
Sec. 2.3.2 *Coordinated metering*, benchmark: no coordination	-	up to -31%	unknown	unknwon
	TTD	SoD	TTD	SoD
Sec. 2.3.3 *Dec-MILP*, Network 2.46a	18%	14%	unknown	unknown
Sec. 2.3.3 *Dec-MILP*, Network 2.46b	17%	15%	unknown	unknown

Table 2.9: Performance comparisons between the algorithms proposed in this deliverable and various benchmarks. Improvements against the benchmarks are indicated in green and suboptimal values are indicated in red. Note that we minimize the TTS and TWT metrics, whereas we maximize TTD and SoD.

average computation times for measurement events (**AverageDensityAndSpeedPerLocation**), since the GP model needs to be re-estimated after a certain time. If no re-estimation because of predicted events is required, the computations will be performed during evaluation of some measurement event after a certain time, that is, after a certain number of measurements. Also, it is noticeable, that the computation time for **PredictedRampOverflow** events are very similar to the distribution of computation times for **PredictedCongestion** events. The exact distribution of computation times depends also on the attribute values which will differ in a closed-loop simulation or execution. The similarity between both distributions is therefore likely only an artifact created by the way events have been created randomly for this evaluation.

2.4.2 Improvements in Terms of Traffic Metrics

For each of the algorithms presented in this section, we have quantified the quantitative improvements, as summarised in Table 2.9. In particular, the algorithms are evaluated against a *benchmark* and the *optimal* solution, if available. The choices for the benchmarks are given in the table itself. Note that the *optimum* refers to the global optimal solution under perfect model knowledge and perfect traffic demand prediction. This performance will be unobtainable for any real-world controller and it merely provides an estimate of the potential for further improvement. Depending on the scenario, we compare algorithms w.r.t. Total Time Spent (TTS) and Total Waiting Time (TWT), as defined in Section 2.2.1. Note we seek to *minimize* both quantities, therefore negative values in the table indicate an improvement of the proposed algorithms over the benchmark performance. Conversely, we seek to maximize the Total Travel Distance (TTD) and the Service of Demand (SoD), as defined in Section 2.3.3. All results reported in Table 2.9 can also be found in the respective sections. Note that for some control problems, there is no efficient method to compute the optimal solution because these problems are nonconvex and far too large to be solvable by available tools. Therefore, the relation of the obtained results to the optimal solution are reported as "unknown".

In the traffic use case, existing solutions for freeways such as *Alinea* are proprietary systems and the decision making algorithm used in these approaches are not publicly available. To circumvent this problem, we suggest the use of the first version of the SPEEDD prototype as the state-of-the-art.

For inner city traffic, this problem is only more acute. Design principles for existing traffic controls in the city of Geneva are opaque. In fact, many of the existing urban traffic control solutions around the world have been hand-tuned over several years, and the design principles are absent or lost. Also, these systems have been designed at some point for existing traffic conditions and have not been thoroughly or rigorously updated since. In this context, an important contribution of the decision making component in SPEEDD are the systematic design approaches developed in this project.

3

DM for Credit Card Fraud

In the use-case of credit card fraud detection, the question of reliable and accurate detection of fraud is predominant and decision making is reduced to a supporting role, providing additional detection algorithms or addressing the question of how to automate the decision tasks that in current state-of-the-art solutions are left to human experts.

3.1 Inverse Optimization

Decision making for the credit card use case deals with fraud detection, based on inputs from event processing for this use case as well as other sources of information. In current practice, the decision makers are often fraud analysts. We examine how to automate this last and important step in fraud detection for the credit card use case. The event processing algorithms identify fraud patterns, or their likelihoods, and this information can be used to decide on whether a fraud has occurred or not. However, credit card companies often use humans, with access to the entire transaction history and other information such as user profiles, to take the final call on fraud. Using models for cognitive processing, we wish to identify the rationale behind the analysts' decisions, so as to replicate them using automated controllers.

A cognitive processing model for an analyst's decisions is depicted in Figure 3.1. The cognitive model is inherently dependent on the visual processes involved: how is the data presented to the analyst and how is such information processed by a human observer? The answer to the first question is determined by the design of the graphical user interface. Data can be presented in the form of lists, numerical values, bar charts, colour maps, etc. Each visual display method has an associated cognitive model, which describes how quickly and accurately the human eye and brain absorb this information. Using this information, the cognitive process updates a belief vector that summarizes the analyst's current understanding of the case at hand and takes a suitable decision. The decision could be to examine another data source on the graphical user interface, or to click on the visual aid to access exact numerical values, or to take a final call on the absence or presence of fraud. By virtue of the fraud analyst's expertise, these decisions can be assumed to be optimal for some unknown cost function. If this cost function were known, it may be possible to design an optimal control policy for this cost, and to use this control policy to replace the expert analyst. We show how inverse optimization can be used to identify this cost function for a simple visual cognitive process.

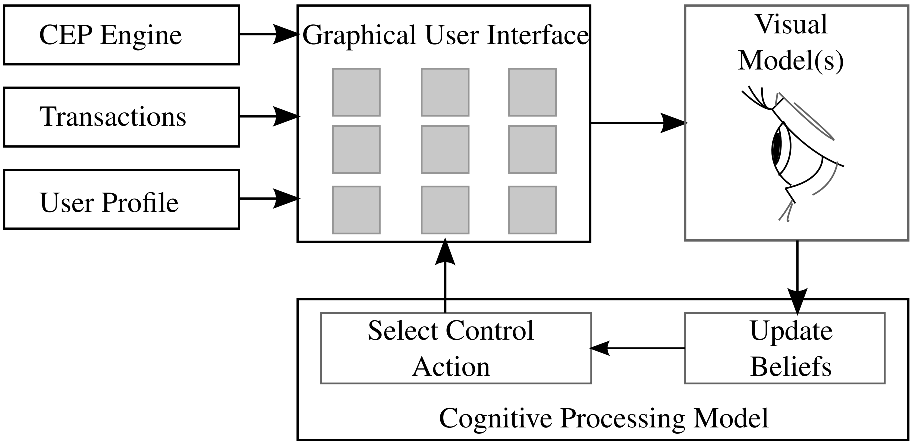


Figure 3.1: A visual and cognitive processing model for a credit card fraud expert who scans and operates the graphical user interface.

3.1.1 Problem Formulation

We describe the model for the cognitive process from Figure 3.1 for a static visual display consisting of an $n \times m$ matrix of encoded symbols. We have kept our formulation generic to motivate its use for various display methods and the associated cognitive models.

Physical Process: We consider a static visual display represented by $X \in \mathcal{X}^{n\times m}$, where $n \times m$ is the size of the visual array and the elements in the array are encoded using the alphabet $\mathcal{X} = \{1, \dots, M\}$. The state corresponding to this physical process at time $k \geq 0$ is $x_k = \text{vec}\,\{X\}$. The visual array corresponds to a static display, and thus, $x_{k+1} = x_k$.

