L-Università ta' Malta
Faculty of Engineering

DOCTOR OF PHILOSOPHY DISSERTATION

Towards Autonomic Control of Urban Traffic Junctions

ING. LUANA CHETCUTI ZAMMIT

Supervised by:

PROFESSOR SIMON FABRI

Co-supervised by:

PROFESSOR MARIA ATTARD

A dissertation submitted in partial fulfilment of the requirements for the Doctor of Philosophy

by the

Faculty of Engineering

September 2019

L-Università ta' Malta

FACULTY/INSTITUTE/CENTRE/SCHOOL OF ENGINEERING

DECLARATIONS BY POSTGRADUATE STUDENTS

Student's I.D. /Code 47788 M

Student's Name & Surname LUANA CHETCUTI ZAMMIT

Course DOCTOR OF PHILOSOPHY

Title of Dissertation

TOWARDS AUTONOMIC CONTROL OF URBAN TRAFFIC JUNCTIONS

(a) Authenticity of Dissertation

I hereby declare that I am the legitimate author of this Dissertation and that it is my original work.

No portion of this work has been submitted in support of an application for another degree or qualification of this or any other university or institution of higher education.

I hold the University of Malta harmless against any third party claims with regard to copyright violation, breach of confidentiality, defamation and any other third party right infringement.

(b) Research Code of Practice and Ethics Review Procedures

I declare that I have abided by the University's Research Ethics Review Procedures.

As a Master's student, as per Regulation 58 of the General Regulations for University Postgraduate Awards, I accept that should my dissertation be awarded a Grade A, it will be made publicly available on the University of Malta Institutional Repository.

L Chetcuti Z

Signature of Student

LUANA CHETCUTI ZAMMIT

Name of Student (in Caps)

09/09/2019

Date

08.02.2018

L-Università ta' Malta
Faculty of Engineering

Copyright Notice

1) Copyright in text of this dissertation rests with the Author. Copies (by any process) either in full, or of extracts may be made only in accordance with regulations held by the Library of the University of Malta. Details may be obtained from the Librarian. This page must form part of any such copies made. Further copies (by any process) made in accordance with such instructions may not be made without the permission (in writing) of the Author.

2) Ownership of the right over any original intellectual property which may be contained in or derived from this dissertation is vested in the University of Malta and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

3) Publication rights over the academic and/or research results presented in this dissertation are vested jointly in both the Author and his/her academic Supervisor(s), and unless such rights are explicitly waived in writing, both parties must be listed among the authors in any academic publication that is derived substantially from this work. Furthermore, any other public communication / disclosure of any form that focuses on the project must acknowledge that this work has been carried out by the Author and the Supervisor(s) (named explicitly) through the University of Malta.

Preface

During the course of research leading to this thesis, the following peer-reviewed articles have been published so far:

- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Joint state and parameter estimation for a macro traffic junction model. Proceedings of the 24th Mediterranean Conference on Control and Automation, Greece, June 21-24, 2016.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Simultaneous traffic flow and macro model estimation for signalized junctions with multiple input lanes. Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems, Portugal, April 22-24, 2017.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Online state and parameter estimation for a 4-arm junction. Proceedings of the 25th Mediterranean Conference on Control and Automation, Malta, July 3-6, 2017.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Online State and Multi-dimensional Parameter Estimation for a Macroscopic Model of a Traffic Junction. Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems, Japan, October 16-19, 2017.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Real-Time Parametric Modelling and Estimation of Urban Traffic Junctions. IEEE Transactions on Intelligent Transportation Systems, 99:1-11, 2019.

Abstract

As traffic demands are increasing and reaching critical levels worldwide, advanced traffic signal management is becoming an important requirement. The evolution and generation of traffic signal control concepts integrated with advances in control, communications and computational technologies provide intelligent control of traffic lights that adapt themselves to meet the time-varying traffic demands or to changing road conditions.

Despite these recent advances, current systems can become suboptimal because the controller parameters of such systems are not tuned to changing traffic behaviour. Hence such systems can fail when networks are subject to major unanticipated irregularities, such as roadworks, accidents and extreme weather conditions. An adaptive system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is required. The need to design self-managing systems, which self-handle the complexity and uncertainties and thus reduce human intervention to a minimum is of utmost importance. Autonomic systems can self-handle these complexities by modelling the network behaviour and adapting to the changes as required. Hence this work is directed towards the development of autonomic systems for urban signalized junctions.

The aims of this research are: i) to model in real-time the traffic dynamics within signalized junctions, with little prior knowledge of the underlying traffic parameters. In literature, the model parameters are typically assumed known apriori from past traffic measurements or by applying nonlinear recursive estimation algorithms. However, difficulties with nonlinear estimation algorithms such as divergence issues, motivated the development of joint state and parameter estimation algorithms from a different perspective, by developing novel variations on the expectation-maximization algorithm. A quasi real-time joint estimation of model states, parameters and noise is first proposed, whereby the standard expectation-maximization algorithm is modified to carry out estimation in quasi real-time, making use of a short uniform window of sensor measurements of fixed time length, which looks back in time. Full real-time algorithms are developed to reflect changing traffic conditions making use of Robbins-Monro

stochastic approximation posed first as a single variant estimation algorithm, followed by a multivariate estimation algorithm. These proposed algorithms require partial derivative of the likelihood function with respect to each parameter to be worked out analytically. In practice, this approach is impractical for larger and complex junctions due to the complexity of the derivatives involved in deriving such equations. Hence a derivative-free approach is proposed, allowing for easier generalisation to other junctions and scalability to more complex junctions; ii) develop real-time multiple model adaptive estimation methods to estimate states and parameters of signalized traffic junctions suitable for structurally diverse dynamic regimes arising abruptly from for example, road works. Several proposed adaptive systems in literature, relied on traffic surveillance technologies to warn the commuters of any detected irregularities and in most cases rely on human experience to evaluate the impact on the network performance and to provide route diversion recommendations that might not guarantee the optimal use of the available network capacity. A less infrastructure demanding multiple model adaptive estimation method is proposed to estimate states and parameters of signalized traffic junctions subject to time-varying parameters, *jump* dynamics and unpredictable disturbances; iii) the development of an autonomic control scheme. Although many so-called "adaptive" systems were proposed in literature, the controller parameters of these systems are not autonomously tuned to changing traffic behaviour so that the controller is able to adapt itself to changing traffic conditions and maintain optimal levels of performance. Hence, a truly adaptive system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is developed in this work. This is based upon model predictive control using linear and quadratic programming optimization techniques; iv) to integrate the latter two novelties (multiple model adaptive estimation and autonomic control) to obtain a multiple model adaptive control scheme for *jump* structural changes in junction dynamics. This scheme is validated by simulating typical signalized 3-arm and 4-arm junctions with arms and lane closure, with the ability to autonomously adjust to changing traffic conditions.

Acknowledgements

I am most grateful to my supervisor Professor Simon Fabri and my colleague Dr Kenneth Scerri for all their patience, guidance and enthusiasm. It was an honour for me to work under their exemplary supervision.

I wish to thank other close colleagues within the Department of System and Control Engineering especially Dr. Ing. Marvin K. Bugeja and Mr Noel Aguis and Mr Jean Gauci for their continuous support.

I wish to thank all my friends for their moral support, help, friendship and constant encouragement.

I would like to thank my loving husband Jeffrey for standing by my side at all times, believing in me and supporting me in any way possible.

Finally a special thanks goes to my family for all their support throughout my degree. They supported and believed in me whilst working on the thesis. I appreciate all that they have done for me.

Contents

List of Figures

List of Tables

Nomenclature

BMILP	Binary-Mixed-Integer-Linear-Program
CA	Collision Avoidance
CTM	Cell Transmission Model
DP	Dynamic Programming
EKF	Extended Kalman Filter
EM	Expectation-Maximisation
EMMPF	Efficient Multiple Model Particle Filter
GHR	Gazis Herman Rothery
GPS	Global Positioning System
IMM	Interactive Multiple Model
ITS	Intelligent Transportation Systems
LQ	Linear-Quadratic
LTW	Lighthill and Whitham
ML	Maximum Likelihood
MOVA	Modernized Optimization Vehicle Actuation
MPC	Model Predictive Control
SAST	Stepwise Adjustment of Signal Timing
SFM	Store-and-Forward Model
SMC	Sequential Monte Carlo

TOL	Traffic Optimization Logic
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

Chapter 1

Introduction

1.1 The Problem Background

Transportation is a vital part of modern life, giving us the freedom to travel efficiently over short and long distances for personal development and professional activities, or for leisure and holiday activities. This develops better contact and understanding between people. With increasing demands to travel over land in a short period of time, traffic problems have become more severe in many countries worldwide. Road transportation problems can result in congestions, delays, accidents and environmental problems. Traffic congestion appears when too many vehicles attempt to use a common transportation infrastructure with limited capacity. Queueing of such vehicles can result. Hence such situations lead to a degraded use of the available transportation infrastructure [9].

The management of road congestion is of utmost importance to attain sustainable economic activity and development. Concerns about such traffic conditions have made transportation a major issue in the agenda of the european union. The european commission white paper entitled the *Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system* [10] indicates that in Europe, there is a serious risk that economic competitiveness is seriously affected by increasingly intolerant chronic delays and the poor quality of some transport services. In fact, the cost of road congestion in Europe is estimated to be equivalent to 1% of the GDP [10]. To mitigate such problems, effective traffic management strategies and road planning techniques are required. Efficiency in transport networks is also a main priority for trans-

port policy at EU level as expressed through the european commission's white paper [10]. Efficient transportation is a precondition for maintaining the EU's prosperity, with the aim to reduce congestion and emissions and increase employment and growth rate.

Such efficiencies can be enhanced through the deployment of Intelligent Transportation Systems (ITS) within the road structure. The directive 2010/40/EU of the european parliament and of the european council of 7 July 2010 [11] proposes the framework for the deployment of intelligent transport systems, in order to guarantee:

- the optimal use of the available road network capacity (priority area I within directive 2010/40/EU) [11];
- the continuity of traffic management ITS services (priority area II within directive 2010/40/EU) [11];
- road safety (priority area III within directive 2010/40/EU) [11] and
- the linkage of the vehicle with the transport infrastructure, ensuring the facilitation of control of vehicles through the network infrastructure. (priority area IV within directive 2010/40/EU) [11].

An important component of ITS and advanced traffic management and road planning techniques is the timing of signal-controlled traffic light intersections that aim to optimize the flow of vehicles through a given road network [12]. Traffic light optimization is a complex problem, as an optimal solution is desired at each junction in the network. Furthermore, the complexity increases with the number of junctions because the state of one traffic light influences the flow of traffic towards other junctions. In addition, the complication increases as the flow of traffic changes constantly during the day, depending also on time of the week, and time of the year. Accidents, roadworks and weather conditions, such as flooding, further influence the complexity and performance of the system.

There is thus a growing need for managing traffic signals [12]. Traffic signals

are used to manage road space allocation, by assigning the right of way to different sets of mutually compatible traffic movements during distinct time intervals. The importance of traffic signals dates back to 1914 in the USA, after the war, when road congestion started to increase. Three generations evolved during the years.

The first generation [13, 14] used fixed-time methods to control the traffic signal duration. Their signal sequence and duration was predetermined and preset. A signal timing plan was selected according to a fixed schedule (e.g. time of day, day of week) from a set of predetermined plans, which were developed off-line on the basis of historical traffic data. The duration and order of all green phases remained fixed and did not adapt to fluctuations in traffic demand. The second generation systems [15, 16] were introduced in the early 1980s. These were characterised by actuated control, whereby, signal timings were adjusted depending on the detection of traffic in real-time. This was done, by installing inductive loops under the road surface for traffic sensing to indicate the presence or absence of vehicles. Regardless of demand, green was retained for at least the specified minimum duration. If a vehicle was detected, the green interval would be extended from the time of actuation by the fixed length of the extension time as shown in Figure 1.1. This could occur repeatedly, with the end of the interval delayed by the extension time from the time of each actuation. The interval would be terminated either when no additional actuation occurs during the latest extension time or when the specified maximum interval length would have been reached. The third generation [17–22] is distinguished by dynamic decision-making resulting from a closed-loop system as shown in Figure 1.1. This means that the system continuously measured the traffic flow and provided green timing feedback for the controller to quickly adjust the resulting green time. This was developed in the late 1980s. These third generation systems were described as adaptive systems, even though they were only responsive to traffic conditions and the controller parameters were not tuned to changing traffic behaviour. An adaptive control system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is thus desirable. In ITS, such systems are called autonomic systems, which will be described in more detail in the next

section.

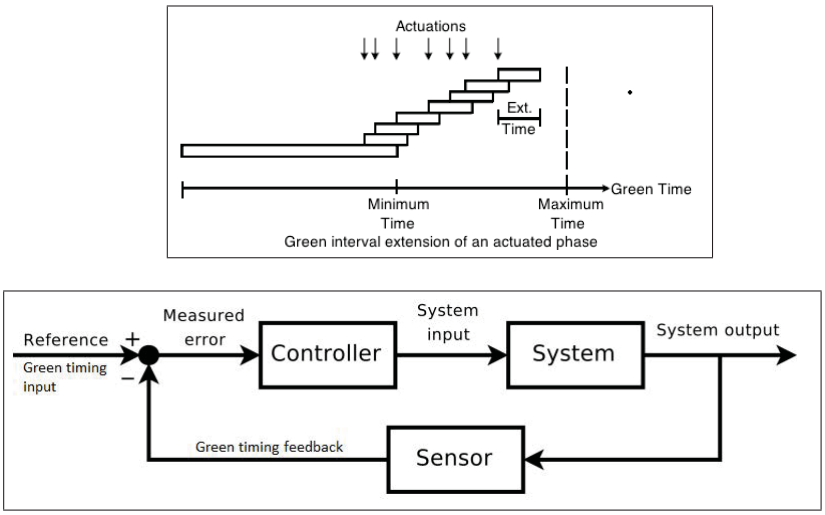

Figure 1.1: Second generation system shown in the top image and third generation system shown in the bottom image

1.2 Motivation

ITS have been implemented through the evolution and generation of traffic signal control concepts. Such evolution integrates advances in control, communications and computational technologies and provide intelligent control of traffic lights that adapt themselves for example to time-varying traffic conditions and direction in demand. Despite such advances in ITS, and despite the response of the previously mentioned three generation systems to the traffic demand, such systems can fail when networks are subject to unanticipated or unusual irregularities, such as roadworks, accidents and extreme weather conditions such as flooding [23,24]. Even third generation systems can fail when networks are subject to such irregular conditions because, although this generation is responsive to traffic demand, it is not adaptive to structural network changes [23], because the controller paradigm is not tuned to changing traffic behaviour. To overcome such limitations, traffic surveillance technologies are installed to monitor network conditions and to warn the commuters of any detected irregularities. Ac-

cording to Etmadnia *et al.* and Stoilova in [25,26], such technologies rely on human intervention to overcome these irregularities. Thus in most cases, humans rely on their experience to evaluate the impact on network performance and to provide route diversion recommendations, that might not guarantee the optimal use of the available network capacity. Such systems, together with surveillance technologies, can be viewed as forming large scale, heterogeneous, control systems, complicated by the dependencies on human behaviour [27]. According to McCluskey [28], to effectively manage or enable the optimization of such a socio-technical system is a daunting task, exacerbated by rising public environmental and operational expectations.

According to Stoilova [26], current traffic light systems that rely on human intervention require corporate data centers which have multiple vendors and platforms. Installing, configuring, and integrating such systems is time consuming and error prone. Problem determination in large, complex network systems can require weeks of team programming, while recovery from traffic incidents and cascading failures is manual, resulting in high labour costs. Moreover, according to McCluskey, trained human experts require much effort to make informed decisions and plans in such complex, real-time systems [28]. Realizing the limitations of the current practice, the need to create self-managing systems is of utmost importance [23, 25, 26, 28–30]. According to Ullas and Nayak [29], an efficient and smart way to control traffic routing is by using the autonomic principles, that minimize human involvement. The term "autonomic principles" is inspired by the human nervous system which *self*-handles complexity and uncertainties, with minimal conscious human intervention.

This means that the complexities and the uncertanties in dynamic heterogeneous and interconnected systems are managed by the systems themselves, thus reducing any human interventions. In the traffic scenario, such systems can self-handle complexity and self-tune the controller parameters to adapt to changing traffic conditions. As the complexity of traffic systems grows, the need to build into them the means to manage and maintain themselves becomes more necessary. Road traffic systems are heterogeneous systems that are expensive to maintain,

and difficult to optimise. Challenges associated with road management include congestion costs, environmental damage, safe network management and support of system costs. Hence autonomic traffic light systems are required, aiming at improving network efficiency and reducing the chances of human error.

Autonomic traffic light systems can have several *self-* properties [31] such as i) *self-configuring* by readjusting themselves automatically when subject to changing circumstances such as network irregularities, ii) *self-healing* to ensure effective recovery of the network when a fault occurs, such as sensor faults, iii) *self-anticipating* with the ability to predict likely outcomes and simulate *self* actions, iv) *self-optimizing* the traffic flow in the network to guarantee the optimal use of the available network capacity and v) *self-aware* with the ability to learn internal states describing the current behaviour of the system with knowledge of past states. Such *self-* properties do not involve any human intervention in the process. Table 1.1 contrasts the differences in system properties between third generation adaptive traffic systems and autonomic systems.

Table 1.1: Adaptive vs autonomic properties

Properties	Adaptive systems	Autonomic systems
adaptive to traffic demand	yes	yes
requires human intervention	yes	reduced to a minimum
self-handling complexities	no	yes
self-configuring	no	yes
self-healing	no	yes
self-anticipating	no	yes
self-optimizing	yes but might not be optimal	yes

As a consequence of such *self-* properties, the interface with the transport infrastructure is facilitated. The interaction with people is set more at a *service* level rather than at *command* level. As a result, coordination between a traffic control centre manager and future autonomic systems will be enhanced by communicating high-level goals, priorities and tasks which the systems will solve. Moreover,

smoother travelling behaviour ensues, because of autonomic transport systems facilitating traffic flow optimization, [32, 33], which will help in reducing consumed fuel, whilst keeping pollutant emissions and noise levels to a minimum. Environmental benefits can also be achieved when incidents, or a combination of unexpected events of different nature, including network irregularities, result.

1.2.1 Autonomicity

To the best of the author's knowledge, autonomicity has only recently been introduced to urban traffic networks. Several so-called *self-organizing* [34–36] and *organic* [37, 38] systems have been proposed which make use of autonomic [39] and organic computing [40] to handle the growing complexity in today's traffic systems. While autonomic computing has a strong focus on server architectures [39], organic computing investigates the use of biological-inspired computing [40] to traffic systems such as particle swarm optimization and ant colony optimization [41].

Such techniques heavily rely on cooperative agents that share or exchange information to achieve better system-wide performance [34, 42–46]. Coordination is a complicated timely process that typically consists of several operations: exchanging local information, detecting interactions, deciding whether or not to coordinate, proposing, analysing, refining and forming commitments, sharing results and so on [47]. Communication can also be related to transmission of knowledge stored in the form of learned policies, particularly in reinforcement learning [48]. However, agents inevitably suffer from communication noise, which introduces another layer of complexity for the traffic systems [49], relying heavily on communication and coordination among agents.

Unlike *self-organising* systems, *organic* systems make use of biological-inspired agents to communicate and coordinate among agents. Significant contributions using approaches based on swarm intelligence have been made, where agents behave like a social insect and the stimuli to select one phase or plan is given by a *pheromone* trail with an intensity related to the number and duration of vehicles

in the link [50].

In both *self-organising* and *organic* systems, communication is a timely process, with the complexity increasing significantly with the number of agents in the system. In addition, the traffic system may sometimes exhibit features of a typical *jump system* where the system structure or parameters are not constant or varying slowly, but prone to change value abruptly due to structural network changes caused for example by flooding, road accidents etc. [51] which could lead to arm closure. Hence the complexity of communication may hinder the performance of such systems because of the slow response to detect abrupt changes in the system parameters and to immediately update model parameters.

A multiple-model adaptive approach turns out to be computationally efficient, reliable and much simpler than other non-multiple-model adaptive approaches for such *jump systems* [51]. The system is *intelligent* in multiple-model approach as it is furnished with the characteristic of learning through memorization and not just adaptation, hence *self-configuring* and being *self-aware* of such abrupt changes, unlike the other techniques discussed before. The system model equations can learn and switch in time from one model to another to represent different traffic mode scenarios. In addition, *self-organized model allocation* is carried out by the system, whereby, in addition to learning the various different mode dynamics and detecting traffic mode switches, the system will automatically configure itself to grow a new model to represent traffic behaviour under novel conditions. Furthermore, multimodality lends itself very well to dealing with transitions scheduled by some operating condition. Different operating conditions exist for traffic such as *saturated and unsaturated operating conditions*. Hence multimodality is appropriate for handling different traffic conditions in a network, by segmenting the space into a number of non-overlapping partitions, each of which correspond to a particular mode. In turn, each mode captures the dynamics within a particular range of operating traffic conditions.

Traffic networks are characterized by highly complex structures and dynamic behaviour typified by time-varying parameters, *jump* dynamics and unpredictable

disturbances. The task of controlling this kind of system is therefore a considerable challenge, when expected to operate safely, reliably and efficiently within several operating conditions and with as little human intervention as possible. This thesis will show, among other things, that a Multiple model adaptive control approach is a good step forward towards increasing autonomicity in such systems.

1.3 Objectives of this research

The main aim in this study is the development of autonomic control for urban traffic junctions. In order to reach this main aim, the following sub-tasks were carried out:

- Familiarization with analysis of models used to describe traffic signal behaviour within road junctions.
- Development of discrete state-space models for signalized junctions such as typical junctions in Malta. - The use of the matrix notation greatly simplifies the mathematical representation of multiple-input-multiple output systems with particular emphasis to signalized traffic junctions. This objective is sub-divided into the following sub-objectives:
  - The development of multiple-input-multiple output state-space representations of traffic behaviour within signalized road junctions.
  - The development of novel algorithms to estimate the model parameters and the states describing the traffic flow dynamics in a signalized junction under different traffic conditions in real-time. This objective is further sub-divided into:

i) Development of quasi real-time joint estimation algorithm to estimate the model states, parameters and noise in signalized traffic junctions based on measurements of traffic flow.
ii) Development of several online algorithms to jointly estimate the model states, parameters and noise using expectation-maximization and Robbins-Monro stochastic approximation techniques, also extending this to multivariate estimation problems.
iii) Extension of the above to a derivative-free multidimensional stochastic approximation methodology, leading to an algorithm that is more generalizable and scalable to multiple junctions.
iv) The development of real-time multiple model adaptive estimation methods to estimate states and parameters of signalized traffic junctions suitable for structurally diverse dynamical regimes arising abruptly from unexpected traffic congestions or network irregularities due to, for example, road works, blockages due to accidents and extreme weather conditions.

- Familiarization and design of autonomic controllers for the traffic control system under normal traffic conditions. Hence the following sub-objectives were reached:
  - The study and familiarization with automatic controllers relevant to traffic junctions such as model predictive control [52].
  - The design and evaluation of novel autonomic control systems with self-tuning controllers making use of model predictive control with linear or quadratic programming.
- Integration of all the above work to develop a multiple model adaptive control scheme with self-tuning contollers that is able to handle better *jump* changes in junction dynamics.
- Simulation studies of the implemented models and comparisons of the system's behaviour subject to several irregularites as discussed before.

1.4 Thesis Contributions and Overview

The work reported in this thesis has led to the following new contributions to autonomic traffic control systems:

- The development of real-time joint state and parameter estimation algorithms to simultaneously estimate the model states, parameters and noise covariance of the state-space model describing the traffic flow dynamics in a signalized junction. Very few publications report on joint state and parameter estimation applied to signalized traffic junctions. Sometimes the model parameters are assumed known apriori from past traffic measurements. Other works apply nonlinear estimation algorithms such as the extended Kalman filter that carries with it some disadvantages, as will be discussed in more detail in Chapter 2, such as divergence issues. This motivated the development of joint state and parameter estimation algorithms from an alternative perspective. The proposed algorithms make use of the state-space model presented in Chapter 4 and the standard expectation-maximization algorithm was modified to obtain real-time estimation of the model parameters jointly with state and noise estimation, as discussed in detail in Chapters 5 and 6.

- The development of real-time multiple model adaptive estimation methods to estimate states and parameters of signalized traffic junctions suitable for structurally diverse dynamical regimes arising abruptly from unexpected traffic congestions or network irregularities due to, for example, road works, blockages due to accidents and extreme weather conditions. As discussed in Chapter 1, adaptive systems relied on traffic surveillance technologies to warn the commuters of any detected irregularities and in most cases rely on human experience to evaluate the impact on network performance and to provide route diversion recommendations that might not guarantee the optimal use of the available network capacity. The autonomic systems that were introduced to urban traffic networks heavily rely on communication and cooperative agents that share or exchange information, leading to a complicated and time-demanding process. Hence

multiple model adaptive estimation is introduced to estimate states and parameters of signalized traffic junctions subject to time-varying parameters, *jump* dynamics and unpredictable disturbances more efficiently. This approach is discussed in Chapter 7.

- The design of an autonomic controller for signalized junctions. Many adaptive systems were proposed in literature. Although such systems are responsive to changing traffic conditions, the controller parameters are not tuned to these changes. An adaptive system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is required. The proposed autonomic controller is based upon control using linear and quadratic programming optimization techniques, as discussed in Chapter 8.

- The integration of the latter two novelties, that is the integration of multiple model adaptive estimation methods with the design of autonomic controller with self-tuning characteristic. This approach leads to a multiple model adaptive control scheme for *jump* structural changes in junction dynamics as will be discussed in detail in Chapter 9.

The above contributions lead to an autonomic traffic control system because: i) the system is able to learn and estimate the model states, parameters and noise covariance of the state-space model describing the traffic flow dynamics in a signalized junction, hence being self-aware of the traffic behaviour; ii) the system is intelligent in multiple-model approach being able to learn through memorization and hence self-configuring and being self-aware of any *jump* changes, unlike the adaptive techniques. The system is also capable of performing self-organized model allocation, whereby, in addition to learning the various mode dynamics and detecting mode switches, the system will automatically configure itself and grow a new model set to represent traffic behaviour under novel different conditions; iii) the system is able to self-optimize the traffic flow in the network to guarantee the optimal use of the available network capacity. The autonomic controller is able to self-tune and adjust its parameters to adapt to changing traffic conditions; iv) a multiple model adaptive control scheme is designed so that the

system is able to self-optimize the traffic flow also when subject to *jump* structural changes in junction dynamics. Such *self-* properties do not involve any human intervention in the process. Two jump dynamic scenarios will be tested namely arm closure or lane closure as will be discussed in detail in Chapter 7 and 9.

This Chapter has delivered a brief overview of the study, its objectives and novel contributions to autonomic control of urban traffic junctions. Chapter 2 will give a literature review of several relevant traffic characteristics studied over the years, and techniques for modelling the dynamics of traffic flow. This review is enhanced in Chapter 3, where the common traffic signal control methods applied in literature are compared and contrasted.

The rest of the thesis is divided into two parts. Part 1 (Chapters 4 to 7) describes the development of a joint state, parameter and noise estimation algorithm for multiple-input-multiple output state space models of traffic junctions. The concept of real-time multiple model estimation is also developed for situations exhibiting *jump* structural changes in the junction dynamics. Part 2 (Chapters 8 and 9) describes the development of model predictive control algorithms with linear and quadratic programming and targets the integration of all the works discussed in the previous chapters to obtain a multiple model adaptive control scheme for *jump* structural changes in junction dynamics. The results and analysis of the implemented algorithms are discussed in the respective chapters. Finally, Chapter 10 gives a discussion and conclusions of this work.

Based on some of the work reported in this thesis, the following peer-reviewed articles have been published so far:

- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Joint state and parameter estimation for a macro traffic junction model. Proceedings of the 24th Mediterranean Conference on Control and Automation, Greece, June 21-24, 2016.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Simultaneous traffic flow

and macro model estimation for signalized junctions with multiple input lanes. Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems, Portugal, April 22-24, 2017.

- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Online state and parameter estimation for a 4-arm junction. Proceedings of the 25th Mediterranean Conference on Control and Automation, Malta, July 3-6, 2017.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Online State and Multi-dimensional Parameter Estimation for a Macroscopic Model of a Traffic Junction. Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems, Japan, October 16-19, 2017.
- L. Chetcuti Zammit, S. G. Fabri and K. Scerri, Real-Time Parametric Modelling and Estimation of Urban Traffic Junctions. IEEE Transactions on Intelligent Transportation Systems, 99:1-11, 2019.

Chapter 2

A review of traffic flow dynamic modelling

The major control measure in intersections in urban road networks are traffic lights. Traffic lights were originally installed in order to ensure the safe crossing of vehicles and pedestrians. Eventually, with steadily increasing traffic demands, it was realized that traffic lights can also be utilized to optimize traffic flow and reduce the travel time of vehicles in the network [53] as a job scheduling process [54]. This chapter first describes the relationships among traffic stream characteristics in a network and their measurement techniques, leading to a review of several modelling techniques for capturing the dynamics of traffic flow within a junction.

2.1 Traffic models

Considering traffic flow patterns, three levels of modelling perspectives can be studied: *macroscopic*, *mesoscopic* and *microscopic*. *Macroscopic* models describe traffic at a high level of aggregation, as a flow, without distinguishing its constituent parts [55–58]. The traffic stream is thus represented in an aggregate manner using characteristics such as flow-rate measured in vehicles per unit time, vehicle density measured as vehicles per unit distance, occupancy measured as the percentage of time a point on the road is occupied by vehicles, and average vehicle speed measured as average distance per unit time. Individual vehicle manoeuvers are not explicitly represented [59]. On the other hand, a *microscopic* model describes both the space-time behaviour of the system's entities, i.e. vehi-

cles and drivers, as well as their individual interactions, at a high level of detail. Examples of the description of traffic patterns at a microscopic level include individual vehicle space coordinates and their time-dependence, a time headway measured in metres and a space gap between two vehicles following each other, also measured in metres, single vehicle speed measured in metres per unit time and vehicle length specified in metres [55]. Moreover, a *mesoscopic* model fills the gap between the family of *microscopic* models and the family of *macroscopic* models and hence describes vehicle flow in aggregate terms such as in probability distributions, and their activities and interactions at a much lower level of detail [55]. The choice of a particular type of model depends on the nature of the problem of interest. *Microscopic* models impose high computational demands due to the level of detail for representing individual vehicle manoeuvers, while *macroscopic* models require lower computational costs due to lower complexity. Nevertheless, in *macroscopic* models, the essentials of traffic behaviour can still be accurately reflected [55].

There exist many measurement techniques for measurement of traffic flow variables, based on sensors or road detectors. Inductive loop technologies, magnetometers, magnetic induction coils, microwave radar, active or passive infrared sensors, acoustic arrays and video image processors can all be used to measure traffic counts, presence or occupancy and speed. Ultrasonic techniques can also be used to measure traffic counts and presence, but are unable to measure speed. Moreover, microwave radar, active infrared, acoustic array and video image processors have the additional ability of detecting traffic in multiple lanes. However the cost of such sensors increases drastically. The cheapest sensors out of all the considered ones, are the inductive loop technologies. Further detail on the different types of sensors are discussed in the traffic control systems handbook [60].

2.2 Road Traffic Control

2.2.1 Basic Notation

An intersection consists of a number of approaches and a crossing area as shown in Figure 2.1, taken with permission from [1]. An approach may have one or more lanes. Approaches are used by corresponding traffic streams. The saturation flow for each junction approach and turn is defined as the maximum flow crossing the stop line of an approach when the corresponding stream has right of way, the upstream demand or the waiting queue is sufficiently large, and the downstream links are not blocked by queues. Two streams are compatible when they can safely cross the intersection simultaneously, else they are called antagonistic. A *phase* is defined as the time period for which a group of one or more traffic links receive an identical signal. The shaded area in the timeline representation of a phase in Figure 2.1 shows the green time duration corresponding to that traffic link. A set of one or more traffic phases that receive a green signal during a particular period of the cycle constitutes a *stage*. Hence, the time between successive starts of the green stage is usually considered as a *cycle* [1]. Figure 2.1 illustrates such basic notations.

A junction exhibits saturated traffic conditions when the flow rate and speed drop to zero. The vehicles are queuing and the junction is exposed to the maximum number of vehicles that could be carried by the arm. On the other hand, a junction exhibits unsaturated conditions when the number of vehicles exiting the arm is less than the maximum dictated by the saturated flow and vehicles are not impeded by other traffic as they travel at maximum speed.

The traffic control system can be seen as a resource allocation system, whereby, the vehicles are considered as customers to be served by a specific number of servers. The servers correspond to the different stages of a junction and these cannot operate simultaneously, but are scheduled an activation period so as to maximize the number of served customers and hence increase the throughput of the system [54].

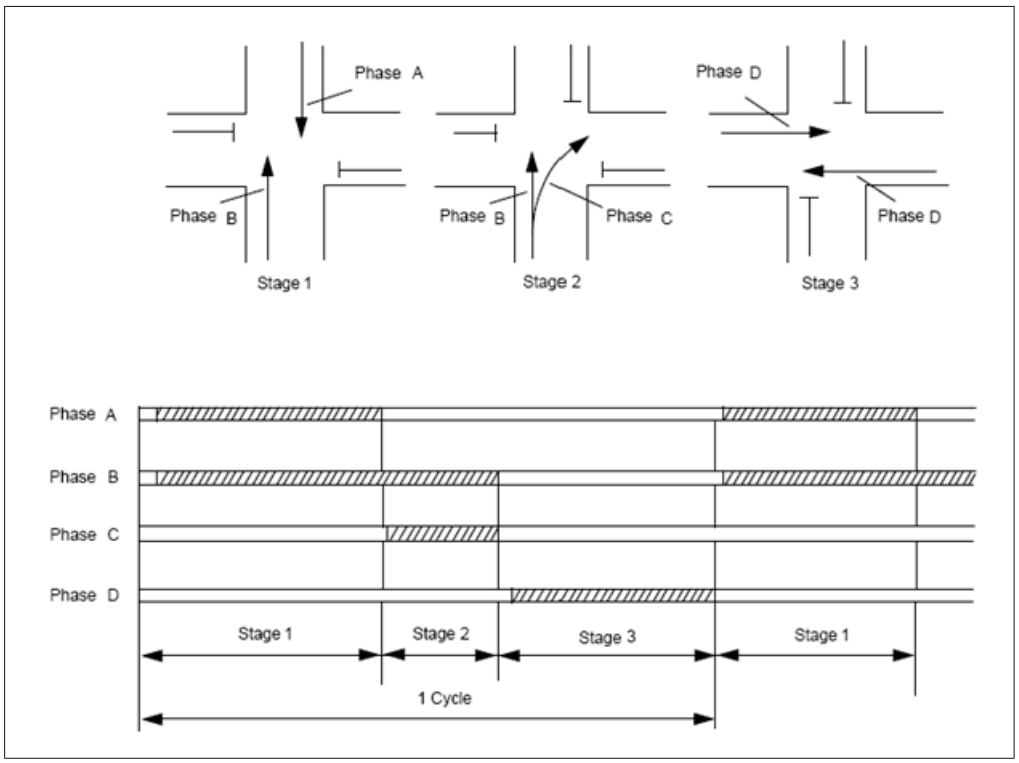

Figure 2.1: Traffic signal control terminologies [1]

2.2.2 The Causes of Traffic Congestion

The primary cause of over-saturated status at an intersection is that traffic demand exceeds the capacity. The capacity is the maximum flow of vehicles that can be discharged through the intersection from a particular lane under the prevailing traffic, roadway, and signalization conditions. Moreover, at signalized intersections, the reasons of congestions can be identified as spillback, residual queues and storage blocking [2]. Spillback occurs when a queue from a downstream intersection occupies all the space on the link and prevents vehicles from entering the upstream link on green. The upstream and downstream intersections define the order of the intersections following the direction of movement of vehicles. Spillback will cause a de-facto red to the movement on green, as represented in Figure 2.2. As shown in Figure 2.3 residual queues at a downstream intersection may impede traffic progression and can cause unnecessary delays. Moreover, storage blocking occurs when turning traffic uses up the entire space

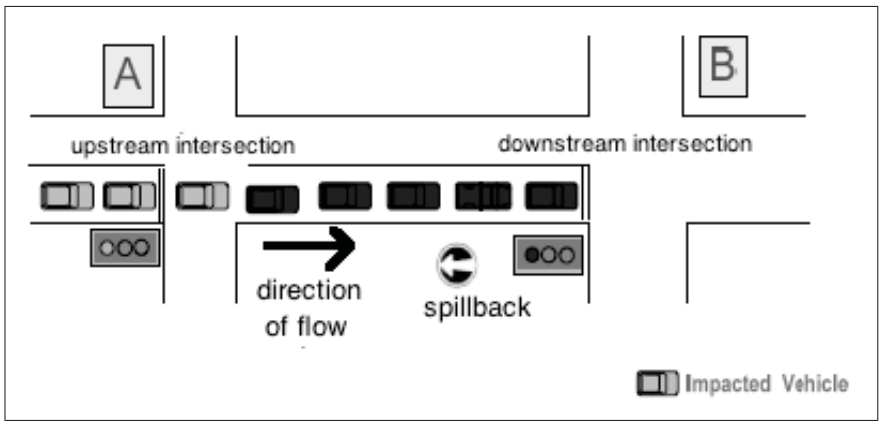

Figure 2.2: Spillback at intersection [2]

of the storage lane and blocks the through traffic as shown in Figure 2.4.

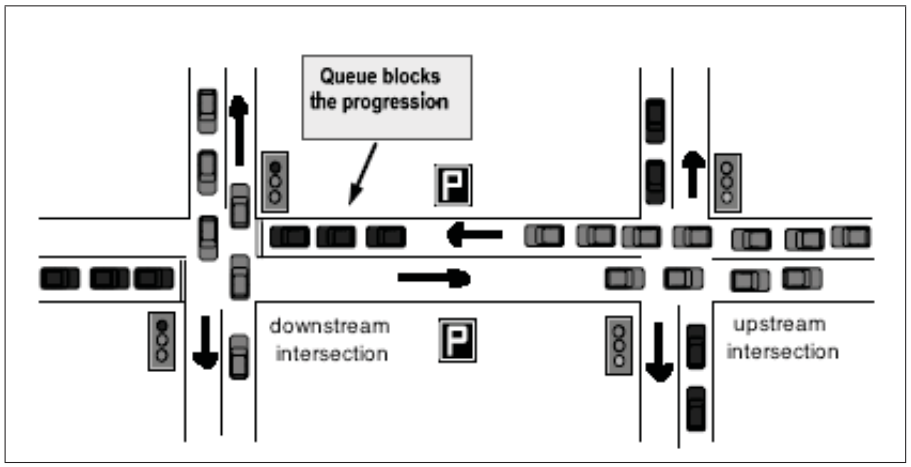

Figure 2.3: Residual queues at intersection [2]

Such situations can be remedied by making infrastructural changes to the intersection. The changes can be either to increase the width of the input or output arms or to change the position of the traffic lights. The infrastructural changes are the most financially expensive solutions to implement and cause long road closure. Land restrictions may also inhibit any infrastructural changes [61].

An alternative option is to introduce fees for vehicles entering the more con-

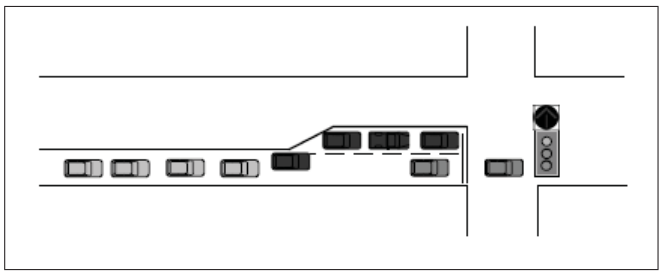

Figure 2.4: Storage blocking [2]

gested arms. This solution is often employed in many countries and has significant financial gains. The main disadvantage of this solution is that it is not always politically attractive and may leave the traffic problem as is, simply shifted either to alternative arms and intersections or to new modes of transportation chosen by travellers [62].

Another option is to modify the traffic lights duration. These can be set off-line or on a trial and error basis, which is not ideal as the traffic lights duration may not correspond to the actual current demand in traffic. The other change can be carried out by adding sensors at strategic locations inside the signalized intersection and using data from the sensors to control the traffic light timings in real-time, thus making the system traffic-responsive. Such a solution is less financially expensive than one based upon infrastructural changes. In order for the latter option to be possible, the dynamics of the traffic behaviour within a junction needs to be analysed as will be discussed in the next section.

2.3 Modelling Strategies

To model the dynamics of traffic behaviour within an intersection, two main distinct strategies can be used: analytic modelling and data-driven modelling [63]. In analytic models, the estimation of delay experienced by vehicles at intersections [64] and the road capacity and the saturation flow rate [65,66] are estimated based on traffic and driver behaviour. In many cases, analytic models describe traffic behaviour at a *microscopic* level, as explained in Section 2.1.

A *microscopic* traffic behaviour model describes the space-time behavior of the vehicles and drivers individually. The interactions among the modelled components are then properly connected to form the complete system model at a high level of detail. An example of an analytic *microscopic* traffic model is the *Gazis Herman Rothery* (GHR) model [67, 68] which dates back to the late fifties and early sixties. This model describes the acceleration of the observed vehicle depending on its actual speed and the relative space and speed, respectively to the vehicle in front of the observed one. Moreover, the *safety distance or collision avoidance model* (CA) , studied in the works of Peter *et al.* and Papaioannou *et al.* [69,70] tries to model and specify a safe following distance through the manipulation of the basic Newtonian equations of motion. The *Helly Linear Model*, due to Helly [71, 72], proposes a model that includes additional terms for the adaptation of the acceleration according to whether the vehicle in front was braking. For analytic models, highly dynamic traffic parameters such as speed or distance are required [67–70], which can result in high cost and computational burden to gather such dynamic information for each vehicle in the network. Thus data-driven modelling strategies are also applied. This involves analysing the data based on the traffic flow intensities at a *macroscopic* level within a signalized intersection, and then the complete model is directly inferred from these observations to capture the important characteristics of the system. When attempting to control the traffic flow at an intersection, a *macroscopic* model is preferred since the main aim is to provide a general evaluation of traffic flow inside an intersection, rather than the individual driver behaviour which varies from one person to another [3]. The lower computational power and the smaller number of estimates required leads to advantages when opting to use *macroscopic* models.