Measurement Model: We apply the spatial smearing and feature noise model for the measurement process, resulting in the measurements

$$y_k = C_{\rho_k} x_k + v_{\rho_k,k} \ . \tag{3.1}$$

Here, $\rho_k \in \mathcal{R}$ is a non-autonomous switching process taking values in the set $\mathcal{R} := \{1, \dots, nm\}$. A mode $\rho_k = r$ denotes the new fixation, following a saccade, on the r^{th} element of the state x_k, or on the corresponding $(i_r, j_r)^{\text{th}}$ element of the visual array X, for $i_r = \text{mod}(r, n)$ and $j_r = \lfloor r/n \rfloor + 1$.

As per the spatial smearing model, the measurement is a weighted average of the elements of the cell corresponding to the new fixation and its neighbours. Thus, the q^{th} component of C_{ρ_k} is

$$C_{\rho_k,q} = \begin{cases} w_{\rho_k,q} & \text{if } q \in \mathcal{N}(\rho_k)\ , \\ 0 & \text{otherwise}\ . \end{cases}$$

Here, $\mathcal{N}(\rho_k)$ denotes the set of indices of the cell ρ_k and its neighbours, and is defined as $\mathcal{N}(\rho_k) := \{q \in \{1, \dots, nm\} \mid i_q = i_{\rho_k} \pm 1\ ,\ j_q = j_{\rho_k} \pm 1\}$. The weights $w_{\rho_k,q}$ are determined by the Gaussian kernel with standard deviation $\sigma(\theta_{\rho_k,q}, d_{\rho_k,q})$, where $\theta_{\rho_k,q}$ and $d_{\rho_k,q}$ are the angle and distance between the fovea, which is fixated at the centroid of the ρ_k^{th} cell, and the centroid of the q^{th} cell, respectively. The standard deviation $\sigma(\theta_{\rho_k,q}, d_{\rho_k,q})$ is defined as

$$\begin{aligned} \sigma(\theta_{\rho_k,q}, d_{\rho_k,q}) &= 1 - \text{acuity}(\theta_{\rho_k,q}, d_{\rho_k,q})\ , \\ \text{acuity}(\theta_{\rho_k,q}, d_{\rho_k,q}) &= \begin{cases} (d_{\rho_k,q}\theta_{\rho_k,q})^{-0.29} - 0.32 & \text{if} \theta_{\rho_k,q} \neq 0\ , \\ 1 & \text{otherwise}\ . \end{cases} \end{aligned} \tag{3.2}$$

As per the feature noise model, the q^{th} component of the measurement noise process $v_{\rho_k,k}$ is given by

$$v_{\rho_k,k,q} = \begin{cases} \nu_{\rho_k,k,q} & \text{if } q \in \mathcal{N}(\rho_k) \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Here, $\nu_{\rho_k,k,q}$ is an independent and identically distributed zero-mean Gaussian noise process with standard deviation $\sigma(\theta_{\rho_k,q}, d_{\rho_k,q})$.

Observer: An observer generates the minimum mean squared error estimate $\hat{x}_k := \mathbf{E}[x_k | \boldsymbol{y}_0^k, \boldsymbol{\rho}_0^k]$, perhaps using a Kalman filter.

Controller: The controller generates a control signal $u_k \in \mathbb{U}$, where $\mathbb{U} := \mathcal{R} \bigcup \{`present', `absent'\}$. The control signal chooses to saccade to a new fixation in $\mathcal{R}$ or detect the absence or presence of a target element in the visual array. When the controller chooses to saccade, the new fixation is given by $\rho_{k+1} = u_k$ for $k = -1, 0, \dots, N-1$.

In general, any admissible control policy, such as $u_k = \gamma_k(\boldsymbol{y}_0^k, \boldsymbol{u}_0^{k-1})$, can be used. Accordingly, for $k = -1$, the control signal will be chosen based on a priori information alone. However, on grounds of simplicity, the control policies may also be restricted to use the current estimate $\hat{x}_k$ alone.

Control Cost: The controller is typically chosen to minimize a cost

$$J = \sum_{k=0}^{N-1} \mathbf{E}[c_k(x_k, u_k)] + \mathbf{E}[c_N(x_N)] \ , \tag{3.3}$$

where $c_k(x_k, u_k)$ is the cost incurred at time k.

Problem of Interest: Assume that the controller is restricted to use the estimate $\hat{x}_k$ in place of the entire measurement and control history. Then, we wish to explore the following questions:

1. Given κ instances of the data pair $\{\hat{x}_k, u_k\}$, corresponding to κ observations of the inputs and outputs of an expert decision maker, can we impute a cost function $\hat{J}_\kappa$ for the above problem?

2. Can we provide a performance guarantee for a controller $\gamma^*(\hat{J}_\kappa)$, designed to minimize the imputed cost, in comparison to the expert decision maker?

3. Can we impute a cost function for various visual display aids, and comment on different designs for the graphical user interface in terms of fraud detection performance?

Brief Summary of Inverse Parametric Optimization

The problem described in (3.1)–(3.3) can be posed as an optimization problem. To do so, let us defined the probability space $(\Omega, \mathcal{F}, \mathbf{P})$ for the primitive random process $v_{\rho_k,k}$ driving our system. Then, we have

$$\begin{aligned} \{\gamma_0, \dots, \gamma_{N-1}\} \in \ & \underset{u_{-1}, u_0, \dots, u_{N-1}}{\arg\min} \sum_{k=0}^{N-1} \mathbf{E}[c_k(x_k, u_k)] + \mathbf{E}[c_N(x_N)] \\ \text{such that} \quad & x_{k+1} = x_k \ , \\ & y_k = C_{u_{k-1}} x_k + v_{u_{k-1},k} \ , \\ & u_k = \gamma_k(\boldsymbol{y}_0^k, \boldsymbol{u}_0^{k-1}) \ , \\ & \gamma_k \in \mathcal{U}(\Omega, \mathcal{F}_k^y, \mathbf{P}) \ , \qquad k = 0, \dots, N-1 \ , \end{aligned}$$

where $\mathcal{U}(\Omega, \mathcal{F}_k^y, \mathbf{P})$ is the space of $\mathbb{U}$-valued functions that are measurable with respect to $\mathcal{F}_k^y := \sigma(y_0, \ldots, y_k)$, the σ-algebra generated by the history of the observation process up to time k. The above formulation represents our aim of selecting causal control policies that minimize the cost in (3.3).

In general, we obtain a parametric optimization problem, defined as one in which the problem data, and consequently the solution, varies with respect to some known parameter. For such problems, one wishes to identify some function $f : \mathbb{R}^m \to \mathbb{R}^n$ such that

$$\begin{aligned} f(p) \in \arg\min_{z} \, & J(z,p) \\ \text{such that } & (z,p) \in \Gamma \ , \end{aligned} \tag{3.4}$$

where $z \in \mathbb{R}^n$ is the decision variable, $p \in \mathbb{R}^m$ is the parameter, $J : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the cost function and $\Gamma \in \mathbb{R}^n \times \mathbb{R}^m$ is the constraint set. The solution function f maps each parameter to an optimal solution for that parameter, and is to be identified. A overview of methods for constructing a solution function f for several common problem classes can be found in Pistikopoulos et al. (2007).

3.1.2 Methodology

The problems considered by SPEEDD are typically subject to significant noise or uncertainty. Therefore, we extend the previously derived formulation to allow for a robust approach in the sense of Task 4.2. Consider the inverse optimization problem in which, instead of a "best fit", we seek to minimize the conditional value at risk (CVar) at level α given as

$$\text{CVar}_\alpha(l_\theta) = \inf_{\tau} \ \tau + \frac{1}{\alpha}\mathbb{E}^{\mathbb{P}}\big[\max\{l_\theta(s,x) - \tau, 0\}\big]$$

for some loss function $l_\theta(s,x)$ describing the quality of fit given some prior probability measure $\mathbb{P}$. The prior distribution $\mathbb{P}$ is not known and therefore, it has to be estimated. We assume that a set of N independent data samples $\hat{\xi}_i = (\hat{s}_i, \hat{x}_i)$ is available. The sample distribution is then given as

$$\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i}$$

We attempt to obtain a solution to the inverse optimization problem that is robust to changes in the (unknown) prior distribution. More precisely, we consider the set of all distribution within a ball

$$\hat{\mathcal{P}} := \mathbb{B}_\epsilon^p(\hat{\mathbb{P}}_N) = \big\{\mathbb{Q} : W_p(\mathbb{Q}, \hat{\mathbb{P}}_N) \leq \epsilon\big\}$$

of radius ϵ centered around the sample distribution $\hat{\mathbb{P}}$, defined by the p-Wasserstein metric. We encode the preference for a robust solution in the loss function

$$l_\theta(s,x) := F_\theta(s,x) - \min_{y \in \mathbb{X}(s)} F_\theta(s,y)$$

which can be interpreted as (minimizing) the suboptimality of the chosen parameters x in comparison to the best parameters y for the parameter dependent, certainty-equivalent objective $F_\theta(s,x)$. Putting all the pieces together, we define the robust inverse optimization problem

$$\underset{\theta \in \Theta}{\text{minimize}} \ \sup_{\mathbb{Q} \in \hat{\mathcal{P}}} \inf_{\tau} \ \tau + \frac{1}{\alpha}\mathbb{E}^{\mathbb{P}}\big[\max\{F_\theta(s,x) - \min_{y \in \mathbb{X}(s)} F_\theta(s,y) - \tau, 0\}\big] \tag{3.5}$$

In Esfahani et al. (2015), the problem given by the previous equation is considered for linear, piecewise-affine and quadratic parametric objective functions $F_\theta(s,x)$ and affine constraints. The following theorem summarizes the main result:

Theorem 3.1.1 (Theorem 5.3, Esfahani et al. (2015))**.** *(Safe Conic Approximation of 3.5 for Quadratic Candidate Objective). Consider the set of quadratic objective functions given as $F_\theta(s,x) = x^\top Q_{xx} x + x^\top Q_{xs} s + x^\top q$ with $\theta = (Q_{xx}, Q_{xs}, q), Q_{xx} \preceq \mathbb{I}_n$, and affine constraints $\mathbb{X}(s) = \{(x,s) : Wx \geq Hs + h\}$. Then, the following conic program provides a safe approximation for the distributionally robust inverse optimization problem 3.5 over the 2-Wasserstein ball:*

$$\begin{array}{lll}
\textit{minimize} & \tau + \dfrac{1}{\alpha}\left(\epsilon^2\lambda + \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} s_i\right) & \\
\textit{subject to} & \theta \in \Theta, \lambda \in \mathbb{R}_+, \tau, r_i, \rho_{i1}, \rho_{i2} \in \mathbb{R}, \phi_{i1}, \phi_{i2} \in \mathcal{C}^*, \mu_{i1}, \mu_{i2}, \gamma_i \in \mathcal{K}^* & \forall i \leq N \\
& \chi_{i1}, \chi_{i2} \in \mathbb{R}^m, \zeta_{i1}, \eta_{i1}, \zeta_{i2} \in \mathbb{R}^n & \forall i \leq N \\
& \chi_{i1} = \dfrac{1}{2}\left(-C^\top \phi_{i1} + H^\top(\mu_{i1} + \gamma_{i1}) - 2\lambda \hat{s}_i\right) & \forall i \leq N \\
& \zeta_{i1} = \dfrac{1}{2}(-q - W^\top \mu_{i1} - 2\lambda \hat{x}_i),\ \eta_{i1} = \dfrac{1}{2}(q - W^\top \gamma_{i1}) & \forall i \leq N \\
& \rho_{i1} = \tau + r_i + \lambda(\hat{x}_i^\top \hat{x}_i + \hat{s}_i^\top \hat{s}_i) + d^\top \phi_{i1} + h^\top(\mu_i + \gamma_i) & \forall i \leq N \\
& \chi_{i2} = \dfrac{1}{2}\left(-C^\top \phi_{i2} + H^\top \mu_{i2} - 2\lambda \hat{s}_i\right) & \forall i \leq N \\
& \zeta_{i2} = \dfrac{1}{2}(-W^\top \mu_{i2} - 2\lambda \hat{x}_i),\ \eta_{i2} = -\dfrac{1}{2} W^\top \gamma_{i2} & \forall i \leq N \\
& \rho_{i2} = r_i + \lambda(\hat{x}_i^\top \hat{x}_i + \hat{s}_i^\top \hat{s}_i) + d^\top \phi_{i2} + h^\top \mu_i & \forall i \leq N \\
& \begin{pmatrix} \lambda\mathbb{I} & -\frac{1}{2}Q_{xs}^\top & \frac{1}{2}Q_{xs}^\top & \chi_{i1} \\ -\frac{1}{2}Q_{xs} & \lambda\mathbb{I} - Q_{xx} & 0 & \zeta_{i1} \\ \frac{1}{2}Q_{xs} & 0 & Q_{xx} & \eta_{i1} \\ \chi_{i1}^\top & \zeta_{i1}^\top & \eta_{i1}^\top & \rho_{i1} \end{pmatrix} \succeq 0, \begin{pmatrix} \lambda\mathbb{I} & 0 & \chi_{i2} \\ 0 & \lambda\mathbb{I} & \zeta_{i2} \\ \chi_{i2}^\top & \zeta_{i2}^\top & \rho_{i2} \end{pmatrix} \succeq 0 & \forall i \leq N.
\end{array}$$

Once we have identified a cost function, the theory of partially observed Markov decision processes provides us with a structural characterization of the optimal controller. We can utilize this to systematically identify controllers that approximate the optimal one. In this section, we briefly outline the important existing results.