Data-driven modelling can be performed following a six step procedure as explained in [63]. The first step is to identify the purpose of the model. This is a vital consideration, since no model irrespective of its complexity can capture the system's complete behaviour. The collection stage is another important step because the accuracy of the model depends on the collected data. The available data indicates the system characteristics to be modelled. Prior to using the

collected data, preprocessing and analysis of the available data is carried out to eliminate noise, outliers and offsets which could have been introduced during the collection stage and do not contribute to the system's behaviour. Based on the purpose of the model as defined in the first step, the appropriate model structure is chosen to carefully represent the repeatable patterns in the observation stage. To develop the model structure in data-driven modelling, the model parameters should be defined. Some of the model parameters are known while some of them are unknown and thus these have to be estimated from the available data. The estimation techniques chosen depend on the model structure and on the availability of data. The final step is to assess the model's performance through appropriate validation tests by comparing the model predicted output to the real world output thus ensuring similar behaviour. The aim of data-driven modelling is to identify a representation that minimizes the discrepancy between the real world behaviour and the model predicted behaviour.

Over the years, several published works focused on deriving data-driven *macroscopic* models aiming to represent the dynamics of traffic behaviour within a signalized intersection to assist traffic engineers to better analyze, understand and even forecast the traffic processes [4, 6, 7, 73–85]. Three discrete macroscopic modelling techniqes are predominant in literature, the *Cell Transmission Model* (CTM) [73, 74] , the *Store-and-Forward Model* (SFM) [4, 75–79, 86] and a set of approaches based on queues and the conservation of vehicles principle, such as in [6, 7, 80–85]. These techniques will be reviewed in the following sections.

2.3.1 Cell Transmission Model

The CTM by Daganzo *et al.* [73, 74] assumes that the road can be divided into homogeneous segments producing a chain of cells of arbitrary length. Moreover, a finite time interval T is defined in order to observe the traffic process at discrete moments $t=0, T, 2T, ...$ The number of vehicles transmitted from one cell to another is calculated by the model. Vehicle-conservation is applied by assuming a piecewise linear relationship between the traffic flow and density for each cell. This model is very similar to the *Lighthill and Whitham* (LTW) model, developed

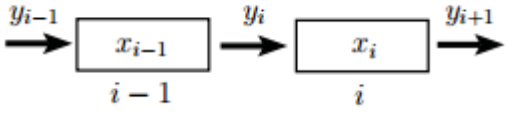

Figure 2.5: Successive cells in CTM [3]

by Lighthill in 1955 [87,88], except that the traffic model is discretized into cells.

The transmission depends on how many vehicles are transferred from the upstream cell to the downstream cell as shown in Figure 2.5. The following notation is identified in the CTM:

- $x_i(k)$ denotes the vehicle number in the i^{th} cell at time t=kT, where k=$1,2,....$
- $x_i^{max}(k)$ is the maximal number of vehicles in the i^{th} cell at time t=kT. $x_i^{max}(k)$ can be calculated by $x_i^{max}(k)$=$l_i\rho_{max}(k)$ where l_i is the length and $\rho_{max}(k)$ is the maximal possible density of the i^{th} cell.
- $Q_i(k)$ denotes the maximal throughput of cell transmission, i.e. how many vehicles can flow from cell $i-1$ to cell i during the time interval of $[kT,(k+1)T]$. This also means that the outflow capacity of cell $i-1$ is equal to the inflow capacity of cell i.
- $y_i(k)$ is the vehicle number transmitting from cell $i-1$ to cell i during $[kT,(k+1)T]$.

By applying the vehicle-conservation principle, Equation (2.3.1) holds

$$x_i(k+1) = x_i(k) + [y_i(k) - y_{i+1}(k)]T \tag{2.3.1}$$

Furthermore, a restrictive condition is introduced for the inflow traffic to hold the vehicle-conservation law given in Equation (2.3.2):

$$y_i(k) = min\{x_{i-1}(k), Q_i(k), x_i^{max}(k) - x_i(k)\} \tag{2.3.2}$$

The last term in this condition represents the empty space in the i^{th} cell at time step $t = kT$. The first term in equation (2.3.2) refers to the source cell, the second the transmission and the third the target cell. Such a model can be applied to urban traffic dynamics whereby, a separate cell is introduced for each turning movement. A lot of measurements are required, as observations are needed at each cell. Hence, this approach is not easily applicable for online traffic control [89].

2.3.2 Store-and-Forward Model

The SFM technique was first suggested by Gazis and Potts in 1963, which attempts to describe the network traffic flow process in an efficient and simplified manner by making use of a directed graph to represent the urban road network [4, 75–79, 90]. It is based on the SFM paradigm which was developed in telecommunication systems, where information is sent to an intermediate node, kept and then forwarded to either an intermediate node or to its final destination, given that the integrity of the message is verified.

Based on two intersections j, denoted by, M and N, connected by a link z as shown in Figure 2.6, Aboudolas *et al.* [4] developed the model given in equation (2.3.3).

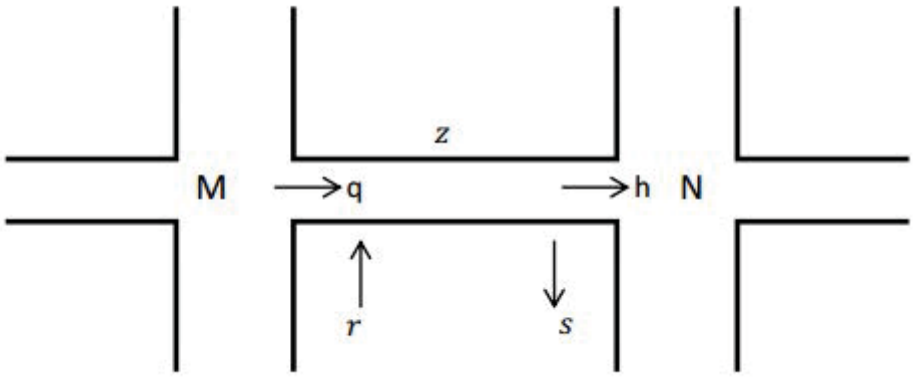

Figure 2.6: The junction under consideration [4]

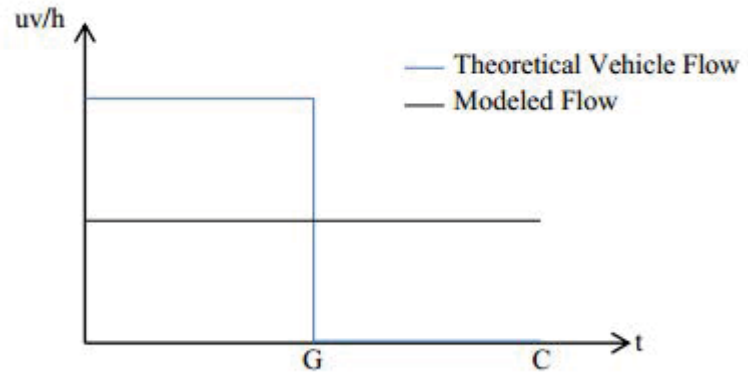

Figure 2.7: The real and modelled flows during one cycle period [4]

$$x_z(k+1) = x_z(k) + T[q_z(k) - h_z(k) + r_z(k) - s_z(k)] \tag{2.3.3}$$

where z and j represent the links and junctions respectively, T is the control interval, $x_z(k)$ represents the number of vehicles present in link z at time kT (uv), where *uv* denotes unit vehicle, $q_z(k)$ and $h_z(k)$ are the inflow and outflow respectively for link z (uv/h) and $r_z(k)$ and $s_z(k)$ are the demand and exit flow respectively. These two latter variables model any non-controlled intersection between the intersections M and N, such as a parking lot (uv/h).

Moreover, constraints (2.3.4) and (2.3.5) hold at each junction j, where C is the cycle time which is assumed to be fixed and equal for all junctions. L_j is the fixed time which is lost when the signals in the traffic lights change and $g_{j,i}$ is the green time of stage i at junction j which has to be set greater than a certain threshold $g_{j,i,min}$. Furthermore, the signal control plan of junction j (including the fixed lost time L_j) is based on a fixed number of stages denoted by F_j.

$$\sum_{i \in F_j} g_{j,i} + L_j \leq C \tag{2.3.4}$$

$$0 \leq x_z(k) \leq x_{z,max} \tag{2.3.5}$$

Provided that the number of vehicles present in the link z is sufficiently high and that space is available in the adjacent links, the real outflow is equal to the saturation flow provided that the link z has right of way. Otherwise, the real outflow is equal to zero. A simplification of the store-and-forward model is to consider the total outflow rate of any lane as its average value as shown in Figure 2.7 by taking into consideration Equation (2.3.6).

$$u_z(k) = \frac{G_z(k)S_z}{C} \tag{2.3.6}$$

S_z gives the saturation flow of link z and $G_z(k)$ is the green time of link z, given by $G_z(k) = \sum_{i \in u_z} g_{j,i}(k)$.

This simplification is essential as it gives a single value for the outflow rather than a constantly fluctuating value. The main advantage of SFM is the low computational requirements due to the linear characteristic. Therefore, this model can be rewritten in linear time invariant state space form, allowing for the use for a number of highly efficient optimization and control methods. However, due to the simplification introduced by Equation 2.3.6 the modelled outflows are considered saturated. The unsaturated traffic conditions are not considered by the model. Hence, the model is only applicable for saturated traffic conditions, when saturation flow rates are approximately constant.

2.3.3 The Queuing Theory Approach

Kleinrock in [91] describes a queuing system as a class of dynamic systems referred to as *systems of flow*. Such a system is defined as one in which some commodity flows, moves or is transferred through one or more finite-capacity channels in order to go from one point to another. In this case, the commodity represents the vehicles in the network moving through numerous roads and junctions, to get to their desired destination [92–94]. Detectors are installed to capture the number of vehicles entering or leaving the junction. Since the roads and junctions are of finite capacity, a queue will occur if this finite capacity is exceeded.

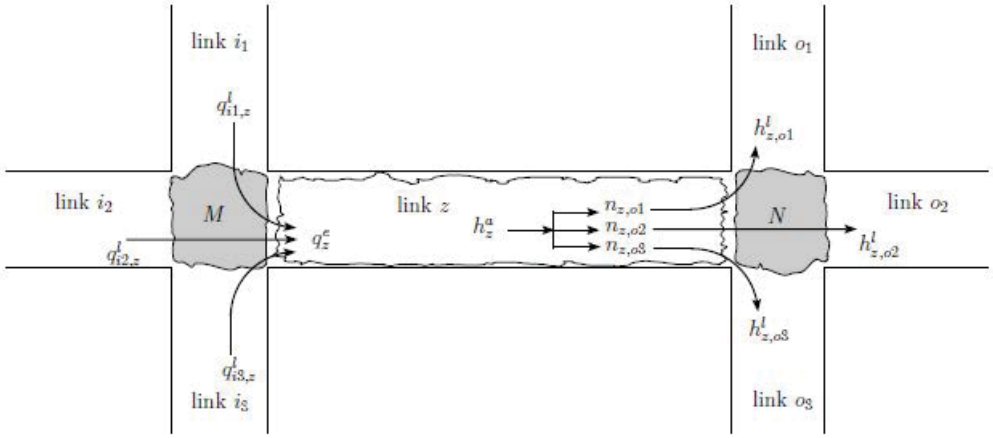

Figure 2.8: Simplified BLX model [5]

The BLX model of Lin, Schutter, Xi and Hellendoorn [5, 80, 95] is one application of queueing theory to model traffic at a junction. This system, unlike the SFM approach, is applicable in all traffic conditions: undersaturated, saturated and oversaturated. Figure 2.8 shows the representation of a simplified BLX model. The notation of Figure 2.8 is defined below:

- I_M is the set of input links of junction M where I_M=i_1, i_2, i_3.
- O_N is the set of output links of junction N where O_N=o_1, o_2, o_3.
- $x_z(k)$ represents the number of vehicles in link z at time step k in uv.
- $n_z(k)$ represents the queue length in link z in uv.
- $n_{z,o}(k)$ represents the portion of $n_z(k)$ turning to link $o \in O_N$ in uv.
- $h^l_z(k)$ represents the flow leaving link z in uv/cycle.
- $h^l_{z,o}(k)$ represents the portion of $h^l_z(k)$ turning to link $o \in O_N$ in uv.
- $q^a_z(k)$ represent the flow arriving at the end of the queue of link z uv/cycle.
- $h^a_z(k)$ represent the flow of unit vehicles in link z, between intersections M and N in uv/cycle.

- $h^a_{z,o}(k)$ represent the portion of $h^a_z(k)$ turning to link $o \in O_N$ in uv.
- $q^e_z(k)$ represent the flow entering link z in uv/cycle.
- $q^e_{i,z}(k)$ represent the portion of $q^e_z(k)$ arriving from link i in uv.
- $a_{z,o}(k)$ represent the turning rate from link z to o.
- S_z represent the saturation flow rate leaving link z in uv/cycle.
- $g_{z,o}(k)$ represent the green time during time step k for the traffic from link z to o.
- v^{free}_z represent the free flow speed in link z in uv/cycle.
- C_z represent the capacity of link z as vehicle number uv.
- N^{lane}_z represent the number of lanes in link z.
- c is the cycle time of the junction. c can be different for intersection M and N.
- l_{veh} is the average vehicle length.

The vehicle conservation law also applies. Therefore the number of vehicles in link z at step $(k+1)$ is defined as in Equation (2.3.7):

$$x_z(k+1) = x_z(k) + \Big[q^e_z(k) - \sum_{o \in O_N} h^l_{z,o}(k)\Big]c \tag{2.3.7}$$

Flow $h^a_z(k)$ arrives at the end of the queue after time delay $\tau(k)c + (k)$ as defined within Equation (2.3.8):

$$h^a_z(k) = \frac{c-\gamma(k)}{c} q^e_z(k)(k-\tau(k)) + \frac{\gamma(k)}{c} q^e_z(k)(k-\tau(k)-1) \tag{2.3.8}$$

where

$$\tau(k) = floor\Big(\frac{(C_z - n_z(k))l_{veh}}{N^{lane}_z v^{free}_z c}\Big) \tag{2.3.9}$$

$$\gamma(k) = rem\Big(\frac{(C_z - n_z(k))l_{veh}}{N^{lane}_z v^{free}_z c}\Big) \tag{2.3.10}$$

with *floor(x)* in Equation (2.3.9) referring to the largest integer smaller than or equal to x, and *rem(x)* in Equation (2.3.10) is the remainder. The BLX model represents a more detailed model of the queue dynamics compared to the SFM approach. At the same time, such detail in the dynamics may lead to high computational power when applied to large-scale traffic network, especially in case of real-time optimization of traffic lights duration.

Another approach to applying the queuing theory is presented in the work of [6, 7, 81–85]. The basic variables and parameters used to describe traffic flow in the latter work are shown in Figure 2.9 and include:

- T_s is the sample time period which is assumed to be the total time taken from a green to red transition of any of the arms.
- $O_{i,t}$ is the occupancy which relates the proportion of time when a detector is occupied and therefore activated in a sample period, with respect to the total measuring period at arm i (%).
- $\gamma_{i,t}$ is the input intensity at arm i at cycle t, which is defined as the quantity of vehicles passing by a detector per cycle (uv/cycle). This detector is placed in the input or output lane, respectively as shown in Figure 2.9.
- $\zeta_{i,t}$ represents the queue length of arm i at cycle t (uv).
- $\delta_{i,t}$ is the queue indicator for arm i at cycle t and can take a value of either 1 (congestion) or 0 (no congestion).
- $\alpha_{i,j}$ is the turning ratio which gives a proportion between the vehicles passing through the junction from approach i to j to the total approach intensity.
- $\gamma_{O_{j,t}}$ represents the total flow of vehicles which an output arm will be subjected to, from the input arms (uv/cycle).
- $z_{i,t}$ is the green time ratio for arm i at cycle t and varies between 0 and 1.
- S_i is the saturation flow of arm i (uv/cycle).

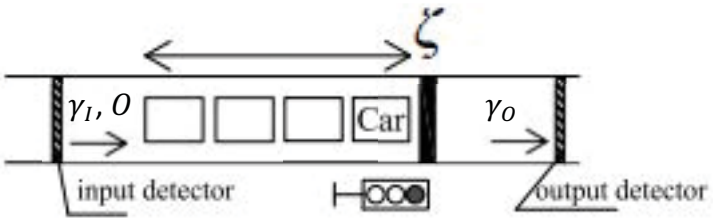

Figure 2.9: An example of a controlled one-way road [6, 7]

- κ_t and β_t are unknown time varying parameters, which define the occupancy relation to the previous occupancy and the queue length.

The queue indicator δ for an arm i depends on the saturation flow of the arm and the relative green time for the arm, as follows:

$$\delta_{i,t} = \begin{cases} 1, & \text{if } \gamma_{I_{i,t}} z_{i,t} + \zeta_{i,t} > S_i z_{i,t} 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.3.11}$$

A queue indicator of value 1 indicates that the vehicles inside an arm i at cycle t are greater than the saturation flow of vehicles which may exit the arm during the respective green time. A queue indicator of value 1 will thus result in a queue in arm i in the following sample period, whilst a queue indicator of 0 indicates that there will be no queue in arm i.

In the developed models [81–85], the dynamics of the queue length, occupancy and the flow in the output arms are given respectively as indicated in Equations (2.3.12), (2.3.13) and (2.3.14):

$$\zeta_{i,t+1} = \delta_{i,t}\zeta_{i,t} - [\delta_{i,t}S_i + (1-\delta_{i,t})\gamma_{I_{i,t}}]z_{i,t} + \gamma_{I_{i,t}} \tag{2.3.12}$$

$$O_{i,t+1} = \beta_{i,t}O_{i,t} + \kappa_{i,t}\zeta_{i,t} \tag{2.3.13}$$

$$\gamma_{O_{j,t+1}} = \Delta\zeta_{i,t+1} + \gamma_{I_{i,t}} \tag{2.3.14}$$

where $\Delta\zeta_{i,t+1} = -\zeta_{i,t+1} + \zeta_{i,t}$.

Equation (2.3.12) is based on the conservation principle of traffic flow, where the queue length at cycle $t+1$ depends on the previous queue length, $\delta_{i,t}\zeta_{i,t}$, the departed vehicles, $[\delta_{i,t}S_i + (1-\delta_{i,t})\gamma_{I_{i,t}}]z_{i,t}$ and the arrived vehicles $\gamma_{I_{i,t}}$ in a cycle. Equation (2.3.12) gives a linear expression to the number of vehicles exiting through an arm, denoted by $[\delta_{i,t}S_i + (1-\delta_{i,t})\gamma_{I_{i,t}}]z_{i,t}$. When the queue indicator takes a value of 0, the number of vehicles exiting an arm varies linearly with the queue length and the input intensity. This assumption holds until saturation is reached and the queue indicator takes a value of 1. When δ is equal to 1, the number of vehicles exiting through an arm is fixed to a constant value given by $S_i z_{i,t}$. This piecewise linear behaviour is represented in Figure 2.10 (blue plot).

Pecherkova *et al.* [6, 7] introduce a nonlinear expression to replace this piecewise linear characteristic for the number of vehicles exiting through an arm, given as $S_i z_{i,t}\left(1 - e^{-\frac{\zeta_{i,t}+\gamma_{I_{i,t}}}{S_i z_{i,t}}}\right)$. This exponential function represents a continuous approximation to the linear piece-wise throughput of the intersection as shown in black in Figure 2.10. This function eliminates the non-smooth change present in the piecewise linear function when the queue indicator switches from 0 to 1 or vice versa, at the cost of introducing nonlinear dynamics in the model. Equation (2.3.13) shows that the occupancy at any time is directly related to the queue length and previous occupancy. It is often assumed that the parameters $\beta_{i,t}$ and $\kappa_{i,t}$ do not vary with time for a given arm [81–85]. Equation (2.3.14) states that the flow of an output arm is equal to the change in queue length in one cycle and intensity of an input arm. For multiple input and output arms, Equation (2.3.14) changes to Equation (2.3.15) by taking into consideration the arm turning ratios:

$$\gamma_{O_{i,t+1}} = \sum_{i=1}^{\wp} (\alpha_{j,i}(\Delta\zeta_{j,t+1} + \gamma_{I_{j,t}})) \tag{2.3.15}$$

where $\wp$ denotes the number of input arms. Hence,

$$\gamma_{O_{i,t+1}} = \sum_{i=1}^{\wp} (\alpha_{j,i}(\zeta_{j,t} - \zeta_{j,t+1} + \gamma_{I_{j,t}})) \tag{2.3.16}$$

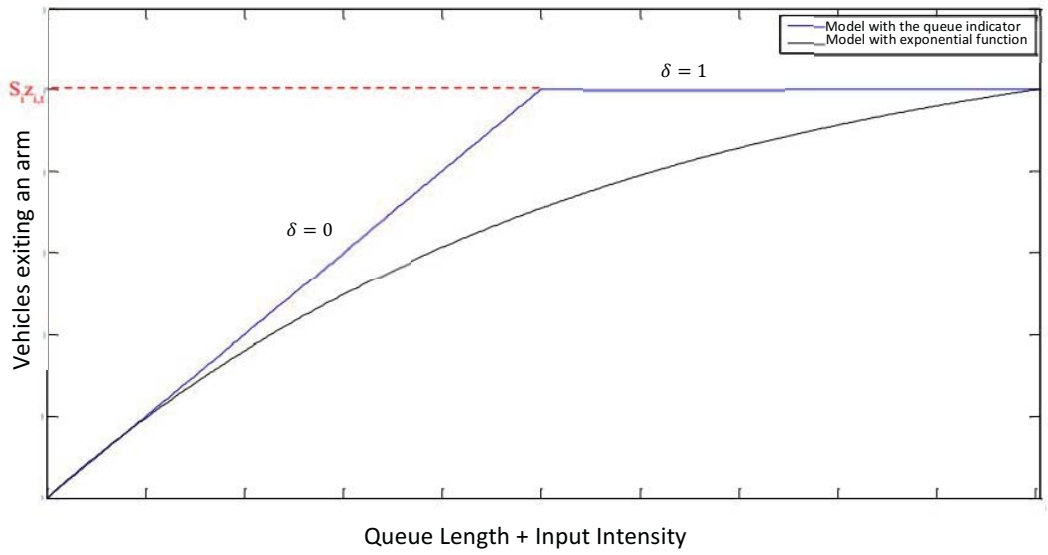

Figure 2.10: The linear and the non-linear models [8]

The input intensity is typically assumed to be a random walk during steady state periods and is thus given by Equation (2.3.17).

$$\gamma_{I_{i,t+1}} = \gamma_{I_{i,t}} + w_{i,t} \tag{2.3.17}$$

where $w_{i,t}$ is a stochastic disturbance representing the change in input intensity which occurs from one cycle to the other.

2.4 Overview of macroscopic data-driven models

Section 2.3 presented a review of existing traffic models suitable for inferring the dynamics of traffic behaviour. Three main classes of models were described. Table 2.1 summarizes the main features of each model with their advantages and possible limitations. It is important to realize that no single representation is ideal. All models have their advantages and limitations. On the contrary each

Table 2.1: Summary of advantages and limitations of traffic models suitable for inferring the dynamics of the traffic behaviour

Advantages and limitations of traffic flow models		
Model	**Advantages**	**Limitations**
CTM	• directly applied to represent urban traffic dynamics; • little knowledge of underlying traffic parameters;	• modelled on a spatial grid of homogenous cells; • possibly not applicable for online traffic control. • excessive computational cost for high dimensional networks
SFM	• simple model to represent; • low computational complexity.	• valid for saturation traffic scenarios only; • knowledge of underlying traffic parameters.
Queuing Theory	• applicable to all traffic conditions; • represenation of dynamics are more accurate than SFM due the the simplification introduced in the latter approach [4]; • represenation of linear and nonlinear functions for vehicles exiting through an arm; • low computational costs.	• knowledge of underlying traffic parameters • linear controllers not suitable when representing nonlinear functions for vehicles exiting through an arm.

model, based on its characteristics, is better suited for particular situations, depending on the characteristics of such traffic situation. For example for low dimensional traffic networks, CTM offer an excellent framework. However, online traffic control cannot be provided through this model because it requires observations at each cell, resulting in a lot of measurements. If online traffic control is one important requirement, the Queuing Theory approach seems the most appropriate choice. For low computational costs, the Queuing Theory or SFM can be applied.

Furthermore, the model equations presented in Section 2.3.3 can be represented in discrete-time stochastic state-space form because the use of the matrix notation greatly simplifies the mathematical representation of multiple-input-multiple output system for a signalized traffic junction. For example the model equations (2.3.12), (2.3.13) and (2.3.14) for arm i are represented in state-space form as in [84] as follows:

$$\begin{aligned}
\begin{bmatrix} \zeta_i(t+1) \\ O_i(t+1) \end{bmatrix} &= \begin{bmatrix} \delta_{i,t} & 0 \\ \kappa_i(t) & \beta_i(t) \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ O_i(t) \end{bmatrix} + \begin{bmatrix} -\delta_{i,t} \cdot S_i - (1-\delta_{i,t}) \cdot \gamma_{I_{i,t}} \\ 0 \end{bmatrix} z_i(t) + \begin{bmatrix} \gamma_{I_{i,t}} \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \\
\begin{bmatrix} \gamma_{O_i}(t) \\ \tau_{O_i}(t) \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ O_i(t) \end{bmatrix} + \begin{bmatrix} \zeta_i(t) + \gamma_{I_{i,t}} \\ 0 \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix}
\end{aligned} \tag{2.4.1}$$

where the state variables in this model comprise of the queue length in arm i denoted as $\zeta_i(t)$ and the occupancy $O_i(t)$. The sensor measurement vector comprise of the outflow given by $\gamma_{O_i}(t)$ and $\tau_{O_i}(t)$ which are simply the state variables $O_i(t)$ corrupted by the additive noise $v_1(t)$ and $v_2(t)$. The input is given by ratio of the green signal time to the total cycle time, given by z_i for arm i. κ_i, β_i and S_i constitute the model parameters. $[w_1 \quad w_2]^T$ and $[v_1 \quad v_2]^T$ represent white, zero-mean, Gaussian distributed process and measurement noise respectively. In the thesis, these vectors are sometimes also denoted as $\mathbf{w}(t)$ and $\mathbf{v}(t)$, respectively.

The state-space model equation (2.4.1) can be expanded according to the number

of arms n in the intersection as given in [84] with model parameter $\alpha_{i,j}$, representing the turning ratio, that is, the ratio of vehicles turning from arm i to arm j in relation to the number of vehicles in arm i. To reflect the changing traffic conditions, the unknown state variables together with the unknown model parameters need to be estimated as will be discussed in the following section.

2.5 Online Joint Estimation Strategies

Joint state and parameter estimation (or dual estimation) refers to the problem of simultaneously estimating in real-time the state variables, such as queue lengths, and the model parameters, such as turning ratios that may change throughout the day.

According to Cappe [96], online estimation refers to the idea of computing model estimates on-the-fly, without storing the data, by continuously updating the estimates as new observations become available. Online algorithms have the potential to speed up learning by making updates more frequently and recursively [97]. When using state-space models, the problem of performing combined online estimation of both states and model parameters adds another degree of difficulty. This problem was classically posed within a nonlinear estimation framework by augmenting the state vector with the unknown parameters of the state space equations leading to a nonlinear set of equations. Early work by Kopp and Orford [98] proposed the use of an Extended Kalman filter (EKF) to solve this nonlinear filtering problem. Wang *et al.* [99] made use of EKF to estimate online traffic states and parameters for freeway traffic conditions. However, the EKF, while being computationally feasible, is prone to divergence and also requires the analytical derivation of partial derivatives [100].

Several approaches have been proposed to mitigate the EKF divergence problem. Schmidt [101] and Neal [102] proposed the addition of terms to the gain of the Kalman filter. However, such approaches require extensive tuning of the Kalman filter gains and can add to the computational burden, which is a disadvantage in practical online applications. Magill [103] and Hilborn and Laini-

otis [104] approached the problem using decision theory by assuming that the unknown parameters exist within a finite set and applied parallel Kalman filters to estimate such parameters. This approach can be impractical for certain applications, as it can be difficult to represent all potential numerous systematic behaviours with a finite set of model parameters [105]. Wan and Nelson [106] proposed an alternative method, where a pair of Kalman filters were run in parallel, one for state estimation and one for parameter estimation, i.e. a dual EKF method. Note however, that the error in each filter could be compensated for by the other, leading to a non-unique solution [107]. These observations, particularly divergence issues, motivated the development of Sequential Monte Carlo (SMC) methods, also known as particle filters. SMC methods were applied in several works for solving the optimal state and / or parameter estimation problem, such as in the work of Cappe in [108] and in the work of Wang and Work in [109].

Difficulties with nonlinear estimation algorithms motivated the development of joint state and parameter estimation algorithms from a linear perspective. Nelson and Stear [110], separate the estimation problem into two using two linear estimators: the estimation of the parameters of a linear system with no state noise, and the estimation of the states of a linear system given the estimated parameters. Although this method achieves combined parameter and state estimation for multivariate linear discrete-time systems, the coupling of the covariance equation for the state estimates with the parameter estimates was not tackled in [110]. The Maximum Likelihood (ML) principle also separates the estimation problem into two interconnected linear problems: one for the state and the other for the parameters and can also be applied for joint state and parameter estimation as in [111]. The solution is obtained by iterating between two systems of linear equations [111].

Maximum likelihood (ML) is a general-purpose estimation method with attractive properties. It provides a consistent approach to parameter estimation problems. This means that ML estimates can be developed for a large variety of estimation situations because they have desirable mathematical and optimality properties [112]. Despite the advantages of ML methods, their practical deploy-

ment is not always straightforward, due to non-convex optimization problems often faced [113]. An elegant and powerful method for finding ML solutions is the Expectation-Maximisation (EM) algorithm [114, 115], as applied in [113, 116]. In the latter works, the classical EM is applied based on an iterative off-line batch estimation and is not appropriate for online estimation. Despite the successful application of EM methods to several fields such as image processing [117], speech recognition [118] and various applied statistical environments such as epidemiology [119], their application to traffic flow models is very innovative.

The earliest approach to online parameter estimation from noisy data follows the stochastic approximation method posed as a single variant estimation algorithm by Robbins and Monro [120]. Blum in [121] extended the Robbins-Monro algorithm to solve multivariate problems. Furthermore, Spall proposed a derivative-free multidimensional stochastic approximation [122, 123]. This approach has been used, with some variations, in many different applications such as for tidal models [124] and for ranking in machine learning and information retrieval problems [125]. However, according to the author's knowledge, this approach has never been applied to transport applications as proposed in this thesis.

Moreover, very few publications report on joint state and parameter estimation applied to signalized traffic junctions. In [81, 82, 84, 85, 126] only the estimation of model states was carried out online and the model parameters were assumed known a priori from past traffic measurements.

In [6, 7] joint state and parameter estimation is carried out by augmenting the states and parameters as one vector [127] and using the EKF. While being computationally feasible, this approach can possibly lead to divergence because of the use of EKF [128]. Moreover, the latter work does not carry out estimation of the noise parameters, this being assumed known. Furthermore, the work in [6] does not directly tackle the estimation of the turning ratios and saturation parameters of the traffic junction. Instead, model parameters that have no direct physical significance are introduced, which require prior tuning. In this work, parameters with physical significance are included in the model such as turning

ratios and saturation parameters. This is detailed in Chapter 5 which uses an EM estimation to obtain real-time estimation of the model parameters jointly with state estimation. This avoids the need for carrying out prior analysis and pre-tuning of the model to traffic conditions.

It is not uncommon to have traffic conditions in signalized junctions that are characterised by different dynamical regimes that change over abruptly in time due to for example traffic incidents or unanticipated network obstructions. Under such conditions the model structure and parameters are neither constant, nor varying slowly, but switch value abruptly; a phenomenon known as *jump* dynamics [51]. The joint estimation of model states and parameters is not straightforward when the system is prone to such *jump*-type dynamic changes. Hence, in this work, multiple model estimation techniques are proposed to handle such situations. These techniques are reviewed in the next section.

2.6 Multiple model estimation

Abrupt changing conditions in the traffic system are represented as different regimes or modes where each mode is further represented by its own distinct model. Hence the dynamics of the traffic system are not represented by one model but by a set of different models, one for each regime, that might be impossible to identify *a priori* for example in the case of a traffic accident. Consequently, a system that is able to detect in real-time any switching among the multiple regimes as well as learning and modelling the dynamics of each of these regimes while estimating the state variables that are not measured directly, will be of great benefit. This is called multiple model estimation [129] and it lends itself very well to such scenarios.

Several research efforts have been directed to multiple model approaches used in different applications [130–133]. Multiple mode switching detection has been tackled using two predominant techniques: deterministic approaches [130, 131] or stochastic techniques [132,133]. The essence of both deterministic and stochastic multiple mode switching detection is based on choosing the model which

yields some best defined estimation accuracy out of all models in the set at any point in time.

Of interest to this work, multiple model adaptive estimation has been previously applied for traffic incident detections for freeway conditions. For example, a multiple model extended Kalman filter was used in [134], where known changing conditions and known models parameters are assumed. A limited number of known changing conditions are assumed, hence resulting in a finite set of models. An EKF is proposed for each model to sequentially estimate the traffic states. The residual data is produced by comparing measured data with the estimated states and probabilities are assigned to each model based on how closely the residual characteristics match their respective anticipated values. Similarly, [135] applied an interactive multiple model ensemble Kalman filter, with a known number of changing conditions and known models. An ensemble Kalman filter is proposed to solve the sequential state estimation problem and to accommodate the switching dynamics and nonlinearity of the traffic incident model. Unlike the work of [134], the Interactive Multiple Model algorithm (IMM) makes use of Markov chains to model the evolution of the transitions from one model to another. Furthermore, [136] and [137] applied multiple model particle filtering for the traffic state estimation to improve the accuracy of the estimate when data is limited. [138] extends previous works by proposing an Efficient Multiple Model Particle Filter (EMMPF) that uses a single sample in each particle filter to infer the correct model and then all model particles are evolved forward in time using the most likely model determined in the selection step.

All the above works on traffic estimation and incident detection which made use of the extended Kalman filter and particle filter have been applied and performed well in freeway traffic conditions. In this thesis, a novel stochastic multiple model adaptive estimation algorithm is developed and applied for urban signalized traffic junctions (Chapter 7), where, similar to previously cited traffic literature, the algorithm needs to learn the traffic regimes and determine which mode is active at any given time. However, whereas the reviewed literature assumed that the monitoring data contained a limited number of previously mod-

elled regimes, this work will not be limited to previously known modes. As proposed in [51] for general systems, the algorithm is adapted for urban traffic junctions and is able to learn the various modes in real-time and hence automatically configure and grow its model set if new modes are detected. Furthermore, an online joint state and parameter estimation technique is applied to estimate the traffic states, apart from the model parameters. In this work, noise parameters are not assumed known a priori and will be estimated online, avoiding the need to calibrate such parameters based on the type of sensors present in the junction.

The proposed multiple model adaptive estimation algorithm makes use of an online dual estimation algorithm to jointly estimate in real-time traffic states such as queue lengths, occupancies and flows, as well as the model parameters such as turning ratios, saturation flows and noise parameters. The approach uses an EM algorithm, modified for real-time estimation, with a Kalman filter implementing the expectation step and a multivariate gradient-based approach for the maximisation step, as proposed in Section 6.3 and in [139]. The algorithm computes residual data by comparing the measured information with the estimated states and assigns probabilities to each model in the set. A new mode is learnt if the residual characteristics do not match any of the learnt modes in the set as will be discussed in Chapter 7.

2.7 Conclusion

This chapter has presented a critical literature review of existing traffic models suitable for inferring the dynamics of traffic behaviour from observations so as to model traffic flow within a junction. The advantages and limitations were discussed for each of the three modelling paradigms presented. Further insight to the choice of model representation that best models traffic behaviour in typical intersections in Malta will be addressed in Chapter 4.

Moreover, this Chapter presented a review of online joint estimation techniques for model states, parameters and noise, together with multiple model estimation techniques for *jump* dynamic systems. These techniques were adapted to de-

sign new algorithms for urban traffic junctions. Further insight on the proposed estimation algorithms, will be addressed in Chapters 5, 6 and 7.

Chapter 3

Traffic Signal Control Methods

The primary objective of a traffic signal plan is to move vehicles through an intersection safely and efficiently. This plan should accommodate real-time fluctuations in the traffic demand to reduce congestion. This Chapter will review several traffic signal control methods so as to accommodate changing traffic demand. This review will aid in the familiarization with, and subsequent development, of a novel traffic control system for urban signalized junctions.

3.1 Traffic Signal Control

The general aim for traffic signal control is discussed by Wood in his work in [140]. Wood classified the objectives of urban traffic management and control based on tactical considerations and more strategic ones. Tactical traffic management is important to ensure efficient operation of the junction with current and expected arrivals of traffic. On the other hand, strategic traffic management is broader and can include priority to different groups of travellers such as buses or pedestrians.

Such objectives can be achieved through the local operation of signals or coordinately throughout the network. An important control decision for operating signal controllers, identified by Heydecker [141] is the duration of the green indication. A substantial body of work has been targeted to the study of intelligent traffic measures which are adaptive to the traffic demand, thus aiming for traffic responsive systems, whilst optimizing the traffic signal plans accordingly by optimizing the green signal duration. Such intelligent traffic control plans aid

in controlling the flow of vehicles at intersections within the network, while improving the ability of a traffic signal to efficiently serve vehicles, reduce traffic delay and thus increasing the average network speed. This Chapter reviews important literature in this field.

3.2 Review of Traffic Signal Control Methods

Several traffic control techniques are presented in literature. These include *non-optimization based methods and optimization based methods*, *reservation and market-based system*, *slot-based system*, *car-to-car communication*, *artificial intelligence* and *autonomic systems* as will be discussed in the following sections.

3.2.1 Non-Optimization Based Methods and Optimization Based Methods

Two of the control techniques predominant in literature include *non-optimization based methods* and *optimization based methods* [1]. Research on such methodologies started before the 1980's, but were commercialised in the 1990's.

The *non-optimization based methods* use a set of heuristic rules to relate signal timings to traffic conditions. The London Department of Transport implemented a microprocessor based traffic signal controller for isolated, linked and urban traffic control installations in 1984 [142]. Such a controller is used to estimate when the flow-rate over the stop-line falls below the saturated flow. The stop-line is the line appearing at the end of the road for vehicles to stop. A stage will be extended until vehicles are no longer detected during a critical time period on any link that will lose right of way. Moreover, such controllers were also reviewed in [143] where acyclic traffic controllers were designed by the Centre of Transport Studies in University College of London. These controllers use similar heuristics within a phase-based framework where such controllers can alter green indication and select phases according to their traffic demands. Although non-optimized based systems are simple to implement, these might not guarantee the optimal use of the available network capacity [1].

3.2.1.1 Off-line optimization based methods

On the other hand, the control objectives of *optimization based methods* are commonly set to optimise some measures of generalised control performance over a time period, such as traffic flow, queue lengths or traffic delay. In the early stages, many publications were focused on optimization based methods which operate off-line, using either historically measured data or assuming that the mean rate at which vehicles arrive at an intersection is constant. In 1957, Webster [144] developed an approximate delay formula with traffic arriving at a uniform rate. An unconstrained formulation was developed to minimise approximated average vehicle delay. Furthermore, Allsop [145,146] developed a traffic simulator program for calculating traffic signal settings by minimising delays for fixed-time traffic signals at a single road junction. Linear convex programming was used by Allsop [146] to find a solution for a constrained formulation problem where the main aim was to maximise reserve capacity and minimise average delay rate. Similar to Webster's method, this required pre-specified stage sequenced and inter-stage structures. By taking individual phases into consideration in the design, more flexible signal timing allocation can be achieved. Moreover, in Little's work in [147], optimization of traffic flow is obtained by traffic signal synchronization, so that a car starting at one end of a main artery and travelling at preassigned speed can keep on going without stopping for a red light. The portion of a signal cycle for which this is possible is called the bandwidth for that direction. A mixed-integer linear program is formulated to find the common signal period, between two signalized intersections, speeds between signals, and the relative phasing of the signals, in order to maximize the sum of the bandwidths for the two directions.

In Improta and Cantarella's work in [148], a Binary-Mixed-Integer-Linear-Program (BMILP) is implemented to identify the optimum phase sequence to be used at a given intersection. Heydecker [149] showed that in case of more complicated junction layouts, the duration of green signal indications for phases can be used as control variables.

Although, in recent years, data availability and computational efficiency are increasing, *off-line optimization based methods* still remained popular. Ribeiro [150] applied fixed time plans calculated by a commercialised traffic simulator TRANSYT. Ribeiro proposed a novel technique called Grouped Network for TRANSYT to calculate signal timing plans that can be used during specific periods of the day (morning peak, evening peak, etc.) and are less sensitive to variations, or fluctuations, in traffic flows. A computational approach was developed by Smith *et al*. [151] which takes into consideration the time-of-day approach where a day is segmented into a number of intervals and a different timing plan is applied at each interval, assuming a constant arrival rate of vehicles at each timing plan. This model aims at minimizing the mean of the delays per vehicle as well as the variabilty of performance. Moreover, Yafeng [152] developed a robust optimal traffic signal timing making use of the 90^{th} percentile traffic flow observation as an average flow measurement to optimize signal timings. Wong *et al*. [153] used clustering techniques to determine common time intervals during the day which share common traffic conditions. The time-of-day approach was applied similar to [151]'s approach and a common timing plan was applied to common time interval clusters.

All publications developed in *off-line optimization based methods* rely on either historical data or on average traffic conditions, such as the average arrival time of vehicles. Typically, such conditions do not accurately describe the current traffic flow, as this normally changes over time. Signal plans that were developed using historical data were shown to suffer from a 3% decay in performance per year [154]. Hence *online optimization based methods* highly contribute to the development of intelligent signal control which relies on realistic traffic performance in time.