A partially observed Markov decision process consists of a state process $s_k \in \mathcal{S}$, an observation process $o_k \in \mathcal{O}$, an action process $a_k \in \mathcal{A}$, for $k = 1, 2, \ldots, N$, and a single decision maker, where:

1. The action at time k is chosen by the decision maker as a function of the entire observation and action history, i.e.,
$$a_k = d_k\left(\boldsymbol{o}_0^k, \boldsymbol{a}_0^{k-1}\right) .$$
2. After the action at time k is taken, the new state and observation are generated according to the transition probability law
$$\mathbf{P}\left(s_{k+1}, o_{k+1} | \boldsymbol{s}_0^k, \boldsymbol{o}_0^k, \boldsymbol{a}_0^k\right) = \mathbf{P}\left(s_{k+1}, o_{k+1} | s_k, a_k\right) .$$
3. At each time instant, an instantaneous cost $c_k(s_k, a_k)$ is incurred.
4. The optimization problem for the decision maker is to choose a decision strategy $\boldsymbol{d} := \{d_0, \ldots, d_N\}$ to minimize the total cost
$$J = \sum_{k=1}^{N} \mathbf{E}\left[c_k(s_k, a_k)\right]$$

The following well known results provide the structure of optimal strategies and a dynamic program for POMDPs (Whittle, 1982).

Theorem 3.1.2. *Let θ_k be the conditional probability distribution of the state s_k, at time k, given the observation and action history, as given by*

$$\theta_k(s) = \mathbf{P}(s_k = s | \boldsymbol{o}_0^k, \boldsymbol{a}_0^{k-1}),\ s \in \mathcal{S}\ .$$

Then:

1. *$\theta_{k+1} = \eta_k(\theta_k, a_k, o_{k+1})$, where η_k is the standard non-linear Bayes' filter. Thus, we have*

$$\theta_{k+1}(s) = \frac{\sum_{s'} \theta_k(s')\, \mathbf{P}(s, o_{k+1}|s', a_k)}{\sum_{\hat{s},\tilde{s}} \theta_k(\hat{s})\, \mathbf{P}(\tilde{s}, o_{k+1}|\hat{s}, a_k)}\ .$$

2. *There exists an optimal strategy of the form $a_k = \hat{d}_k(\theta_k)$, which can be found using dynamic programming.*

3.1.3 Results

We present result for two different test cases. First, we consider a problem motivated by Linear Quadratic Control, in which a receding horizon controller is implicitly defined as the solution to a finite horizon optimal control problem parametrized by the system state. The cost function in this case reads

$$J = \sum_{k=0}^{M-1} x_k^T Q x_k + u_k^T R u_k + x_M^T S x_M,$$

where S is the infinite horizon cost. This solution is exact under constraints on u_k and x_k as long as the horizon M is long enough, such that all constraints remain inactive after time step M. The problem can be solved using the batch approach, in which it is converted to the system

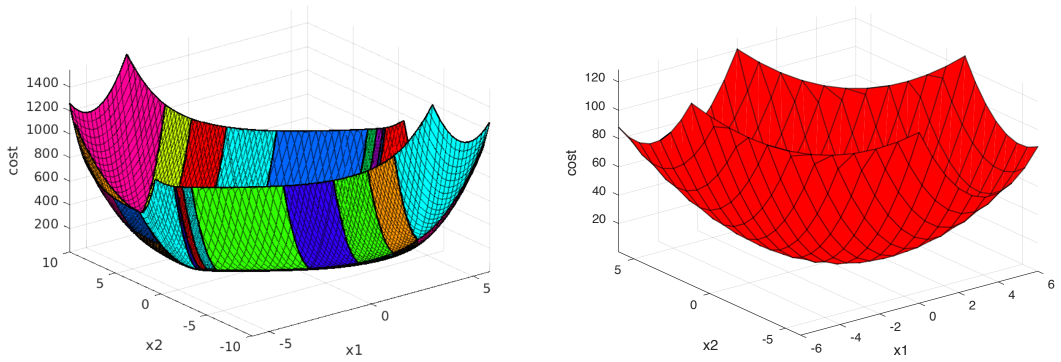
$$\underbrace{\begin{bmatrix} x(0) \\ x_1 \\ \vdots \\ \vdots \\ x_M \end{bmatrix}}_{\mathcal{X}} = \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ \vdots \\ A^M \end{bmatrix}}_{\mathcal{S}^x} x(0) + \underbrace{\begin{bmatrix} 0 & \dots & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A^{M-1}B & \dots & \dots & B \end{bmatrix}}_{\mathcal{S}^u} \underbrace{\begin{bmatrix} u_0 \\ \vdots \\ \vdots \\ u_M \end{bmatrix}}_{U_0}$$

With this system, the cost function is

$$J(x(0), U_0) = \mathcal{X}^T \bar{Q} \mathcal{X} + U_0^T \bar{R} U_0,$$

where $\bar{Q} = \text{blockdiag}\{Q, \dots, Q, S\}, \bar{Q} \succeq 0$ and $\bar{R} = \text{blockdiag}\{R, \dots, R\}, \bar{R} \succ 0$. Now, the optimal inputs U_0 can be determined by solving the quadratic optimization problem

$$\begin{aligned} \text{minimize} \quad & \mathcal{X}^T \bar{Q} \mathcal{X} + U_0^T \bar{R} U_0 \\ \text{subject to} \quad & \mathcal{X} = \mathcal{S}^x x(0) + \mathcal{S}^u U_0 \\ & \bar{C} U_0 \succeq \bar{d} \\ & \bar{W} \mathcal{X} \succeq \bar{h}, \end{aligned}$$



(a) Real, piece-wise quadratic cost.

(b) Identified cost.

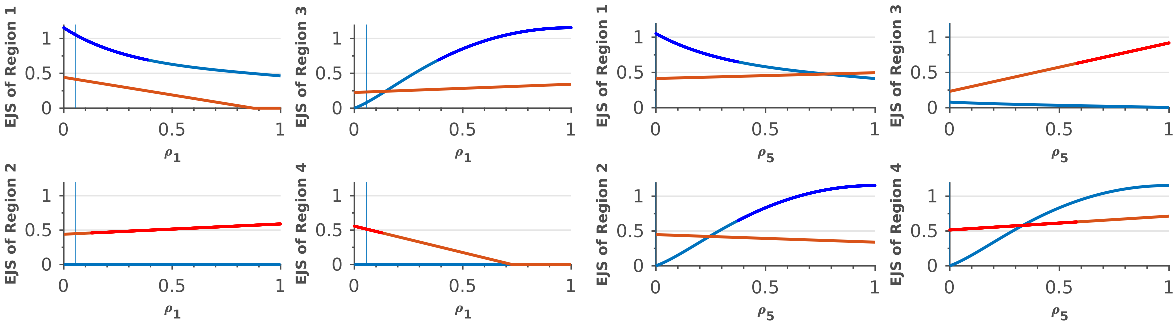
Figure 3.2: Example of real and identified costs based on 20 samples for a small LQR-type problem. The real cost function is piecewise-quadratic, but convex and can reasonably well be approximated by another convex, piecewise quadratic function defined on a smaller set of pieces, as provided by the inverse optimization problem.

where $\bar{C}$, $\bar{W}$, $\bar{d}$ and $\bar{h}$ are blockdiagonal matrices similar to $\bar{R}$. In Figure 3.2, we compare the exact solution of a small problem with two states and one input to the controller obtained by inverse optimization based on a small set of data samples. Note that we do only formulate a one-step inverse optimization problem, that is, the true objective function is *not* contained in the set of candidate objective that the inverse optimization problem considers. Nevertheless, the robust approach allows to find a close approximation.

Second, we study a visual search problem as outlined in the previous section, but we restrict ourself to the simplifications made in Naghshvar and Javidi (2013). Most importantly, the general detection problem in which a target might or might not be present is simplified to a version where we assume that some target is always present and we seek to locate it in one of finitely many observation regions. The signal s is now the prior distribution of the probability of the target being located in a particular region and the decision x is the decision on which region to observe in a next step. A key distinction to the aforementioned problem is that now, the parametric optimization problem is no longer convex, let alone quadratic. An immediate consequence is that a convex inverse optimization problem is no longer able to provide a close estimate of the objective function. Since the problem is now high-dimensional, it is not easy to visualize the objective function as a whole as in Figure 3.2 for the LQ-problem, but we can depict low-dimensional projections of the ground-truth optimizer in comparison to the solution of the optimization problem using the identified cost. Figure 3.3 depicts the results. It can be seen that the identified optimizer does not correspond to the true optimizer at all. This seems to be due to the non-convex nature of the optimization problem. If a parametrix optimization problem is not strictly convex, than the optimizer may be discontinuous in the parameters, essentially making the robust inverse optimization intractable.

The results can be slightly improved if the feature space is expanded. We consider the same setup as used to create Figure 3.3, but add nonlinear features to the original ones, the prior distribution. The results are depicted in Figure 3.4. The quality of fit is improved significantly.

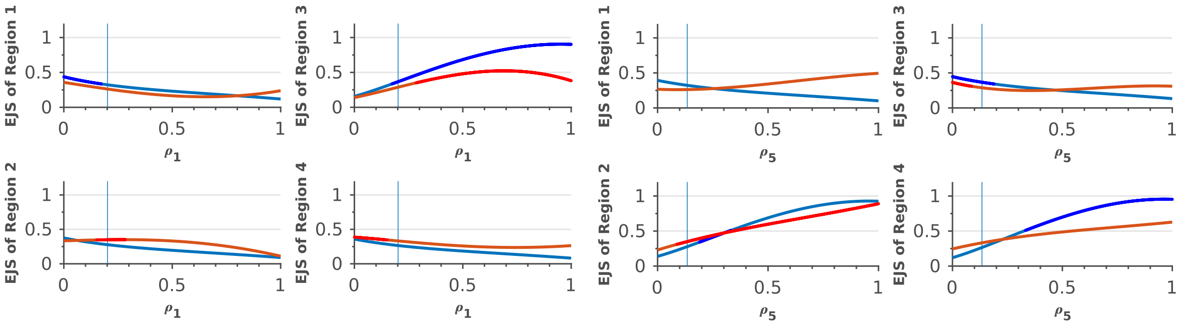
To evaluate the potential of the chosen approach, we also conducted a systematic comparison of the inverse optimization method with a neural network. We consider the visual search problem over a finite number of regions as described and use either inverse optimization or a neural network to learn the problem based on a set of data pairs. The training data are either just the prior distribution or they are also enhance with nonlinear features. Furthermore, the training targets are either given as continuous



(a) Changes in the true (blue) and identified (red) optimal actions for all observation regions if the prior is shifted towards the boundary.

(b) Changes in the true (blue) and identified (red) optimal actions for all observation regions if the prior is shifted towards the center.

Figure 3.3: Projections of the optimizer x^*, depending on one-dimensional changes to the prior, i.e., the signal s. The identified optimizer does not correspond to the true optimizer at all.



(a) Changes in the true (blue) and identified (red) optimal actions for all observation regions if the prior is shifted towards the boundary.

(b) Changes in the true (blue) and identified (red) optimal actions for all observation regions if the prior is shifted towards the center.

Figure 3.4: Projections of the optimizer x^*, based on parameters that include artificially created, nonlinear features. The quality of fit is significantly improved in comparison to Figure 3.3.

	Only prior.		Prior + NL features	
	Targets $\in [0,1]^n$	Targets $\in \{0,1\}^n$	Targets $\in [0,1]^n$	Targets $\in \{0,1\}^n$
Inverse Optimization	0.458	0.400	0.613	0.385
Neural Network	0.356	0.356	0.432	0.321

Table 3.1: Probability of correct classification.

values according to Naghshvar and Javidi (2013) (Which uses a method based on the Extrinsic Jenssen-Shannon Divergence as an approximation to the exact, unknown optimal policy) or as zero-one data which only indicate the chosen observation region. The probabilities for correct classification for 20 data pairs as training samples are reported in Table 3.1. Such a small data set was chosen on purpose, as inverse optimization is most suitable for situations in which only limited amounts of data are available. It can be seen that for such a small data set, inverse optimization outperforms the neural network. However, the probability of correct classification is poor and except for the unrealistic case, when both continuous training targets and nonlinear features are made available, it is not good enough for further usage. If the amount of training data is increased, one can unsurprisingly notice a higher probability of correct classification, but the neural network also easily outperforms inverse optimization.

3.1.4 Discussion

We have defined a setup in which inverse optimization is applied to a problem motivated by visual search, in order to identify the decision maker's cost function. A solution method has been proposed and it has been evaluated in a case study. It turns out that the available methods in inverse optimization are unsuitable for problems as they appear in visual search. This is caused in part by the nonconvexity of these problems. However, even more severe is the fact that the structure of the solution to these problems is not known, except for its Markov property. It seem that most inverse optimization approaches are able to outperform generic, black-box approaches like neural networks if (and only if) they can use the knowledge that the training data have been generated by an optimization problem of a particular structure. The knowledge of the structure of the parametric optimization problem restricts the a-priori search space of suitable models and can be interpreted as additional prior knowledge, which is ultimately responsible for the better performance. This is for example the case for the LQ-control problems. Conversely, visual search problems present no exploitable structure.