3.2.1.2 Online optimization based methods

Online optimization based methods consider dynamic traffic flow for time signal optimization. The previous mentioned work aimed at minimising traffic delays or maximizing intersection capacity. By contrast, most of the works on online

optimization focus on minimizing queue lengths within junctions. Haddad *et al*. [155] developed an optimal steady-state controller for isolated traffic intersections aiming at reducing the queue lengths within such intersections. In addition, Ioslovich *et al*. [156, 157] represented a continuous traffic model through the use of differential equations and the methods of optimal control theory were applied to develop an analytic solution to minimise queue lengths within an intersection. Ioslovich *et al*. [156] represented the queue lengths accumulating within an intersection using Gazi's continuous differential equations presented in [75]. The optimal minimum time control for an isolated intersection was found using the Pontryagin Maximum Principle [158]. Aboudolas *et al*. [159] further discuss optimization based methods for traffic signal control in large-scale congested urban road networks.

In [160–162], a binary choice logic is implemented, where time is divided into successive small intervals and a binary decision is made either to extend the current green signal timing by one interval or to terminate it depending on any sensed vehicles. More examples of such an approach include Miller's algorithm [163], Traffic Optimization Logic (TOL) [164], Modernized Optimization Vehicle Actuation Strategy (MOVA) [165] and Stepwise Adjustment of Signal Timing (SAST) [166]. However, the drawback of binary choice logic is that it considers a very short future time interval of 3 to 6 seconds. For longer length decision-making, other optimization based methods need to be considered such as dynamic programming, approximate dynamic programming or linear quadratic control as will be discussed in the next sections. In addition, a rolling horizon concept can also be included as in [167] to determine optimal signal switching sequences for a future timing period namely model predictive control.

3.2.1.3 Dynamic Programming

For longer length decision-making, Dynamic Programming (DP) [17–21, 168–172] can be applied. A complex problem can be decomposed into a series of sub-problems with discrete time steps between them and a number of state vari-

ables at each time step to specify the sub-problem. Bell *et al*. [173] described the state of a traffic signal control system as: the state of traffic and the state of the controller. The state of traffic at a junction can be described by the number of vehicles queuing in the links and the arrival rate of vehicles in the near future. The state of the controller can be specified by the green signals, the times of green completion and the expiry of any permitted duration. In his work, Bell *et al*. indicated that normal backward dynamic programming techniques are not much adequate for real-time control. This is due to unnecessarily large number of state sequences and the commencement of calculations at the end of the look-ahead where in this case, information on arrivals is least certain. These look-ahead policies make decisions by explicitly optimizing over some horizon by combining some approximation of future information, with some approximation of future actions. This also reaffirms conclusions drawn in [15, 17, 20].

DYPIC [169] is a DP approach and is used only for analytical purposes. The computational difficulty of a DP solution restricts the implementation of DYPIC for engineering purposes [174]. Robertson and Bretherton in [169] proposed a quadratic function to approximate an exact value function. A heuristic solution based approximation function was developed which makes use of a rolling horizon. A project horizon is predetermined which consists of N time intervals. Traffic flow data are measured for the first H intervals, known as the *head* period, and also traffic flow data are estimated from a model for the next $N-H$ intervals, called the *tail* period for the first iteration. An optimal policy is calculated for the entire horizon, but is only implemented for the *head* period. Finally, when the head period expires and new information becomes available, the project horizon is shifted into the future by R intervals, considered as the roll period and the same process is repeated.

PRODYN [20] adopts also a DP approach with a heuristic solution. It extends over Robertson and Bretherton's heuristic solution [169] to distributed network control while applying a rolling horizon approach. This approach aggregates state variables into a few subsets, and the value of being in a subset is only evaluated when it is actually being visited. PRODYN calculates the optimal trajectory

of control policy in a planning horizon of 75s by evaluating all the subsets that are visited. The process is then repeated as the rolling horizon rolls forward one step in time. In [20], it was shown that PRODYN yields an average reduction in total travel time of 10%. UPTOPIA [21] also makes use of the rolling horizon concept. In addition, a hybrid control system is implemented combining online dynamic optimization and offline optimization. A system hierarchy is constructed made up of an area level and a local level. A reference plan is generated by the area controller, this reference plan is adapted by the local controllers by dynamically coordinating signals in adjacent intersections. The rolling horizon approach is 120s long and the process is repeated every 3 seconds. OPAC [19] and RHODES [22] are another two promising American systems that also apply the rolling horizon concept and are based on dynamic programming. These have been implemented primarily for individual intersections with an extension to networks. Both systems have their limitations. OPAC optimization methodology can be classified as trial-and-error enumeration that cannot guarantee a globally optimum solution [19]. RHODES requires a fixed sequence of phases and a much longer projection time. Nevertheless, both systems have demonstrated the potential of using DP to optimize the signal plan for a certain future horizon [22].

Liu *et al*. [171] implemented a reverse causal-effect modelling approach for an oversaturated intersection. Smooth continuous functions of time are applied to model both traffic arrivals and departures, as if there were no interruptions to traffic flow from signals. This approach has the added advantage of applying differential calculus in optimization unlike step functions. Moreover, this approach also simplifies future extension to a system of intersections. A dynamic linear programming model is also used to maximize the total vehicular output from the intersection during the entire period of congestion subject to prevailing capacity and other operational constraints.

DP is based on estimation of some measures of the value of state (also known as value function). The previously mentioned works, all implement a deterministic dynamic program solution, where the state at the next step is completely determined by the state and policy decision at the current step. A stochastic dynamic

program solution is also implemented [175, 176] where the state at future steps is not completely determined by the state and policy decision at the current step.

In both deterministic and stochastic dynamic programming, the simplifications in the implementation of a complex problem are countered by the computational requirement associated with the size of the state space. The intensive computation requirement hinders DP to real-time operation, where possible actions have to be evaluated within a short time in real time applications. The computational requirements for finding an optimal solution numerically and storing the optimal value function over the entire state space is in exponential order to the size of the state space [177, 178]. Hence for most problems of practical interest, DP remains computationally infeasible. Additionally, the availability of future traffic information such as future arrivals is very unrealistic. In order to bring the principles of DP to real-time control, a reduction in the dimensionality of the control problem needs to be introduced and secondly, accumulating limited sensor information progressively to improve knowledge of the underlying control process will also aid in traffic information knowledge.

3.2.1.4 Reinforcement Learning

Another learning technique which is closely related to DP is reinforcement learning. Reinforcement learning is used to estimate values for the parameters of specified functional relationships between actions and their effects [179]. In reinforcement learning, a learning agent is learning from a control process while using DP to estimate values of performance. This agent must discover which action yields the best performance by trying them out. Its actions may affect not only the immediate performance but also the next state and, through that, all subsequent performance. Trial-and-error attempts and delayed effects can also result when discovering which action yields the best performance. Another limitation of the reinforcement learning is that the system heavily relies on the scenario being trained on.

Several works have been directed to the application of reinforcement learning

to traffic signal control plans including works in [22, 180–196]. A difficult aspect in applying reinforcement learning to traffic light control is the selection of features, that is the number of states, each of which describing the exact situation around an intersection, is huge [197]. This implies that a form of featurization and subsequent function approximation is needed to even represent the value function for these settings. Recent years, however, have seen the development of reinforcement learning methods through deep q-learning, that use deep learning techniques [198] where deep neural networks are trained as function approximators in a reinforcement learning setting. Deep reinforcement learning has been recently applied in the works in [199–202] with application to autonomic traffic lights control and was also shown to be potentially promising.

Similar to artificial intelligence and genetic algorithms, this approach is computationally intensive, as many simulations need to be performed. The computational power requirement increases with the network size. With sufficiently large networks, high computational real-time costs may be involved [180, 181].

3.2.1.5 Approximate Dynamic Programming

Approximate Dynamic Programming (ADP) is derived from DP, whereby its aim is to significantly reduce the computational requirement required by the original DP problem [174]. ADP adopts a forward process, using information such as arrival of vehicles which becomes available between time steps to help in the decision-making. Monte Carlo simulation can also be used to make an optimal decision without detailed information and simulate policy approximations and value function approximations [203]. Furthermore, the approximations that are used in ADP improve progressively, because the exact values of functional parameters are not known apriori. The ADP algorithm updates the parameters of the approximation function upon each observation of state transition.

Cai [174] proposed an ADP solution for an isolated intersection, where in his work the ADP approach uses a linear function to approximate the value function and thus reduce the dimension of the state-space to the size of a few functional

parameters. A forward rolling process is adopted that makes use of limited online formation, 10 seconds of future vehicle arrivals. To update the approximation, perturbation learning is used. This perturbs the system with incremental changes in state variables. Cai's results showed that the performance of this ADP controller is comparable with existing adaptive control methods. The same ADP approach was extended by Heydecker *et al*. [204] to more complex intersection layouts, with the results indicating significant vehicle delay reductions when compared to the fixed-time plans.

Li *et al*. [205] presented another approach to develop ADP solutions to traffic signal control. This approach uses an action dependent heuristic DP method which was first developed by Werbos [206]. Two neural networks are used, one to map state to action, and the other one to map state to discounted future values. Cai [207] generalised the ADP approach to isolated intersections. Artificial neural networks were applied to approximate the value function, and reinforcement learning was applied as the learning algorithm. The artificial neural network based approximation structure can approximate both linear and nonlinear value functions. This is useful for traffic applications because many delay functions [144, 208] are nonlinear.

Nevertheless, practical use of ADP remains limited by a lack of systematic guidelines for implementation and for approximating value functions. This is reflected by the amount of trial and error involved in each of the success stories found in the literature and on the difficulty of duplicating the same success to other applications [209].

3.2.1.6 Linear-Quadratic Optimal Control

For the derivation of multivariable regulators, the urban traffic control problem can be formulated as a Linear-Quadratic (LQ) optimal control problem [210–213]. The formulation of an LQ controller for urban traffic required a generic mathematical model for the traffic flow, upon which an optimal control approach was designed that aims at minimizing and balancing the link queues so as to

reduce the risk of queue spillback. The optimization problem considered in [210–213] was of the Quadratic-Programming (QP) type with linear constraints. Nominal green times were considered, which were optimal for a given historical demand. The aim in these works was to modify in real-time these nominal values depending on the demand such that the capacity of the controlled network is utilised in a balanced way whilst reducing the queues in the junction.

In [211] two evaluation criteria were calculated for comparison: i) the total time spent and ii) the relative queue balance, while in [210, 212, 213], the evaluation criteria was based on minimizing the queue length. In [213] a Hybrid Signal Control Strategy is developed which make use of either the LQ regulator or a demand-based control law to calculate in real-time the green times of each junction. The controller was chosen according to a saturation criterion that depended on the flow measurements in the junction.

For the LQ controller in the works of [210–213] a quadratic optimization criterion was considered. The control law is given by:

$$g(k) = g^N - \mathbf{L}x(k) \tag{3.2.1}$$

where k is a discrete time index reflecting corresponding signal cycles, g is a vector of green times of all stages, g^N is the vector of nominal green times corresponding to prespecified fixed signal plan for the network; x is the vector containing the number of vehicles in the links, $\mathbf{L}$ is the gain matrix. $\mathbf{L}$ is either computed offline or else kept constant to reduce the computational complexity involved in estimating it in an online manner.

Although the LQ is computationally efficient, it lacks the robust property to cope with parameter perturbations and external disturbances. Moreover, the LQ controller does not allow for direct consideration of constraints. Furthermore, for a real-time application, the QP algorithm may be embedded in a rolling horizon leading to a model predictive procedure as will be discussed in the next section.

3.2.1.7 Model Predictive Control

Model Predictive Control (MPC) [214] is a methodology that implements and repeatedly applies optimal control over a prediction horizon in a rolling horizon way. In each control step, only the first control sample of the optimal control sequence is implemented. Next, the horizon is shifted one sample and the optimization is restarted again with new information of the measurements. The optimization is redone based on the prediction model of the process and an estimate of the disturbances.

MPC can predict and find the optimal solution for the future. Different from optimal control, MPC has the ability to deal with the uncertainty of the process, which can be caused by the unpredictable disturbances, the slow variation over time of the parameters and model mismatches in the prediction model. Since the optimal control effort is updated regularly, any model uncertainty can be dealt within a fair range. Hence, accuracy of the model used for the prediction can be compensated with regular updating of states and model parameters [215]. To maintain the on-line computational feasibility, the optimization is calculated over a finite time span instead of up to infinity, called prediction horizon. MPC can also deal with multi-input and multi-output problems with constraints. Another advantage of MPC is that one can easily select and replace the prediction model based on the control requirements.

MPC is a multivariable control algorithm that uses:

1. past control moves
2. an internal dynamic model to represent the process and
3. an optimization cost function J with a receding prediction horizon, to calculate the optimum control moves.

Hence the control process of the MPC controller can be described by the prediction model, the optimization problem and the rolling horizon.

An example of the cost function J is given by:

$$J = \sum_{t=1}^{N} w_{x_t}(r_t - x_t)^2 + \sum_{t=1}^{N} w_{u_t}\Delta u_t^2 \tag{3.2.2}$$

where x_t is the t^{th} controlled variable, r_t is the t^{th} referenced variable, Δu is the change in control variable u from the previous time step $(t-1)$ to the current time step t, w_{x_t} is the weighting coefficient reflecting the relative importance of x_t and w_{u_t} is the weighting coefficient penalizing relative big changes in Δu_t, N is the prediction horizon.

An optimization problem can be formulated based on the optimal criteria required in the model, e.g. reducing queue length of vehicles within an intersection. The optimal control input is derived from the optimization step and is transferred to the process and implemented. When estimating the next control step, the prediction model is fed with real measured traffic states, the whole prediction horizon is shifted one step forward and the optimization starts all over again. This rolling horizon scheme closes the control loop, enables the system to get feedback from the real traffic network and enables the MPC controller to be adaptive to the uncertainty and disturbances.

Several works have been directed to the design of an MPC controller for traffic networks [5, 44, 79, 216–237]. MPC has many advantages, like robustness to disturbances, long-term sight and ease of dealing with constraints. However, despite these advantages, the real-time computational complexity of MPC increases exponentially, when the network scale and the predictive time horizon grow [227].

Systems controlled by MPC as applied to signalized junctions have applied a quadratic formulation of the objective function [5,210–214,227]. Hence quadratic programming is used to optimize the objective function. The quadratic formulation represents a smooth convex function with a unique minimum. The quadratic formulation weights large cost functional values much higher than small values.

Therefore, systems controlled by MPC, based on a quadratic formulation, usually show a non-negligible overshoot in the step response [238].

3.2.2 Reservation and Market-Based System

Contrary to ADP, a reservation-based approach can easily be generalised to different junctions as no trial and error attempts are involved. A reservation-based approach does not rely on approximating value functions, but relies on communication. Specific time slots are reserved by vehicles that communicate with upcoming intersections, so as to pass through [239, 240]. Such a system aims at maximizing the efficiency of moving vehicles through intersections with minimal centralized infrastructure. Intersections are outfitted with a wireless communication system, for communicating with oncoming traffic and giving permission for vehicles to pass. Vehicles must only traverse intersections when allowed to, but otherwise are free to decide for themselves how to drive. Each car is treated as an autonomic agent, and in particular need not surrender control to any centralized decision maker [239, 240].

The interaction between the vehicle agent and the traffic light is straightforward. First, the vehicle agent calculates when it will reach the traffic light given its current velocity. The vehicle agent then sends a message to the intersection informing it of the time at which the vehicle is expected to arrive. The intersection then responds with the range of times during or after the time specified by the vehicle agent, at which point the lights will be green. The vehicle agent can then make any adjustments necessary to ensure that the vehicle enters the intersection when the lights are green.

Given the increase of autonomicity, several works have been directed to the study of reservation-based systems [239–242]. These autonomic systems heavily rely on cooperative agents, coordination and communication to share or exchange information to achieve better system-wide performance [34, 42–46]. A multi-agent approach has been built, where, for each intersection, there is a corresponding intersection manager and for each vehicle, there is a vehicle agent. Intersec-

tion managers are responsible for directing the vehicles through the intersection, while the vehicle agents are responsible for controlling the vehicles to which they are assigned. This approach has been simulated in [239–242] for the full fleet of vehicles with simplified characteristics of real-world intersection traffic. Vehicles are not allowed to turn and all vehicles travel at the same speed. Their results indicate that the Reservation-based system allows traffic to flow through the intersection more efficiently than the traffic light mechanism with fixed plans.

On the other hand, in a market-based approach, signal coordination is determined by a form of currency held by vehicles within the network. To access the intersection, vehicles must bid and pay [243, 244]. In [244], the work extends over Dresner and Stone's approach [239–242] to networks of intersections. Market-inspired control methods are implemented as a paradigm for urban road traffic management. Vehicle agents trade with infrastructure agents in a virtual marketplace, purchasing reservations to cross intersections when commuting through the city. This work concludes that in situations of similar traffic load, an increase in the infrastructure's monetary benefit usually implies a decrease in the vehicles' average travel times [243, 244].

3.2.3 Slot-Based system

Similar approaches to reservation and market-based systems have recently been introduced, such as the slot-based system in slot-based intersections, whereby, vehicles are accelerated or decelerated such that they arrive at the signalized intersection when gaps in the conflicting traffic flows have been created for them [245]. However, a one-by-one service policy is not efficient at high vehicle arrival rates. Hence, platoons of vehicles are formed and all vehicles in the platoon are served before giving way to a conflicting flow. Different types of slot-based intersections have recently been proposed as in [246–249] and based on simulation results, it has been suggested that they might be more efficient than traditional traffic lights [246–248].

Even though, the reservation-based, market-based and slot-based systems require

no approximations, the systems demonstrated some serious safety flaws on an extremely advanced and precise automated driving system [250]. Moreover, high costs are involved in renovating aged vehicles to such communication standards.

3.2.4 Car-to-Car Communication

In car-to-car communication, also known as Vehicle-to-Vehicle (V2V) communication, vehicles are equipped with speed sensors or with Global Positioning System (GPS) units that can provide location information with accuracy, and transceivers to be able to send and receive information between vehicles. Similar to the reservation-based method, this autonomic system heavily relies on communication. The traffic signal control listens to the communication between vehicles, in Vehicle-to-Infrastructure (V2I) communication. It estimates the density of vehicles around it and adjusts the signal timings accordingly [251–254]. In [254] a clustering based data dissemination protocol is developed, where the direction information at the intersection can be computed a priori. At an intersection, vehicles can take up three different directions: straight (S), right (R) and left (L). Clusters S, R and L are formed on a particular lane at a distance of twice the radio range from the intersection. A clusterhead is elected to transmit the density information within each cluster to the traffic signal control. Once the packet is transmitted to the traffic signal control, the signal control would identify density information including details of lane, direction field, arrival time and cluster length. An optimum cycle length is defined based on the sum of lost time and the ratio of density to the cluster length. The lost time is the sum of the intergreen time for all the phases of traffic signal control, as no vehicles will cross the intersection during this interval. Simulation results of car-to-car communication showed that a better level of services was obtained in terms of the average waiting time at the intersections.

In [255], an algorithm is developed for estimating queue length using car-to-car communication by applying similar clustering methods as discussed previously. Moreover in [184], a learning method is developed with a road user based function to determine optimal decisions for each traffic light. This decision is

based on a cumulative vote of all vehicles waiting at an intersection where each car votes using its estimated gain of setting the light green. This way of co-learning allows the driver to choose the route with the lowest expected waiting time. In [256], the traffic light controller uses intelligent agents at isolated intersections. Machine learning algorithms are applied for the agents to learn a traffic control policy, based on the attribution of rewards, which may result in a slow convergence of the optimum timing pattern. To reduce the convergence times for obtaining an optimal solution, a greedy control algorithm is defined in [257] with a scarcity measurement approach. Since it adapts a greedy approach, this algorithm may not reach an optimal solution if too many fluctuations exist in the number of vehicles within the traffic network. Other works on car-to-car communication include [258–263].

Although such systems provide autonomicity within traffic system to reduce driving times, high costs are involved in renovating and upgrading aged vehicles to such communication standards. Moreover, more tests need to be developed in real world scenarios to ensure the safety of such a system on our roads. For example, failure of data communication may cause the whole system to collapse. Hacking and data security are another point of concern [264, 265].

3.2.5 Artificial Intelligence

Artificial intelligence methodologies, as applied to traffic control, can be classified into four distinct parts, namely *evolutionary computation*, *fuzzy logic*, *artificial neural networks* and
swarm intelligence. Reinforcement learning as discussed in Section 3.2.1.4 can also form part of artificial intelligence. The use of the four former parts in traffic control scenarios will be discussed in the following sections.

3.2.5.1 Evolutionary Computation

Several publications use evolutionary computation to optimize signal plans in a traffic network, such as in [266–268]. A set of chromosomes are used to represent the signal plans in a network. Following this, an evolutionary algorithm is

applied to each individual within the population, each time computing an individual's fitness. [269].

Literature has also shown that the genetic algorithm is a convenient and effective technique for the optimization of traffic control variables [270–272]. Similar to evolutionary computation, the genetic algorithm involves two basic steps:

- a selection mechanism that implements a survival of the fittest strategy;
- a genetic recombination of the selected chromosomes to produce offspring for the new population.

Recombination is affected through the genetic operators of crossover and mutation.

Evolutionary computation and genetic algorithms are highly computationally intensive, as many simulations need to be performed. The cost of simulation increases as the network size increases. With sufficiently large networks, real-time control may be impossible to achieve as the computation may not be completed in a suitable amount of time [273].

3.2.5.2 Fuzzy Logic

Fuzzy logic is adopted in traffic signal control to model a human expert's knowledge in traffic control based on inputs such as quantity of traffic on the arrival side and quantity of traffic on the queuing side, expressed in fuzzy terms such as *low*, *medium* and *high*. The output variable is the extension time needed for the green light on the arrival side. Based on the current traffic conditions, fuzzy rules derived from the expert's knowledge can be formulated so that the output of the fuzzy controller will extend or not extend the current green light time [274].

Before the late 1980s, research in the field of fuzzy logic-based traffic signal control was centered on simplified junctions or networks. In [275] a fuzzy logic for a single intersection of two one-way streets is implemented and optimal cycle lengths are obtained. Nakatsuyama *et al*. [276] implemented a fuzzy logic

controller in two consecutive junctions with one-way movements. Such works determined the extension and termination of green signal for the downstream intersection based on the upstream traffic. Kelsey *et al*. [277] implemented a fuzzy logic controller for a typical cross intersection and chose three inputs into the fuzzy controller including the average traffic density behind the green lights, the average traffic density behind the red lights and the length of the current cycle. Four membership functions were defined to describe the traffic densities, as zero, low, medium and high. Three membership functions were defined to describe the input current cycle time in seconds, as, short, medium and long. The fuzzy controller behaviour was simulated for one hour and its behaviour was compared to a controller with fixed time plans. The simulation results showed an increase in the average flow rate and a decrease in the average waiting time when compared to the fixed time plans.

Chiu *et al*. [278] investigated the inadequacy of an offline global optimization approach-based urban traffic control system in responding to unpredictable changes in traffic demand, as this system generates timing plans offline on a central computer based on average traffic conditions for a specific time of day. Fuzzy decision rules were applied for adaptive traffic control in a highly distributed architecture whereby the timing parameters at each intersection were adjusted using only local information and coordinated with adjacent intersections. Such approach resulted in significant reductions in waiting time and number of stops. A fuzzy logic-based traffic signal control was proposed by Hoyer *et al* [279], designed to tackle twelve main direction traffic flows of an intersection. They applied fuzzy rule based decisions to activate a two-state, three-state or four-state control. The simulation results showed a decrease in the average travel time when compared to fixed time plans applied to the same simulation example. Tan *et al*. [280] described the design and implementation of an intelligent traffic signal controller for a four arm isolated traffic junction. Two fuzzy input variables were considered: the number of vehicles on the arrival side and the number of vehicles on the queueing side. The output fuzzy variable was set to be the extension time on the green signal on the arrival side. The simulation results showed a decrease in the average travel time when compared to fixed time plans applied to the same

simulation example. Other works on fuzzy logic include [281–313].

Fuzzy control has been popular over the years in traffic signal control, having a simple process structure, with low maintenance cost, control adaptivity, fast evaluation time and savings in material costs. Decisions in fuzzy logic are usually made using a rule-base which can be developed by expert knowledge, trial-and-error or an automatic method such as a genetic algorithm [250]. However as noted in [314] an effective rule-base for complex intersections might be difficult to generate. The performance of the controller greatly relies on the effectiveness of the rules developed and it can be difficult to determine if the rules being used are helpful [250]. Moreover, evaluating the solution quality and finding optimality bounds is not a trivial task [315].

3.2.5.3 Artificial Neural Networks

Neural networks represent another approach for the development of traffic signal controllers as in [316–318]. Online training algorithms can be applied to estimate the neural network connection weights. For example in Spall's work in [182], Simultaneous Perturbation Stochastic Approximation (SPSA) was applied as the online training algorithm. In [182] and [286], a traffic signal controller using neural networks was developed for very simple networks with static traffic volumes.

Neural networks can be used with other combination of learning methods, such as dynamic programming [319] and reinforcement learning [320]. Zhiyong and Baiwu in [321] proposed a hierarchical fuzzy control method, and applied it to the arterial coordinated control. Its fundamental principle was to use neural networks to map fuzzy relations in arterial coordinated control for improving the precision of fuzzy controller. Similarly, this approach was also applied in [322] for isolated intersections. Performance deterioration can result when traffic conditions change, as these systems must relearn another effective method of control.

3.2.5.4 Swarm Intelligence

Swarm intelligence has also been applied to traffic signal control. Oliveira *et al*. [323] used the swarm principle of task allocation to select signal plans based on a theoretical pheromone stimulus emitted by waiting vehicles. With the insertion of the neighbours' influence on the agents' decision, group formation based on direct communication is formulated, prioritizing the global optimization, with a local view based on pheromone dissipation, focused on a more local optimization.

Swarm particle optimization is used in [324] to search for optimal signal plans within a traffic network. Other recent literature which applied swarm intelligence in real-time include the works in [325–328].

Similar to evolutionary computation, large amount of simulations are required and may require re-optimization with changing traffic conditions [324].

3.2.6 Autonomic Systems

Great progress has been achieved in traffic responsive control systems as discussed in the previously mentioned sections and as reflected by commercial traffic control software, such as MAXBAND by Little [147, 329], TRANSYT by Robertson [13], SCOOT by Hunt *et al.* [15], SCATS by Sims and Finlay [330], STAUKO by Boettger [331], OPAC by Gartner [19], PRODYN by Vincent *et al.* [20], CRONOS by Boillot *et al.* [332], RHODES by Mirchandani and Head [22] and UTOPIA by Mauro and Di Taranto [21], which make use of the previously mentioned control techniques. However the controller parameters of such methods are not tuned to changing traffic behaviour (for example non-typical demands) so as to ensure consistent and optimal control. Such methods are also unable to self-adapt to any network irregularities, such as roadworks, accidents and disturbing weather conditions such as flooding. To adapt to the changes in the network, these implementations use traffic surveillance technologies to monitor network conditions and to warn the commuters of any detected irregularities. On the other hand, an autonomic network is required to self-manage network be-

haviour under such conditions.

The autonomic paradigm is inspired by the human autonomic nervous system that handles complexity and uncertainties, and aims at realizing dynamic systems and applications capable of managing themselves with minimum human intervention. Autonomicity has been introduced to manage large-scale distributed computing infrastructures [39, 43, 333, 334]. To the author's best knowledge, it has only recently been applied to urban traffic networks [335] through the use of a spatially distributed set of controllers capable of self-stabilizing [336] and self-optimizing [25] the network behaviour. Three main types of autonomic architectures have been studied including: the centralized architectures [337] through the use of one single controller that controls the entire network; the decentralized architecture [338] which makes use of multiple controllers with each controller managing the traffic network within a predefined area; and the hybrid architecture which makes use of multiple controllers forming a hierarchical structure where centralization occurs at a higher level that commands the decentralized controllers in the lower level of the model [339], thus providing further coordination among the controllers.

The activation of the controllers in the network changes, depending on the system perceived real-time traffic conditions. Due to the diversity and the dynamic behaviour of traffic, it is required that the system adapts to the operational traffic dynamics and provides reasonably accurate actions within reasonable time instances so as to optimize the traffic flow in the network. Emulation of the human's autonomic behaviour is obtained by activating controllers that are more suited for dealing with specific traffic scenarios; that is, if a part of the network is subject to any irregularity, for example traffic accidents or flooding, the system intelligently activates specific controllers with the aim of recovering from these irregularities with little human intervention, thus ensuring the continuity of an efficient traffic management scheme [335].

Gershenson [34] presented an initial application, with an agent-based solution able to control traffic signals effectively. Cools *et al.* [42] tested the controller on

a modelled intersection area of the city of Brussels. Lammer and Helbing [340] tested this approach to hypothetical grid-like networks.

3.3 Conclusion

This chapter has presented a review of traffic signal control methods suitable for controlling traffic signal plans at intersections. Table 3.1 summarizes the main advantages and limitations of each approach. Autonomicity has been recently introduced and there has been little research into the many challenges of implementing such behaviour in transportation systems. The presented works on autonomicity heavily rely on cooperative agents that share or exchange information, resulting in a timely-coordination process [47]. In addition, the traffic system may sometimes exhibit features of a typical *jump system*, where the system structure or parameters are prone to change value abruptly due to environmental changes, such as network structural changes caused by flooding, road accidents etc. [51]. Hence the complexity of communication may hinder the performance of such systems because of the slow response to detect abrupt changes in the system parameters and to immediately update the model parameters. The following work in this thesis will aim to address such research gaps, thus leading to novel contributions for autonomic control, as will be presented in Chapters 8 and 9.

Table 3.1: Summary of advantages and limitations of the approaches to control

Approach	Advantages	Limitations
Non-Optimization and Optimization (Optimization: Adaptive)	• easy to implement; • computationally efficient; • low storage requirements; • constrained optimization is possible; • robust in control; • Optimization methods include: DP, ADP, LQ and MPC; • DP and ADP suitable for long term decision making. • ADP results in reduction in computation requirements when compared to DP; • Deep Reinforcement learning potentially promising to autonomic traffic lights control. • Ease of dealing with constraints when applying MPC.	• control performance is not optimized in non-optimization approach; • limitations of the search space to reduce computation time in optimization approach; • offline approach relies on either historical data or on average traffic conditions; • human intervention required when network irregularities occur; • LQ lacks the robust property to cope with parameter perturbations and external disturbances; • LQ has no direct consideration for constraints unlike MPC.
Reservation and Market-Based System (Autonomic)	• do not rely on approximations; • human intervention reduced to a minimum when network irregularities occur.	• renovation and infrastructure costs; • possibility of serious safety flaws.

Approach	Advantages	Limitations
Car-to-Car Communication (Autonomic)	• do not rely on approximations; • human intervention reduced to a minimum when network irregularities occur.	• renovation and infrastructure costs; • possibility of serious safety flaws; • if communication is down the system fails completely.
Artificial Intelligence (Adaptive)	• do not rely on approximations; • fuzzy control is simple to implement. • AI includes: Evolutionary Computation, Fuzzy Logic, Artificial Neural Networks, Swarm Intelligence and Reinforcement Learning;	• intensive computational speed requirements; • large amount of simulations to determine the performance of the proposed signal plans; • deteriorating performance when traffic conditions change particularly for neural networks; • difficulty in generating an effective rule-base in fuzzy logic.

Part I

Model Development and Estimation

Chapter 4

Model Development

4.1 Introduction

Many countries, including Malta, are struggling with severe daily traffic congestions that cause a huge amount of social and economic loss [341]. As discussed in Section 2.2.2, the timing of signal-controlled traffic light intersections is a major component in optimizing traffic flow within a road network and reducing traffic congestion. Traffic-responsive signals are required to efficiently control traffic in real-time. Sensors are installed at strategic locations on the signalized intersection to generate data that is required to control the traffic light timings in real-time. Such an infrastructural installation, and its associated maintenance costs, are an additional burden that should be kept to a minimum. A more effective solution is to use more efficiently the current infrastructure and to optimize the flow of vehicles through the road network in the presence of time-varying traffic conditions. The design and implementation of such systems requires the use of computationally efficient numerical models that continuously estimate the dynamics of traffic flow, thus allowing the controllers to self-tune and self-optimize.

Over the years, several mathematical models were developed to model the traffic dynamics within a signalized junction, as discussed in Chapter 2.3. Large sets of observation in time are required to capture the constantly changing traffic demand [6, 81]. This leads to high computational demand when estimating models from data. A further complication is brought about by the spatial diversity of traffic flow behaviour within a junction. Traffic flow depends highly on traffic conditions within a junction, whether the junction is saturated or not, and

hence homogenous behaviour of traffic cannot be assumed. In modelling approaches where the spatial domain is divided into a grid of cells, the number of cells increases with the size of the junction. In addition, the underlying traffic parameters are assumed to be known or else estimated from past traffic information.

The main objectives of this chapter are to develop linear heterogenous models that are able to describe several different traffic conditions, saturated and unsaturated traffic conditions, inferred from measured data, with junction models not divided into a grid of cells, and with computational costs kept to a minimum. Faced with such requirements, the choice of the models presented in Chapter 2.3 is quite limited. CTM is not highly adequate for urban scenarios having traffic junctions discontinuities, because of a grid of spatial cells that are assumed to be homogenous. Also, computational demand increases with larger networks [89], resulting in high computational effort for CTM to provide real-time traffic control. On the other hand, such computational burden can be eliminated through the use of SFM. However such a model is only applicable to saturated traffic scenarios [4]. The queuing theory approach can be used to model traffic behaviour during different traffic conditions with high accuracy in the modelling of the dynamics. This approach is nonlinear in nature, capturing the nonlinear characteristics of vehicles flowing through an arm, but can easily be piecewise linearized [6, 81] making it adequate for the design of appropriate controllers in both linear and nonlinear domains. The approach of Pecherkova *et al.* [6] and Kratochilova *et al.* [81] was chosen for this study satisfying the above requirements. Furthermore, the approach leading to this model was validated by [6] thus boosting our confidence that the model is a good and reliable representation of a true junction.

4.2 Representation of the Model

Consider an arm i as shown in Figure 4.1 with one sensor placed at the input to measure the inflow towards the arm, another placed on the stop-line to measure the number of departed cars and the third sensor placed at the output of the arm to measure the outflow away from the arm.

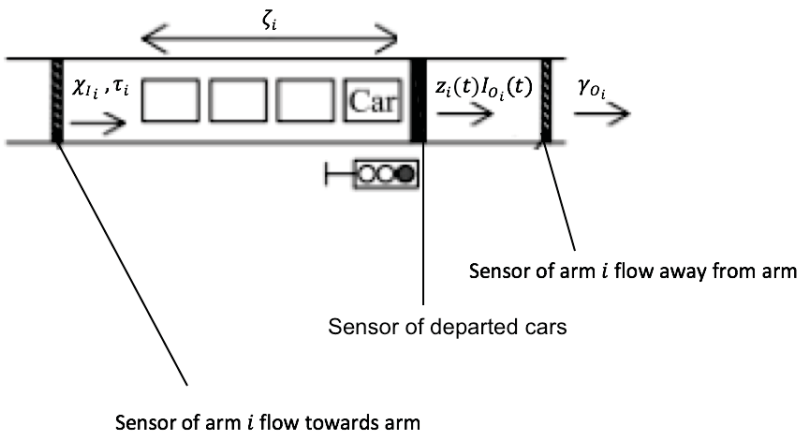

Figure 4.1: An example of an arm i

The state variables in this model comprise:

- the queue length in arm i, denoted as $\zeta_i(t)$ which represents the number of vehicles in arm i waiting to pass towards the arm at the start of the red phase of each cycle (in $[uv]$). t denotes time in cycles, where a cycle is the time taken to fulfill one complete sequence of traffic signal phases, and integer t denotes the cycle index;
- the inflow in arm i, i.e. the number of unit vehicles (uv) entering arm i in a cycle, denoted as $\gamma_{I_i}(t)$ in [uv/cycle];
- the occupancy $\phi_i(t)$ which relates the proportion of time when a sensor is occupied (and therefore activated) in a cycle with respect to the total measuring period given in [%];

Hence, the state vector $\mathbf{x}$ is given by Equation (4.2.1).

$$\mathbf{x}(t) = \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} \tag{4.2.1}$$

The state equation is later presented in Equation (4.2.5).

The three sensors mentioned previously, are usually implemented as inductive loops [60] and their readings typically include sensor noise $\mathbf{v}(t)$ assumed in this model, to be additive white, zero-mean, Gaussian measurement noise with covariance $\mathbf{R}$, where $\mathbf{R}$ is a symmetric positive-definite matrix. $r_{l,j}$ represent the elements of the inverse of $\mathbf{R}$ and l, j represent the row and column index of the matrix respectively. Thus, the sensor measurement vector $\mathbf{y}$ is taken to be $\chi_{I_i}(t)$ and $\tau_i(t)$, which are simply the state variables $\gamma_{I_i}(t)$ and $\phi_i(t)$ corrupted by the additive noise $\mathbf{v}(t)$, i.e.

$$\mathbf{y}(t) = \begin{bmatrix} \chi_{I_i}(t) \\ \tau_i(t) \end{bmatrix} = \begin{bmatrix} \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \mathbf{v}(t) \tag{4.2.2}$$

Further notation includes:

- z_i representing the ratio of the green signal time to the total cycle time for arm i,
- $\mathbf{w}(t)$ is a white, zero-mean, Gaussian noise process with covariance $\mathbf{Q}$, capturing model inaccuracy, where $\mathbf{Q}$ is a symmetric positive-definite matrix, with $q_{l,j}$ representing the elements of the inverse of $\mathbf{Q}$ and l, j represent the rows and columns of the matrix respectively,
- S_i represents the saturated flow for arm i in [uv/cycle].

The model assumes that the flow into the junction (γ_{I_i}) is a Markovian random process where the flow into the junction at time $t+1$ is dependent on the inflow at the previous cycle, $\gamma_{I_i}(t)$ and white, zero-mean, Gaussian noise. The occupancy $\phi_i(t+1)$ is considered to depend upon the occupancy at the previous cycle $\phi_i(t)$ and the queue length $\zeta_i(t)$, parameterized by two variables $\kappa_i(t)$ and $\beta_i(t)$, as similarly described in Section 2.3.3. Note that, although $\phi_i(t)$ does not affect the estimation of $\gamma_{I_i}(t)$ and $\zeta_i(t)$, the occupancy is included in the model because it is useful to detect unusual situations such as lane blockage during unsaturated conditions.

If a continuous green signal is shown throughout the cycle, the number of unit

vehicles exiting arm i in a cycle towards the junction is denoted as $I_{O_i}(t)$. Thus, during a cycle with green ratio $z_i(t)$, the departed vehicles during that cycle, is given by $z_i(t)I_{O_i}(t)$, where:

$$I_{O_i}(t) = S_i\left(1 - e^{\frac{-(\zeta_i(t)+\gamma_{I_i}(t))}{S_i z_i(t)}}\right) \tag{4.2.3}$$

This nonlinear function as presented in [6] and plotted in Figure 2.10 (black plot) represents the number of vehicles exiting an arm and depends on three quantities: (i) the queue length $\zeta_i(t)$ (ii) the inflow $\gamma_{I_i}(t)$ and (iii) the maximum number of passing vehicles that could be carried by the arm in saturated traffic conditions, given by $S_i z_i(t)$ with green ratio $z_i(t)$. Similar to the piecewise linear function (blue plot), as the queue length and the inflow increase, the intensity of departed vehicles during the green signal increases until it approaches a maximum number determined by saturated traffic flow conditions.

According to this model, the queue length at cycle $t+1$ is equal to the previous queue length, $\zeta_i(t)$, less the departed vehicles $z_i(t)I_{O_i}(t)$, and the arriving vehicles $\gamma_{I_i}(t)$ in a cycle, leading to:

$$\zeta_i(t+1) = \zeta_i(t) - z_i(t)I_{O_i}(t) + \gamma_{I_i}(t) + w_1(t) \tag{4.2.4}$$

where $w_1(t)$ represents the first element of process noise $\mathbf{w}$, introduced before.

$\gamma_{O_i}(t)$, the outflow from the arm, is equal to the departed vehicles during the green signal $z_i(t)I_{O_i}(t)$ plus some additive noise $v_3(t)$.

All these considerations applied to one of the arms are combined to form the state-space equations presented in Equation (4.2.5).

$$\begin{bmatrix} \zeta_i(t+1) \\ \gamma_{I_i}(t+1) \\ \phi_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ \kappa_i(t) & 0 & \beta_i(t) \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} - \begin{bmatrix} I_{O_i}(t) \\ 0 \\ 0 \end{bmatrix} z_i(t) + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix}$$

$$\begin{bmatrix} \chi_{I_i}(t) \\ \tau_i(t) \\ \gamma_{O_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I_{O_i}(t) \end{bmatrix} z_i(t) + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \tag{4.2.5}$$

Let **A**, **B**, **C** and **D** denote the state-space matrices of Equation (4.2.5), for arm i, where **A** represents the state matrix, **B** represents the input matrix, **C** represents the output matrix and **D** represents the direct transmission matrix, respectively. Note that Equation (4.2.5) represents the state-space representations for both saturated and unsaturated traffic conditions because $I_{O_i}(t)$ is calculated from Equation (4.2.3), which represents the departed vehicles during the green signal, $I_{O_i}(t)$ as a nonlinear function.

As an alternative, as previously described in Section 2.3.3, the departed vehicles are sometimes also represented as a piecewise linear function as in Figure 2.10 [81]. For unsaturated traffic conditions, the model assumes that the outflow is equal to the inflow if no queue exists. Otherwise, the outflow increases in direct proportion with the queue length and inversely with green time so that the queue is reduced efficiently. This leads to the following equation for unsaturated conditions i.e. when $I_{O_i}(t) < S_i$:

$$I_{O_i}(t) = \gamma_{I_i}(t) + \frac{\zeta_i(t)}{z_i(t)} \tag{4.2.6}$$

$$\implies \gamma_{O_i}(t) = z_i(t)\gamma_{I_i}(t) + \zeta_i(t) \tag{4.2.7}$$

For saturated traffic conditions, $I_{O_i}(t)$ is equal to S_i which implies that $\gamma_{O_i}(t) = z_i(t)S_i$.