Even though we conclude that the inverse optimization approach is unsuitable for visual search type problems, we still believe it might be used in CCF decision support advantageously. However, it might be more suitable for an active learning setting, in which we seek to extract and formalize the prior knowledge that a domain expert has gained through experience. Consider a situation in which an expert of CCF is presented with partial information about a suspicious transaction and is then asked to select additional information that (s)he would consider before deciding if the transaction in question is fraudulent or not. Mathematically speaking, we consider a classification problem in which only parts of the feature vector are revealed and the expert then selects the most expressive features, given the values of the available features, to make the final decision. The question of which features are most expressive is closely related to the location of the decision boundary in the feature space, between fraudulent and non-fraudulent decisions. If the decision boundary is given as the solution of an explicitly defined optimization problem, one can use the decision by the expert to infer the objective of the optimization problem and therefore estimate the decision boundary via inverse optimization. In the next section, we review an approach for distributionally robust classifiers (originally presented in deliverable D4.1), which ultimately reduces to a conic program. The structure of this optimization problem is known, but it does not allow for immediate application of Theorem 3.1.1 because of a constraint which is bilinear in parameters and optimization variables. Currently, methods for extending the inverse optimization approach to such a problem class are investigated in a student project.

3.1.5 Distributionally Robust Classifiers

The problem of credit card fraud detection is a heavily unbalanced classification problem, since only a small minority of transactions are fraudulent. It is well known that unbalanced data pose a challenge for machine learning. Common state of the art solutions include over-/ under-sampling as well as the replacement of individual data points by probability distributions, which approximate the data prior to training. This approach has recently been extended by exploiting results from robust optimization to create distributionally robust classifiers (Van Parys et al. (2014); Stellato (2014)). A brief summary of the derivation of distributionally robust classifiers will be provided in the following.

Problem Formulation

Robust optimization addresses optimization problems, in which at least one of the problem parameters is a priori uncertain and will only be revealed after a decision has been made. In robust optimization, it is assumed that even though we do not know the value of those parameters a priori, we know that they belong to a given set and we seek to optimize the worst case outcome.

Consider the problem to separate samples drawn from two probability distributions by a hyperplane, such that the probability of misclassification is minimized for either side. Assume further that the mean and the covariance of the probability distributions are known, but not the complete probability density function. The problem of minimizing the classification error can then be formulated as a robust optimization problem, in which we find a hyperplane such that we minimize the classification error for the worst-case probability distribution that coincides with the given first and second moments. In practice, it often seems to be unjustified to allow for entirely arbitrary probability density functions. In particular, many probability distributions encountered in practice are unimodal. Intuitively, unimodality of the probability density function means that obtaining samples ”close” to the mean is more likely than obtaining samples ”further away” from the mean.

Let the μ_1 and μ_2 denote the mean of the first and the second probability distribution, and likewise let Σ_1 and Σ_2 denote the respective variances. By $\mathbb{P} \in \mathcal{P}(\mu, \Sigma) \cap \mathcal{P}_\alpha$ we denote a probability distribution $\mathbb{P}$ that has mean μ and variance Σ, and an α-unimodal probability density function[1]. The problem of minimizing the worst-case classification error can be formulated as:

$$\begin{array}{lll} \max\limits_{a,\gamma,b} & \gamma & \\ \text{subject to} & \mathbb{P}(-a^\top x > -b) \leq 1 - \gamma, & \forall\, \mathbb{P} \in (\mu_1, \Sigma_1) \cap \mathcal{P}_\alpha \\ & \mathbb{P}(+a^\top x > +b) \leq 1 - \gamma, & \forall\, \mathbb{P} \in (\mu_2, \Sigma_2) \cap \mathcal{P}_\alpha \end{array} \tag{3.6}$$

Results

The previously introduced optimization problem includes an infinite number of constraints, because of the ”for all” classifier, therefore a straightforward numerical solution is impossible. However, by employing results from Van Parys et al. (2014), Stellato (2014) reformulates the problem as an equivalent,

[1] In Van Parys et al. (2014), α-unimodality is introduced as a generalization of the concept of unimodality.

nonlinear, but finite dimensional problem:

$$\begin{aligned}\min_{a,b} \quad & \omega \\ \text{subject to} \quad & (-b + a^\top \mu_1) \geq \left\|(\Sigma_1)^{\frac{1}{2}} a\right\| \\ & (-b + a^\top \mu_2) \geq \left\|(\Sigma_2)^{\frac{1}{2}} a\right\|\end{aligned}$$

This problem can be reformulated into a finite dimensional Second-order Cone Problem (SOCP):

$$\begin{aligned}\min_{a} \quad & \left\|(\Sigma_1)^{\frac{1}{2}} a\right\| + \left\|(\Sigma_2)^{\frac{1}{2}} a\right\| \\ \text{subject to} \quad & a^\top (\mu_1 - \mu_2) = 1\end{aligned} \tag{3.7}$$

Note in particular that this problem is convex. Numerical solvers for SOCPs exist, so that the resulting optimization problem can be solved (numerically) for the optimizer a^*. The remaining quantities can be computed as

$$\omega^* = \frac{1}{\left\|(\Sigma_1)^{\frac{1}{2}} a^*\right\| + \left\|(\Sigma_2)^{\frac{1}{2}} a^*\right\|}$$

and

$$b^* = a^* \mu_1 - \omega^* \left\|(\Sigma_1)^{\frac{1}{2}} a^*\right\|$$

Note that the objective has been the minimization of the classification error for either probability distribution. A tractable problem formulation for the objective of minimizing the overall misclassification error is also given in Stellato (2014).

We intend to use this problem in an inverse-optimization setting as described in the previous section in order to learn the prior distributions $(\mu_1, \Sigma_1$ and (μ_2, Σ_2) from samples.

4

Conclusions

This report documents the final version of the DM algorithms in the SPEEDD project. An extensive range of control algorithms for the traffic use case and a description of DM functionality in the use-case of CCF are described.

First, we have investigated how Gaussian Processes can be used to learn the traffic dynamics of a freeway from data. A range of tools addressing non-stationary noise, inhomogeneous process variance and time delays in measurement data have been developed to this end. Such GP models allow to incorporate model uncertainty directly into the models used for decision making, making them especially suitable for SPEEDD. To enable usage of such models, we have demonstrated how reinforcement learning techniques based on GPs can be extended to high-dimensional systems such as traffic networks. This extension is possible via usage of SGPs, which model systems that exhibit a distributed structure and can be decomposed into small-dimensional, uncertain subsystems coupled by linear and known dynamics.

Second, we have presented a range of low-level decision making algorithms for traffic control. This includes controllers like the best-effort-controller for freeways, which is provably robust to variations in the traffic demand, and a corresponding, one-step-ahead control scheme suitable to maximize local traffic flows in arbitrary traffic networks, in particular in inner-city road networks.

Third, local feedback laws are complemented by high-level coordination algorithms in the proposed hierarchical control scheme, which are designed to achieve some system-wide control objective. We have presented a generic algorithm for model-free decision making in the presence of disturbances and explored its application to the ramp metering use-case. In more use-case specific approaches, we have analyzed the need for coordination in both the freeway and the inner city traffic control scenario. We have proven that there exist certain situations in freeway ramp metering in which coordination is not necessary to achieve flow optimality, due to the convexity of the underlying optimization problem. However, global problems for optimization of road traffic in arbitrary networks are inherently nonconvex, and we have developed a coordination scheme which allows to approximate the solution to the nonconvex problem via successive linear optimization problems.

A fourth contribution addresses the use case of Credit Card Fraud detection. For CCF, the answer to the question of whether to accept or to block a certain transaction depends on whether a fraud attempt was detected or not. If a fraud attempt is detected with sufficiently high confidence, the decision is obvious. Therefore, DM is reduced to a support role in this use case. Indeed, we have presented a method to formalize and solve the problem of reverse engineering a utility function that a credit card analyst might be unconsciously optimizing. The intention is to use the gained knowledge to streamline the decision making process for the analyst in automatically providing the right information at the right time.

Bibliography

M. Alamir. *Collaborative Ramp Metering Control: Application to Grenoble South Ring*. PhD thesis, Lund University, Lund, Sweden, 2014.

R. E. Allsop. Transport networks and their use: how real can modelling get? *Phil. Trans. R. Soc. A*, 366:1879 – 1892, 2008.

D. Angeli and E. D. Sontag. Monotone control systems. *IEEE Transactions on automatic control*, 48 (10):1684–1698, 2003.

D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE transactions on automatic control*, 57(7):1615–1626, 2012.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

M. Burger, M. Van Den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. Considerations for model-based traffic control. *Transportation Research Part C: Emerging Technologies*, 35:1–19, 2013.

C. I. Chen, J. B. C. Jr, and J. G. Paquet. Entrance ramp control for travel-rate maximization in expressways. *Transportation Research*, 8(6):503–508, 1974.

M. Colombino and R. Smith. A convex characterization of robust stability for positive and positively dominated linear systems. 2015.

M. Colombino and R. S. Smith. Convex characterization of robust stability analysis and control synthesis for positive linear systems. In *53rd IEEE Conference on Decision and Control*, pages 4379–4384. IEEE, 2014.

C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994a.

C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994b.

C. F. Daganzo. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

L. B. de Oliveira and E. Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120 – 139, 2010.

C. C. de Wit, F. Morbidi, L. L. Ojeda, A. Y. Kibangou, I. Bellicot, and P. Bellemain. Grenoble traffic lab: An experimental platform for advanced traffic monitoring and forecasting. *Control Systems, IEEE*, 35(3):23–39, 2015.

M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

G. Dirr, H. Ito, A. Rantzer, and B. Rüffer. Separable lyapunov functions for monotone systems: constructions and limitations. *Discrete Contin. Dyn. Syst. Ser. B*, 2015.

W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549 – 558, 2006.

M. Ellis, H. Durand, and P. D. Christofides. A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24(8):1156–1178, 2014.

P. M. Esfahani, S. Shafieezadeh-Abadeh, G. A. Hanasusanto, and D. Kuhn. Data-driven inverse optimization with incomplete information. *arXiv preprint arXiv:1512.05489*, 2015.

N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou. A multi band approach to the arterial traffic signal optimization. *Transportation Research B*, 25(1):55–74, 1991.

R. Gibbens and Y. Saatci. Data, modelling and inference in road traffic networks. *Phil. Trans. R. Soc. A*, 366:1907–1919, 2008.

G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006a.

G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006b.

G. Gomes, R. Horowitz, A. A. Kurzhanskiy, P. Varaiya, and J. Kwon. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies*, 16(4):485–513, 2008.

P. Grandinetti, C. Canudas-de Wit, and F. Garin. An efficient one step ahead optimal control for urban signalized traffic networks based on an averaged cell transmission model. In *Proceedings of the 14th IEEE European Control Conference*, Linz, Austria, 2015a.

P. Grandinetti, C. C. de Wit, and F. Garin. An efficient one-step-ahead optimal control for urban signalized traffic networks based on an averaged cell-transmission model. In *Control Conference (ECC), 2015 European*, pages 3478–3483. IEEE, 2015b.

P. Grandinetti, F. Garin, and C. Canudas-de Wit. Towards scalable optimal traffic control. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2175–2180. IEEE, 2015c.

L. Grüne and M. Zanon. Periodic optimal control, dissipativity and mpc. In *MTNS 2014*, pages 1804–1807, 2014.

I. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL http://www.gurobi.com.

X. He and L. Zhang. Economic impact of highway investment at the metropolitan level: Empirical analysis with consideration for induced demand and induces supply. *Transportation Research Board Annual Meeting*, 2015.

A. Hegyi, B. D. Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):102 – 112, 2005.

M. Hirsch and H. Smith. Monotone maps: a review. *Journal of Difference Equations and Applications*, 11(4-5):379–398, 2005.

M. Hirsch, H. Smith, et al. Monotone dynamical systems. *Handbook of differential equations: ordinary differential equations*, 2:239–357, 2005.

H. Ito, B. S. Rüffer, and A. Rantzer. Max-and sum-separable lyapunov functions for monotone systems and their level sets. In *53rd IEEE Conference on Decision and Control*, pages 2371–2377. IEEE, 2014.

I. Karafyllis, M. Kontorinaki, and M. Papageorgiou. Global exponential stabilization of freeway models. *arXiv preprint arXiv:1408.5833*, 2014.

I. Karafyllis, M. Kontorinaki, and M. Papageorgiou. Global exponential stabilization of freeway models. *International Journal of Robust and Nonlinear Control*, 26(6):1184–1210, 2016. ISSN 1099-1239. doi: 10.1002/rnc.3412. URL http://dx.doi.org/10.1002/rnc.3412. rnc.3412.

J. Lebacque. The godunov scheme and what it means for first-order traffic models. *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 647–677, 1996.

M. J. Lighthill and G. B. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 229, pages 317–345. The Royal Society, 1955.

J. Little. The synchronization of traffic signals by mixed integer linear programming. *Operation Research*, 14(4):568–594, 1966.

J. Lofberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

E. Lovisari, G. Como, A. Rantzer, and K. Savla. Stability analysis and control synthesis for dynamical transportation networks. *arXiv preprint arXiv:1410.5956*, 2014.

J. R. Marden, H. P. Young, and L. Y. Pao. Achieving pareto optimality through distributed learning. *SIAM Journal on Control and Optimization*, 52(5):2753–2770, 2014. doi: 10.1137/110850694. URL http://dx.doi.org/10.1137/110850694.