With this piecewise linear approximation, the state space model for unsaturated traffic conditions is given as:

$$\begin{bmatrix} \zeta_i(t+1) \\ \gamma_{I_i}(t+1) \\ \phi_i(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1-z_i(t) & 0 \\ 0 & 1 & 0 \\ \kappa_i(t) & 0 & \beta_i(t) \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix}$$
$$\begin{bmatrix} \chi_{I_i}(t) \\ \tau_i(t) \\ \gamma_{O_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & z_i(t) & 0 \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \tag{4.2.8}$$

whereas for saturated conditions it becomes:

$$\begin{bmatrix} \zeta_i(t+1) \\ \gamma_{I_i}(t+1) \\ \phi_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ \kappa_i(t) & 0 & \beta_i(t) \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} - \begin{bmatrix} S_i(t) \\ 0 \\ 0 \end{bmatrix} z_i(t) + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \end{bmatrix}$$
$$\begin{bmatrix} \chi_{I_i}(t) \\ \tau_i(t) \\ \gamma_{O_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ S_i(t) \end{bmatrix} z_i(t) + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \tag{4.2.9}$$

Note that switching between the two models of Equations (4.2.8) and (4.2.9) is required when both unsaturated and saturated traffic conditions are present.

In this work both models of Equations (4.2.8) and (4.2.9) and Equation (4.2.5) will be used. In the estimation algorithms presented in Chapters 5 and Sections 6.1 to 6.2 of Chapter 6, the model of Equations (4.2.8) and (4.2.9) will be applied. This is because the estimation algorithms presented there required the partial derivative of the likelihood function with respect to each parameter to be worked out analytically. Solving for the saturation parameters from the partial derivatives of the likelihood function with respect to such parameters was too complex because of the high nonlinearity in Equation (4.2.3). On the other hand, in the estimation algorithms presented in Section 6.3, analytical derivatives are not required. Hence, the model of Equation (4.2.5) was used avoiding the need of switching when both unsaturated and saturated traffic conditions were present.

Furthermore, Equations (4.2.5), (4.2.8) and (4.2.9) assume that the inflow measurements $\chi_{I_i}(t)$ are independent of queue length. In actual fact, as a consequence

of sensor positioning, the measured inflow may be affected by the distance between the two sensors located on the input lane and the queue length. When the queue length exceeds this distance, incoming vehicles cannot be sensed so as to update the true inflow rate. Instead, under this condition, the measured inflow is perceived to be equal to the rate of departed vehicles towards the junction per cycle, namely $z_i(t)I_{O_i}(t)$ plus noise $v_1(t)$, as represented in Equation (4.2.10), where $I_{O_i}(t)$ can be represented both by the nonlinear function of Equation (4.2.3) and by the piecewise linear function of Equation (4.2.7) for unsaturated traffic conditions while for saturated traffic conditions, $I_{O_i}(t)$ is equal to S_i.

$$\begin{bmatrix} \chi_{I_i}(t) \\ \tau_i(t) \\ \gamma_{O_i}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \zeta_i(t) \\ \gamma_{I_i}(t) \\ \phi_i(t) \end{bmatrix} + \begin{bmatrix} I_{O_i}(t, z_i(t)) \\ 0 \\ I_{O_i}(t, z_i(t)) \end{bmatrix} z_i(t) + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \tag{4.2.10}$$

All considerations previously discussed are applicable to an arm i considered in isolation. This shall now be extended to a junction with multiple arms. Consider a typical 3-arm or 4-arm junction as shown in Figures 4.2 and 4.3 respectively. Each arm is comprised of five lanes, with three lanes carrying opposite traffic flow to the other two lanes. In Figure 4.3 three sensors (indicated in blue) are installed per arm, with one sensor placed at the input lanes to measure the inflow towards the junction, another placed on the stop-line of each arm to measure the outflow towards the junction and the third sensor placed at the output lanes of the arm to measure the outflow away from the junction. Similarly for the 3-arm junction of Figure 4.2.

For multiple arms, γ_{O_i} represents the flow for each arm away from the junction, that is the number of vehicles exiting the other arms towards the junction and entering into arm i. Let the turning ratio α_{ij} represent the ratio of vehicles turning from arm i to arm j in relation to the number of vehicles in arm i. Hence, for an n-arm junction, the flow for each arm away from the junction is given by

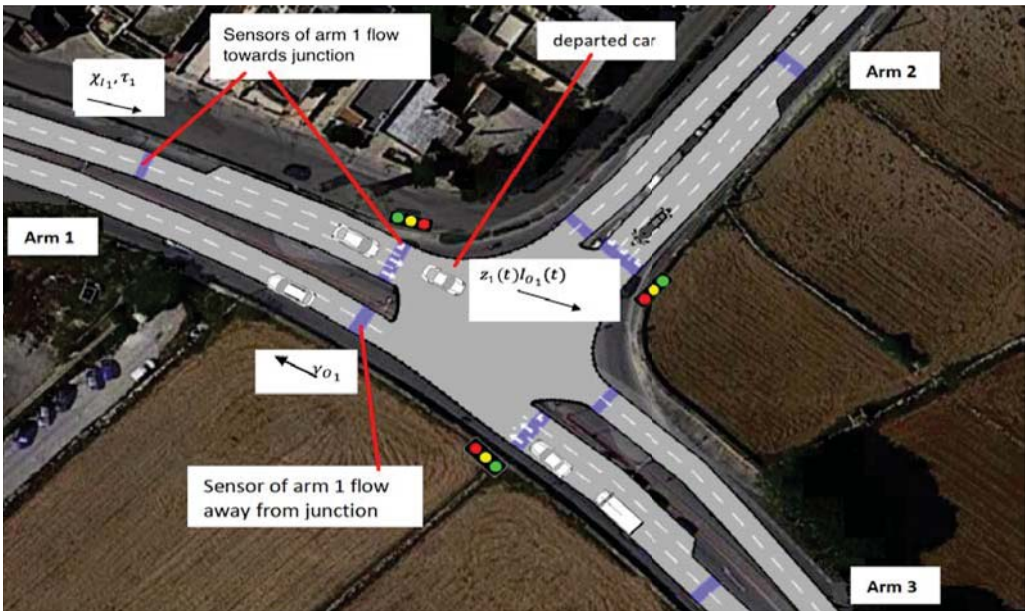

Figure 4.2: 3-arm signalized junction

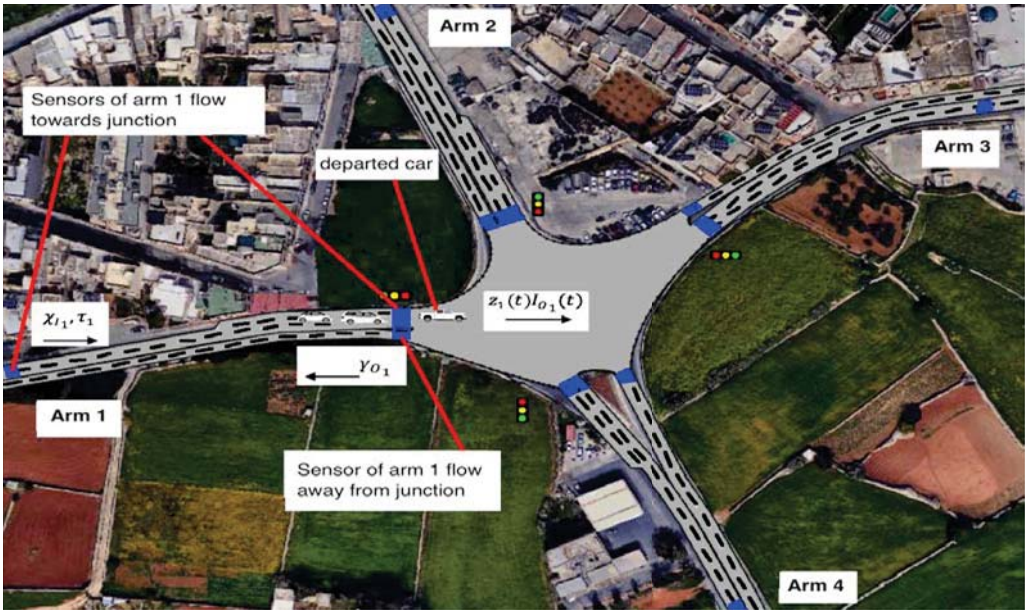

Figure 4.3: 4-arm signalized junction

Equation 4.2.11.

$$\gamma_{O_i}(t) = \sum_{\substack{j=1 \\ j \neq i}}^{n} \alpha_{ji}(t) z_j(t) I_{O_j}(t) + v_3(t) \quad \forall i = 1...n \tag{4.2.11}$$

The turning ratios α_{ij} are subject to the constraints given by Equations (4.2.12) and (4.2.13), where the addition of the ratio of vehicles turning from the same arm towards other arms should add up to one by the Conservation of Flow Principle, while the value itself should be between 0 and 1. Hence,

$$\text{For every given } i, \sum_{i \neq j} \alpha_{ij}(t) = 1 \quad \forall j = 1...n \tag{4.2.12}$$

$$0 \leq \alpha_{ij}(t) \leq 1 \quad \forall\, i, j = 1...n \tag{4.2.13}$$

Extending these and the previous equations to an n-arm junction yields the following state-space representation:

$$
\begin{bmatrix} \zeta_1(t+1) \\ \zeta_2(t+1) \\ \vdots \\ \zeta_n(t+1) \\ \gamma_{l_1}(t+1) \\ \gamma_{l_2}(t+1) \\ \vdots \\ \gamma_{l_n}(t+1) \\ \phi_1(t+1) \\ \phi_2(t+1) \\ \vdots \\ \phi_n(t+1) \end{bmatrix} =
\begin{bmatrix}
1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
\kappa_1(t) & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \beta_1(t) & 0 & \cdots & 0 \\
0 & \kappa_2(t) & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \beta_2(t) & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & \kappa_n(t) & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \beta_n(t)
\end{bmatrix}
\begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \\ \vdots \\ \zeta_n(t) \\ \gamma_{l_1}(t) \\ \gamma_{l_2}(t) \\ \vdots \\ \gamma_{l_n}(t) \\ \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_n(t) \end{bmatrix}
$$

$$
- \begin{bmatrix}
I_{O_1}(t) & 0 & \cdots & 0 \\
0 & I_{O_2}(t) & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & I_{O_n}(t) \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\ddots & \ddots & \cdots & \ddots \\
0 & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix} +
\begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \\ w_4(t) \\ w_5(t) \\ w_6(t) \\ \vdots \\ w_{3n}(t) \end{bmatrix}
$$

$$
\begin{bmatrix} \chi_{I_1}(t) \\ \chi_{I_2}(t) \\ \vdots \\ \chi_{I_n}(t) \\ \tau_1(t) \\ \tau_2(t) \\ \vdots \\ \tau_n(t) \\ \gamma_{O_1}(t) \\ \gamma_{O_2}(t) \\ \vdots \\ \gamma_{O_n}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \\ \vdots \\ \zeta_n(t) \\ \gamma_{I_1}(t) \\ \gamma_{I_2}(t) \\ \vdots \\ \gamma_{I_n}(t) \\ \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_n(t) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \ddots & \ddots & \cdots & \ddots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \ddots & \ddots & \cdots & \ddots \\ 0 & 0 & \cdots & 0 \\ 0 & \alpha_{2,1}(t)I_{O_2}(t) & \cdots & \alpha_{n,1}(t)I_{O_n}(t) \\ \alpha_{1,2}(t)I_{O_1}(t) & 0 & \cdots & \alpha_{n,2}(t)I_{O_n}(t) \\ \ddots & \ddots & \cdots & \ddots \\ \alpha_{1,n}(t)I_{O_1}(t) & \alpha_{2,n}(t)I_{O_2}(t) & \cdots & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \\ v_5(t) \\ v_6(t) \\ \vdots \\ v_{3n}(t) \end{bmatrix}
\tag{4.2.14}
$$

Table 4.1 summarizes all the notations used for an arm *i* or an *n*-arm junction:

Table 4.1: Summary of notations used for an arm i or a n-arm junction

Notation	Definition
ζ_i	queue in arm i
γ_{I_i}	number of vehicles entering arm i
χ_{I_i}	γ_{I_i} + noise
ϕ_i	occupancy at arm i
τ_i	ϕ_i + noise
z_i	ratio of green time / cycle time
$z_i I_{O_i}$	number of vehicles exiting arm i in cycle for 1 arm number of vehicles exiting arm i towards junction in cycle for multiple arms
γ_{O_i}	$z_i I_{O_i}$ + noise for 1 arm $\sum_{\substack{j=1 \\ j\neq i}}^{n} \alpha_{ji}(t) z_j(t) I_{O_j}(t)+$ noise $\forall i = 1...n$ for multiple arms

4.3 Conclusion

The state space model for a 3-arm and 4-arm junction have been presented, where these could be extended to n-arm junctions. In this work, we aim to estimate recursively in real-time, the state variables $\zeta_i(t)$, $\gamma_{I_i}(t)$ and $\phi_i(t)$, representing queue length, inflow and occupancy, together with model parameters κ_i, β_i, saturated flows S_i, turning ratios α_{ij} and noise covariances $\mathbf{Q}$ and $\mathbf{R}$ which are all assumed unknown, based upon sensor measurements χ_{I_i}, τ_i, γ_{O_i}, representing measured inflow, occupancy and outflow, and knowledge of green time ratio z_i. Note that although the inflow and the occupancy state variables are sensed, these measurements are affected by sensor noise $\mathbf{v}(t)$. They shall thus still be estimated recursively, as discussed in the next chapter, so as to obtain smoother estimates. Furthermore, there is also the possibility of estimating predictions for the inflow γ_{I_i} and the occupancy ϕ_i, which might be useful for controllers integrated to this estimation procedure.

Chapter 5

Quasi-real-time Joint estimation of states, model parameters and noise covariances

5.1 Introduction

Faced with the requirement of continuously tuning the model to reflect changing traffic conditions in real-time, a joint state and parameter estimation procedure is necessary. Based on the observations discussed in Section 2.5 on joint estimation strategies, an elegant and powerful method for joint estimation is the EM algorithm [114, 115]. A set of novel online joint state, parameter and noise estimation algorithms, based on the EM algorithm but modified for real-time implementation, will be proposed in this chapter.

5.2 Quasi Real-Time Joint Estimation of States, Parameters and Noise Covariances

The key feature of the EM algorithm is to exploit the concavity of the log likelihood function so as to guarantee its maximization [113, 116, 342–349]. Let G be the log likelihood function with respect to the parameter vector $\boldsymbol{\theta}$. Then G is concave if for all $\lambda \in$ (0,1) the following condition is satisfied:

$$G((1-\lambda)\boldsymbol{\theta}_{k-1} + \lambda\boldsymbol{\theta}_k) \geq (1-\lambda)G(\boldsymbol{\theta}_{k-1}) + \lambda G(\boldsymbol{\theta}_k) \tag{5.2.1}$$

The EM algorithm iteratively computes $\boldsymbol{\theta}$ such that each consecutive $\boldsymbol{\theta}$ estimate, $\hat{\boldsymbol{\theta}}$, either remains the same or improves on the previous estimate. Hence, the process of maximiz-

ing the likelihood function is the same as maximizing the difference in the log likelihood between iterations as indicated in Equation (5.2.2) given in [116]:

$$G(\boldsymbol{\theta}) - G(\hat{\boldsymbol{\theta}}) = \ln p(\mathbf{Y}_N|\boldsymbol{\theta}) - \ln p(\mathbf{Y}_N|\hat{\boldsymbol{\theta}}) \tag{5.2.2}$$

where $\boldsymbol{\theta}$=[α_{12}, α_{13}, ..., α_{1n}, α_{21}, α_{23}, ..., α_{2n}, ..., α_{n1}, α_{n2}, ..., $\alpha_{n(n-1)}$, κ_1, κ_2, ..., κ_n, β_1, β_2, ..., β_n, S_1, S_2, ..., S_n] in our case, $\hat{\boldsymbol{\theta}}$ represents the initial parameter vector guess and $\mathbf{Y}_N$ denotes some observed data set up to time N. The decomposition in Equation (5.2.2) shows that if a value for $\boldsymbol{\theta}$ is found that increases $logp(\mathbf{Y}_N|\boldsymbol{\theta})$, then this must also increase the log-likelihood $G(\boldsymbol{\theta})$. Suppose we are able to calculate the likelihood of the observed data $\mathbf{Y}_N$ and some hidden data $\mathbf{X}_N$, it follows from [116] that:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left(E_{\mathbf{X}_N|\mathbf{Y}_N,\boldsymbol{\theta}} G(\boldsymbol{\theta})\right) = \arg\max_{\boldsymbol{\theta}} \left(E_{\mathbf{X}_N|\mathbf{Y}_N,\boldsymbol{\theta}}[\ln p(\mathbf{Y}_N, \mathbf{X}_N|\boldsymbol{\theta})]\right) \tag{5.2.3}$$

by definition of likelihood function $G(\boldsymbol{\theta})$ in Equation (5.2.2), where

$$E_{\mathbf{X}_N|\mathbf{Y}_N,\boldsymbol{\theta}}[\ln p(\mathbf{Y}_N, \mathbf{X}_N|\theta] = E_{\mathbf{X}_N|\mathbf{Y}_N,\boldsymbol{\theta}}\Big(\sum_{k=t}^{N} \ln p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\theta}) + \sum_{k=t}^{N} \ln p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}) + \ln p(\mathbf{x}_0, \boldsymbol{\theta})\Big) \tag{5.2.4}$$

where the outputs, $\mathbf{y}_k$ represent the observed data and $\mathbf{x}_k$ represent the state variables. Making use of model Equations (4.2.5) to (4.2.10) and the fact that the noise is assumed to be normally distributed and zero mean, the probabilities in Equation (5.2.4) can be expanded further as follows: the observed data $\mathbf{y}_k$ is distributed normally with mean $\mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{z}_k$ and covariance $\mathbf{R}$. Similarly, the state variable $\mathbf{x}_k$ is distributed normally with mean $\mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{z}_{k-1}$ and covariance $\mathbf{Q}$. The initial state $\mathbf{x}_0$ is also assumed to be distributed normally with mean μ and variance $\mathbf{P}_0$. This leads to the expression for the log-likelihood as follows [116]:

$$
\begin{aligned}
&2E\{G(\boldsymbol{\theta}_k,\hat{\boldsymbol{\theta}}_k)\} = -\ln|\mathbf{P}_0| \\
&-Tr(\mathbf{P}_0^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{x}_0-\boldsymbol{\mu})(\mathbf{x}_0-\boldsymbol{\mu})^T+\mathbf{P}_0|\mathbf{Y}_N\big)) \\
&-N\ln|\mathbf{Q}|-N\ln|\mathbf{R}| \\
&-\sum_{k=t}^{N}\Big(Tr\Big(\mathbf{Q}^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{x}_k-\mathbf{A}\hat{\mathbf{x}}_{k-1}-\mathbf{B}\mathbf{z}_{k-1}-\mathbf{w}_{k-1}) \\
&\qquad\qquad (\mathbf{x}_k-\mathbf{A}\hat{\mathbf{x}}_{k-1}-\mathbf{B}\mathbf{z}_{k-1}-\mathbf{w}_{k-1})^T|\mathbf{y}_k\big)\Big)\Big) \\
&-\sum_{k=t}^{N}\Big(Tr\Big(\mathbf{R}^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{y}_k-\mathbf{C}\hat{\mathbf{x}}_k-\mathbf{D}\mathbf{z}_k-\mathbf{v}_k) \\
&\qquad\qquad (\mathbf{y}_k-\mathbf{C}\hat{\mathbf{x}}_k-\mathbf{D}\mathbf{z}_k-\mathbf{v}_k)^T|\mathbf{y}_k\big)\Big)\Big)
\end{aligned}
\tag{5.2.5}
$$

where $\hat{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$, $\hat{\mathbf{x}}_{k-1}\sim N(\mu,\mathbf{P}_0)$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are the state space matrices from Equations (4.2.8) and (4.2.9), the covariance matrices
$\mathbf{E}\{\mathbf{w}(t)\mathbf{w}^T(t)\}$=$\mathbf{Q}$ and $\mathbf{E}\{\mathbf{v}(t)\mathbf{v}^T(t)\}$=$\mathbf{R}$.

The EM algorithm alternates between two steps. The Expectation (E) step, computes the best estimate of the likelihood function using the current estimate for the parameters using a Kalman filter and/or Kalman smoother recursions [350]. The Maximization (M) step computes parameters maximizing the expected log-likelihood found in the E step. Therefore,

- **E Step** - Calculate $2E\{G(\boldsymbol{\theta}_k,\hat{\boldsymbol{\theta}}_k)\}$)
- **M Step** - Compute $\boldsymbol{\theta}_{k+1}=\arg\max_{\boldsymbol{\theta}}\left(E_{\mathbf{X}_N|\mathbf{Y}_N,\boldsymbol{\theta}}G(\boldsymbol{\theta}_k,\hat{\boldsymbol{\theta}}_k)\right)$ where k represents the k^{th} iteration of the EM algorithm.

5.2.1 Standard batch-based EM

Let $\hat{\mathbf{x}}$ denote the estimate of state vector $\mathbf{x}$. Then, $\hat{\mathbf{x}}$ is given by first running the Kalman filter recursions [350] followed by the Kalman smoother recursions [350] to calculate the smoothed estimates. To solve for $\boldsymbol{\theta}$, we find the parameters that maximize the log-likelihood Equation (5.2.5), or equivalently those that minimize the negative log-likelihood given by $-2E\{G(\boldsymbol{\theta}_k,\hat{\boldsymbol{\theta}}_k)\}$ in Equation (5.2.5) conditioned upon $\hat{\mathbf{x}}$, the estimated states. In a standard batch EM implementation, these parameters are iteratively estimated over a batch of N observations.

Differentiating the negative log likelihood with respect to each of the elements of θ[1], and equated to zero in order to find the parameters that give the minimum value. Note that derivatives for parameters describing 3 or 4-arm junctions will be derived according to the following methodology. For example, using model equations (4.2.5) or (4.2.8) and (4.2.9) and taking the partial derivative of the objective function with respect to κ_1 and equating to 0, gives:

$$\kappa_1 \sum_{k=t}^{N} \zeta_1(k-1)^2 = \sum_{k=t}^{N} \{\phi_1(k)\zeta_1(k-1)\} - \beta_1 \sum_{k=t}^{N} \phi_1(k-1)\zeta_1(k-1) - \sum_{k=t}^{N} w_6(k-1)\zeta_1(k-1) \quad (5.2.6)$$

which can easily be solved for κ_1. Similarly for β_1:

$$\beta_1 \sum_{k=t}^{N} \phi_1(k-1)^2 = \sum_{k=t}^{N} \{\phi_1(k)\phi_1(k-1)\} - \kappa_1 \sum_{k=t}^{N} \phi_1(k-1)\zeta_1(k-1) - \sum_{k=t}^{N} \phi_1(k-1)w_6(k-1) \quad (5.2.7)$$

For simplification reasons, Equations (5.2.6) and (5.2.7) assume that $\mathbf{Q}$ and $\mathbf{R}$ are diagonal matrices, with equal diagonal elements. Doing the same for parameter S_1, using Equation (4.2.5) and taking the partial derivative of the objective

[1]The following matrix properties are useful for this [351]:

- $(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$
- $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$
- $\frac{\partial tr(\mathbf{AB}(x))}{\partial x} = tr(\mathbf{A}\frac{\partial \mathbf{B}(x)}{\partial x})$

function with respect to S_1 and equating to 0, gives Equation (5.2.8).

$$\begin{aligned}
&\zeta_1(k)\Big[1-e^{-}\frac{\zeta_1(k)+\gamma_{I_1}(k)}{S_1 z_1(k)}\Big(1-\frac{\zeta_1(k)+\gamma_{I_1}(k)}{S_1 z_1(k)}\Big)\Big]-\\
&\zeta_1(k-1)\Big[1-e^{-}\frac{\zeta_1(k)+\gamma_{I_1}(k)}{S_1 z_1(k)}\Big(1-\frac{\zeta_1(k)+\gamma_{I_1}(k)}{S_1 z_1(k)}\Big)\Big]+\\
&S_1 z_1(k-1)-z_1(k-1)\Big[2S_1+\frac{\zeta_1(k)+\gamma_{I_1}(k)}{z_1(k)}\Big]e^{-}\frac{\zeta_1(k)+\gamma_{I_1}(k)}{S_1 z_1(k)}+\\
&z_1(k-1)\Big[S_1+\frac{\zeta_1(k)+\gamma_{I_1}(k)}{z_1(k)}\Big]e^{-}\frac{2(\zeta_1(k)+\gamma_{I_1}(k))}{S_1 z_1(k)}=0
\end{aligned} \tag{5.2.8}$$

Unfortunately in this case, solving for S_1 from Equation (5.2.8) is not straightforward because S_1 appears nonlinearly in the equation. Hence estimation of parameters using the linear piecewise approximation as discussed in Chapter 4 with model Equations given by (4.2.8) and (4.2.9) is preferred. Taking this approach, for S_1 under saturated traffic conditions leads to:

$$\begin{aligned}
&\sum_{k=t}^{N}\Big(\zeta_1(k-1)z_1(k-1)+\gamma_{I_1}(k-1)z_1(k-1)\\
&-\zeta_1(k)z_1(k-1)+\alpha_{12}\gamma_{O_2}(k)z_1(k)+\alpha_{13}\gamma_{O_3}(k)z_1(k)\\
&-\alpha_{13}\alpha_{23}S_2 z_2(k)z_1(k)-\alpha_{12}\alpha_{32}S_3 z_3(k)z_1(k)\\
&+z_1(k-1)w_1(k-1)-\alpha_{12}z_1(k)v_8(k)-\alpha_{13}z_1(k)v_9(k)\Big)\\
&=S_1\sum_{k=t}^{N}\Big(z_1(k-1)^2+\alpha_{12}^2 z_1(k)^2+\alpha_{13}^2 z_1^2(k)\Big)
\end{aligned} \tag{5.2.9}$$

which can be easily solved for S_1. Similarly for α_{21} under unsaturated traffic conditions, we obtain:

$$\begin{aligned}
&\sum_{k=t}^{N}(\gamma_{O_1}(k)\gamma_{I_2}(k)z_2(k))-\sum_{k=t}^{N}(\gamma_{O_1}(k)\zeta_2(k))-\sum_{k=t}^{N}\gamma_{I_2}(k)z_2(k)v_7(k)-\sum_{k=t}^{N}\zeta_2(k)v_7(k)\\
&=\sum_{k=t}^{N}(z_2(k)^2\gamma_{I_2}(k)^2\alpha_{21})+\sum_{k=t}^{N}(z_2(k)\gamma_{I_2}(k)\zeta_2(k)\alpha_{21})\\
&\quad+\sum_{k=t}^{N}(z_3(k)z_2(k)\gamma_{I_3}(k)\gamma_{I_2}(k)\alpha_{31})+\sum_{k=t}^{N}(z_3(k)\gamma_{I_3}(k)\zeta_2(k)\alpha_{31})
\end{aligned} \tag{5.2.10}$$

and for α_{21} under saturated traffic conditions we get:

$$\sum_{k=t}^{N}\Big(S_2\gamma_{O_1}(k)z_2(k)-S_2z_2(k)v_7(k)-\alpha_{31}S_2S_3z_2(k)z_3(k)\Big) = \alpha_{21}\sum_{k=t}^{N}(S_2^2z_2^2(k)) \tag{5.2.11}$$

In this approach, the model parameters $\boldsymbol{\theta}$ are considered as separate from the noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$. This differs from standard EM methodology where $\mathbf{Q}$ and $\mathbf{R}$ are typically grouped with the model parameters [343]. Our approach thus allows the estimation algorithm to be tuned more selectively according to the differing characteristics of the model parameters and the elements of the covariance matrices, such as their different order of magnitude and their numerical constraints.

Following from Chen's work [116] and taking the partial derivative of the negative log-likelihood given by $-2E\{G(\boldsymbol{\theta}_k,\hat{\boldsymbol{\theta}}_k)\}$ in Equation (5.2.5) with respect to $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ respectively[2], and equating both to 0, leads to Equations (5.2.12) and (5.2.13). The quantities $\mathbf{P}_{k|N}$, $\mathbf{P}_{k-1,k}$, $\mathbf{P}_{k,k-1}$ and $\mathbf{P}_{k-1|N}$ are pre-computed from the Kalman smoother recursions [116].

$$\begin{aligned}\hat{\mathbf{Q}} = &\\ &\frac{1}{(N-1)}\sum_{k=2}^{N}\Big[(\hat{\mathbf{x}}_{k|N}-\hat{\mathbf{A}}(k-1)\hat{\mathbf{x}}_{k-1|N}-\hat{\mathbf{B}}(k-1)\mathbf{z}(k-1))\\ &\quad(\hat{\mathbf{x}}_{k|N}-\hat{\mathbf{A}}(k-1)\hat{\mathbf{x}}_{k-1|N}-\hat{\mathbf{B}}(k-1)\mathbf{z}(k-1))^T\\ &+\mathbf{P}_{k|N}-\hat{\mathbf{A}}(k-1)\mathbf{P}_{k-1,k}-\mathbf{P}_{k,k-1}\hat{\mathbf{A}}(k-1)^T\\ &+\hat{\mathbf{A}}(k-1)\mathbf{P}_{k-1|N}\hat{\mathbf{A}}(k-1)^T\Big]\end{aligned} \tag{5.2.12}$$

[2]To find the partial derivative of the objective function with respect to $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ respectively, the following properties were used [116, 351, 352]:

- $\ln(\mathbf{R}^{-1}) = -\ln(\mathbf{R})$
- $\frac{\partial \ln|(\mathbf{R})|}{\partial \mathbf{R}^{-1}} = -\frac{\partial \ln|(\mathbf{R}^{-1})|}{\partial \mathbf{R}^{-1}}$
- $\frac{\partial \ln|(\mathbf{R}^{-1})|}{\partial \mathbf{R}^{-1}} = [(\mathbf{R}^{-1})^{-1}]^T = \mathbf{R}$

$$\begin{aligned}\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N}\Big[&(\mathbf{y}(k) - \hat{\mathbf{C}}(k-1)\hat{\mathbf{x}}_{k|N} - \hat{\mathbf{D}}(k-1)\mathbf{z}(k)) \\ &(\mathbf{y}(k) - \hat{\mathbf{C}}(k-1)\hat{\mathbf{x}}_{k|N} - \hat{\mathbf{D}}(k-1)\mathbf{z}(k))^T \\ &+ \hat{\mathbf{C}}(k-1)\mathbf{P}_{k|N}\hat{\mathbf{C}}(k-1)^T\Big]\end{aligned} \tag{5.2.13}$$

This process is repeated for all other variables describing the traffic dynamics within a signalized junction for unsaturated and saturated traffic conditions. The algorithm is given in Table 5.1. However, such a batch approach is not suitable for real-time control, since the Kalman smoothed estimates are conditioned on the whole data set N. Hence a modified algorithm for joint estimation of states, noise and model parameters is proposed, as discussed in the following section.

Table 5.1: EM algorithm for estimation of model states, parameters and noise realisations

Initialise estimates for $\boldsymbol{\theta}$

Iterate for k=1,2,...,N where N represent the size of the available data batch

E-step

Run Kalman-filter recursions followed by the Kalman smoother recursions in order to compute $\hat{\mathbf{x}}_{k|N}$ with N representing number of observations.

M-step

If $(\zeta_n(k) + \gamma_{I_n}(k) \cdot z_n(k)) < (S_n z_n(k))$

Minimise $-2E\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_k)\}$ over $\boldsymbol{\theta}$ via the choices in unsaturated conditions including Equations (5.2.6) - (5.2.7), (5.2.10), (5.2.12), (5.2.13).

If $(\zeta_n(k) + \gamma_{I_n}(k) \cdot z_n(k)) > (S_n z_n(k))$

Minimise $-2E\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_k)\}$ over $\boldsymbol{\theta}$ via the choices in saturated conditions including Equations (5.2.6) - (5.2.7), (5.2.9), (5.2.11), (5.2.12), (5.2.13).

Repeat the E and M steps until the log likelihood of the objective function converges to a constant value up to a small predefined tolerance bound.

Update **A**, **B**, **C**, **D**, with $\hat{\boldsymbol{\theta}}_k$ to reflect the traffic conditions per arm.

5.2.2 The modified EM algorithm - Quasi Real-Time Implementation

In the dynamic traffic situations considered in this work, estimation is ideally required to reflect traffic conditions in real-time. Hence, the standard batch mode EM algorithm of Table 5.1 is modified to carry out quasi real-time estimation.

Table 5.2: Quasi real-time EM algorithm for estimation of model states, parameters and noise covariance

Initialise estimates for $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$
Commencing from t=$\bar{n}+1$
Iterate for every time step t and measure $\mathbf{y}(t)$.
Iterate for k=$(t-\bar{n})$,...,t

E-step
Run Kalman-filter recursions followed by the Kalman smoother recursions in order to compute $\hat{\mathbf{x}}_{k|\bar{n}}$.

M-step
Minimise $-2E\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_k)\}$ over $\boldsymbol{\theta}$ for unsaturated conditions including (5.2.6) - (5.2.7), (5.2.10) and for saturated conditions including Equations (5.2.6) - (5.2.7), (5.2.9), (5.2.11), with N replaced by $\bar{n}$.

Repeat the E and M steps until the log likelihood of the objective function converges to a constant value up to a small predefined tolerance bound.

Update $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$, with $\hat{\boldsymbol{\theta}}_k$ to reflect the traffic conditions per arm.
Example: $\hat{\mathbf{A}}=\frac{(\hat{\mathbf{A}}_{unsat}\cdot\bar{n}_{unsat})(\hat{\mathbf{A}}_{sat}\cdot\bar{n}_{sat})}{\bar{n}}$

If (t>$\bar{m}$) where $\bar{m}$ represents the size of the second window
Iterate for k=$(t-\bar{m})$,...,t
Maximise $\mathbf{Q}$ and $\mathbf{R}$ as in Equations (5.2.12) and (5.2.13) with N replaced by $\bar{m}$.

t=t+1

This algorithm and the new notations are explained in the following text. This estimation algorithm requires continuous storage of a reduced set of past measurement data, hence denoting a quasi real-time estimation as opposed to a purely online estimation where estimation is carried out on the fly, without storing the measurement data. However, only a reduced batch of data points is necessary at any given time, as opposed to classical EM.

In the quasi real-time estimation, an iterative algorithm is proposed which makes use of two uniform windows, one of fixed time length $\bar{n}$ for joint states and parameter estimation, and another of fixed time length $\bar{m}$ for noise covariance estimation, hence separating the estimation of $\mathbf{Q}$ and $\mathbf{R}$ as shown in Table 5.2. The uniform windows of fixed time length $\bar{n}$ and $\bar{m}$ respectively, less than N, look back in time and move on a time grid as shown in Figure 5.1. $\bar{n}$ and $\bar{m}$ are both significantly less than N to obtain quasi real-time estimation. Simulations show that a $\bar{n}$ of order 20 and a $\bar{m}$ of order 1500 for N being equal to 7097 cycles give good reliable results. Joint parameter and state estimation is carried out for those particular time points falling inside the first window, by applying the EM algorithm presented in Table 5.2. At every time iteration, the window slides forward by one instant and the procedure is repeated again, with initial values for the parameters and the states being fed from the previous time window. This algorithm is beneficial to modelling the traffic behaviour within a junction, since traffic information from sensors, including the input intensity and the output intensity need not be available as a batch of N samples, but rather within a shorter window of $\bar{n}$ samples. In the second time window, noise covariance estimation is carried out for those particular time points falling inside this window. Tests to establish suitable window lengths $\bar{m}$ and $\bar{n}$ are carried out through simulations as will be described in Section 5.2.3.

5.2.2.1 Transitions between saturated and unsaturated flows

The presented model is subject to unsaturated or saturated traffic conditions per arm. An arm can exhibit unsaturated or saturated conditions irrespective of other arms within the same junction. An arm is saturated if the condition $(\zeta_n(k) +$

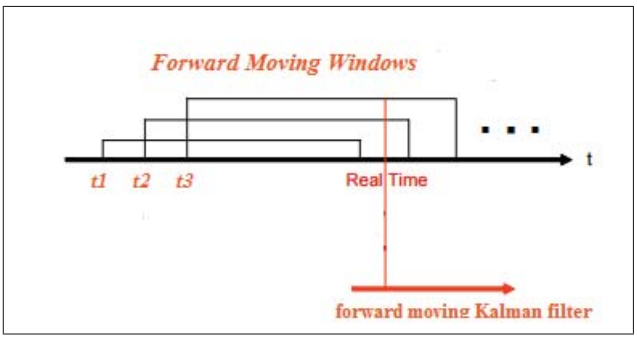

Figure 5.1: Real-time joint parameters and states estimation

$\gamma_{I_n}(k) \cdot z_n(k)) \geq (S_n z_n(k))$ is satisfied. Conversely, an arm is usaturated if $(\zeta_n(k) + \gamma_{I_n}(k) \cdot z_n(k)) < (S_n z_n(k))$. Hence, for example, for a 3-arm junction, eight (2^3) different combinations of switching conditions exist. Thus during execution of the modified EM algorithm the state space model's matrices given in Equations (4.2.8) and (4.2.9) are all updated to reflect the different traffic conditions per arm. For one time instance, falling within a window length $\bar{n}$, the number of times an arm is in unsaturation or in saturation is noted and average weighting of model parameters and state space matrices are obtained. For example, $\mathbf{A} = \frac{(\mathbf{A}_{unsat} \cdot \bar{n}_{unsat})(\mathbf{A}_{sat} \cdot \bar{n}_{sat})}{\bar{n}}$, where $\mathbf{A}_{unsat}$ and $\mathbf{A}_{sat}$ represent the state transition matrix for unsaturation and saturation respectively, while $\bar{n}_{unsat}$ and $\bar{n}_{sat}$ represent the number of times an arm is unsaturated or saturated, where $\bar{n} = \bar{n}_{unsat} + \bar{n}_{sat}$. The average weighting of model parameters and state space matrices are obtained for each time instance, so as to update the Kalman filter and smoother recursions.

5.2.2.2 System Identification

One fundamental question of system identification is whether or not the unknown variables can be uniquely estimated from a given set of mathematical expressions. If unique estimates of the parameters cannot be obtained, the system is not uniquely identifiable [353].

For example, for a 3-arm junction, to solve for the six turning ratio parameters in the maximization stage (α_{12}, α_{13}, α_{21}, α_{23}, α_{31} and α_{32}), six equations are available under unsaturated traffic conditions, as given by Equations (4.2.12)

and (4.2.13). Hence a unique solution can be found and the system is identifiable.

However, for saturated traffic conditions, to estimate the turning ratio parameters and the saturations in the maximization stage, nine parameters need to be solved: α_{12}, α_{13}, α_{21}, α_{23}, α_{31}, α_{32} and S_1, S_2, S_3. However only six equations are available as per Equations (4.2.12) and (4.2.13). Hence, the system of equations is underdetermined and the unknowns are not uniquely identifiable. To overcome this problem in the saturated traffic case, the saturation and turning ratio estimations are carried out in an alternating manner. At a given time instance, the estimation of turning ratios is carried out first, whilst feeding in any *prior* information for the saturation parameters as if they were known. At the next instant, the saturation parameters are updated given the turning ratio parameters estimated in the previous step. Priority is given to first estimating the turning ratios based on some *prior* knowledge of the saturation parameters, because the latter are less likely to change when subject to changing traffic scenarios, unlike the turning ratios.

5.2.3 Results

The proposed algorithm was tested on a 3-arm junction, with geometry similar to the junction presented in Figure 4.2. The unknown variables to be estimated include the state vector $\mathbf{x} \triangleq [\zeta_1\ \zeta_2\ \zeta_3\ \gamma_{I_1}\ \gamma_{I_2}\ \gamma_{I_3}\ \phi_1\ \phi_2\ \phi_3]^T$, the vector of model parameters $\boldsymbol{\theta} \triangleq [\alpha_{12}, \alpha_{13}, \alpha_{21}, \alpha_{23}, \alpha_{31}, \alpha_{32}, \kappa_1, \kappa_2, \kappa_3, \beta_1, \beta_2, \beta_3, S_1, S_2, S_3]^T$ and noise covariances $\mathbf{Q}$ and $\mathbf{R}$.

Two cases were tested: i) the estimation of states and model parameters, with measurement noise covariance assumed known, and ii) the estimation of states and model parameters together with process and measurement noise covariance. The geometry of the network was imported from OpenStreetMap which is interfaced through Aimsun. Measurements of vehicles entering and leaving each arm were simulated in Aimsun [354] as *Traffic State* per second. Traffic light information such as phases and a cycle time of 90 seconds were introduced to reflect typical traffic characteristics that this simulated junction is based on. The simula-

tion was executed for 7097 cycles to replicate traffic count measurements similar to a physical ITS junction purposely fitted with sensors. Sensors were placed in the junction similar to those shown in Figure 4.2, with the sensors measuring the flow towards the junction placed 60 metres away from each other, for each arm. Randomly generated Gaussian distributed measurement noise was added to the high resolution dataset from Aimsun for evaluation purposes, which was then used as input to the quasi real-time EM algorithm. The actual measurement noise covariance $\mathbf{R}$ for the 3-arm signalized junction is given by a 9 by 9 symmetric positive-definite matrix with randomly generated Gaussian elements, as given in Equation (5.2.14).

$$\mathbf{R} = \begin{bmatrix} 0.2963 & 0.0849 & 0.0616 & 0.0921 & 0.0891 & 0.1215 & 0.0832 & 0.0750 & 0.0673 \\ 0.0849 & 0.3145 & 0.0488 & 0.1078 & 0.0706 & 0.1128 & 0.0940 & 0.0824 & 0.0990 \\ 0.0616 & 0.0488 & 0.2163 & 0.0610 & 0.0662 & 0.0756 & 0.0605 & 0.0740 & 0.0741 \\ 0.0921 & 0.1078 & 0.0610 & 0.3543 & 0.0746 & 0.0630 & 0.1072 & 0.0720 & 0.1128 \\ 0.0891 & 0.0706 & 0.0662 & 0.0746 & 0.3016 & 0.0735 & 0.1133 & 0.1054 & 0.0913 \\ 0.1215 & 0.1128 & 0.0756 & 0.0630 & 0.0735 & 0.3098 & 0.0668 & 0.0951 & 0.1206 \\ 0.0832 & 0.0940 & 0.0605 & 0.1072 & 0.1133 & 0.0668 & 0.3577 & 0.0984 & 0.1039 \\ 0.0750 & 0.0824 & 0.0740 & 0.0720 & 0.1054 & 0.0951 & 0.0984 & 0.2933 & 0.0841 \\ 0.0673 & 0.0990 & 0.0741 & 0.1128 & 0.0913 & 0.1206 & 0.1039 & 0.0841 & 0.3285 \end{bmatrix} \tag{5.2.14}$$

Tests were first carried out to determine a suitable window size for the modified EM algorithm. The expectation stage was executed separately from the maximization stage, assuming known parameters for this stage. 1000 realisations were executed for each window size with different traffic conditions. In order to evaluate the quality of the estimation, a Root Mean Square Error (RMSE) measure for some estimate $\hat{p}$ was defined to be given by the square root of the averaged mean square error per window sample as $J \triangleq \sqrt{\frac{\sum_{\bar{n}}(p(t)-\hat{p}(t))^2}{\bar{n}}}$, where p is the actual value. For each realisation and for each window sample, the RMSE measure J was calculated, expressed as a percentage of the mean measurement $p(t)$ over the window sample. The resultant %RMSE measures are then aver-

aged over all window samples to get one value for evaluation purposes. Table 5.3 shows the resultant %RMSE measures for the states including the queue length ζ_i and the inflow γ_{I_i} for each arm for different window lengths. A window of length 20 resulted in an average %RMSE measure of 10.7%, averaged over all states, while a window length of 40 resulted in an average %RMSE measure of 9.0%. Thus, doubling the window size resulted in only a minor improvement in state estimation.