A. Menon and J. S. Baras. Convergence guarantees for a decentralized algorithm achieving pareto optimality. In *Proceedings of the 2013 American Control Conference*, pages 1935–1940, 2013a.

A. Menon and J. S. Baras. A distributed learning algorithm with bit-valued communications for multi-agent welfare optimization. In *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, pages 2406–2411, 2013b.

A. Messner and M. Papageorgiou. Metanet: A macroscopic simulation program for motorway networks. *Traffic Engineering & Control*, 31(8-9):466–470, 1990.

F. Morbidi, L. L. Ojeda, C. C. De Wit, and I. Bellicot. A new robust approach for highway traffic density estimation. In *Control Conference (ECC), 2014 European*, pages 2575–2580. IEEE, 2014.

M. A. Müller, D. Angeli, and F. Allgöwer. On necessity and robustness of dissipativity in economic model predictive control. *IEEE Transactions on Automatic Control*, 60(6):1671–1676, 2015a.

M. A. Müller, L. Grüne, and F. Allgöwer. On the role of dissipativity in economic model predictive control. *IFAC-PapersOnLine*, 48(23):110–116, 2015b.

M. Naghshvar and T. Javidi. Two-dimensional visual search. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 1262–1266. IEEE, 2013.

J. W. Ng and M. P. Deisenroth. Hierarchical mixture-of-experts model for large-scale gaussian process regression. *arXiv preprint arXiv:1412.3078*, 2014.

L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit. Adaptive kalman filtering for multi-step ahead traffic flow prediction. In *IEEE American Control Conference (ACC)*, pages 4724–4729, 2013.

M. Papageorgiou and A. Kotsialos. Freeway ramp metering: An overview. In *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, pages 228–239. IEEE, 2000.

M. Papageorgiou, H. Hadj-Salem, and J.-M. Blosseville. Alinea: A local feedback control law for on-ramp metering. *Transportation Research Record*, (1320):58–64, 1991.

M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

I. Papamichail and M. Papageorgiou. Traffic-responsive linked ramp-metering control. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):111–121, 2008.

I. Papamichail, A. Kotsialos, I. Margonis, and M. Papageorgiou. Coordinated ramp metering for freeway networks–a model-predictive hierarchical control approach. *Transportation Research Part C: Emerging Technologies*, 18(3):311–331, 2010a.

I. Papamichail, M. Papageorgiou, V. Vong, and J. Gaffney. Heuristic ramp-metering coordination strategy implemented at monash freeway, australia. *Transportation Research Record: Journal of the Transportation Research Board*, 2178(1):10–20, 2010b.

D. Pisarski and C. Canudas-de Wit. Analysis and design of equilibrium points for the cell-transmission traffic model. In *American Control Conference (ACC), 2012*, pages 5763–5768. IEEE, 2012a.

D. Pisarski and C. Canudas-de Wit. Optimal balancing of road traffic density distributions for the cell transmission model. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 6969–6974. IEEE, 2012b.

D. Pisarski and C. Canudas-de Wit. Optimal balancing of freeway traffic density: Application to the grenoble south ring. In *Control Conference (ECC), 2013 European*, pages 4021–4026. IEEE, 2013.

E. N. Pistikopoulos, M. C. Georgiadis, and V. Dua, editors. *Multi-Parametric Programming: Theory, Algorithms and Applications, Volume 1*. Wiley-VCH Verlag GmbH & Co. KGaA, 2007.

B. S. Pradelski and H. P. Young. Learning efficient nash equilibria in distributed systems. *Games and Economic Behavior*, 75(2):882 – 897, 2012.

C. Ramesh, M. Schmitt, and J. Lygeros. Distributed learning in the presence of disturbances. Submitted to the European Control Conference (2016), October 2015.

A. Rantzer. Scalable control of positive systems. *European Journal of Control*, 24:72–80, 2015.

A. Rantzer and B. Bernhardsson. Control of convex-monotone systems. In *CDC*, pages 2378–2383, 2014.

A. Rantzer, B. S. Rüffer, and G. Dirr. Separable lyapunov functions for monotone systems. In *52nd IEEE Conference on Decision and Control*, pages 4590–4594. IEEE, 2013.

C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

P. I. Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.

D. Robertson. Transyt method for area traffic control. *Traffic Eng. Control*, 10:276–281, 1969.

K. Savla, E. Lovisari, and G. Como. On maximally stabilizing traffic signal control with unknown turn ratios. 19(1):1849–1854, 2014.

M. Schmitt, P. Goulart, A. Georghiou, and J. Lygeros. Flow-maximizing equilibria of the cell transmission model. In *European Control Conference*, 2015.

M. Schmitt, C. Ramesh, P. Goulart, and J. Lygeros. Convex, monotone systems are optimally operated at steady-state. In *submitted to American Control Conference 2016*, 2016a.

M. Schmitt, C. Ramesh, and J. Lygeros. Sufficient optimality conditions for distributed, non-predictive ramp metering in the monotonic cell transmission model. submitted to Journal of Transportation Research, Part B: Methodological., 2016b.

E. Smaragdis, M. Papageorgiou, and E. Kosmatopoulos. A flow-maximizing adaptive local ramp metering strategy. *Transportation Research Part B: Methodological*, 38(3):251–270, 2004.

B. Stellato. Data-driven chance constrained optimization, 2014. URL http://e-collection.library.ethz.ch/view/eth:46900.

Y. Stephanedes. Implementation of on-line zone control strategies for optimal ramp metering in the minneapolis ring road. 1994.

M. Sweet. Traffic congestion's economic impacts: Evidence from u.s. metropolitan regions. *Transportation Research Board Annual Meeting*, 2015.

B. P. Van Parys, P. J. Goulart, and D. Kuhn. Generalized gauss inequalities via semidefinite programming. Technical report, 2014.

P. Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013.

Y. Wang, E. B. Kosmatopoulos, M. Papageorgiou, and I. Papamichail. Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2024–2039, 2014.

J. A. Wattleworth. Peak-period analysis and control of a freeway system. Technical report, Texas Transportation Institute, 1965.

P. Whittle. *Optimization over Time*. John Wiley & Sons, Inc., New York, NY, USA, 1982. ISBN 0471101206.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

H. P. Young. The evolution of conventions. *Econometrica*, 61(1):pp. 57–84, 1993. ISSN 00129682. URL `http://www.jstor.org/stable/2951778`.

H. P. Young. Learning by trial and error. *Games and Economic Behavior*, 65(2):626–643, March 2009.

H. M. Zhang and S. G. Ritchie. Freeway ramp metering using artificial neural networks. *Transportation Research Part C: Emerging Technologies*, 5(5):273–286, 1997.

L. Zhang and D. Levinson. Optimal freeway ramp control without origin–destination information. *Transportation Research Part B: Methodological*, 38(10):869–887, 2004.

A. K. Ziliaskopoulos. A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation science*, 34(1):37–49, 2000.