Similarly, Monte Carlo runs with 1000 realisations were executed for the maximization stage assuming known states for this stage and each realisation represented different traffic conditions. For each realisation and for each window sample, the RMSE measure J was calculated, expressed as a percentage of the mean parameter $p(t)$ over the window sample. Similar to the previous stage, the resultant %RMSE measures are then averaged over all window samples to get one value for evaluation purposes. Table 5.4 shows the resultant %RMSE measures for the turning ratios α_{ij} and saturation parameters S_i for different window lengths. A window of length 20 resulted in an average %RMSE measures of 0.6%, averaged over all parameters, while a window length of 40 resulted in an average %RMSE measures of 0.3%. In this case, the improvement in parameter estimation with double window size is more significant than for state estimation.

Table 5.3: Average RMSE measure J for the states expressed as a percentage of the respective measurement for different window sizes

Window size $\bar{n}$	20	40
%RMSE for ζ_1	9.2 (4 vehicles)	8.5 (3 vehicles)
%RMSE for ζ_2	11.3 (5 vehicles)	8.6 (4 vehicles)
%RMSE for ζ_3	9.8 (4 vehicles)	7.9 (3 vehicles)
%RMSE for γ_{I_1}	11.5 (3 vehicles)	8.5 (3 vehicles)
%RMSE for γ_{I_2}	12.3 (4 vehicles)	10.8 (3 vehicles)
%RMSE for γ_{I_3}	10.3 (3 vehicles)	9.8 (3 vehicles)

From Tables 5.3 and 5.4 it was noted that the accuracy of the estimation of states and parameters improved with increasing window length. This can also be observed in Figure 5.2 showing the distribution of the estimation of one of the pa-

Table 5.4: Average RMSE measure J for the parameters expressed as a percentage of the respective measurement for different window sizes

Window size $\bar{n}$	20	40
%RMSE for α_{12}	0.2	0.03
%RMSE for α_{13}	0.2	0.02
%RMSE for α_{21}	0.4	0.01
%RMSE for α_{23}	0.6	0.1
%RMSE for α_{31}	0.1	0.01
%RMSE for α_{32}	0.3	0.04
%RMSE for S_1	1.3	0.9
%RMSE for S_2	0.8	0.7
%RMSE for S_3	1.5	1.1

rameters namely α_{12}, with different time lags over the Monte Carlo runs, whose true value was 0.2. Figure 5.2 shows the mean and standard deviation above each frequency plot. A mean of 1.7% decrease in accuracy from 10.7% over all the states of Table 5.3 and a mean decrease of 0.3% from 0.6% over all the parameters of Table 5.4, is observed when applying a window of length 20. Furthermore, the joint estimation algorithm requires 5.086 minutes of simulation time to complete the full 7097 cycles with a window size of 20 cycles, while a window size of 40 requires 10.341 minutes when running Matlab 2015 on a Intel Processor 2.7GHz Intel Core i5. In addition, a window size of 20 reduces the storage requirements by half, as measurements need to be made available as a batch of 20 samples rather than 40 samples. A balance between computation efficiency, storage requirements and estimation accuracy was aimed for. Hence, a window of length 20 is preferred over 40, having less computational effort.

To determine $\bar{m}$, Monte Carlo runs with 1000 different realisations were executed for the maximization stage of $\mathbf{Q}$ and $\mathbf{R}$. The matrix Euclidean norm of the estimated covariances, $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$, represented as $\|\hat{\mathbf{Q}}\|$ and $\|\hat{\mathbf{R}}\|$ were calculated as a measure of estimation accuracy for different window sizes. For evaluation purposes, the Euclidean norm of the randomly generated Gaussian distributed measurement noise $\mathbf{R}$ which was added to the high resolution dataset from Aimsun given by: $\sqrt{\sum_{l=1}^{b}\sum_{j=1}^{b}|a_{l,j}|^2}$ was made equal to 1, where $a_{l,j}$ represents the element

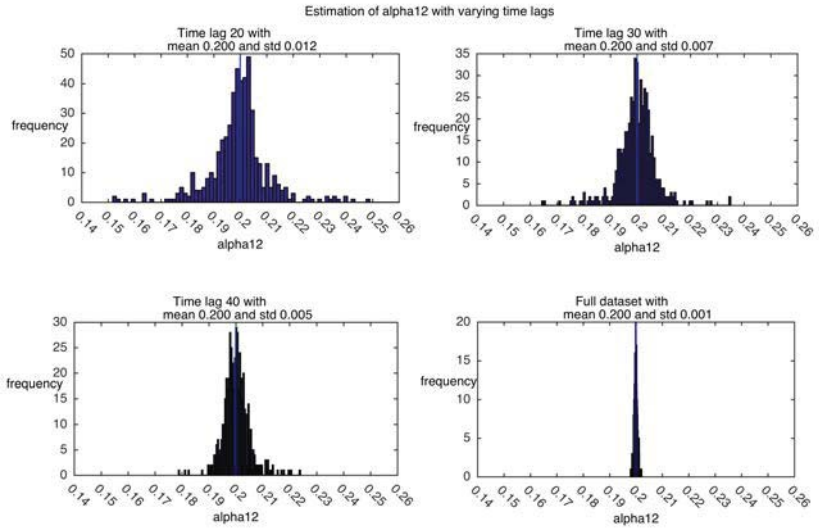

Figure 5.2: Estimation of α_{12} for saturated traffic conditions, with different time lags

of $\mathbf{R}$ at the l^{th} row and j^{th} column. From Table 5.5, $\bar{m}$ equal to 1500 and 2000 respectively, gave similar results. Hence a smaller window of size 1500 is preferred.

Table 5.5: Estimated covariance $\hat{\mathbf{R}}$

Window size $\bar{m}$	$\|\hat{\mathbf{R}}\|$	$\|\mathbf{R}\|$	% Difference
1000	0.730	1	-27
1500	1	1	0
2000	1	1	0

The training data generated by Aimsun for 7097 cycles, which includes χ_{I_i}, τ_i, and γ_{O_i}, $i = 1,2,3$, was used to generate the measurement vector $\mathbf{y}$ in the state space model. The EM algorithm was then executed to jointly estimate the states, the model parameters and noise covariances with $\bar{n}$=20 and $\bar{m}$=1500. Table 5.6 shows some of the results obtained for the turning ratios of this junction compared with the actual values, averaged over the whole training time (7097cycles), with

a % difference ranging from -2.6% to 8.4%.

Table 5.6: Estimated results for turning ratios

Turning ratio	Estimated mean	Actual mean	% Difference
α_{12}	0.216	0.2	8
α_{13}	0.784	0.8	-2
α_{21}	0.832	0.845	-1.5
α_{23}	0.168	0.155	8.4
α_{31}	0.810	0.805	0.6
α_{32}	0.19	0.195	-2.6

To further test the model under different traffic conditions, fresh validation datasets were generated from Aimsun. The mean of the estimated turning ratios presented in Table 5.6 were fed to Aimsun to generate these datasets. The means were also fed to the state space model so that both Aimsun and the state space model were presented with the same parameters. The results from the model and Aimsun were compared.

The previously defined measure J, for the fresh validation dataset with added randomly generated Gaussian distributed measurement noise, is calculated. Results for both known and estimated measurement covariance is shown in Table 5.7, expressed as a percentage of $\sqrt{\frac{\sum_{\bar{n}} p^2(t)}{\bar{n}}}$ to yield a normalized measurement over one window. For comparison reasons, one figure of merit was computed for both tests. A mean % RMSE was computed by taking the resultant mean value over all the 12 % RMSE estimates of Table 5.7 for each test. An average of 1.3% was obtained when the noise covariances are known, whilst the average with estimation of noise covariances was 1.5%. This shows a reasonable decrease in accuracy of 0.2% with the estimation of noise covariances compared to the result obtained when the noise covariances are known.

Table 5.7: Results on validation datasets with and without noise estimation

Estimate	% RMSE with known noise	% RMSE with noise estimation
ζ_1	1.1	1.3
ζ_2	1.7	1.7
ζ_3	1.1	1.2
γ_{O_1}	0.8	0.8
γ_{O_2}	0.7	0.9
γ_{O_3}	0.9	1.4
ϕ_1	1.4	1.6
ϕ_2	2.9	3.0
ϕ_3	1.2	1.7
Average % RMSE	1.3	1.5

5.2.4 Conclusions

To contribute towards the autonomicity of traffic light systems, a novel self-estimation algorithm is proposed to estimate the model parameters, states and noise covariances in quasi real-time. As already discussed, estimation of the model parameters $\boldsymbol{\theta}$ are considered separate from the noise covariances $\mathbf{Q}$ and $\mathbf{R}$, thus allowing the estimation algorithm to be tuned according to the differing characteristics of the model parameters and the elements of the covariance matrices, such as their different orders of magnitude and their numerical constraints. This algorithm reduces the computational and storage demands for a junction model estimator since traffic information from sensors, including the inflow and the outflow, need not be available as a batch of N samples, covering for example typical traffic operations over twenty-four hours, but rather within much shorter time windows of $\bar{n}$ or $\bar{m}$ samples.

Chapter 6

Real-time Online Joint Estimation of States, Parameters and Noise Covariance

6.1 Introduction

Chapter 5, for joint state, parameter and noise covariance estimation for traffic junctions, made use of uniform windows, that look back in time and move on a time grid, dictated by uniform time lags, in order to perform joint state and parameter estimation. Although the time windows were short when compared to classical EM methodology, this cannot be considered to represent a truly real-time approach because it still requires continuous storage of a reduced set of measurement data inside a moving window. In practice this means that traffic information from sensors needs to be available and stored as a batch of window samples during all time instants. By contrast, in this Chapter we develop improvements by using the measurement data only once and avoiding altogether batch data storage, leading to a full recursive real-time algorithm exhibiting shorter computation times and reduced storage requirements in the estimation of possibly time-varying conditions that could be implemented as an online algorithm. This leads to a more efficient implementation with smaller demands on computational hardware requirements. This technique makes use of stochastic approximation theory proposed by Robbins and Monro [120], where online parameter estimation is carried out from noisy data as will be discussed in the next section. Robbins and Monro stochastic approximation performs well in the presence of heavy-tailed

noise and outliers, unlike for example the Kalman filter which assumes the presence of Gaussian noise and results in degraded performance in the presence of outliers and heavy-tailed noise [355].

6.1.1 Robbins-Monro stochastic approximation

Let the equation $f(\rho_t) = 0$, for some given function f have a unique root $\rho = \rho^*$. In general, if $f(\rho) > 0$ for $\rho > \rho^*$ and $f(\rho) < 0$ for $\rho < \rho^*$, the Robbins-Monro algorithm [120] iteratively calculates $\hat{\rho}_t$, where $\hat{\rho}$ is an estimate of the root ρ^*, according to:

$$\hat{\rho}_{t+1} = \hat{\rho}_t - \eta_t g_t \tag{6.1.1}$$

where g_t is a noisy observation of $f(\hat{\rho}_t)$, and η_t is a sequence of positive numbers (steps). Provided the sequence of steps satisfies the following conditions: $\lim_{t\to\infty} \eta_t = 0$, $\sum_{t=0}^{\infty} \eta_t = \infty$ and $\sum_{t=0}^{\infty} \eta_t^2 < \infty$, this algorithm is guaranteed to converge to the solution $f(\rho) = 0$, where $\eta_t = \frac{1}{t}$ is typically used [120].

The utility of this stochastic approximation method to the traffic flow model presented in Chapter 4, is explored in this work as will be discussed in the next sections.

6.1.2 Robbins-Monro algorithm for the model's parameters

The Robbins-Monro stochastic approximation is applied to estimate the model unconstrained parameters κ_1, ..., κ_n, β_1, ..., β_n and the positively constrained parameters S_1, ..., S_n. Each parameter is estimated separately. This is done so as to simplify the algorithm due to the relatively large number of parameters to be estimated. To solve for such parameters, the ML objective function given by [116] was considered, that is, the parameters that minimize the negative-log likelihood as defined in Equation (6.1.2) conditioned upon $\mathbf{y}_t$. Note that Equation (6.1.2) is derived from (5.2.5), rewritten in recursive form and the most recent value at instance t is extracted.

$$
\begin{aligned}
-2E\{G(\boldsymbol{\theta}_t,\hat{\boldsymbol{\theta}}_t,\hat{\mathbf{x}}_t,\hat{\mathbf{Q}},\hat{\mathbf{R}})\} &= -2E\{G(\boldsymbol{\theta}_{t-1},\hat{\boldsymbol{\theta}}_{t-1},\hat{\mathbf{x}}_{t-1},\hat{\mathbf{Q}},\hat{\mathbf{R}})\} + \ln|\mathbf{Q}| \\
+\ln|\mathbf{R}| + \Big(Tr\Big(\mathbf{Q}^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{x}_t - \mathbf{A}\hat{\mathbf{x}}_{t-1} - \mathbf{B}\mathbf{z}_{t-1} - \mathbf{w})\big. & \qquad (6.1.2) \\
\big.(\mathbf{x}_t - \mathbf{A}\hat{\mathbf{x}}_{t-1} - \mathbf{B}\mathbf{z}_{t-1} - \mathbf{w})^T|\mathbf{y}_t\big)\Big)\Big) & \\
+\Big(Tr\Big(\mathbf{R}^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{y}_t - \mathbf{C}\hat{\mathbf{x}}_t - \mathbf{D}\mathbf{z}_t - \mathbf{v})(\mathbf{y}_t - \mathbf{C}\hat{\mathbf{x}}_t - \mathbf{D}\mathbf{z}_t - \mathbf{v})^T|\mathbf{y}_t\big)\Big)\Big) &
\end{aligned}
$$

Let the subscript $_0$ denote the initial conditions. Hence the state and parameters are initialised as follows: the initial state $\mathbf{x}_0$ is sampled from a normal distribution with a mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}_0$. $\boldsymbol{\mu}$ represents a reasonable prior (for example, the initial queue lengths $\boldsymbol{\mu}$ is set to 10 vehicles) and covariance $\mathbf{P}_0$ represents a large initial positive definite covariance matrix. Equation (6.1.3) shows the initial value of $-2G(\boldsymbol{\theta}_t,\hat{\boldsymbol{\theta}}_t,\hat{\mathbf{x}}_t,\hat{\mathbf{Q}},\hat{\mathbf{R}})$ at time $t = 0$.

$$
\begin{aligned}
-2E\{G(\boldsymbol{\theta}_0,\hat{\boldsymbol{\theta}}_0,\mathbf{x}_0,\hat{\mathbf{Q}}_\mathbf{0},\hat{\mathbf{R}}_\mathbf{0})\} &= \ln|\mathbf{P}_0| + \ln|\mathbf{Q}_0| + \ln|\mathbf{R}_0| \\
&+ Tr(\mathbf{P}_0^{-1}\mathbf{E}_{\hat{\boldsymbol{\theta}}}\big((\mathbf{x}_0 - \boldsymbol{\mu})(\mathbf{x}_0 - \boldsymbol{\mu})^T + \mathbf{P}_0|\mathbf{y}_t\big)) \qquad (6.1.3)
\end{aligned}
$$

Substituting the dynamic equations of the model given by Equations (4.2.8) and (4.2.9) in the objective function Equation (6.1.2), the partial differentiation of the objective function with respect to each parameter is performed and equated to zero in order to find the minimum value. Its solution is given recursively by applying the Robbins-Monro Equation (6.1.1).

Thus for example for unsaturated traffic conditions, let $f(\kappa_1)$ represent the partial derivative of the objective function with respect to κ_1, which when equated to 0 gives Equation (6.1.4) for a 3-arm junction:

$$
\begin{aligned}
&2\Big(-\hat{q}_{6,1}\hat{\xi}_1(t)\hat{\xi}_1(t-1) - \hat{q}_{6,2}\hat{\xi}_2(t)\hat{\xi}_1(t-1) - \hat{q}_{6,3}\hat{\xi}_3(t)\hat{\xi}_1(t-1) \\
&- \hat{q}_{6,4}\hat{\gamma}_{I_1}(t)\hat{\xi}_1(t-1) - \hat{q}_{6,5}\hat{\gamma}_{I_2}(t)\hat{\xi}_1(t-1) - \hat{q}_{6,6}\hat{\phi}_1(t)\hat{\xi}_1(t-1) \\
&- \hat{q}_{6,6}\hat{\xi}_1(t-1)w_6(t-1) - \hat{q}_{6,7}\hat{\phi}_2(t)\hat{\xi}_1(t-1) - \hat{q}_{6,8}\hat{\phi}_3(t)\hat{\xi}_1(t-1) \\
&+ \hat{q}_{6,6}\big(\hat{\kappa}_1\hat{\xi}_1(t-1)\hat{\xi}_1(t-1) + \hat{\beta}_1\hat{\phi}_1(t-1)\hat{\xi}_1(t-1)\big) \\
&- \hat{q}_{3,6}\hat{\xi}_1(t-1)\hat{\gamma}_{I_3}(t)z_3(t)\Big) = 0
\end{aligned}
\tag{6.1.4}
$$

The Robbins-Monro algorithm yields the sequential solution presented by Equation (6.1.5).

$$
\begin{aligned}
\hat{\kappa}_1(t) = \hat{\kappa}_1(t-1) - 2\eta_t \\
&\Big(-\hat{q}_{6,1}\hat{\xi}_1(t)\hat{\xi}_1(t-1) - \hat{q}_{6,2}\hat{\xi}_2(t)\hat{\xi}_1(t-1) - \hat{q}_{6,3}\hat{\xi}_3(t)\hat{\xi}_1(t-1) \\
&- \hat{q}_{6,4}\hat{\gamma}_{I_1}(t)\hat{\xi}_1(t-1) - \hat{q}_{6,5}\hat{\gamma}_{I_2}(t)\hat{\xi}_1(t-1) - \hat{q}_{6,6}\hat{\phi}_1(t)\hat{\xi}_1(t-1) \\
&- \hat{q}_{6,6}\hat{\xi}_1(t-1)w_6(t-1) - \hat{q}_{6,7}\hat{\phi}_2(t)\hat{\xi}_1(t-1) - \hat{q}_{6,8}\hat{\phi}_3(t)\hat{\xi}_1(t-1) \\
&+ \hat{q}_{6,6}\big(\hat{\kappa}_1\hat{\xi}_1(t-1)\hat{\xi}_1(t-1) + \hat{\beta}_1\hat{\phi}_1(t-1)\hat{\xi}_1(t-1)\big) \\
&- \hat{q}_{3,6}\hat{\xi}_1(t-1)\hat{\gamma}_{I_3}(t)z_3(t)\Big)
\end{aligned}
\tag{6.1.5}
$$

Note that $\eta(t)$ is set to $\frac{1}{t}$. This process is repeated for all other unconstrained variables describing the traffic dynamics within a signalized junction for unsaturated and saturated traffic conditions.

Similarly, the covariance matrices of the noise are estimated by differentiating the objective function (6.1.2) with respect to $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ respectively, equating to zero to find the minimum value, and solving recursively by the Robbins-Monro Equations (6.1.6) and (6.1.7).

$$
\begin{aligned}
\hat{\mathbf{Q}}(t) = \hat{\mathbf{Q}}(t-1) - \frac{1}{t}\Big[&\hat{\mathbf{Q}}(t-1) \\
&- (\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{A}}(t-1)\hat{\mathbf{x}}_{t-1|t-1} - \hat{\mathbf{B}}(t-1)\mathbf{z}(t-1)) \\
&\quad (\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{A}}(t-1)\hat{\mathbf{x}}_{t-1|t-1} - \hat{\mathbf{B}}(t-1)\mathbf{z}(t-1))^T \\
&+ \mathbf{P}_{t|t} - \hat{\mathbf{A}}(t-1)\mathbf{P}_{t-1,t|t} - \mathbf{P}_{t,t-1|t}\hat{\mathbf{A}}(t-1)^T \\
&+ \hat{\mathbf{A}}(t-1)\mathbf{P}_{t-1|t-1}\hat{\mathbf{A}}(t-1)^T\Big]
\end{aligned}
\tag{6.1.6}
$$

$$\begin{aligned}\hat{\mathbf{R}}(t) = \hat{\mathbf{R}}(t-1) - \frac{1}{t}\Big[&\hat{\mathbf{R}}(t-1) \\ &- (\mathbf{y}(t) - \hat{\mathbf{C}}(t-1)\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{D}}(t-1)\mathbf{z}(t)) \\ &\quad (\mathbf{y}(t) - \hat{\mathbf{C}}(t-1)\hat{\mathbf{x}}_{t|t} - \hat{\mathbf{D}}(t-1)\mathbf{z}(t))^T \\ &+ \hat{\mathbf{C}}(t-1)\mathbf{P}_{t|t}\hat{\mathbf{C}}(t-1)^T\Big]\end{aligned} \tag{6.1.7}$$

6.1.3 Robbins-Monro algorithm for the model's constrained parameters

In view that some of the model parameters are subject to numerical constraints (e.g. turning ratios have to be from 0 to 1), a modified projected Robbins-Monro algorithm [356] is applied. This is used to estimate the traffic flow model parameters that are subject to constraints (4.2.12) and (4.2.13) for a 3-arm junction or a 4-arm junction, namely the turning ratios α_{12}, α_{13}, ..., α_{1n}, α_{21}, α_{23}, ..., α_{2n}, α_{n1}, α_{n2}, ..., $\alpha_{n(n-1)}$ for a n-arm junction must be bounded between 0 and 1 and must sum up to 1.

In general the projected Robbins-Monro stochastic approximation is given by:

$$\begin{aligned}\tilde{\rho}_{t+1} &= \tilde{\rho}_t - \eta_t g_t \\ \hat{\rho}_{t+1} &= \pi_F(\tilde{\rho}_{t+1})\end{aligned} \tag{6.1.8}$$

where π_F is the function that projects $\tilde{\rho}$ onto a constrained region F such that $\hat{\rho}$ $\in F$ for all t [356].

Substituting the dynamic equations of the model given by Equations (4.2.8) and (4.2.9) in the objective function Equation (6.1.2), the partial differentiation of the objective function with respect to each constrained parameter is performed and equated to zero in order to find the minimum value. Its solution is given recursively by applying the projected Robbins-Monro Equation (6.1.8), where $\eta(t)$ in this case is set to $\frac{h}{t}$, where $h = \frac{(\hat{\alpha}_{12}(t) - \bar{\alpha}_{12})}{f(\hat{\alpha}_{12}(t))}$ and $\bar{\alpha}_{12}$ is some rough approximation of $\alpha_{12}(t)$ assumed to be known a priori [357]. This is done so as to increase the convergence rate of the algorithm and it is repeated to estimate all other turning ratios.

6.1.3.1 Projection for a 3-arm junction

In order for the traffic flow turning ratio parameter estimates to satisfy constraints (4.2.12) and (4.2.13), π_f is designed to project these estimates onto a subspace F of a straight line that satisfies Equations (4.2.12) and (4.2.13), indicated in red in Figure 6.1. Let the estimates of the turning ratios obtained recursively from Equation (6.1.8) be denoted by $\tilde{\alpha}_{12}(t)$, $\tilde{\alpha}_{13}(t)$, etc. To project $\tilde{\alpha}_{12}(t)$ and $\tilde{\alpha}_{13}(t)$ onto F, 3 regions were defined as shown in Figure 6.1, with region 1 defined by: $\tilde{\alpha}_{13} - \tilde{\alpha}_{12} - 1 > 0$, region 2 defined by $\tilde{\alpha}_{13} + \tilde{\alpha}_{12} - 1 \neq 0$ and region 3 defined by $\tilde{\alpha}_{13} - \tilde{\alpha}_{12} + 1 < 0$. $\tilde{\alpha}_{12}(t)$ and $\tilde{\alpha}_{13}(t)$ falling within region 1 are projected onto the

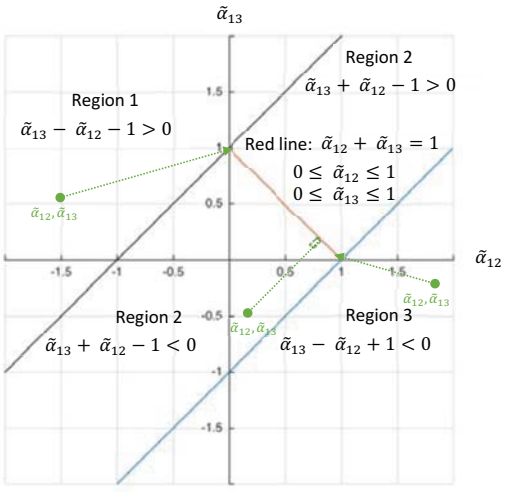

Figure 6.1: Regions for projection in the space of α_{12}, α_{13}

point $\hat{\alpha}_{12}(t) = 0$ and $\hat{\alpha}_{13}(t) = 1$ in F. Similarly $\tilde{\alpha}_{12}(t)$ and $\tilde{\alpha}_{13}(t)$ falling within region 3 are projected onto the point $\hat{\alpha}_{12}(t) = 1$ and $\hat{\alpha}_{13}(t) = 0$ in F. $\tilde{\alpha}_{12}(t)$ and $\tilde{\alpha}_{13}(t)$ falling within region 2 are projected onto F by estimating a point on the line $\hat{\alpha}_{12} + \hat{\alpha}_{13} = 1$ satisfying the shortest distance to point $\tilde{\alpha}_{12}(t)$ and $\tilde{\alpha}_{13}(t)$. This is given by the perpendicular distance from point $(\tilde{\alpha}_{12}(t), \tilde{\alpha}_{13}(t))$ to the line. In this case, the projection of the point $(\tilde{\alpha}_{12}(t), \tilde{\alpha}_{13}(t))$ on the line is estimated as follows [358]:

$$\hat{\alpha}_{12}(t) = \frac{(\tilde{\alpha}_{12}(t) - \tilde{\alpha}_{13}(t)) + 1}{2} \tag{6.1.9}$$

$$\hat{\alpha}_{13}(t) = \frac{(\tilde{\alpha}_{13}(t) - \tilde{\alpha}_{12}(t)) + 1}{2} \tag{6.1.10}$$

These 3 projections are all marked in green in Figure 6.1. This process is repeated for all the other pairwise combinations of turning ratios describing the traffic dynamics within a signalized junction for unsaturated and saturated traffic conditions.

6.1.3.2 Projection for a 4-arm junction

For a 4-arm junction, π_F is designed to project the estimates of the turning ratios $\tilde{\rho}$ onto a subspace F of a plane, satisfying Equations (4.2.12) and (4.2.13). This is the brown region enclosed within the red perimeter in Figure 6.2 that lies on the plane $\tilde{\alpha}_{12}(t) + \tilde{\alpha}_{13}(t) + \tilde{\alpha}_{14}(t) = 1$ shown in blue. Let the estimates of the turning ratios from arm 1 of a 4-arm junction be denoted by $\tilde{\alpha}_{12}(t)$, $\tilde{\alpha}_{13}(t)$ and $\tilde{\alpha}_{14}(t)$. The projection of the point $\tilde{\alpha}_{12}(t)$, $\tilde{\alpha}_{13}(t)$, $\tilde{\alpha}_{14}(t)$ onto the plane $\tilde{\alpha}_{12}(t) + \tilde{\alpha}_{13}(t) + \tilde{\alpha}_{14}(t) = 1$ is given as follows [359]: If the estimates from

$$\hat{\alpha}_{12_p}(t) = \tilde{\alpha}_{12}(t) - \left(\frac{\tilde{\alpha}_{12}(t) + \tilde{\alpha}_{13}(t) + \tilde{\alpha}_{14}(t) - 1}{3}\right) \tag{6.1.11}$$

$$\hat{\alpha}_{13_p}(t) = \tilde{\alpha}_{13}(t) - \left(\frac{\tilde{\alpha}_{12}(t) + \tilde{\alpha}_{13}(t) + \tilde{\alpha}_{14}(t) - 1}{3}\right) \tag{6.1.12}$$

$$\hat{\alpha}_{14_p}(t) = \tilde{\alpha}_{14}(t) - \left(\frac{\tilde{\alpha}_{12}(t) + \tilde{\alpha}_{13}(t) + \tilde{\alpha}_{14}(t) - 1}{3}\right) \tag{6.1.13}$$

Equations (6.1.11) to (6.1.13) satisfy Equations (4.2.12) and (4.2.13) then the projection of the point is within region F on the plane. However, this is not always the case. In the latter case, one needs to project point $\hat{\alpha}_{12_p}$, $\hat{\alpha}_{13_p}$, $\hat{\alpha}_{14_p}$ onto the closest point on the perimeter of region F.

Let s_1 be a point along the line AC shown in Figure 6.2 in 3D space, given by $s_1 = (t_1, 0, 1-t_1)$, where parameter t_1 varies between 0 and 1 to completely cover this line. Note that points A (0,0,1) and C (1,0,0) are two points on the line when t_1=0 or t_1=1 respectively. Moreover the squared Euclidean distance between a point on the line with parameter t_1 and the point $\hat{\alpha}_{12_p}, \hat{\alpha}_{13_p}, \hat{\alpha}_{14_p}$ is denoted as $d^2 = [\hat{\alpha}_{12_p} - t_1]^2 + [\hat{\alpha}_{13_p}]^2 + [\hat{\alpha}_{14_p} - 1 + t_1]^2$. To minimize the distance, set

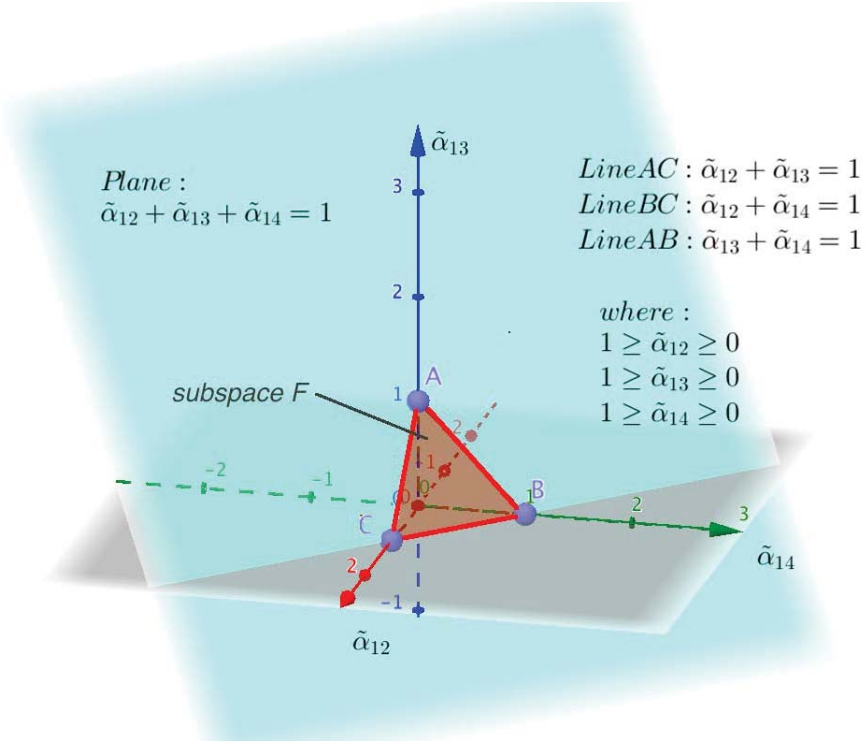

Figure 6.2: Projection in the space of α_{12}, α_{13} and α_{14}

$\frac{d(d^2)}{dt_1} = 0$ and solve for t_1.

Similarly, s_2 is the point along the line BC, given by $s_2 = (t_2, 1-t_2, 0)$, and s_3 is the point along the line AB given by $s_3 = (0, t_3, 1-t_3)$, where t_2 and t_3 vary between 0 and 1 to completely cover these lines. The solutions for d, t_2 and t_3 are obtained similarly as above. The point which gives the minimum Euclidean distance d out of all the 3 cases and which satisfies the magnitude constraints for t_1, t_2 and t_3 is used to determine the projected point $\hat{\alpha}_{12}$, $\hat{\alpha}_{13}$, $\hat{\alpha}_{14}$, by substituting for t_1, t_2 or t_3 into s_1, s_2 or s_3.

If the magnitude constraints for t_1, t_2 and t_3 are not satisfied, the point $\hat{\alpha}_{12_p}, \hat{\alpha}_{13_p}, \hat{\alpha}_{14_p}$ is projected onto one corner of the triangular perimeter of F, being either: (1,0,0), (0,0,1) or (0,1,0), whichever is closest.

This process is repeated for all the other pairwise combinations of turning ratios under unsaturated and saturated traffic flow conditions. This leads to the online EM estimation algorithm as described in Table 6.1, where n=1 to 3 for a 3-arm junction and n=1 to 4 for a 4-arm junction. The algorithm is applied to saturated and unsaturated traffic conditions.

Table 6.1: Online EM algorithm for estimation of model states, parameters and noise covariance

Initialise estimates for $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$
Iterate for every time step t and measure $\mathbf{y}(t)$.

E-step
Run Kalman-filter recursion in order to compute $\hat{\mathbf{x}}_{t|t}$.

M-step
If $(\hat{\zeta}_n(t)+\hat{\gamma}_{I_n}(t)z_n(t))<(\hat{S}_n(t-1)z_n(t))$
Minimise -2E$\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_t)\}$ over $\boldsymbol{\theta}$ via the choices in unsaturated conditions using Equation (6.1.8) and satisfying Equations (4.2.12) and (4.2.13).
If $(\hat{\zeta}_n(t)+\hat{\gamma}_{I_n}(t)z_n(t))>(\hat{S}_n(t-1)z_n(t))$
Minimise -2E$\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_t)\}$ over $\boldsymbol{\theta}$ via the choices in saturated conditions using Equation (6.1.8) and satisfying Equations (4.2.12) and (4.2.13).
Update $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$, with $\hat{\boldsymbol{\theta}}_t$ to reflect the traffic conditions per arm.
Minimise -2E$\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_t)\}$ over $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ as in Equations (6.1.6) and (6.1.7).
Increment t.

The same switching conditions presented in Section 5.2.2.1 and system identification conditions presented in Section 5.2.2.2 still apply.

6.1.4 Results

The proposed algorithm was tested and validated by simulating a signalized 4-arm junction having geometry as in Figure 4.3 with a cycle time of 110 seconds. Aimsun micro traffic simulation software [354] was used to generate data

at one second intervals and the resulting traffic characteristics of this junction were recorded as shown in the third column of Table 6.3.

From the Aimsun simulation of the junction, the saturation parameters were also determined. These correspond to the maximal number of vehicles flowing through the intersection when subject to maximal inflow, and were found to be $S_1 = 50$ uv/cycle, $S_2 = 57$ uv/cycle, $S_3 = 48$ uv/cycle and $S_4 = 50$ uv/cycle. The high resolution dataset generated from Aimsun was used as input to the online EM algorithm after adding randomly generated Gaussian distributed measurement noise for evaluation purposes, with the Euclidean norm of $\mathbf{R}$ set equal to 1×10^{-3}, in order to satisfy the noise signal in the model equations.

This data which includes χ_{I_i}, τ_i, and γ_{O_i} for $i = 1,2,3,4$, was used to form the sensor measurement vector $\mathbf{y}$ of the state space model. These measurements were used to learn the model using the online EM algorithm, presented in Section 6.1.3.2, starting with random initial conditions for the states, model parameters and the noise covariances. In order to test and validate the model under different traffic conditions, further tests were performed by generating fresh validation datasets from Aimsun. The means of the model parameters estimated from the training phase were fed to the state space equations and the online EM algorithm was executed. The states and parameters from the model were compared with the validation dataset, as shown in Table 6.2 containing the estimation results of the states and in the second column of Table 6.3 containing the estimation results of the turning ratio parameters, saturation parameters and noise covariance matrix $\hat{\mathbf{R}}$. Table 6.2 and 6.3 show that the estimates compare highly with the actual states and model parameters. Note that in Table 6.3, $\|\mathbf{R}\|$ represents the Euclidean norm of the estimated noise covariance $\mathbf{R}$. The % error between the estimated and the actual model parameters is limited to within 6% in all parameters except for $\hat{\alpha}_{13}$ with a % error of 16.5. Such a difference could be due to the low number of vehicles (3 uv/cycle) choosing to turn from arm 1 to arm 3. This can be compared to the larger number of vehicles (10 uv/cycle) choosing to turn from arm 2 to arm 3, resulting in a % error of -0.2 on $\hat{\alpha}_{23}$.

Table 6.2: Estimated states from the validation dataset

Estimate	% RMSE
ζ_1	1.1
ζ_2	1.6
ζ_3	1.1
γ_{O_1}	0.5
γ_{O_2}	0.6
γ_{O_3}	0.8
ϕ_1	1.6
ϕ_2	1.9
ϕ_3	1.2
Average % RMSE	1.2

Table 6.3: Estimated parameters from the validation dataset

Model parameters	Estimated mean results	Actual mean	% Error
$\hat{\alpha}_{12}$	0.519	0.5	3.8
$\hat{\alpha}_{13}$	0.167	0.2	-16.5
$\hat{\alpha}_{14}$	0.314	0.3	4.7
$\hat{\alpha}_{21}$	0.203	0.2	1.5
$\hat{\alpha}_{23}$	0.599	0.6	-0.2
$\hat{\alpha}_{24}$	0.198	0.2	-1.0
$\hat{\alpha}_{31}$	0.199	0.2	-0.5
$\hat{\alpha}_{32}$	0.302	0.3	0.7
$\hat{\alpha}_{34}$	0.499	0.5	-0.2
$\hat{\alpha}_{41}$	0.188	0.2	-6.0
$\hat{\alpha}_{42}$	0.287	0.3	-4.3
$\hat{\alpha}_{43}$	0.515	0.5	5.0
$\hat{S}_1$	50.707	50	1.4
$\hat{S}_2$	56.675	57	-0.6
$\hat{S}_3$	47.263	48	-1.5
$\hat{S}_4$	51.530	50	3.1
$\|\|\hat{\mathbf{R}}\|\|$	9.998×10^{-4}	1×10^{-3}	-0.02

6.1.5 Conclusion

This Chapter proposed an online real-time joint self-estimation method for the state variables, model parameters and noise covariances, which makes use of the stochastic approximation of Robbins and Monro with projection and the EM algorithm.

The proposed online self-estimation algorithm is superior to those presented in Section 5.2.2 in terms of computational and memory demands. This is because it is a recursive algorithm where measurement data is only stored once avoiding the need of batch data storage. This leads to a more efficient implementation with smaller demands on computational hardware requirements as proposed in [360]. Furthermore, the results are superior to those reported in Section 5.2.2, with the mean % error of the online parameter estimates, equal to -0.8% when compared with the mean % error of the quasi-real-time implementation of 1.3% for the 4-arm junction.

The stochastic estimation algorithm proposed in this section was posed as a single variant estimation algorithm, where each parameter was estimated separately from the others. Due to the high level of dependency among these parameters, a multidimensional online estimation algorithm will be presented in the next section, to estimate all the parameters simultaneously.

6.2 Online State and Multidimensional Parameter and Noise Estimation

The novel contribution in this section is the development of a multidimensional online joint state, parameter and noise estimation algorithm. Given the large number of model parameters to be estimated and the high level of dependency among these parameters, this online estimation algorithm is presented to estimate all the parameters simultaneously, as published in the author's work in [361].

6.2.1 Multidimensional stochastic approximation algorithm for the model's parameters

In such a multidimensional approach, let $f^p(\boldsymbol{\theta})$=0, where $f^p(\boldsymbol{\theta})$ represent the partial derivatives of the objective function with respect to the p^{th} parameter, p=1,...,h, and h represents the total number of parameters to be estimated [121]. In addition, let $\mathbf{f}(\boldsymbol{\theta})$ represent the vector field of the partial derivatives of the objective function with respect to each parameter, where $\mathbf{f}(\boldsymbol{\theta}) \triangleq [f^1(\boldsymbol{\theta})\ f^2(\boldsymbol{\theta})\ ...\ f^h(\boldsymbol{\theta})]^T$. The stochastic approximation approach [121] iteratively calculates $\hat{\boldsymbol{\theta}}_t$, where $\hat{\boldsymbol{\theta}}$ is an estimate of the root $\boldsymbol{\theta}^*$, according to:

$$\hat{\boldsymbol{\theta}}_{t+1} = \hat{\boldsymbol{\theta}}_t - \eta_t \frac{\mathbf{g}_t}{c_t} \tag{6.2.1}$$

where $\mathbf{g}_t$ is a noisy observation of $\mathbf{f}(\boldsymbol{\theta}_t)$. η_t and c_t are a sequence of positive numbers (steps), where c_t=$\frac{c}{C_t}$, $C_t = t^d$, $\eta_t = \frac{m}{t^b}$ and c and m are positive constants. Furthermore, d and b are chosen such that the sequence of steps satisfy the following conditions: $\lim_{t\to\infty} c_t = 0$, $\sum_{t=0}^{\infty} \eta_t = \infty$, $\sum_{t=0}^{\infty} (\frac{\eta_t}{c_t})^2 < \infty$ and $\sum_{t=0}^{\infty} \eta_t c_t < \infty$ in order to guarantee convergence of this algorithm by the Kiefer-Wolfwitz minimization process [121].

Substituting the dynamic equations of the model given by Equations (4.2.8) and (4.2.9) in the objective function Equation (6.1.2), the partial differentiation of the objective function with respect to each parameter at cycle t is performed and equated to zero in order to find the minimum value, similar to Section 6.1. Its solution is given recursively by applying Equation (6.2.1). The same algorithm presented in Table 6.1 still applies but in this case, the parameters are estimated simultaneously as one vector of parameters. The projection algorithms presented in Sections 6.1.3.1 and 6.1.3.2 still apply.

6.2.2 Results

The proposed algorithm was tested and validated on a signalized 3-arm junction, with geometry similar to the junction presented in Figure 4.2. Measurements of vehicles entering and leaving each arm were simulated in Aimsun as *Traffic*

State per second for the junction. Traffic light information such as phases, cycle time and the average turning ratios were introduced in Aimsun to reflect realistic traffic characteristics of this junction as shown in Table 6.4. Based on the parameters, data was generated every second.

Table 6.4: Model parameters

Model parameters	α_{12}	α_{13}	α_{21}	α_{23}	α_{31}	α_{32}	Cycle time
Actual mean	0.504	0.496	0.845	0.155	0.805	0.195	110 seconds

From the Aimsun simulation of the junction, the saturation parameters were determined to be compared to the estimated saturation parameters. These parameters correspond to the maximal number of vehicles flowing through the intersection when subject to maximal inflow of vehicles and were found to be equal to S_1= 54 uv/cycle, S_2= 53 uv/cycle and S_3= 65 uv/cycle. The high resolution dataset generated from Aimsun was used as input to the online EM algorithm after adding randomly generated Gaussian distributed measurement noise for evaluation purposes, with the Euclidean norm of $\mathbf{R}$ equal to 1, in order to satisfy the noise signal in the model equations. This data which includes χ_{I_i}, τ_i, and γ_{O_i} for $i = 1,2,3$, with each having $N = 7097$ cycles at 110 seconds per cycle, was used to form the sensor measurement vector $\mathbf{y}$ in order to train the model by the online EM algorithm starting with random initial conditions for the states, model parameters and the noise covariances. Tests were first carried out to establish values for the algorithm parameters C_t and η_t and constant c. The online EM algorithm was executed a number of times, varying one algorithm parameter at a time, while satisfying the conditions described in Section 6.2.1.

In order to validate the model under different traffic conditions, fresh validation datasets were generated from Aimsun. The means of the model parameters estimated from the training phase were fed to the state space equations and the online EM algorithm was executed with the tuned parameters. The states and parameters from the model were compared to the validation dataset.

Figure 6.3 shows the estimation results of one turning ratio parameter, α_{12}, (expected mean value of 0.504) from the training dataset for several different values of c_t. C_t is equal to $t^{0.1}$, while varying c from 1 to 100. Similarly, results were ob-

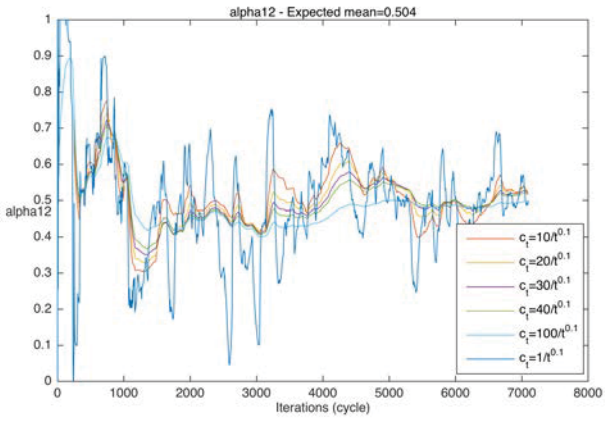

Figure 6.3: Estimation of α_{12} from the training dataset

tained for the changes in the other algorithm parameter η_t. Figure 6.3 shows that the estimation of α_{12} and similarly for the other turning ratios are consistent for all choices of the algorithm parameters. If the model is required to detect rapid fluctuations in the turning ratios, then the algorithm parameters should be chosen to allow tracking of such behaviour. Hence η_t and c_t should be slow decaying, to detect the instantaneous fluctuations in the turning ratios. On the other hand, if a fast convergence rate is required, whilst detecting less rapid fluctuations in the traffic behaviour, then η_t and c_t should be fast decaying. In this study, since the aim was for a fast convergence rate, c_t was set to $\frac{100}{t^{0.1}}$, where $c = 100$ and $C_t = t^{0.1}$. Similarly, from the tests carried out for η_t, this was set to $\frac{1}{t}$.

Figure 6.4 shows the estimation results of α_{12}, the chosen turning ratio parameter from the validation dataset, with $c_t = \frac{100}{t^{0.1}}$ where $c = 100$ and $C_t = t^{0.1}$ and $\eta_t = \frac{1}{t}$. This figure shows that after an initial settling period, the estimate approaches the actual model parameters shown in Table 6.4. The initial instantaneous value in the turning ratio α_{12} during the first 100 cycles, correspond to the flows away from the junction in arms 2 and 3. The flow in arm 3 away from the junction was

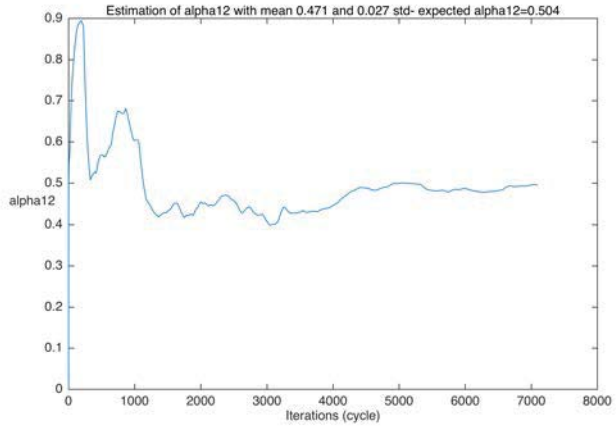

Figure 6.4: Estimation of α_{12} from the validation dataset

approximately zero during the first 100 cycles, hence all vehicles were leaving the junction through arm 2, resulting in an actual and hence estimated turning ratio α_{12} equal to 1. Table 6.5 shows the mean difference between the actual model parameters and the estimated model parameters from the validation dataset for the last 4000 cycles after the estimates show significant departure from the initial conditions. The mean % error of the parameter estimates in Table 6.5 for the online implementation was found to be equal to -4.4% with the % error of the parameter estimates varying between 8.0% and -9.2%, except for that of $\hat{\alpha}_{23}$, being -43.9%, attributed to the low number of vehicles (3 uv/cycle) choosing to turn from arm 2 to arm 3.

A Monte Carlo run of 50 realisations was performed on the 3-arm junction to compare the single variate estimation algorithm results as presented in Section 6.2. A reduction in the mean % error was observed for a 3-arm junction. The Monte Carlo runs resulted in a mean % error of -1.5% for the multivariate parameter estimation as opposed to a mean % error of 3.1% in the single variate parameter estimation. Furthermore, a one-way analysis of variance was used to determine whether there are any statistically significant differences between the two % error means. There is a statistically significant difference between the two % error means with a p-value of 12×10^{-4} for the 3-arm junction.

Table 6.5: Estimated parameters from the validation dataset

Parameters	Estimated mean results	Actual mean	% Error
$\hat{\alpha}_{12}$	0.471	0.504	-6.6
$\hat{\alpha}_{13}$	0.519	0.496	4.6
$\hat{\alpha}_{21}$	0.913	0.845	8.0
$\hat{\alpha}_{23}$	0.087	0.155	-43.9
$\hat{\alpha}_{31}$	0.823	0.805	2.2
$\hat{\alpha}_{32}$	0.177	0.195	-9.2
$\hat{S}_1$	54.432	54	0.8
$\hat{S}_2$	53.342	53	0.6
$\hat{S}_3$	64.622	65	-0.6
$\hat{\mathbf{R}}$	0.999	1	-0.1

6.2.3 Conclusion

This section proposes an online real-time self-estimation method for the state variables, model parameters and noise covariances, that makes use of multidimensional stochastic approximation with projection together with a modified EM algorithm. Such an implementation is highly efficient in computational and memory requirements compared to the quasi real-time approach of Chapter 5. The multidimensional stochastic approximation requires 2.311 minutes of simulation time to complete the full 7097 cycles for a 3-arm junction, while the quasi real-time approach of Chapter 5 requires 5.086 minutes when running Matlab 2015 on a Intel Processor 2.7GHz Intel Core i5. In addition, the quasi real-time approach of Chapter 5 requires batch storage of data of 20 cycles for the estimation of states and parameters and 1500 cycles for the estimation of noise covariances, as opposed to the multidimensional stochastic approximation which makes us of the measurement data only once and requires no batch data storage. It was further observed that a smaller mean % error of -1.0% resulted in the online implementation, compared with a mean % error of 2.8% in the quasi-real-time approach for a 3-arm junction.

Furthermore, this section provides a multivariate estimation to the model parameters. A reduction in the mean % error was observed for a 3-arm junction.

The Monte Carlo runs resulted in a mean % error of -1.5% for the multivariate parameter estimation as opposed to a mean % error of 3.1% in the single variate parameter estimation. Furthermore, a one-way analysis of variance showed that there is a statistically significant difference between the two % error means with a p-value of 12×10^{-4} for the 3-arm junction.

6.3 Online Derivative-Free State and Multidimensional Parameter and Noise Estimation

All algorithms proposed in the previous sections of this Chapter required the partial derivative of the likelihood function with respect to each parameter to be worked out analytically. In practice, this approach is impractical for larger and complex junctions due to the complexity of the derivatives involved in deriving such equations. On the contrary, a derivative-free approach, as proposed in this section (published in the author's work in [139]) is more practical because the analytical derivatives need not be worked out, allowing for easier generalisation of this approach to other junctions and scalability to more complex junctions.

6.3.1 Simultaneous Perturbation Gradient Approximation

A multivariate stochastic approximation method was applied in a recursive way to find the parameters that minimize the negative-log likelihood Equation (6.1.2). In order to find this, the differentiation of the likelihood function with respect to each parameter was used. In sections 6.1 and 6.2, this required that the partial derivative of the likelihood function with respect to each parameter is worked out analytically. In this section, a derivative-free approach is proposed, whereby the analytical derivatives need not be worked out.

The dynamic equation of the model given by Equation (4.2.5) is substituted in the negative-log likelihood $-2E\{G(\boldsymbol{\theta}_t,\hat{\boldsymbol{\theta}}_t,\hat{\mathbf{x}}_t,\hat{\mathbf{Q}},\hat{\mathbf{R}})\}$ from Equation (6.1.2). Due to the complexity of Equation (6.1.2), analytical solutions for the model parameters can be difficult to derive. Hence due to the stochastic nature of the model, a multivariate stochastic approximation method is applied [122].

Consider this negative-log likelihood G to be a differentiable function of $\boldsymbol{\theta}_t$ and let $\mathbf{g}(\boldsymbol{\theta}_t) = \frac{\partial G}{\partial \boldsymbol{\theta}_t}$. Our goal is to find the minimum point $\boldsymbol{\theta}^*$ which corresponds to a root of $\mathbf{g}(\boldsymbol{\theta}_t) = 0$.

The stochastic approximation of Spall [122] iteratively calculates $\hat{\boldsymbol{\theta}}_t$ which is an estimate of the root $\boldsymbol{\theta}^*$, according to:

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\eta}_t \hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}_{t-1}, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) \tag{6.3.1}$$

where $\boldsymbol{\eta}_t$ is a time varying vector of size h, $\boldsymbol{\eta}_t \in R_{\geq 0} \; \forall \, t$, given by $\frac{\mathbf{a}}{t^{\lambda}}$, where $\mathbf{a}$ is a vector of h positive constants, h represents the total number of model parameters to be estimated and

$$\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}_{t-1}, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) = \left[\frac{f_t^+ - f_t^-}{2c_{t_1}\Delta_{t,1}} \ldots \frac{f_t^+ - f_t^-}{2c_{t_h}\Delta_{t,h}} \right]^T \tag{6.3.2}$$

where $\hat{\mathbf{g}}_t$ is an estimate of $\mathbf{g}(\boldsymbol{\theta}_t)$. The major difference between the approach presented in Section 6.1 and 6.2 and this section is Equation (6.3.2). This equation gives the gradient approximation and replaces the analytical partial derivative of the likelihood function with respect to each parameter [122]. The gradient approximation of Equation (6.3.2) requires two measurements of the likelihood function at instance t, denoted by f_t^+ and f_t^-, with positive and negative perturbation of the parameters, where $f_t^+ = -2E\{G(\hat{\boldsymbol{\theta}}_{t-1} + \mathbf{c}_t\boldsymbol{\Delta}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) + \varepsilon_t^{(+)}\}$ and $f_t^- = -2E\{G(\hat{\boldsymbol{\theta}}_{t-1} - \mathbf{c}_t\boldsymbol{\Delta}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) + \varepsilon_t^{(-)}\}$. The gradient approximation is performed by taking the difference between f_t^+ and f_t^- divided by a sequence of perturbations denoted by $\mathbf{c}_t$. $\mathbf{c}_t$ is a time varying vector of size h, where, $\mathbf{c}_t \in R_{\geq 0} \; \forall \, t$, with each element denoted as c_{t_h}, representing the perturbation on the parameters. To ensure convergence of $\hat{\mathbf{g}}_t$, $\mathbf{c}_t$ is a sequence of perturbation sizes that decreases to 0 and is given by $\frac{\mathbf{c}}{t^{\omega}}$, where $\mathbf{c}$ is a vector of h positive constants. $\varepsilon_t^{(+)}$ and $\varepsilon_t^{(-)}$ represent measurement noise terms that satisfy $E(\varepsilon_t^{(+)} - \varepsilon_t^{(-)} | \hat{\boldsymbol{\theta}}_{1,\ldots,t}\boldsymbol{\Delta}_t) = 0 \; \forall \, t$, where $\boldsymbol{\Delta}_t \in R^h$ is a vector of h mutually independent zero-mean random variables $\{\Delta_{t,1}, \Delta_{t,2}, \ldots, \Delta_{t,h}\}$.

Moreover, to ensure convergence of the stochastic approximation estimation [122], for each element of $\boldsymbol{\eta}_t$, denoted as η_{t_h},

- $\sum_{t=0}^{\infty} \eta_{t_h} = \infty$,
- $\sum_{t=0}^{\infty} (\frac{\eta_{t_h}}{c_{t_h}})^2 < \infty$ and
- $\sum_{t=0}^{\infty} \eta_{t_h} c_{t_h} < \infty$.

To ensure asymptotic normality, each element of $\mathbf{c}$ and $\mathbf{a}, \lambda, \omega > 0$, $\lambda - 2\omega > 0$ and $3\omega - \frac{\lambda}{2} \geq 0$ [122].

Note that $\boldsymbol{\eta}_t$ and $\mathbf{c}_t$ should be chosen to allow for either detection of instantaneous fluctuations in the model parameters, hence slow decaying $\boldsymbol{\eta}_t$ and $\mathbf{c}_t$ as necessary (example $\boldsymbol{\eta}_t$ set to $\frac{1}{t}$ and $\mathbf{c}_t$ set to $\frac{1}{t^{0.02}}$) or else if a fast convergence rate is required whilst detecting less rapid fluctuations in the traffic behaviour, then $\boldsymbol{\eta}_t$ and $\mathbf{c}_t$ should be chosen to represent fast decaying functions (example $\boldsymbol{\eta}_t$ set to $\frac{1}{1000t}$ and $\mathbf{c}_t$ set to $\frac{1}{t^{0.2}}$). The choice of $\mathbf{c}$ is further influenced by the magnitude of the estimated parameters. For example since the magnitude of the turning ratios varies between 0 and 1, then the corresponding elements of $\mathbf{c}$ for the estimation of the turning ratios are chosen to satisfy this range. Hence for example for the estimation of the turning ratios the corresponding elements of $\mathbf{c}$ are set to 1.

To ensure randomness, independence and symmetry in the perturbation, the conditions given in [122] for $\boldsymbol{\Delta}_t$ should be satisfied and hence independent Bernoulli random variables with outcomes ± 1 should be assigned to $\Delta_{t,h}$. The noise terms ε_t^+ and ε_t^- are large enough to achieve a significant effect in degrading $\hat{\mathbf{g}}_t$.

The traffic flow turning ratio parameter estimates from Equation (6.3.1) are not guaranteed to satisfy constraints (4.2.12) and (4.2.13). Hence the projection algorithms presented in Sections 6.1.3.1 and 6.1.3.2 still apply.

Similar to the unconstrained parameters, the noise covariance $\hat{\mathbf{Q}}_t$ is estimated by:

$$\hat{\mathbf{Q}}_t = \hat{\mathbf{Q}}_{t-1} - \eta_t \hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) \tag{6.3.3}$$

where, in this case, $\hat{\mathbf{Q}}_0$ is initialized to be a positive definite matrix, η_t is a sequence of positive numbers, $f_t^+ = -2E\{G(\hat{\boldsymbol{\theta}}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1} + c_t\boldsymbol{\delta}_t, \hat{\mathbf{R}}_{t-1}) + \varepsilon_t^{(+)}\}$, $f_t^- = -2E\{G(\hat{\boldsymbol{\theta}}_{t-1}, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1} - c_t\boldsymbol{\delta}_t, \hat{\mathbf{R}}_{t-1}) + \varepsilon_t^{(-)}\}$ and $\boldsymbol{\delta}_t \in R^{m\times m}$ represents a symmetric matrix. The upper triangle elements of $\boldsymbol{\delta}_t$ are sampled from a Bernoulli distribution [122] with outcome ± 1, while the lower triangle elements were set to ensure symmetry. Note that the l^{th} and j^{th} element is denoted as $\delta_{t_{l,j}}$.

Moreover, in this case, c_t is a sequence of positive numbers, denoting perturbation on all elements of $\mathbf{Q}$ and $\hat{\mathbf{g}}_t$ is a symmetric matrix of size $m \times m$, where

$$\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}_t, \hat{\mathbf{x}}_t, \hat{\mathbf{Q}}_{t-1}, \hat{\mathbf{R}}_{t-1}) = \begin{bmatrix} \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{1,1}}} & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{1,2}}} & \cdots & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{1,m}}} \\ \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{1,2}}} & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{2,2}}} & \cdots & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{2,m}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{1,m}}} & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{2,m}}} & \cdots & \frac{f_t^+ - f_t^-}{2c_t\delta_{t_{m,m}}} \end{bmatrix} \tag{6.3.4}$$

The same conditions as before apply to ε_t, c_t and η_t.

To ensure that $\hat{\mathbf{Q}}_t$ is positive definite, the following transformation is applied to the matrix of Equation (6.3.3), where a Schur method [362] is used for the matrix square root operation:

$$\hat{\mathbf{Q}}_t = (\hat{\mathbf{Q}}_t\hat{\mathbf{Q}}_t)^{\frac{1}{2}} \tag{6.3.5}$$

This process is also repeated for the estimation of the noise covariance $\hat{\mathbf{R}}_t$, thus ensuring that the symmetric and positive definite properties of the noise covariance matrix are retained.

This leads to the online EM estimation algorithm as described in Table 6.6.
Also note that switching between saturated and unsaturated traffic conditions is avoided in this case, unlike in Chapter 5 and Sections 6.1 to 6.2, since the state-space representation of Equation 4.2.5 used in this algorithm (Table 6.6) represents both saturated and unsaturated traffic conditions.

Table 6.6: Online EM algorithm

Initialise estimates for $\hat{\boldsymbol{\theta}}$, $\hat{\mathbf{Q}}$, $\hat{\mathbf{R}}$ and G from Equation (6.1.3) Iterate for every time step t and measure $\mathbf{y}(t)$. **E-step** Run Kalman-filter recursions to compute $\hat{\mathbf{x}}_{t\|t}$. **M-step** Minimise -2E$\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_t)\}$ over $\boldsymbol{\theta}$ for dynamic traffic conditions using Equation (6.2.1) and satisfying Equations (4.2.12) and (4.2.13) (with projection if necessary). Update $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$, with $\hat{\boldsymbol{\theta}}_t$ to reflect the traffic conditions per arm. Minimise -2E$\{G(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_t)\}$ over $\hat{\mathbf{Q}}$ as in Equations (6.3.3) and similarly for $\hat{\mathbf{R}}$. Increment t.

6.3.2 Results

The proposed algorithm for the online joint estimation of states, model parameters and process and measurement noise covariances, was tested and validated on signalized 3-arm and 4-arm junctions, with geometry similar to Figures 4.3 and 4.2 respectively. Aimsun micro traffic simulation software [354] was used to generate traffic data at one second intervals for a typical working day. Traffic light information such as phases, cycle time and the average turning ratios were introduced in Aimsun to reflect traffic characteristics for each junction. The traffic information for the 4-arm and the 3-arm junctions are defined in Table 6.7 and Table 6.8 respectively.

Table 6.7: Model parameters for the 4-arm junction

Model parameter	α_{12}	α_{13}	α_{14}	α_{21}	α_{23}	α_{24}	α_{31}
Value	0.5	0.2	0.3	0.2	0.6	0.2	0.2
Model Parameter	α_{32}	α_{34}	α_{41}	α_{42}	α_{43}	Cycle time	
Value	0.3	0.5	0.2	0.3	0.5	110 seconds	

Table 6.8: Model parameters for the 3-arm junction

Model parameters	α_{12}	α_{13}	α_{21}	α_{23}	α_{31}	α_{32}	Cycle time
Actual mean	0.504	0.496	0.845	0.155	0.805	0.195	104 seconds

The saturation parameters of the junction were determined from Aimsun, as these correspond to the maximum number of vehicles flowing through the intersection when subject to a high inflow of vehicles. For the 4-arm junction, the saturation parameters were found to be equal to S_1= 120 uv/cycle, S_2= 127 uv/cycle, S_3= 80 uv/cycle and S_4= 120 uv/cycle, while for the 3-arm junction these were found to be equal to S_1= 118 uv/cycle, S_2= 120 uv/cycle and S_3= 80 uv/cycle. Randomly generated Gaussian distributed measurement noise was added to the high resolution dataset from Aimsun for evaluation purposes, which was then used as input to the online EM algorithm. The Euclidean norm of $\mathbf{R}$ was set to 1.

The generated data which includes χ_{I_i}, τ_i, and γ_{O_i} for $i = 1,2,3,4$, for a 4-arm junction, with each having $N = 786$ cycles at 110 seconds per cycle, was used to form the sensor measurement vector $\mathbf{y}$ input to the proposed online EM algorithm to jointly estimate the states, the model parameters and noise covariances, starting with random initial conditions for the states, model parameters and the noise covariances. Similarly for the 3-arm junction. Figure 6.5 shows the flow of vehicles away from the junction for arm 1 for a typical working day from 6.00am of one day to 6.00am of the next day for a 4-arm junction. The morning peak period is between 7.00am till 8.00am whilst the evening peak period is between 6.30pm till 7.30pm for the first arm. Similarly for the other arms.

In order to test and validate the model under different traffic conditions, further tests were performed by generating fresh validation datasets from Aimsun. The means of the model parameters estimated from the training phase were used to initialize the state space equations and the online EM algorithm was executed. The states estimated from the model were compared with the validation dataset.

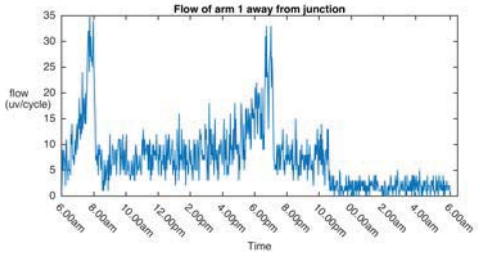

Figure 6.5: Flow away from junction

Similar to the previous sections, a RMSE measure is defined to determine the accuracy of the estimation results, given by $J \triangleq \sqrt{\frac{\sum_N (p(t)-\hat{p}(t))^2}{N}}$.

A prediction error measurement is defined to determine the accuracy of the one-step ahead predictions of the outflow away from the junction. A t-step prediction is defined in [6] whereby if t is set to 1, this denotes one-step ahead predictions. The one-step ahead prediction error is expressed as $K \triangleq \frac{1}{N}\sum_N (\gamma_{O_i}(t) - \hat{\gamma}_{O_i}(t))$, where $\hat{\gamma}_{O_i}(t)$ is the prediction obtained from the model Equations presented in Chapter 4 and the state estimates $\hat{\boldsymbol{x}}(t-1)$.

The joint estimation results obtained for the signalized 4-arm junction will be first presented and analysed. Figure 6.6 shows $\hat{\alpha}_{12}$, estimated from the training dataset for the corresponding elements of $\mathbf{a}$ which affect the step size of the stochastic approximation equation. When the corresponding element of $\mathbf{a}$ was set to $\frac{1}{1000}$, this gave the best estimation results with a percentage difference of 4.8%, observed for the last 100 cycles. Similarly, results were obtained for changes in the other algorithm parameters λ, $\mathbf{c}$ and ω. As discussed in Section 6.3.1, these results show that if the model is required to detect rapid fluctuations in the turning ratios, then the algorithm parameters should be chosen to allow for the detection in the instantaneous fluctuations in the turning ratios. If a fast convergence rate is required, whilst detecting less rapid fluctuations in the traffic behaviour, then $\boldsymbol{\eta}_t$ and $\mathbf{c}_t$ should be fast decaying. In this study, since the aim was for a fast convergence rate, $\boldsymbol{\eta}_t$ was chosen to be equal to $\frac{\mathbf{a}}{t}$ and $\mathbf{c}_t$ was chosen to be equal to

$\frac{\mathbf{c}}{t^{0.2}}$, for the estimation of the turning ratios, where the corresponding element of $\mathbf{a}$ was set to $\frac{1}{1000}$, $\lambda = 1$, $\omega = 0.2$. The choice of $\mathbf{c}$ for the perturbation of the parameters was further influenced by the magnitude of the estimated parameters. Since the turning ratios were of magnitude between 0 and 1, then the corresponding elements of $\mathbf{c}$ for the estimation of the turning ratios were chosen to be within this range. Figure 6.7 shows the perturbation f_t for α_{12}, where the corresponding element of $\mathbf{c}$ for turning ratio α_{12} was equal to 1. This figure shows that the perturbation on the turning ratio is between 0 and 1 because of the magnitude constraint of the turning ratio. Hence, the unconstrained estimates for the turning ratios are not far from the desired constraints. If the perturbation is out of the constrained region as can be observed in Figure 6.7 during the initial stages, then the turning ratio would still be projected onto the desired region by applying the projection function described in Sections 6.1.3.1 and 6.1.3.2.

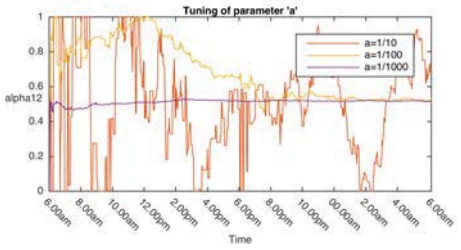

Figure 6.6: Estimation of α_{12} with changing algorithm parameter a

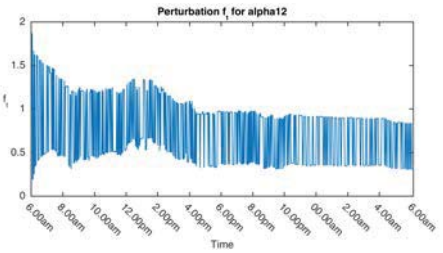

Figure 6.7: Perturbation for turning ratio α_{12}

To determine the standard deviation of the noise ε_t^+ and ε_t^-, the standard deviation of the difference between the positive and negative perturbations on the parameters, i.e. $f_t^+ - f_t^-$ was computed. This was found to have an order of

magnitude of 10^3. Hence, Gaussian distributed noise ε_t^+ and ε_t^- was set to mean of 0 and standard deviation of 1000 to have a significant degrading effect on $\hat{\mathbf{g}}_t$.

Furthermore, Figure 6.8 shows $\hat{\alpha}_{12}$ from the training dataset, with the algorithm parameters chosen to give fast convergence estimations. This figure shows that the estimates approach the actual model parameters shown in Table 6.7. These estimations are further compared in the same figure to a moving average calculated from the data itself, with a window of size 20 cycles. The estimations are smoother when compared to a moving average, filtering out the rapid fluctuations in the traffic behaviour and hence resulting in a fast convergence rate.

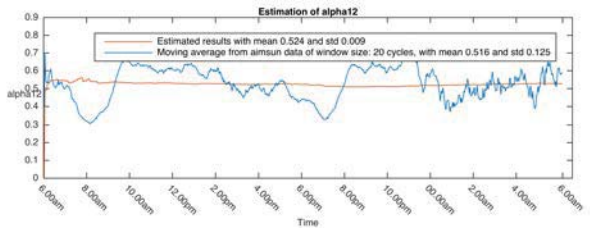

Figure 6.8: Online estimation of turning ratio α_{12} from the training dataset

Table 6.9 shows that the mean difference between the estimated model parameters from the training dataset and the actual model parameters for the last 100 cycles, varies between -14.3% to 10.0%. The highest errors were recorded on α_{13} and α_{14} which could be due to the low number of vehicles choosing to turn from arm 1 to arm 3 (4 uv/cycle) and arm 4 (6 uv/cycle) respectively. Moreover, the average Euclidean norm of the estimated noise covariances $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ were found to be 0.998 and 1.004 respectively. The average estimated Euclidean norm is very similar to the actual Euclidean norm of $\hat{\mathbf{R}}$ with a percentage difference of 0.4% which is deemed to be satisfactory.

Figure 6.9 shows the estimate of the queue length for arm 3 when carrying out the joint estimation of model states and parameters. The estimates approach the measured queue length with a RMSE of 7.0% of the maximum queue length. Similar behaviour was observed for the other arms, with a RMSE of 8.5%, 3.7%

Table 6.9: Estimated model parameters from the training dataset

Model parameters	Estimated mean results	Actual mean	% Difference
$\hat{\alpha}_{12}$	0.524	0.5	4.8
$\hat{\alpha}_{13}$	0.220	0.2	10.0
$\hat{\alpha}_{14}$	0.257	0.3	-14.3
$\hat{\alpha}_{21}$	0.192	0.2	-4.0
$\hat{\alpha}_{23}$	0.626	0.6	4.3
$\hat{\alpha}_{24}$	0.182	0.2	-9.0
$\hat{\alpha}_{31}$	0.181	0.2	-9.5
$\hat{\alpha}_{32}$	0.310	0.3	3.3
$\hat{\alpha}_{34}$	0.508	0.5	1.6
$\hat{\alpha}_{41}$	0.186	0.2	-7.0
$\hat{\alpha}_{42}$	0.298	0.3	-0.7
$\hat{\alpha}_{43}$	0.516	0.5	3.2
$\hat{S}_1$	119.026	120	-0.8
$\hat{S}_2$	126.654	127	-0.3
$\hat{S}_3$	81.142	80	1.4
$\hat{S}_4$	121.113	120	0.9

and 8.9% of the maximum queue length respectively. This leads to an average RMSE of 7.0% for the 4-arm junction. These estimation results are further compared to the estimation results of Pecherkova *et al.* [6], where the RMSE was equal to 3.590 for the queue length, with data exhibiting a maximum of 40 vehicles. This represents a slightly higher RMSE of 9.0% of the maximum queue length.

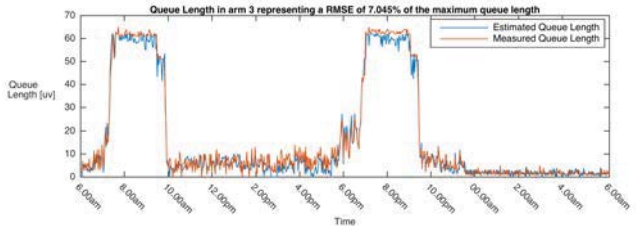

Figure 6.9: Queue Length in arm 3

Figure 6.10 shows the one-step ahead prediction for the outflow of arm 1 for all cycles, with a prediction percentage error of 5.5% of the maximum outflow. Similar behaviour was observed for the other arms, obtaining 3.4%, 6.3% and 5.9% of the maximum outflow for each arm respectively. This leads to an average one-step ahead prediction error measurement of 5.3% for all arms. The one step-ahead prediction error measurement was also worked out for the early morning cycles only i.e. from midnight till 6 am. The prediction error measurement for this time duration was equal to 2.056 for the outflow for arm 1, with data exhibiting a maximum of 8 vehicles. This represents a prediction error of 25.7% of the maximum outflow. Similarly for the other arms, the prediction error measurements for this time duration were 20.5%, 23.9%, and 24.2%. This leads to an average prediction percentage error of 23.6% of the maximum outflow for all 4 arms. On the other hand, the prediction error measurement in [6] was 1.328 for the outflow, with data exhibiting a maximum of 6 vehicles. This represents a prediction percentage error of 22.1% of the maximum outflow, a slightly better prediction error when compared to the average prediction percentage error of 23.6% as obtained in this work, with the maximum outflow having the same order of magnitude as in [6].

Although both Pecherkova *et al.*'s work in [6] and this work give very similar results, the approach presented in this thesis points to a more direct estimation methodology by including parameters that have physical significance in the parametric model, such as the saturation and turning ratio parameters, that are estimated directly by the algorithm. This contrasts with [6] where different model parameters that have no physical significance were used. Thus, no prior analysis was required for the model parameters in this work, unlike in [6], where some parameters with no physical significance required prior tuning. Moreover state and parameter augmentation as applied in [6] is avoided.

Simulation results were also obtained for a 3-arm junction. Figure 6.11 shows $\hat{\alpha}_{21}$ from the training dataset, with the algorithm parameters chosen to give fast convergence. Similar to a 4-arm junction, this figure shows that the estimates

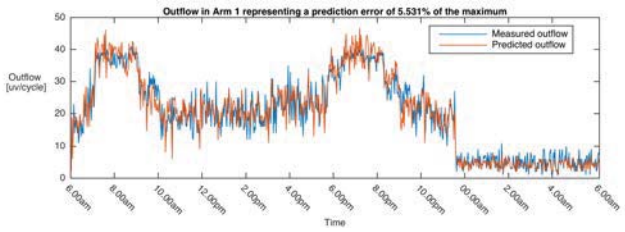

Figure 6.10: Outflow in arm 1

approach the actual model parameters shown in Table 6.8. These estimations are further compared in the same figure to a moving average calculated from the data itself, with a window of size 20 cycles. The estimations are smoother when compared to a moving average, filtering out the rapid fluctuations in the traffic behaviour and hence leading to a fast convergence rate.

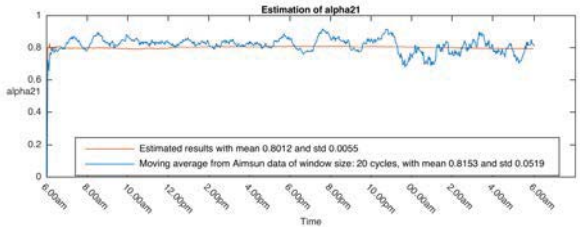

Figure 6.11: Online estimation of turning ratio α_{21} from the training dataset

Table 6.10 shows that the mean difference between the estimated model parameters from the training dataset and the actual model parameters for the last 100 cycles, varies between -10.3% to 8.7%. The highest errors were recorded on α_{23} and α_{32} which is again attributable to the low number of vehicles choosing to turn from arm 2 to arm 3 (8 uv/cycle) and from arm 3 to arm 2 (4 uv/cycle) respectively.

The RMSE of the state estimates on the validation datasets between the online implementation requiring derivatives as proposed in Section 6.2 and the derivative-free online implementation in this section are compared, starting with the same initial conditions for the model parameters for each estimation method. In the online implementation of Section 6.2 the partial derivative of the objective function

Table 6.10: Estimated model parameters from the training dataset

Model parameters	Estimated mean results	Actual mean	% Difference
$\hat{\alpha}_{12}$	0.499	0.504	-1.0
$\hat{\alpha}_{13}$	0.501	0.496	1.0
$\hat{\alpha}_{21}$	0.801	0.845	-5.2
$\hat{\alpha}_{23}$	0.139	0.155	-10.3
$\hat{\alpha}_{31}$	0.763	0.805	-5.2
$\hat{\alpha}_{32}$	0.212	0.195	8.7
$\hat{S}_1$	119.015	118	-0.8
$\hat{S}_2$	119.844	120	-0.3
$\hat{S}_3$	79.379	80	1.4

with respect to each parameter had to be evaluated analytically during the design process. On the contrary, in this approach, the partial derivatives are evaluated numerically, resulting in a derivative-free stochastic approximation algorithm with a faster and less tedious implementation. This makes it attractive for practical implementation as the design process does not require calculations of analytical derivatives, making the approach scalable and generalisable to other junctions. Moreover, the online implementation of Section 6.2 requires vector/matrix multiplications which may contribute to an increase in computation time.

The RMSE estimates were expressed as a percentage of $\sqrt{\frac{1}{N}\sum_N p^2(t)}$ to yield a normalized measurement over time for each state. To compare both approaches, one figure of merit was computed for each implementation, which is the average error over all 12 state variables, for the 4-arm junction and 9 state variables for the 3-arm junction. This gave an average % RMSE of 0.6 for both implementations for a 4-arm junction. For a 3-arm junction the average % RMSE with the online implementation presented in Section 6.2 resulted in 0.723% while the % RMSE with the online implementation proposed in this section resulted in 0.725%. Thus no deterioration in the estimate accuracy is obtained as a result of the numerical approach applied in this section. Furthermore, the online implementation as proposed in Section 6.2 requires 16.376 seconds of simulation time to complete the full 786 cycles for a 4-arm junction, while the online implementation proposed

in this work requires 10.436 seconds when running Matlab 2015 on a Intel Processor 2.7GHz Intel Core i5, which is significantly less. In the work of Section 6.2, the evaluation of complex partial derivatives for each parameter required vector/matrix multiplications at each time instance, significantly increasing the computational effort involved. This contrasts with the reduced computational effort of the derivative-free approach where only simple perturbations are required, which as shown, had no significant impact on the accuracy of the results.

6.3.3 Conclusion

To contribute to the autonomicity of traffic light systems, this Section proposes a derivative-free multidimensional online joint state, parameter and noise estimation algorithm for state space macro models of traffic junctions which makes use of the simultaneous perturbation gradient approximation with projection.

The online implementation in this Section is faster in computational terms when compared to the previous online implementation presented in Section 6.2. In fact the simulation time for the proposed online implementation has been reduced by a factor of 53.5% for a 4-arm junction, without any deterioration in accuracy. Furthermore, the design and implementation proposed in this section are more practical when compared with schemes that make use of the EKF, such as [99], because the partial derivative of the objective function with respect to each parameter need not be worked out analytically, hence resulting in a derivative-free stochastic approximation algorithm that is generalisable and scalable to multiple junctions. It also avoids the divergence problems associated with the EKF, as well as state and parameter augmentation for joint estimation.

The work presented in this Chapter is aimed to form an integral part of the real-time multiple model estimation algorithm to be discussed in detail in the following chapter.

Chapter 7

Real-time multiple model estimation

7.1 Introduction

Traffic conditions of signalized junctions are subject to several potential dynamical regimes that change over abruptly due to traffic incidents or unanticipated network obstructions. Under such conditions the model structure and parameters are not constant or varying slowly, but switch value abruptly, a phenomenon known as *jump* dynamics [51]. Hence the dynamics of the traffic system could be represented not by one model, but by a set of different models each corresponding to a given condition. Furthermore, it might be impossible to identify these models *a priori*. In this chapter, a multiple model algorithm is developed to detect such abrupt changes in the dynamic behaviour of signalized traffic junctions. The algorithm also estimates the dynamics and model parameters of the different active regimes, such as turning ratios, saturation flows and noise parameters, which need not be known beforehand, and detects which mode is active at any given time. In addition, the model state variables and noise covariances are also estimated.

7.2 Multiple Model Estimation

Suppose that H distinct modes of operation are postulated initially, each representing one specific regime. Following the generic notation of [51], only one particular mode from these H could be active at any given time t. Denote this as

mode f, where $f \in [1,..,H]$. The dynamics of each mode f can be represented by model M^f having the same form as in Equation (4.2.5) given in Chapter 4, with state space matrices $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$ and $\mathbf{D}^f$, noise covariances $\mathbf{Q}^f$ and $\mathbf{R}^f$ and model parameters $\mathbf{\Theta}^f$ which could either be known *a priori* or else estimated on-line. Let $E^f(t)$ denote the event that the system dynamics at time t correspond to model M^f with parameters $\mathbf{\Theta}^f$. Also let $S_j(t)$ represent one specific sequence of such events from start up to time t (e.g. $S_1(t) = \{E^1(1), E^2(2), E^2(3), ..., E^1(t)\}$ denotes an example of one possible sequence). Since every element of this sequence has H possibilities, there exist H^t different possible sequences at time t in all, whereby only one of these H^t has actually taken place.

The question that follows is how to identify which of the modes is active at time t. This can be addressed by finding the probability that a model in the set $\{M^1, M^2, ..., M^H\}$ is best representing the current observations. A Kalman filter is matched to each mode and a probabilistic framework is formulated to identify the posterior probability of the event that the actual model sequence conditioned on the observation set, $\mathbf{Y}^t$ is S_f. This approach is faced with evaluation complications and high computation and storage requirements [51], as the number of possible sequences to be considered increases exponentially with time, hence becoming impractical to implement. Three main possible sub-optimal solutions have been proposed to mitigate such complications: *Generalised Pseudo-Bayes (GPB) method*, *the Interacting Multiple Model (IMM) method* and *the lower bounding approach*. The GPB method [363, 364] and the IMM approach [135, 365, 366] introduce the concept of *pruning* to limit the increase in the number of Kalman filters and *merging* for the state estimation information to be propagated to the filters. These approaches offer a reliable suboptimal solution to the problem of switching dynamics but computational complexity could still be rather high.

To reduce the computational demand even further, the lower bounding approach can be considered. To explain this approach, the problem of identifying the most probable model from a given set of possible modes in a non-*jump* dynamics scenario is first considered. Bayes' rule is applied to infer the posterior probability

of model M^f conditioned on $\mathbf{Y}^t$:

$$Pr(M^f|\mathbf{Y}^t) = \frac{p(\mathbf{y}_t|M^f,\mathbf{Y}^{t-1})Pr(M^f|\mathbf{Y}^{t-1})}{\sum_{j=1}^{H} p(\mathbf{y}_t|M^j,\mathbf{Y}^{t-1})Pr(M^j|\mathbf{Y}^{t-1})} \tag{7.2.1}$$

where $Pr()$ denotes probability and $p()$ denotes a distribution.

Assuming linear models with Gaussian noise, a Kalman filter matched to each mode can be used to calculate the mean square estimate of state $\mathbf{x}_t$ according to each model M^f, denoted as $\hat{\mathbf{x}}_t^f$ and the corresponding covariance of the estimation error denoted as $\mathbf{P}^f(t|t-1)$. The likelihood function can then be evaluated by:

$$p(\mathbf{y}_t|M^f,\mathbf{Y}^{t-1}) = -\frac{1}{(2\pi)^{\frac{1}{2}}|\mathbf{Z}_t^f|^{\frac{1}{2}}} exp^{-\frac{1}{2}(\mathbf{y}_t-\hat{\mathbf{y}}_t^f)'(\mathbf{Z}_t^f)^{-1}(\mathbf{y}_t-\hat{\mathbf{y}}_t^f)} \tag{7.2.2}$$

where the corresponding variance $\mathbf{Z}_t^f$ attributed to model M^f is estimated as:

$$\mathbf{Z}_t^f = \mathbf{C}^f\mathbf{P}^f(t|t-1)\mathbf{C}^{f'} + \mathbf{R}^f \tag{7.2.3}$$

$\hat{\mathbf{y}}_t^f$ denotes the estimate of the observation $\mathbf{y}_t$, where:

$$\hat{\mathbf{y}}_t^f = \mathbf{C}^f\hat{\mathbf{x}}_t^f + \mathbf{D}^f\mathbf{z}(t) \tag{7.2.4}$$

Thus $(\mathbf{y}_t - \hat{\mathbf{y}}_t^f)$ in Equation (7.2.2) represents the difference between the observation $\mathbf{y}_t$ and its estimate $\hat{\mathbf{y}}_t^f$, referred to as the residual.

In a *jump* dynamics scenario, where switching between different modes could occur at anytime t, Equation (7.2.1) is used together with a small lower bound placed on the computed candidate probabilities, to prevent a mode from being *locked-out* [367] because its associated probability has gone to zero. Hence the lower bounding approach requires only H Kalman filters at every time instant, one for each possible mode, as opposed to GBP which requires a maximum of H^k Kalman filters pruned down to H after d time steps and IMM which also re-

quires H Kalman filters at every time instant.

The lower bounding approach that was discussed in this section will now be applied to signalized traffic junctions. As noted previously, the model development for this junction as discussed in Chapter 4 given by Equations (4.2.5) and the online joint estimation algorithm for the model states and parameters as discussed in Chapter 6.3 will be used.

7.3 Multiple model approach for signalized junctions

Suppose that a set of candidate models for different regimes of operation are known *a priori*. Individual state estimates $\hat{\mathbf{x}}_t^f$ are obtained by running a Kalman filter for each candidate model. The lower bounding approach together with Bayes' rule Equation (7.2.1) are applied to detect which mode is active at a given time instant. The active mode gives the highest probability in Equation (7.2.1) from all the other modes. The resultant state estimate, denoted as $\hat{\mathbf{x}}_t$ as noted by [51], is either calculated as a combination of the individual state estimates from the bank of Kalman filters given by:

$$\hat{\mathbf{x}}_t = \sum_{f=1}^{H} \hat{\mathbf{x}}_t^f Pr(M^f|\mathbf{Y}^t) \tag{7.3.1}$$

or else as the state estimate from the Kalman filter corresponding to the mode exhibiting maximum posterior probability in Equation (7.2.1), denoted by $\hat{\mathbf{x}}_{t_{max:Pr(M^f|\mathbf{Y}^t)}}$. In the former option, the covariance matrix of the state estimation error is denoted as $\mathbf{P}(t|t)$ and is given by:

$$\mathbf{P}(t|t) = \sum_{f=1}^{H} [\mathbf{P}^f(t|t) + (\hat{\mathbf{x}}_t^f - \hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_t^f - \hat{\mathbf{x}}_t)^T] Pr(M^f|\mathbf{Y}^t) \tag{7.3.2}$$

The model parameters $\boldsymbol{\Theta}^f$ for each active mode and the noise covariances matrices $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ are also updated using the multivariate stochastic approximation

method as discussed in Section 6.3. Note that $\boldsymbol{\Theta}^f$ is equivalent to the estimated vector of model parameters $\hat{\boldsymbol{\theta}}$ for each active model.

Several scenarios that represent different operating regimes (modes) were considered to be tested and validated on a signalized traffic junction, with normal day-to-day traffic conditions and lane closures representing the different modes. Note that for each case, the active mode is to be detected and the model parameters for each active mode are to be learnt if these are not known *a priori*, as detailed below. Also note that the model set is either limited to a known finite number of modes or else allowed to grow in time.

Thus, the following possibilities could be considered:

- Known number of modes and known model parameters
- Known number of modes and:
  - known model parameters for the respective lane closures
  - unknown model parameters
- Unknown number of modes and:
  - known model parameters for the respective lane closures
  - unknown model parameters

Each case is further explained as follows.

7.3.1 Known number of modes and known model parameters

In this case a known total number of modes H is assumed and the model parameters $\boldsymbol{\Theta}$, the model matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ and the process and measurement noise covariances $\mathbf{Q}$ and $\mathbf{R}$ are assumed to be known for each candidate model. The Kalman filter is executed for each candidate and the posterior distribution for each is calculated as detailed in Table 7.1.

Table 7.1: Multiple model approach with known number of modes and known model parameters

Define $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$, $\mathbf{D}^f$, $\mathbf{Q}^f$ and $\mathbf{R}^f$ for each candidate model.
Step 1: At every time step t, measure $\mathbf{y}(t)$.
For each candidate model f with known $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$, $\mathbf{D}^f$, $\mathbf{Q}^f$ and $\mathbf{R}^f$:
Run Kalman-filter recursion in order to compute $\hat{\mathbf{x}}_t^f$
Calculate the posterior probability distribution for each candidate model f given in Equation (7.2.1).
Calculate $\hat{\mathbf{x}}_t$ using Equation (7.3.1) or by $\hat{\mathbf{x}}_{t_{max:Pr(M^f\|\mathbf{Y}^t)}}$ representation.
Calculate $\mathbf{P}(t\|t)$ using Equation (7.3.2) or by $\mathbf{P}(t\|t)_{max:Pr(M^f\|\mathbf{Y}^t)}$.
t=t+1
Repeat from Step 1.

7.3.2 Known number of modes and known or unknown model parameters

In this case the total number of modes H is assumed known and the model parameters Θ and the process and measurement noise covariances $\mathbf{Q}$ and $\mathbf{R}$ for each model are updated in time by applying the online self-estimation algorithm proposed in Chapter 6.3. The posterior distribution for each candidate model is calculated as detailed in Table 7.2.

For comparison reasons this test is carried out for two particular cases: i) when the turning ratios for the respective arm closures are known, meaning that α_{ij} is known where j denotes the closed arm while estimating all the other model parameters within vector Θ^f; For example if an accident occurs in arm 1, which blocks the flow of vehicles leaving the junction through this arm, then the turning ratios α_{21} and α_{31} are known to be equal to 0 while α_{23} and α_{32} are set equal to 1. All other turning ratios and other model parameters are estimated; ii) when all model parameters are unknown including α_{ij} and hence all parameters are estimated.

Table 7.2: Multiple model approach with known number of modes and known or unknown model parameters

Initialise estimates for $\hat{\boldsymbol{\Theta}}^f$, $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{R}}^f$ and $-2E\{G\}$ Iterate for every time step t and measure $\mathbf{y}(t)$. **E-step** Run Kalman-filter recursions for each candidate model f to compute $\hat{\mathbf{x}}_t^f$. **M-step** For each candidate model f, minimise -2E$\{G(\boldsymbol{\Theta}^f,\hat{\boldsymbol{\Theta}}_t^f)\}$ over $\boldsymbol{\Theta}^f$ for dynamic traffic conditions while satisfying Equations (4.2.12) and (4.2.13) (with projection if necessary). Minimise -2E$\{G(\boldsymbol{\Theta}^f,\hat{\boldsymbol{\Theta}}_t^f)\}$ over $\hat{\mathbf{Q}}^f$ and similarly for $\hat{\mathbf{R}}^f$. Calculate the posterior probability distribution for each candidate model f given in Equation (7.2.1). Calculate $\hat{\mathbf{x}}_t$ using Equation (7.3.1) or by $\hat{\mathbf{x}}_{t_{max:Pr(M^f\|\mathbf{Y}^t)}}$ representation. Calculate $\mathbf{P}(t\|t)$ using Equation (7.3.2) or by $\mathbf{P}(t\|t)_{max:Pr(M^f\|\mathbf{Y}^t)}$. For the current active model, that is the model which corresponds to the $max:Pr(M^f\|\mathbf{Y}^t)$ update $\hat{\mathbf{A}}^f$, $\hat{\mathbf{B}}^f$, $\hat{\mathbf{C}}^f$, $\hat{\mathbf{D}}^f$, with $\hat{\boldsymbol{\Theta}}_t^f$ to reflect the traffic conditions per arm. Increment t.

7.3.3 Unknown number of modes and known or unknown model parameters

In this case the total number of modes H is not known *a priori* and the model parameters $\boldsymbol{\Theta}$ and the process and measurement noise covariances $\mathbf{Q}$ and $\mathbf{R}$ are updated in time. Assume that only f distinct modes are known initially. A self-organized model allocation approach [51] is applied, whereby new modes are learnt in real-time, hence growing the multiple model set. Just after the f^{th} model has been selected by the mode estimation algorithm, as a result of a new mode being detected active, a freshly initialized model is introduced by adding a newly initialized parameter vector for this model. The probability density function of the freshly initialized model is initially wide and narrows as the variance decreases as a consequence of learning. This procedure of adding a fresh local model once the previous one has been selected is repeated continuously, hence

growing the candidate models in the multiple model set. Thus there is always one spare model ready to accept a new regime that has not appeared before, as detailed in Table 7.3.

Table 7.3: Unknown number of modes and known or unknown model parameters

Initialise $\hat{\mathbf{\Theta}}^f$, $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{R}}^f$ estimates for f local models and $-2E\{G\}$
Iterate for every time step t.
Step 1:
Measure $\mathbf{y}(t)$.
E-step
Run Kalman-filter recursions for each local candidate models f to compute $\hat{\mathbf{x}}_t^f$, where $f = 1...H$.
M-step
For such candidate model f, $(f = 1...H)$, minimise -2E$\{G(\mathbf{\Theta}^f, \hat{\mathbf{\Theta}}_t^f)\}$ over $\mathbf{\Theta}^f$ for dynamic traffic conditions while satisfying Equations (4.2.12) and (4.2.13) (with projection if necessary).
Minimise -2E$\{G(\mathbf{\Theta}^f, \hat{\mathbf{\Theta}}_t^f)\}$ over $\hat{\mathbf{Q}}^f$ and similarly for $\hat{\mathbf{R}}^f$.
Calculate the posterior probability distribution for such candidate model f given in Equation (7.2.1).
Just after the H^{th} model has been selected, as a result of a new mode being detected active, introduce a new model with randomly selected $\hat{\mathbf{\Theta}}^f$, $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{R}}^f$ and let $H \rightarrow H+1$.
Calculate $\hat{\mathbf{x}}_t$ using Equation (7.3.1) or by $\hat{\mathbf{x}}_{t_{max:Pr(M^f|\mathbf{Y}^t)}}$ representation.
For the current active model, that is the model which corresponds to the $max : Pr(M^f|\mathbf{Y}^t)$ update $\hat{\mathbf{A}}^f$, $\hat{\mathbf{B}}^f$, $\hat{\mathbf{C}}^f$, $\hat{\mathbf{D}}^f$, with $\hat{\mathbf{\Theta}}_t^f$
to reflect the traffic conditions per arm.
Increment t and repeat from Step 1.

For comparison reasons, similar to Section 7.3.2, this test is carried out for two particular cases: i) when the turning ratios for the respective arm closures are known while estimating all the other parameters within vector $\mathbf{\Theta}^f$; ii) when all model parameters are unknown including α_{ij} and hence all parameters are estimated.

7.4 Simulation Experiments

The proposed algorithms for the multiple model approach were tested and validated on signalized 3-arm and 4-arm junctions, with geometry represented in Figure 4.2 and 4.3 respectively and with model parameters presented in Chapter 6.3.2. As in the previous chapters, Aimsun micro traffic simulation software [354] was used to generate traffic data at one second intervals for a typical working day. Randomly generated Gaussian distributed measurement noise was added to the high resolution dataset from Aimsun for evaluation purposes, which was then used as input to the proposed algorithms. The Euclidean norm of $\mathbf{R}$ was set equal to 1 respectively.

This data which includes the inflow in arm i, γ_{I_i}, the occupancy ϕ_i and the outflow from arm i, γ_{O_i}, for $i = 1,2,3$, with each having $N = 830$ cycles at 104 seconds per cycle, was used to form the sensor measurement vector $\mathbf{y}$ input to the proposed algorithms for the 3-arm junction, similar to the previous chapter. Similarly for the 4-arm junction.

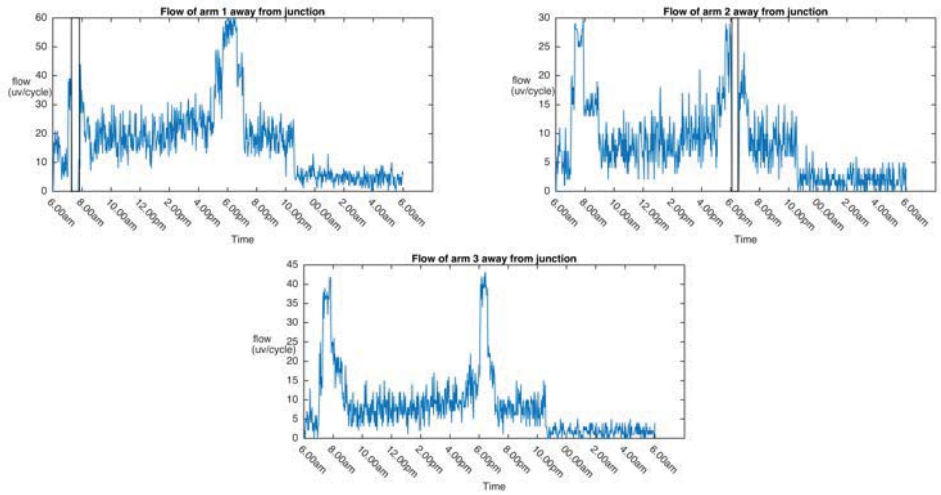

Figure 7.1: Flow away from junction

Figure 7.1 shows the flow of vehicles away from the 3-arm junction for one arm for a typical working day from 6:00 am of one day to 6:00 am of the next day. The morning peak period is between the 37^{th} and 68^{th} cycle (between 7.00am

till 8.00am) whilst the evening peak period is between the 405^{th} and 427^{th} cycle (between 5.45pm till 6.20pm) for the first arm. Similarly for the other arms and for the 4-arm junction. Furthermore, the black windows represented in Figure 7.1, show the duration of arm 1 and arm 2 closures. Arm 1 closure is shown in Figure 7.2, marked with a dark red cross, where all outflow lanes in arm 1 are blocked. Similarly for arm 2 closure.

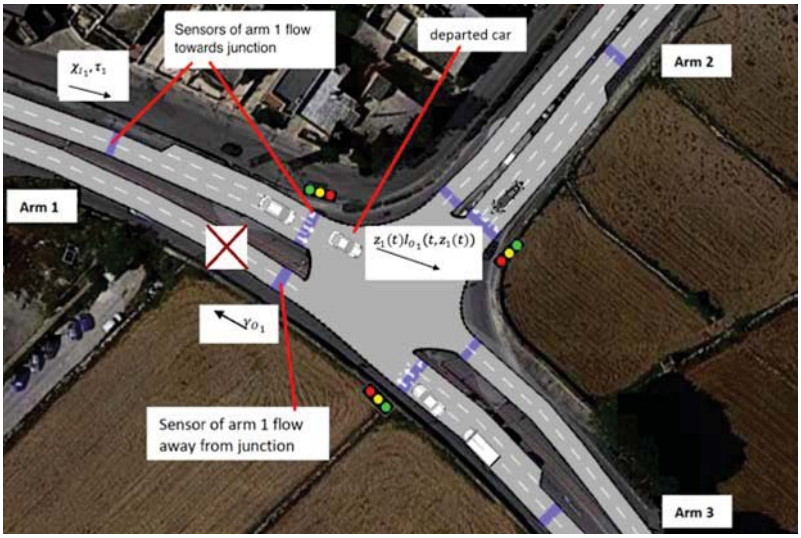

Figure 7.2: 3-arm signalized junction with arm 1 closure

Arm 1 closure occurs between the 43^{rd} and 63^{rd} cycle (between 7.15am till 7.50am) whilst arm 2 closure occurs between the 416^{th} and 433^{rd} cycle (between 6.00pm till 6.30pm) during the morning and evening peak periods respectively. Similarly arm closure scenarios were simulated on the 4-arm junction. Arm 1 closure occurs between the 43^{rd} and 63^{rd} cycle (between 7.18am till 7.55am) whilst arm 2 closure occurs between the 416^{th} and 433^{rd} cycle (between 6.42pm till 7.13pm).

Furthermore, a lane closure scenario was simulated on the 3-arm junction. Figure 7.3 shows the lane closure for arm 1, crossed in dark red, where one of the inflow lanes in arm 1 is blocked. Fresh dataset was generated from Aimsun representing normal traffic conditions and a lane closure was simulated during a short

period. Lane closure occurs between the 52^{nd} and 87^{th} cycle (between 7.30am till 8.30am).

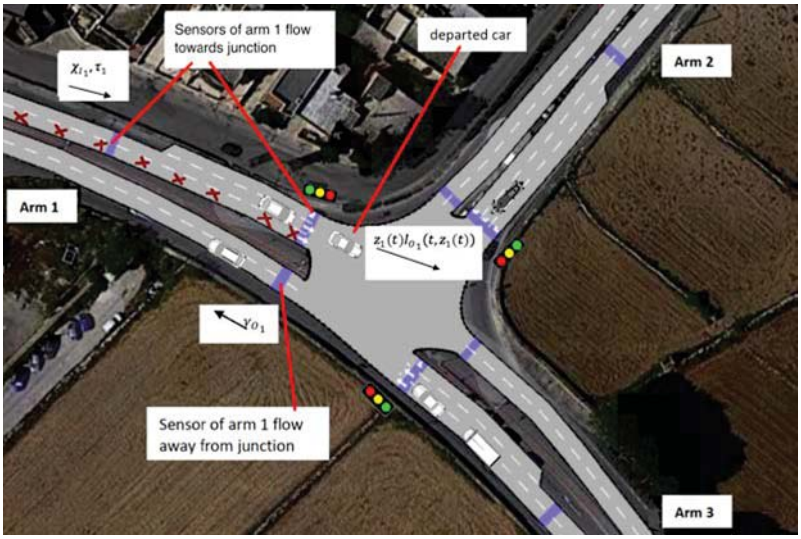

Figure 7.3: 3-arm signalized junction with one lane in arm 1 being closed

A Root Mean Square Error (RMSE) measure is defined to determine the accuracy of the estimation results, similar to Chapter 6.3.2. The RMSE is expressed as a percentage of $\sqrt{\frac{1}{N}\sum_{N} p^2(t)}$ to yield a normalized measurement over time.

7.4.1 Results

Figure 7.4 shows the estimated switching conditions for the 3-arm junction with simulated arm closures, when the algorithm in Table 7.1 is executed, where the total number of candidate models H in this example was set to 3 and the system parameters $\mathbf{\Theta}^f$, the system matrices $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$ and $\mathbf{D}^f$ and the measurement noise covariance $\mathbf{R}^f$ are assumed to be known for each candidate model. The subplots on the right in Figure 7.4 zoom in on the situation between 6.58am and 8.10am so as to clearly show the transition before and after arm 1 closure. The estimated switching conditions compare well with the actual switching traffic conditions, with lane closures detected at the 44^{th} and 64^{th} cycle (between 7.16am till 7.51am) and the 417^{th} and 434^{th} cycle (between 6.01pm till 6.31pm), respectively, essentially a one cycle delay between the actual and the estimated

switching conditions. This is attributed to the Kalman filter state estimations that depend on the estimated states from the previous cycle and the current measurements.

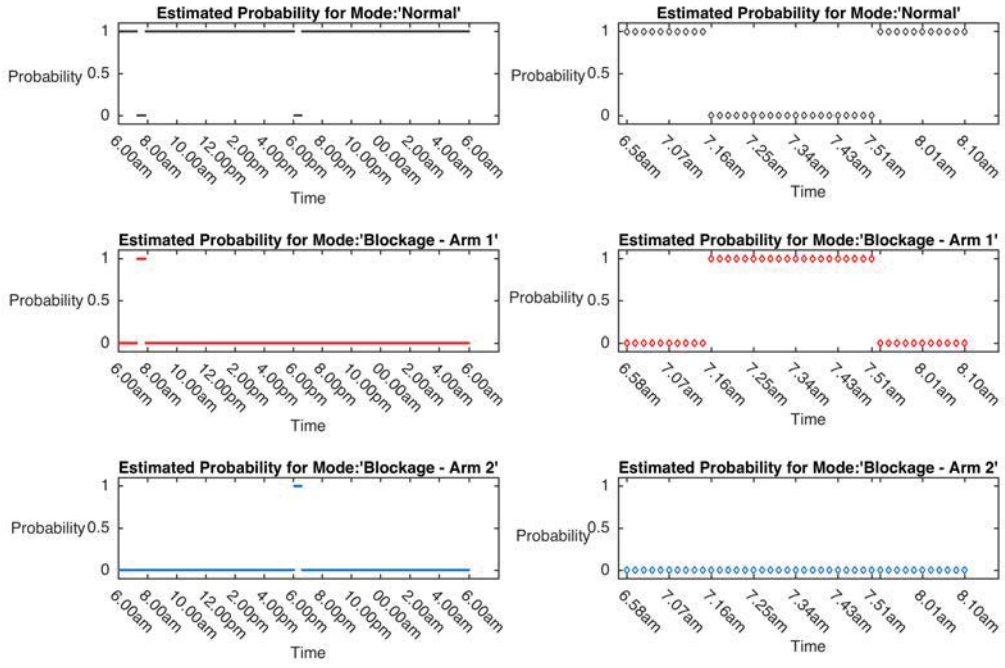

Figure 7.4: Estimated switching conditions with known number of modes and known model parameters

Table 7.4 compares the % RMSE of every individual state estimate obtained when applying Equation (7.3.1) with those obtained when applying $\hat{\mathbf{x}}_{t_{max:Pr(M^j|\mathbf{Y}^t)}}$ respectively for the 3-arm junction, and with the system parameters $\boldsymbol{\Theta}^f$, the system matrices $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$ and $\mathbf{D}^f$ and the measurement noise covariance $\mathbf{R}^f$ assumed to be known. For comparison reasons, one figure of merit was computed for both cases, consisting of the mean % RMSE. This was computed by calculating the mean value of the % RMSE over all 9 state variables. A mean % RMSE of 0.687 was obtained for the first case and 0.686 for the second case. Hence from the estimated results, there is no significant difference observed between

the maximum aposteriori estimates and the estimates resulting from Equation (7.2.1) based upon the combination of Kalman filters. This occurs because the probabilities of the inactive models dropped to 0.01%, hence resulting in no significant differences between the state estimations of the two cases.

Table 7.4: % RMSE of estimates for the maximum aposteriori or for the combination of Kalman filters

Estimates	RMSE on the state estimates from maximum aposteriori	RMSE on the state estimates from the combination of Kalman filters
$\hat{\zeta}_1$	1.564	1.565
$\hat{\zeta}_2$	0.947	0.945
$\hat{\zeta}_3$	1.171	1.171
$\hat{\gamma}_1$	0.510	0.510
$\hat{\gamma}_2$	0.265	0.266
$\hat{\gamma}_3$	0.310	0.311
$\hat{\phi}_1$	0.510	0.509
$\hat{\phi}_2$	0.443	0.443
$\hat{\phi}_3$	0.459	0.458
Average	0.687	0.686

Further tests were carried out to estimate the system parameters Θ^f and the process and measurement noise covariances $\mathbf{Q}^f$ and $\mathbf{R}^f$, with known number of modes H. The algorithm of Table 5.2 is executed for the 3-arm junction with simulated arm closures. For comparison reasons, tests are carried out for two cases as follows: i) when the model parameters for the respective lane closures are known, meaning that α_{21} and α_{31} are set equal to 0, while α_{23} and α_{32} are set equal to 1 during closure of arm 1. Similarly, α_{12} and α_{32} are set equal to 0, while α_{13} and α_{23} are set equal to 1 during closure of arm 2. All other turning ratios and other model parameters are estimated as in Table 5.2; ii) all model parameters are estimated.

The switching conditions observed during both cases when the algorithm in Table 5.2 is executed were similar to the ones obtained in Figure 7.4, with arm closures detected between the 44^{th} and 64^{th} cycle (between 7.16am till 7.51am) and be-

tween the 417^{th} and 434^{th} cycle (between 6.01pm till 6.31pm) respectively. Both the estimated and the expected switching conditions are similar, with a one cycle delay in detecting the switching conditions in the estimation results. Such minor delay is expected due to the Kalman filter state estimations.

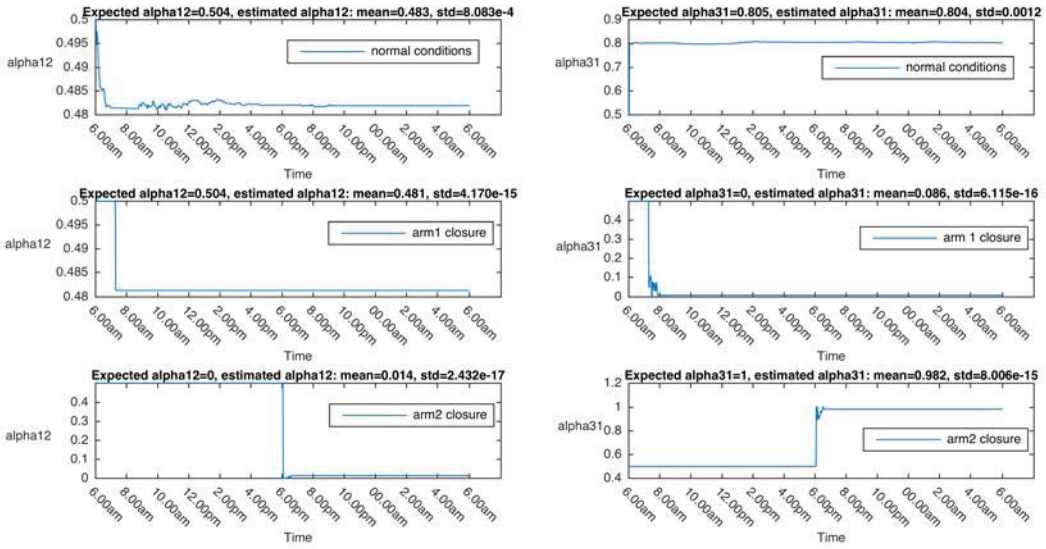

Figure 7.5: Estimated turning ratios

Figure 7.5 shows the estimation results for α_{12} and α_{31} when all model parameters including all the turning ratios are estimated for the 3-arm junction. The estimated values compare well with the expected values. In fact, the estimation of α_{12} resulted in a mean of 0.483, for the last 10 cycles during normal traffic conditions and 0.481 for the last 10 cycles during arm 1 closure, where the expected value for α_{12} for both conditions was 0.504. During arm 2 closure the estimated value was found to be equal to 0.014, when the expected value was 0. Similarly for α_{31} and the other model parameters. These results are superior to the estimation results obtained when executing the online estimation algorithm presented in Chapter 6.3 without multiple model estimation, with the same initial and simulation conditions presented to both estimation algorithms. For example, the estimation of α_{12} resulted in a mean of 0.481, for the last 10 cycles during

normal traffic conditions and a mean of 0.481 for the last 10 cycles during arm 1 closure, similar to the previous case, where the expected value for α_{12} for both conditions was 0.504. However, during arm 2 closure the estimated value was found to be equal to 0.473, when the expected value was 0. Similarly for the other model parameters.

Further tests were carried out, where this time the number of modes H is unknown and hence modes and their models are learnt as they occur in real-time, while estimating the time-variant system parameters $\boldsymbol{\Theta}$ and the process and measurement noise covariances $\mathbf{Q}$ and $\mathbf{R}$. Only one local model is known initially and this denotes 'normal' traffic conditions. Figure 7.6 and 7.7 show the estimated switching conditions for this test for the 3-arm and 4-arm respectively with the actual switching conditions denoting arm closures for such junctions described in Section 7.4. The subplots on the right in Figure7.6 zoom in on the situation between 6.58am and 8.10am and the subplots on the right in Figure 7.7 zoom in on the situation between 7.05am and 8.13am so as to clearly show the transition before and after arm 1 closure.

As shown in Figure 7.6 *spare* models are introduced at the 44^{th} cycle (at 7.16am) and 417^{th} cycles (at 6.01pm) respectively when closures of the arms are detected for the 3-arm junction. Before the 44^{th} cycle, there is no trace for the probability for 'blockage - arm 1', because this model is introduced at 7.16am. Similarly, before the 417^{th} cycle, there is no trace for the probability for 'blockage - arm 2', because this model is introduced at 6.01pm. The system switches back to normal conditions at the 65^{th} cycle (at 7.53am) and at the 435^{th} cycle (at 6.32pm) respectively. This results in one cycle delay in detecting arm 1 and arm 2 closures and another cycle delay in switching back to normal conditions following the respective arm closures. Similarly, for a 4-arm junction, in Figure 7.7, *spare* models are introduced at the 44^{th} cycle (at 7.20am) and 417^{th} cycles (at 6.44pm) respectively when closures of the arms are detected. Before the 44^{th} cycle, there is no trace for the probability for 'blockage - arm 1', because this model is introduced at 7.20am. Similarly, before the 417^{th} cycle, there is no trace for the probability for 'blockage - arm 2', because this model is introduced at 6.44pm.

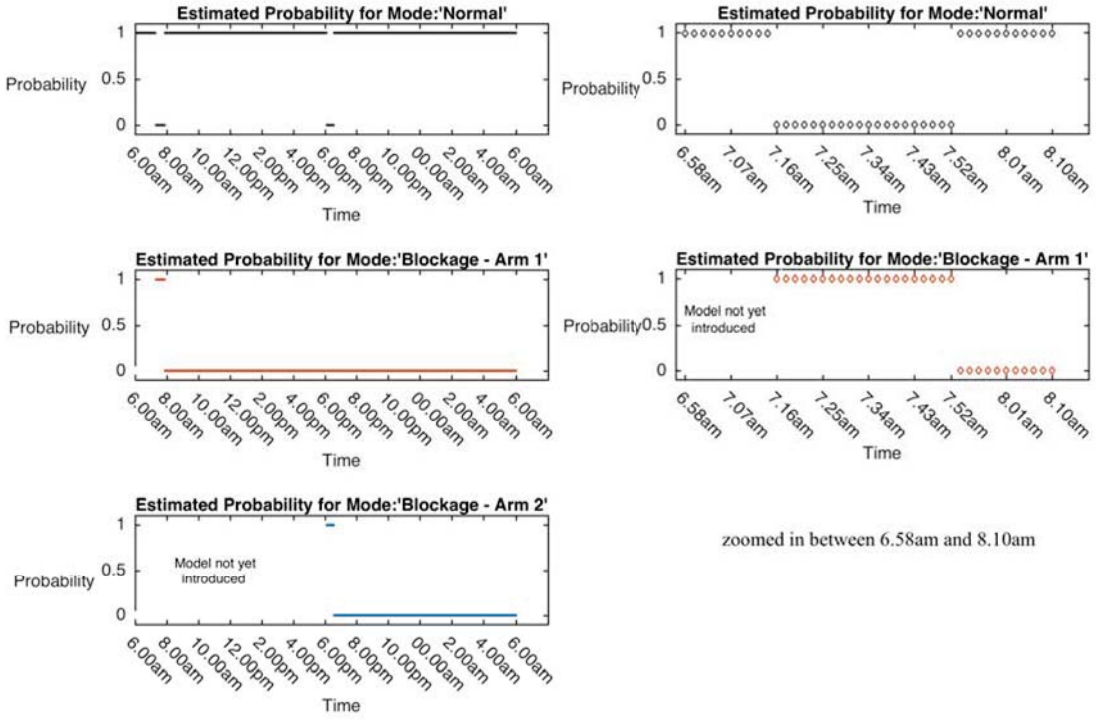

Figure 7.6: Estimated switching conditions with unknown number of modes and unknown model parameters for the 3-arm junction

The system switches back to normal conditions at the 65^{th} cycle (at 7.59am) and at the 435^{th} cycle (at 7.17pm) respectively. This results in one cycle delay in detecting arm 1 and arm 2 closures and another cycle delay in switching back to normal conditions following the respective arm closures in a 4-arm junction as in the previous case.

Furthermore, the RMSE estimates of the system states for known and unknown number of modes were compared. For comparison reasons, an average % RMSE was computed for both cases. The resulting average % RMSE was 1.1% for each case for the 3-arm junction while for the 4-arm junction an average % RMSE of 0.8325% with known number of modes and 0.8421% with unknown number of modes were obtained. Although the total number of modes was not known *a*

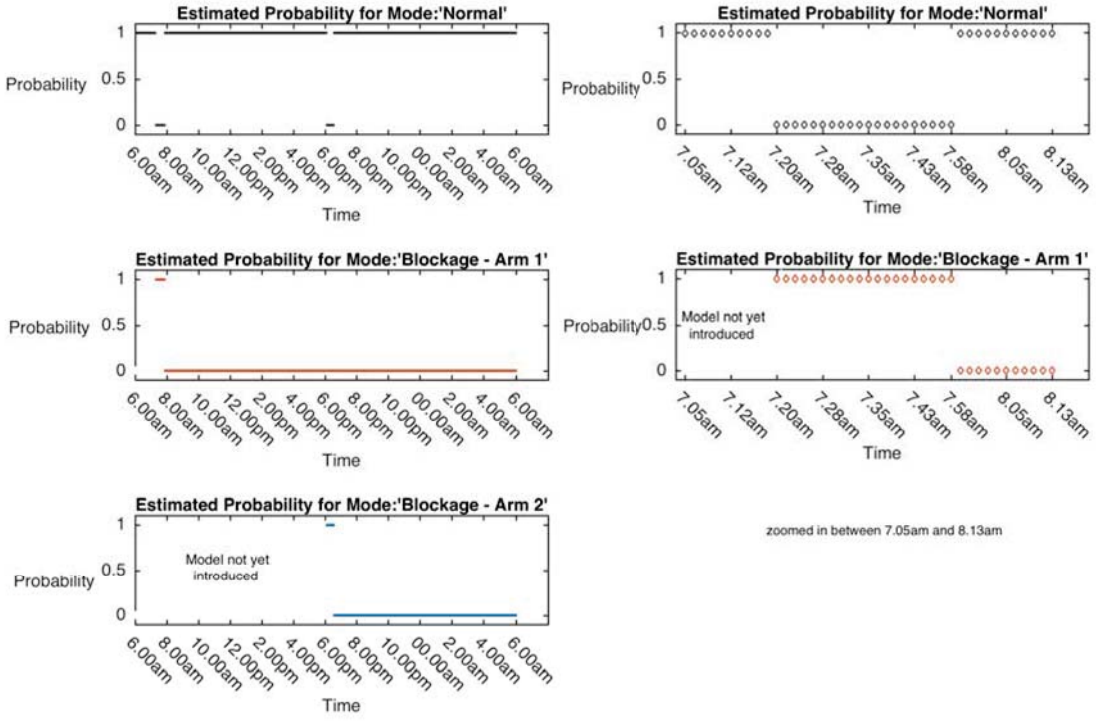

Figure 7.7: Estimated switching conditions with unknown number of modes and unknown model parameters for the 4-arm junction

priori, the average % RMSE did not significantly deteriorate when comparing results to the case when the total number of modes was assumed known *a priori* for both junctions.

Another test was carried out, for the 3-arm junction with simulated lane closure as discussed in Section 7.4. Only one local model is known initially and this denotes 'normal' traffic conditions. The number of modes H is unknown and hence modes and their models are learnt as they occur in real-time, while estimating the time-variant system parameters $\boldsymbol{\Theta}$ and the process and measurement noise covariances $\mathbf{Q}$ and $\mathbf{R}$. Figure 7.8 shows the estimated switching conditions for this test for the 3-arm junction with the actual switching conditions denoting lane closure for such junction described in Section 7.4. The subplot on the

right in Figure 7.8 zoom in on the situation between 6.45am and 9.01am so as to clearly show the transition before and after lane closure. Before the 53^{rd} cycle,

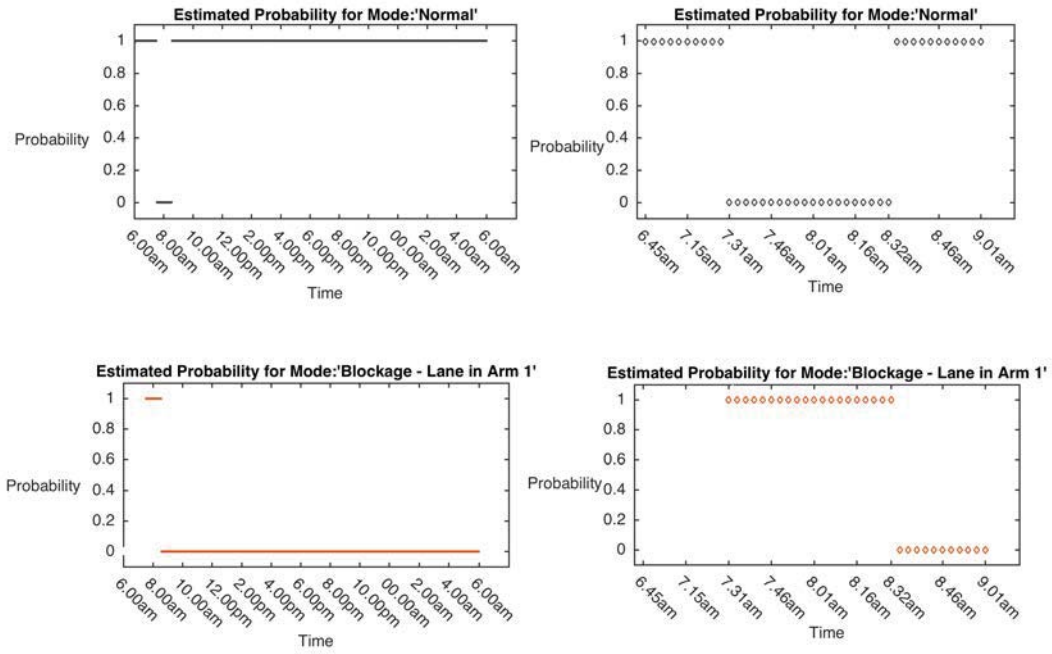

Figure 7.8: Estimated switching conditions for the 3-arm junction subject to lane closure

there is no trace for the probability for 'blockage - lane in arm 1', because this model is introduced at 7.31am. The system switches back to normal conditions at the 88^{th} cycle (at 8.32am). This results in one cycle delay in detecting lane closure and another cycle delay in switching back to normal conditions following the lane closure in the 3-arm junction as in the previous case. Table 7.5 shows the RMSE of every individual state estimate from the maximum aposteriori for the 3-arm junction, expressed as a percentage of $\sqrt{\frac{1}{N}\sum_{N} p^2(t)}$ to yield a normalized measurement over time.

These results are superior to the %RMSE of the state estimates obtained when executing the online estimation algorithm presented in Chapter 6.3 without multiple model estimation, with the same initial and simulation conditions including

Table 7.5: % RMSE of the state estimates expressed as a percentage

$\hat{\zeta}_1$	$\hat{\zeta}_2$	$\hat{\zeta}_3$	$\hat{\gamma}_{I_1}$	$\hat{\gamma}_{I_2}$	$\hat{\gamma}_{I_3}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\phi}_3$
8.352	7.286	9.151	2.509	2.274	3.320	0.602	0.375	0.514

lane closure presented to both algorithms. For example the estimate $\hat{\zeta}_1$ resulted in a percentage RMSE of 20.8% without multiple model estimation as shown in Figure 7.9, with the largest discrepancy between the estimate and the actual measurement being observed during the lane closure, between 7.30am and 8.30am. This represents an increase of 12.5% in error in $\hat{\zeta}_1$ when compared to the estimate shown in Table 7.5. Similarly for the other estimates.

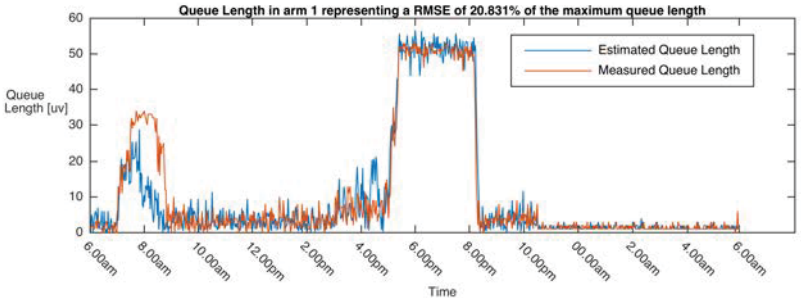

Figure 7.9: Queue length in arm 1

7.5 Conclusion

This chapter proposed a stochastic multiple model joint estimation approach for representing *jump* dynamics within urban signalized traffic junctions. Traffic network irregularities like incidents lead to abrupt changes in the dynamic behaviour of the traffic system. Hence these changing conditions are better represented by different regimes or modes, where each mode is represented by a distinct model.

The proposed algorithm learns the mode dynamics in real-time and determines which mode is active at a given time. An online dual estimation algorithm is

adopted to jointly estimate traffic states in real-time such as queue lengths, occupancies and flows, as well as the model parameters such as turning ratios, saturation flows and noise parameters. The adopted approach makes use of an EM algorithm, modified for real-time estimations, with a Kalman filter estimating the traffic states and a multivariate gradient-based approach estimating the model parameters, as proposed in Section 6.3.1. The algorithm computes residual data by comparing the measured information with the estimated states and assign probabilities to each model in the set. The algorithm is also able to learn the unlimited number of potential dynamical regimes not anticipated *a priori* if the residual characteristics do not match any of the learnt modes in the set that is known so far. Through this approach, the system will automatically configure and grow its model set if new modes appear in the scene. A limitation to the proposed stochastic multiple model approach is that this work assumes a Gaussian likelihood distribution which could be violated when having low traffic counts in the junction. However, this assumption is not critical, as this work is highly crucial during congestions.

The developed algorithm was tested by simulating experiments of 3-arm and 4-arm signalized traffic junction, with several arm closures, with either having i) fixed number of known changing conditions and hence a finite set of models or ii) an unlimited number of potential dynamic regimes where the system automatically learnt the distinct system behaviours as different models. The results in this chapter highlight the accuracy in estimating and detecting the switching conditions, with minor delays. Results also demonstrate the performance of the algorithm with a known finite set of models and with an unlimited number of modes. For the 3-arm junction, both cases performed equally well with an average % RMSE of 1.050% on the estimates of the system states for both tests, while for the 4-arm junction an average % RMSE of 0.8325% and 0.8421% were obtained with known or unknown number of modes respectively.

The results also highlight the accuracy of the estimates when executing the online estimation algorithm with multiple model estimation as proposed in this Chapter compared to the the online estimation algorithm presented in Chapter 6.3,

without multiple model estimation. For example the estimate $\hat{\zeta}_1$ resulted in a percentage RMSE increase of 12.5% without multiple model estimation when compared to the result obtained with multiple model estimation.

This work is aimed to form part of adaptive control loops for traffic light systems that are able to autonomously adjust to changing traffic conditions so as to ensure efficient vehicle flows, as will be shown in the next chapter.

Part II

Autonomic control of signalized junctions

Chapter 8

Model Predictive Control

8.1 Introduction

Previous works discussed in Chapters 5 to 7 assumed knowledge of signal times z_i, representing the ratio of the green signal time for arm i to the total cycle time. This was fixed to a constant value for each arm i. The fixed time plans for the 3-arm and 4-arm junctions of Section 4.2 were taken from the actual junctions and tuned to some predefined traffic flows by the network operator.

Fixed time plans work well when traffic conditions are consistent. However performance is degraded when traffic conditions are subject to changes because of time-dependent high demands or during unusual occurrences such as traffic incidents or unanticipated network obstructions, causing significant changes to the normal traffic conditions. To address these situations, traffic-responsive signal time plans are required which make use of the online data given from the sensors. Models describing the dynamics of the traffic behaviour within a junction can be used to predict future traffic flows. There has been great advancement in developing such traffic responsive systems as discussed in Chapter 1, however the controller parameters are not tuned to changing traffic behaviour so as to ensure consistent and optimal control. An adaptive system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is thus desirable, leading towards an autonomic system.

This work focuses on a MPC based approach which can tune the controller parameters to reflect changing traffic conditions. This consists of a model-based

prediction, online optimization, and a rolling time horizon. The traffic models and the online joint estimation techniques discussed in Part I are required to be able to describe and predict the traffic flow dynamics in the future, and as result these enable the controller to look ahead into the future to avoid short sighted decisions. Moreover, by using the rolling horizon procedure, feedback is introduced, which makes the controllers more robust to disturbances and model mismatch errors.

8.2 Problem Formulation

MPC is required to minimise the queue lengths in the network whilst adhering to some constraints. The control process can be described by the elements described in the following:

8.2.1 Prediction Model

To predict the future queue lengths used for evaluating the objective function based on the information of current queue lengths, a prediction model is required. The prediction model used in this work is simplified from Equation (4.2.5) presented in Chapter 4. The state variables in this prediction model comprise the queue length $\zeta_i(t)$ and the inflow $\gamma_{I_i}(t)$. The occupancy given by $\phi_i(t)$ which was included as a state variable in Equation (4.2.5) is now unnecessary in the prediction model because it does not affect the estimation of future queue lengths. Hence it is removed in the prediction model given by Equation (8.2.1). However, as discussed in Section 4.2, the occupancy is still included in the estimation model because it is useful to detect unusual situations such as lane blockage during unsaturated conditions.

For a 3-arm junction, the state variables formulating the prediction model are reduced from a total of 9 states to a total of 6. The output vector given by Equation (8.2.1) comprises the queue length, since the queue length will be minimised in the network.

$$
\begin{bmatrix} \zeta_1(t+1) \\ \zeta_2(t+1) \\ \zeta_3(t+1) \\ \vdots \\ \zeta_n(t+1) \\ \gamma_{I_1}(t+1) \\ \gamma_{I_2}(t+1) \\ \gamma_{I_3}(t+1) \\ \vdots \\ \gamma_{I_n}(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ldots & \ddots & \ddots & \ddots & \ddots & \ldots & \ddots \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ldots & \ddots & \ddots & \ddots & \ddots & \ldots & \ddots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \\ \zeta_3(t) \\ \vdots \\ \zeta_n(t) \\ \gamma_{I_1}(t) \\ \gamma_{I_2}(t) \\ \gamma_{I_3}(t) \\ \vdots \\ \gamma_{I_n}(t) \end{bmatrix}
$$

$$
- \begin{bmatrix} I_{O_1} & 0 & 0 & \cdots & 0 \\ 0 & I_{O_2} & 0 & \cdots & 0 \\ 0 & 0 & I_{O_3} & \cdots & 0 \\ \ddots & \ddots & \ddots & \ldots & \ddots \\ 0 & 0 & 0 & \cdots & I_{O_n} \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ldots & \ddots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \\ \vdots \\ z_n(t) \end{bmatrix} + \begin{bmatrix} w_1(t) \\ w_2(t) \\ w_3(t) \\ w_4(t) \\ w_5(t) \\ w_6(t) \\ \vdots \\ w_{2n}(t) \end{bmatrix}
$$

$$
\begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \\ \zeta_3(t) \\ \vdots \\ \zeta_n(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0\cdots & 0 & 0 & 0 & 0\cdots & 0 \\ 0 & 1 & 0\cdots & 0 & 0 & 0 & 0\cdots & 0 \\ 0 & 0 & 1\cdots & 0 & 0 & 0 & 0\cdots & 0 \\ \ddots & \ddots & \ddots\ldots & \ddots & \ddots & \ddots & \ddots\ldots & \ddots \\ 0 & 0 & 0\cdots & 1 & 0 & 0 & 0\cdots & 0 \end{bmatrix} \begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \\ \zeta_3(t) \\ \vdots \\ \gamma_{I_1}(t) \\ \gamma_{I_2}(t) \\ \gamma_{I_3}(t) \\ \vdots \\ \gamma_{I_n}(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ \vdots \\ v_n(t) \end{bmatrix} \tag{8.2.1}
$$

Let $\mathbf{A}_c$ be the state matrix, $\mathbf{B}_c$ be the input matrix, $\mathbf{C}_c$ be the output matrix, $\mathbf{x}_c$ be the state vector, $\mathbf{y}_c$ be the output vector and $\mathbf{u}$ be the input vector for the prediction model, where for a 3-arm junction $\mathbf{x}_c(t)=[\zeta_1(t)\zeta_2(t)\zeta_3(t)\gamma_{I_1}(t)\gamma_{I_2}(t)\gamma_{I_3}(t)]$, $\mathbf{y}_c(t)=[\zeta_1(t)\zeta_2(t)\zeta_3(t)]$ and $\mathbf{u}(t)=[z_1(t)z_2(t)z_3(t)]$. Equation (8.2.1) gives the prediction model extended to an n-arm junction.

Following the notation of [238], the stacked vectors of future states $\mathbf{X}$, inputs

U, and outputs **Y** are formulated as:

$$\begin{aligned}\mathbf{X} &= [\mathbf{x}_c{}^T(t+1) \quad \mathbf{x}_c{}^T(t+2) \quad \cdots \quad \mathbf{x}_c{}^T(t+N)]^T \\ \mathbf{U} &= [\mathbf{u}^T(t) \quad \mathbf{u}^T(t+1) \quad \cdots \quad \mathbf{u}^T(t+N-1)]^T \\ \mathbf{Y} &= [\mathbf{y}_c{}^T(t+1) \quad \mathbf{y}_c{}^T(t+2) \quad \cdots \quad \mathbf{y}_c{}^T(t+N)]^T\end{aligned} \tag{8.2.2}$$

where N gives the control prediction horizon.

To tune the model to reflect to changing traffic conditions in real-time, online joint state and parameter estimation is carried out as proposed in Chapter 6.3. The predicted states **X** and the predicted outputs **Y** can be calculated as follows:

$$\mathbf{X} = \mathbf{A}_x\mathbf{x}_c(t) + \mathbf{B}_x\mathbf{U} \tag{8.2.3}$$

$$\mathbf{Y} = \mathbf{A}_y\mathbf{x}_c(t) + \mathbf{B}_y\mathbf{U} \tag{8.2.4}$$

depending on the future inputs **U** and the state $\mathbf{x}_c(t)$. The future input vector **U** is the optimization variable which should be determined by an optimization algorithm to minimize a specified cost function.

The prediction matrices $\mathbf{A}_x$ and $\mathbf{B}_x$ are defined as follows, where $\mathbf{A}_c^2$ represent $\mathbf{A}_c$ squared. Similarly for the other exponents.

$$\begin{aligned}\mathbf{A}_x &= \begin{bmatrix} \mathbf{A}_c & \mathbf{A}_c{}^2 & \cdots & \mathbf{A}_c{}^N \end{bmatrix}^T \\ \mathbf{B}_x &= \begin{bmatrix} \mathbf{B}_c & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{A}_c\mathbf{B}_c & \mathbf{B}_c & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A}_c{}^{N-1}\mathbf{B}_c & \mathbf{A}_c{}^{N-2}\mathbf{B}_c & \cdots & \cdots & \mathbf{B}_c \end{bmatrix}\end{aligned} \tag{8.2.5}$$

The prediction matrices for the output $\mathbf{A}_y$ and $\mathbf{B}_y$ are similarly defined, whereby $\mathbf{A}_c{}^{\upsilon}$ is replaced by $\mathbf{C}_c\mathbf{A}_c{}^{\upsilon}\ \forall\ \upsilon \in 0,1,...,N$.

8.2.2 Optimization Problem

For one arm, the optimization problem is required to minimize the queue length of this arm in time. This problem can be formulated to minimize either one of the following norms [368]:

- l_1 Norm - This is given by the sum of the magnitudes of the queue length for the arm in time, as follows:

$$J_1 = \sum_{t}^{N} |\zeta(t)| \tag{8.2.6}$$

  This is also known as the Manhattan Distance.

- l_2 Norm - This is given by the sum of the square of the magnitudes of the queue length for the arm in time, as follows:

$$J_2 = \sqrt{\sum_{t}^{N} |\zeta(t)|^2} \tag{8.2.7}$$

  This is also known as the Euclidean norm.

- l_∞ Norm - This is given by the largest magnitude of the queue length for the arm in time, as follows:

$$J_\infty = max_t\{|\zeta(t)|\} \tag{8.2.8}$$

The optimization problem can be extended to minimize the queue lengths for all or some arms of the intersection. In this case, the optimization problem is formulated to minimize mixed norms in space and time [238, 369], since the queue lengths in different arms are required to be minimised (space), while also considering their varying behaviour in time. This minimization depends on weighting matrices which are used to prioritize some queue lengths in space and time. Nine

possible combinations (three in space and three in time) are formulated and the one which gives the best performance based on the largest reduction in queue length when compared to those resulting from fixed time plans, will be determined.

The nine possible combinations of mixed norms are:

- $l_{1,1}$ Norm - 1-norm with respect to time and 1-norm with respect to space
- $l_{2,2}$ Norm - 2-norm with respect to time and 2-norm with respect to space
- $l_{\infty,\infty}$ Norm - ∞-norm with respect to time and ∞-norm with respect to space
- $l_{1,2}$ Norm - 1-norm with respect to time and 2-norm with respect to space
- $l_{1,\infty}$ Norm - 1-norm with respect to time and ∞-norm with respect to space
- $l_{2,1}$ Norm - 2-norm with respect to time and 1-norm with respect to space
- $l_{2,\infty}$ Norm - 2-norm with respect to time and ∞-norm with respect to space
- $l_{\infty,1}$ Norm - ∞-norm with respect to time and 1-norm with respect to space
- $l_{\infty,2}$ Norm - ∞-norm with respect to time and 2-norm with respect to space

The next sections will discuss: i) the $l_{1,1}$ Norm; ii) $l_{2,2}$ Norm and iii) $l_{\infty,\infty}$ Norm. The formulation for $l_{1,2}$ Norm, $l_{1,\infty}$ Norm, $l_{2,1}$ Norm, $l_{2,\infty}$ Norm, $l_{\infty,1}$ Norm and $l_{\infty,2}$ Norm are discussed in the Appendix.

8.2.2.1 $l_{1,1}$ Norm - 1-norm with respect to time and 1-norm with respect to space

The cost function based on the $l_{1,1}$ norm is given by:

$$J_{1,1} = \sum_{t}^{N} \sum_{i}^{n} |y_{ci}(t)| = ||\mathbf{Y}||_{1,1} \tag{8.2.9}$$

By Equation (8.2.3) this implies that:

$$J_{1,1} = ||\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t) + \mathbf{R}_c\mathbf{B}_y\mathbf{u}(t)||_{1,1} \tag{8.2.10}$$

where the states are weighted with the diagonal matrix $\mathbf{Q}_c$ and the input is weighted with the diagonal matrix $\mathbf{R}_c$.

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\|_{1,1} \tag{8.2.11}$$

where $\mathbf{F}_1 = \mathbf{R}_c\mathbf{B}_y$ and $\mathbf{g}_1 = \mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t)$. Note that $J_{1,1}$ is not smooth but is a composite of smooth functions [238,368,370]. This function can be transformed into a smooth one by introducing a new variable $\boldsymbol{\Upsilon}$ [370] such that:

$$\min_{\mathbf{U},\boldsymbol{\Upsilon}} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} \tag{8.2.12}$$

$$0 \leq \mathbf{F}_1\mathbf{U} + \mathbf{g}_1 \leq \boldsymbol{\Upsilon} \tag{8.2.13}$$

and $\boldsymbol{\Upsilon} = [\Upsilon_1(t) \quad \Upsilon_2(t) \quad \ldots \quad \Upsilon_n(t) \quad \Upsilon_1(t+1) \quad \ldots \quad \Upsilon_n(t+N)]^T$. Queue lengths given by vector $\mathbf{Y}$ must be greater or equal to zero and must be less than some maximum value, $\boldsymbol{\Upsilon}$.

This minimization problem can be cast as a linear program with linear constraints given by Equation (8.2.13) that need to be satisfied. As mentioned before, queue lengths must be greater or equal to zero and must be less than $\boldsymbol{\Upsilon}$. This maximum value is set to be the maximum number of vehicles that can fit along each arm, without spilling over to the next junction, thus avoiding block-back scenarios. The minimum value of $\boldsymbol{\Upsilon}$ is set to 0. Hence $\boldsymbol{\Upsilon}$ is upper and lower bounded to be positive. Moreover, as recommended by the service operator the minimum green time required to allocate enough time for the traffic light to operate and for a few vehicles to exit the link, set to 0.1. In addition, the green time

for all arms should add up to 0.93, whilst providing a safe transition between conflicting phases. Hence, the maximum green time for an arm should be set to $0.93-((n-1)\times 0.1)$ provided that all other arms are assigned a minimum green time. Such constraints can be expressed in matrix form as follows:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 & \mathbf{0} \\ \mathbf{R}_c\mathbf{B}_y & -\mathbf{I} \\ -\mathbf{R}_c\mathbf{B}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_2 \end{bmatrix} \tag{8.2.14}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ -\mathbf{g}_1 \\ \mathbf{g}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{8.2.15}$$

$$\mathbf{A}_{constraint} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} \leq \mathbf{b}_{constraint} \tag{8.2.16}$$

The maximum and minimum constraints on $\boldsymbol{\Upsilon}$ are defined by the 4^{th} row in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$ in Equations (8.2.14) and (8.2.15) respectively. For example, for a 3-arm junction, with a prediction horizon of 1 cycle, $\boldsymbol{\Upsilon}=[\Upsilon_1(t) \quad \Upsilon_2(t) \quad \Upsilon_3(t) \quad \Upsilon_1(t+1) \quad \Upsilon_2(t+1) \quad \Upsilon_3(t+1)]^T$. The maximum bound on $\boldsymbol{\Upsilon}$ is set to 60 vehicles for each arm. $\mathbf{\Omega}_2$ is given by:

$$\Omega_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.2.17}$$

and $\mathbf{c}_2$ is equal to $[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 60 \quad 60 \quad 60 \quad 60 \quad 60 \quad 60]^T$. The maximum and minimum constraints on $\mathbf{U}$ are defined by the 1^{st} row in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$. For example for a 3-arm junction, with a prediction horizon of 1 cycle, Ω_1 is given by:

$$\Omega_1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.2.18}$$

and $\mathbf{c}_1$ is equal to $[0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.73 \quad 0.73 \quad 0.73 \quad 0.73 \quad 0.73 \quad 0.73]^T$. The constraint in Equation (8.2.13) is defined by the 3^{rd} and 4^{th} rows in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$ in Equations (8.2.14) and (8.2.15).

Moreover, the constraint that the green time addition for all arms should add up to 0.93 is defined by $\mathbf{A}_{eq}$ and $\mathbf{b}_{eq}$ given by Equation (8.2.19). For example for a 3-arm junction, with a prediction horizon of 1 cycle, $\mathbf{A}_{eq}$ is equal to $\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ and $\mathbf{b}_{eq}$ is equal to $\begin{pmatrix} 0.93 \\ 0.93 \end{pmatrix}$.

$$\mathbf{A}_{eq} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \mathbf{b}_{eq} \tag{8.2.19}$$

8.2.2.2 $l_{2,2}$ Norm - 2-norm with respect to time and 2-norm with respect to space

The cost function based on the $l_{2,2}$ norm is given by:

$$J_{2,2} = \sum_{t}^{N} \sum_{i}^{n} |y_{ci}(t)|^2 = ||\mathbf{Y}||_{2,2}^2 \tag{8.2.20}$$

$$= ||\mathbf{Q}_c^{\frac{1}{2}} \mathbf{A}_y \mathbf{x}_c(t) + \mathbf{R}_c^{\frac{1}{2}} \mathbf{B}_y \mathbf{u}(t)||_{2,2}^2 \tag{8.2.21}$$

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_2 \mathbf{U} + \mathbf{g}_2\|_{2,2}^2 \tag{8.2.22}$$

where $\mathbf{F}_2 = \mathbf{R}_c^{\frac{1}{2}} \mathbf{B}_y$ and $\mathbf{g}_2 = \mathbf{Q}_c^{\frac{1}{2}} \mathbf{A}_y \mathbf{x}_c(t)$. This minimization problem is cast as a quadratic program with linear constraints [238]:

$$\min_{\mathbf{U}} \frac{1}{2} \mathbf{U}^T [2\mathbf{F}_2^T \mathbf{F}_2] \mathbf{U} + [2\mathbf{F}_2^T \mathbf{g}_2]^T \mathbf{U} \tag{8.2.23}$$

such that the queue lengths are upper and lower bounded, where:

$$\mathbf{0} \leq \mathbf{Y} \leq \mathbf{c}_2 \tag{8.2.24}$$

For example for a 3-arm junction, with a prediction horizon of 1 cycle, and a maximum bound on the queue length set to 60 vehicles for each arm gives $\mathbf{c}_2 = [60 \quad 60 \quad 60 \quad 60 \quad 60 \quad 60]^T$.

Hence,

$$-\mathbf{g}_2 \leq \mathbf{F}_2\mathbf{U} \leq \mathbf{c}_2 - \mathbf{g}_2 \tag{8.2.25}$$

The constraints for the maximum and minimum bounds on the green time as expressed in Section 8.2.2.1 still apply. Such constraints can be expressed in matrix form, given by Equations (8.2.26) to (8.2.28). The maximum and minimum constraints on $\mathbf{U}$ are defined by the 1^{st} row in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$ as similarly expressed in Section 8.2.2.1. The constraint in Equation (8.2.25) is defined by the 2^{nd} and 3^{rd} row in Equations (8.2.26) to (8.2.28).

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{F}_2 \\ -\mathbf{F}_2 \end{bmatrix} \tag{8.2.26}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 - \mathbf{g}_2 \\ -\mathbf{g}_2 \end{bmatrix} \tag{8.2.27}$$

$$\mathbf{A}_{constraint}\mathbf{U} \leq \mathbf{b}_{constraint} \tag{8.2.28}$$

Moreover, the constraint that the green time addition for all arms should add up to 0.93 is defined by $\mathbf{A}_{eq}$ and $\mathbf{b}_{eq}$ in Equation (8.2.29). For example for a 3-arm junction, with a prediction horizon of 1 cycle, $\mathbf{A}_{eq}$ is equal to $\left(\begin{smallmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{smallmatrix}\right)$ and $\mathbf{b}_{eq}$ is equal to $\left(\begin{smallmatrix} 0.93 \\ 0.93 \end{smallmatrix}\right)$.

$$\mathbf{A}_{eq}\mathbf{U} = \mathbf{b}_{eq} \tag{8.2.29}$$

8.2.2.3 $l_{\infty,\infty}$ Norm - ∞-norm with respect to time and ∞-norm with respect to space

The cost function based on the $l_{\infty,\infty}$ norm is given by:

$$J_{\infty,\infty} = max_t\{max_i|y_{c_i}(t)|\} = ||\mathbf{Y}||_{\infty,\infty} \tag{8.2.30}$$
$$= ||\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t) + \mathbf{R}_c\mathbf{B}_y\mathbf{u}(t)||_{\infty,\infty} \tag{8.2.31}$$

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\|_{\infty,\infty} \tag{8.2.32}$$

where $\mathbf{F}_1$=$\mathbf{R}_c\mathbf{B}_y$ and $\mathbf{g}_1$=$\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t)$. Note that $J_{\infty,\infty}$ is not smooth [238,368,370]. This function can be transformed into a smooth one by introducing a new variable Υ [370], such that:

$$0 \leq max_t\{max_i(\mathbf{F}_1\mathbf{U} + \mathbf{g}_1)\} \leq \Upsilon \tag{8.2.33}$$

This minimization problem can be cast as a linear program with linear constraints:

$$\min_{\mathbf{U},\Upsilon} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} \tag{8.2.34}$$

The same constraints for the queue length and green time as expressed in the previous sections still apply. Such constraints can be expressed in matrix form, where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \Omega_1 & \mathbf{0} \\ \mathbf{R}_c\mathbf{B}_y & -I \\ -\mathbf{R}_c\mathbf{B}_y & 0 \\ \mathbf{0} & \Omega_2 \end{bmatrix} \tag{8.2.35}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ -\mathbf{g}_1 \\ \mathbf{g}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{8.2.36}$$

$$\mathbf{A}_{constraint} \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} \leq \mathbf{b}_{constraint} \tag{8.2.37}$$

The maximum and minimum constraint on Υ are defined by the 4^{th} row in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$ in Equations (8.2.35) and (8.2.36) respectively. For example, for a 3-arm junction, with a prediction horizon of 1 cycle, $\mathbf{\Omega}_2$ is equal to $\left(\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right)$ and $\mathbf{c}_2$ is equal to $[0 \quad 60]^T$. The maximum bound on Υ is set to 60 vehicles. The maximum and minimum constraints on $\mathbf{U}$ are defined by the 1^{st} row in $\mathbf{A}_{constraint}$ and $\mathbf{b}_{constraint}$, with $\mathbf{\Omega}_1$ and $\mathbf{c}_1$ similarly defined in Section 8.2.2.1.

Moreover, the constraint that the green time addition for all arms should add up to 0.93 is defined by $\mathbf{A}_{eq}$ and $\mathbf{b}_{eq}$ in Equation (8.2.38). For example for a 3-arm junction, with a prediction horizon of 1 cycle, $\mathbf{A}_{eq}$ is equal to $\left(\begin{smallmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{smallmatrix}\right)$ and $\mathbf{b}_{eq}$ is equal to $\left(\begin{smallmatrix} 0.93 \\ 0.93 \end{smallmatrix}\right)$.

$$\mathbf{A}_{eq} \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} = \mathbf{b}_{eq} \tag{8.2.38}$$

The formulation for $l_{1,2}$ Norm, $l_{1,\infty}$ Norm, $l_{2,1}$ Norm, $l_{2,\infty}$ Norm, $l_{\infty,1}$ Norm and $l_{\infty,2}$ Norm are similarly described in the Appendix.

8.3 Model Predictive Control for signalized junctions

One of the cost functions discussed in the previous section is chosen so as to solve for the green times which minimize the queue lengths in the junction. Online optimization is obtained by applying linear programming or quadratic programming. The most successful and widely used linear programming algorithms are the simplex method and the interior point methods. These two methods use a different searching procedure to find the optimal points within the feasible region. The simplex method is computationally more efficient when dealing with non-complex systems [371]. Hence, in this work the simplex method will be applied. The *linprog* function in Matlab is used to solve the linear programming algorithm [372]. The most successful quadratic programming algorithms are the interior-point-convex and the trust-region-reflective algorithm. The trust-region-reflective algorithm is used when the concave or convex problem has inequality constraints or only linear equalities, but not both. Since in this work, the formulation has both equality and inequality constraints, the interior-point-convex algorithm will be applied. The *quadprog* function in Matlab is used to solve the quadratic programming algorithm [372].

Model predictive control repeatedly applies optimal control in a rolling horizon way [5, 210–214, 227]. Hence, the optimal control input, $\mathbf{u}(t)$, is obtained from the online optimization discussed before. This control input is implemented as indicated in Table 8.1. At the next cycle, the whole time horizon is shifted one step forward, and the optimization over the new prediction horizon starts over again, based on the prediction model that is fed and the estimated model states and parameters as new information from the sensors become available.

The estimation and control algorithm is presented in Table (8.1). As the input intensities and the queue lengths start increasing, a feasible LP or QP solution that satisfies all constraints might not exist. This is based on the design considerations made apriori by the service operator. Green times larger than 0.73 73 for a 3-arm and 0.63 for a 4-arm junction, as defined by the service operator, cannot

be assigned in order to satisfy the green time constraints as discussed in Section 8.2.2.1. To mitigate this problem, some constraints are progressively relaxed so as to find a solution [373]. This is done by sequentially relaxing the limit on the queue lengths for one link at a time. The queue length constraint of the link with the least importance for that time of day is relaxed first, by increasing its upper bound Υ.

Table 8.1: MPC for signalized junctions

Initialise estimates for $\hat{\mathbf{\Theta}}$, $\hat{\mathbf{Q}}$, $\hat{\mathbf{R}}$ and $-2E\{G\}$
Iterate for every time step t.
Step 1:
Measure $\mathbf{y}(t)$.
E-step
Run Kalman-filter recursions to compute $\hat{\mathbf{x}}_t$.
M-step
Minimise -2E$\{G(\mathbf{\Theta},\hat{\mathbf{\Theta}}_t)\}$ over $\mathbf{\Theta}$ for dynamic traffic conditions while satisfying Equations (4.2.12) and (4.2.13) (with projection if necessary).
Minimise -2E$\{G(\mathbf{\Theta},\hat{\mathbf{\Theta}}_t)\}$ over $\hat{\mathbf{Q}}$ and similarly for $\hat{\mathbf{R}}$.
Control step
Based on the prediction model given by Equation (8.2.1), minimise the control objective function J for the selected mixed norm in Section 2.2 subject to the linear constraints (for example Equations (8.2.14) to (8.2.19) for the $l_{1,1}$ norm and similarly for the other norms) and solve for $\mathbf{U}$.
Apply $\mathbf{u}(t)$ as the control input.
Increment t.
Shift the whole time horizon one step forward.
Repeat from Step 1.

8.4 Simulation Experiments

The proposed control algorithm with online joint estimation of states, model parameters and process and measurement noise covariances, was tested on signalized 3-arm and 4-arm junctions, with geometry similar to Figures 4.3 and 4.2 respectively and with traffic data for these junctions as presented in Section

6.3.2. To implement adaptive traffic control with the standard Aimsun software, the Aimsun API module [374] was used to enable communication between the Aimsun simulation model and the MPC Algorithm of Table 8.1 running under MATLAB. Hence, the traffic data is collected from the traffic simulation through the Aimsun API module at every time step, and the controller calculates online the required model predictions and control actions based on the collected information. The control action is then applied to the simulation and such process completes the communication between the simulation in Aimsun and the MPC algorithm on MATLAB.

In these experiments, the diagonal elements of $\mathbf{Q}_c$ and $\mathbf{R}_c$ are set to 1, thus equally weighting the states and the control input for each arm. Moreover, the upper bound of the queue length for each link, given by Υ, is defined for the 4-arm and 3-arm junction as shown in Tables 8.2 and 8.3 respectively. If no solution is found, the Υ bound is relaxed to values shown in the third column for each arm, with the last column indicating the order in which such relaxation is to occur. 1 indicates the first upperbound to be relaxed and similarly for the rest. If a solution is still not found, then the upper bounds on the queue lengths are removed for all arms.

Table 8.2: The upper bound of the queue length for each link for the 4-arm junction

Arm	Υ [vehicles]	Υ if no solution [vehicles]	Order
1	40	60	3 (morning), 4 (evening)
2	40	60	2
3	40	60	4 (morning), 3 (evening)
4	40	60	1

Table 8.3: The upper bound of the queue length for each link for the 3-arm junction

Arm	Υ [vehicles]	Υ if no solution [vehicles]	Order
1	40	60	2 (morning), 3 (evening)
2	40	60	1
3	40	60	3 (morning), 2 (evening)

8.5 Results

The left hand plots of Figure 8.1 show the queue lengths occurring from the MPC for all 4 arms, for a typical working day from 6.00am of one day to 6.00am of the next day, with $l_{2,2}$ norm being implemented to solve the optimization problem and with a prediction horizon of 1 cycle. During saturated traffic conditions, the MPC calculates the required control action, while during unsaturated traffic conditions, the fixed time plans are implemented. The switching transition threshold defined in Chapter 5 still applies, where an arm is detected to be saturated if the condition $(\zeta_n(k) + \gamma_{I_n}(k) \cdot z_n(k)) \geq (S_n z_n(k))$ is satisfied. This method is adopted because the large queues during saturated traffic conditions are critical and hence need to be minimised. During unsaturated traffic conditions, since the number of vehicles in the arms are very low, the control action is characterised by the fixed green time, similar to the work in [213].

To quantify the effectiveness of the MPC, comparisons are drawn when the fixed time plans are constantly applied to the junction. The queue lengths and green times generated from the MPC are denoted by the plot having the legend as *variable* in Figure 8.1, while the queue lengths and green times generated from the fixed time plans are denoted by the plot with legend *fixed.* As shown in Figure 8.1 the MPC controller aims to minimize the queue length depending on the order defined in Table 8.2. In the morning, priority is given to arm 3, whereby Υ for this arm is last to be relaxed if no solution is found, compared to all other 3 arms. The right hand plots of Figure 8.1 show the green times. Note that MPC

assigns longer green times to this arm to keep the queue length bounded within Υ, while minimum green times are assigned to arms 2 and 4 respectively. Similarly in the evening, whereby Υ for arm 1 is last to be relaxed if no solution is found, compared to all other three arms. Hence significant green time is assigned to this arm while minimum green time is assigned to arm 4. Some green times are assigned to arms 2 and 3 respectively, so as to keep the queue lengths within the upper bound for each arm.

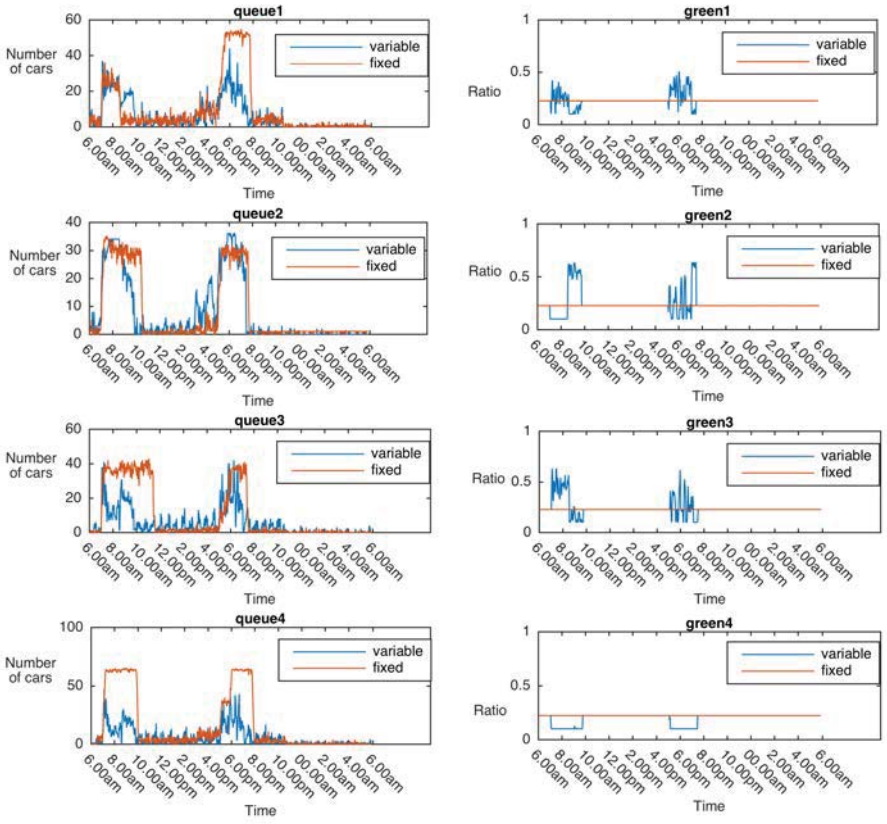

Figure 8.1: Queue lengths and green times for the 4-arm junction with the $l_{2,2}$ norm

Similarly for the 3-arm junction. Figures 8.2 and 8.3 show the queue length for all 3 arms, for a typical working day from 6.00am of one day to 6.00am of the next day, with the $l_{1,1}$ and the $l_{2,2}$ norm being implemented to solve the optimization problem. In the morning, the MPC assigns large green times to arm

3 and minimum green times to arm 2 during many cycles, at the cost of increasing queue lengths in arms 1 and 2. On the other hand, in the evening, Υ for arm 1 is last to be relaxed if no solution is found. Hence significant green time is assigned to such arm at the expense of slightly increasing the queue lengths in arms 2 and 3 respectively.

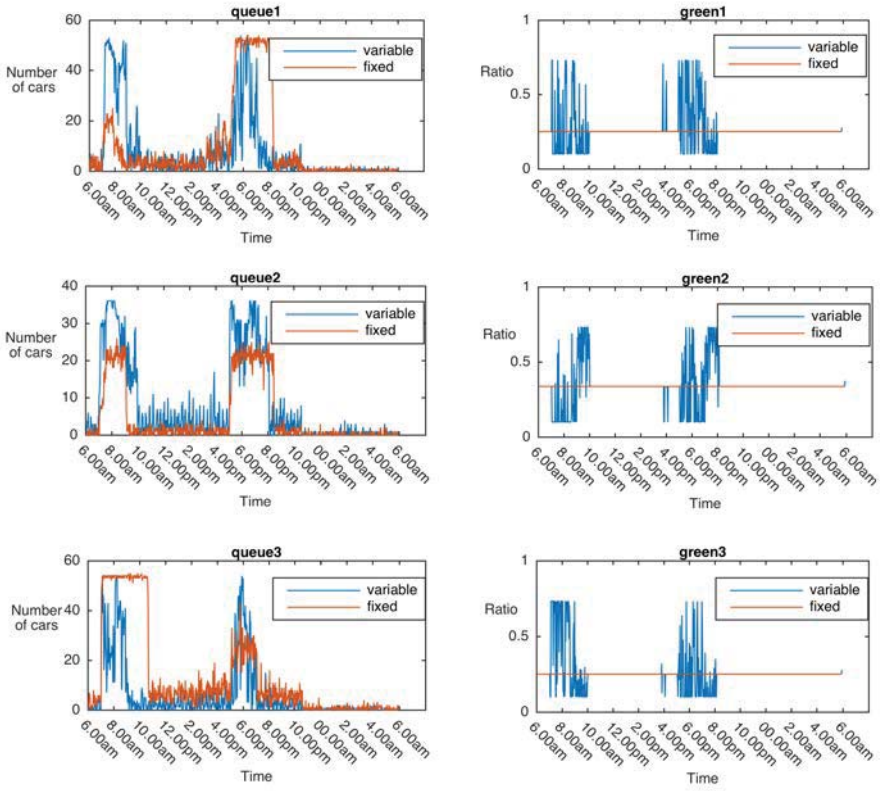

Figure 8.2: Queue lengths and green times for the 3-arm junction with $l_{1,1}$ norm

Figure 8.2 shows the aggressiveness of the $l_{1,1}$ controller action compared to the $l_{2,2}$ control action in Figure 8.3. This could be attributed to the non-smoothness of the $l_{1,1}$ norm cost function compared to the $l_{2,2}$ norm cost function. Table 8.4 shows the % difference in total queue lengths for each arm, for different times of the day, with the $l_{1,1}$ norm being implemented to solve the optimization problem, compared to the total queue lengths obtained with fixed

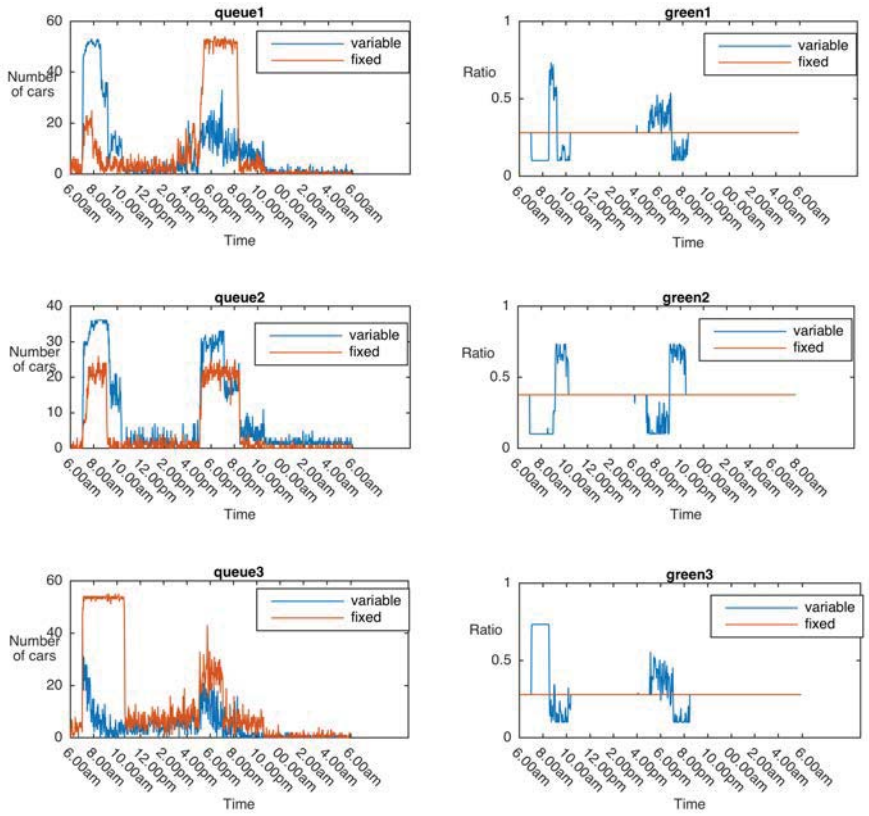

Figure 8.3: Queue lengths and green times for the 3-arm junction with $l_{2,2}$ norm

green times. The % difference in queue length in arm i is calculated as follows: $\frac{(queue\_length_{variable_i} - queue\_length_{fixed_i})}{queue\_length_{variable_i}} \times 100$, where $queue\_length_{variable_i}$ is the queue length in arm i resulting from green times generated by the MPC and $queue\_length_{fixed_i}$ is the queue length in arm i resulting from fixed time plans. During the morning, the total queue length in arm 3 is reduced significantly, -66.8%, as seen in the table, compared to the total queue lengths in arms 1 and 2 respectively, with the constraint Υ for arm 3 last to be relaxed if no solution is found. On the contrary, during the evening, the total queue length in arm 1 is reduced significantly, -58.6% as seen in the table, compared to the total queue lengths in arms 2 and 3 respectively, with the constraint Υ for arm 1 last to be relaxed if no solution is found. This behaviour is similarly observed when ap-

plying the other mixed norms to solve the optimization problem.

Table 8.4: % difference in queue lengths for the 3-arm junction with $l_{1,1}$ norm compared with fixed timing

Description	% Difference in queue length in arm$_i$
% difference morning arm1	67.5
% difference morning arm2	49.8
% difference morning arm3	-66.8
% difference evening arm1	-58.6
% difference evening arm2	19.8
% difference evening arm3	-13.8
% difference over 24 hours arm1	-19.0
% difference over 24 hours arm2	60.5
% difference over 24 hours arm3	-55.1

Table 8.5 shows the performance of the different mixed norms when applied to solve the optimization problem for a 3-arm junction, subject to the same traffic data. The resulting total queue lengths and travel time over the whole day for the MPC are compared to those resulting from fixed time plans. The % difference in total queue lengths and the % difference in travel time between both implementations over the whole day are worked out as given in Table 8.5. The % difference in total queue lengths is worked out as follows:

$$\frac{(\sum_i queue\_length_{variable_i} - \sum_i queue\_length_{fixed_i})}{\sum_i queue\_length_{variable_i}} \times 100$$

. A reduction in queue length was observed for all mixed norms ranging between 12.8% to 21.7% and a reduction in travel time ranging between 9.5% and 44.2%. For the 4-arm junction only 3 norms were tested: $l_{1,1}$, $l_{2,2}$ and $l_{\infty,\infty}$. The reduction in queue length for the mixed norms ranged between 21.2% to 40.3% and the reduction in travel time ranged between 23.7% to 27.3%.

To test the consistency of the performance obtained by the $l_{1,1}$, $l_{2,2}$ and $l_{\infty,\infty}$ MPC controller, a Monte Carlo run of 100 realisations was performed on the 3 and 4-

Table 8.5: Different norm results for the 3-arm junction

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-21.7	-32.0
$l_{2,2}$ norm	-20.3	-44.2
$l_{\infty,\infty}$ norm	-14.3	-27.4
$l_{1,2}$ norm	-20.3	-44.2
$l_{2,1}$ norm	-12.9	-9.5
$l_{1,\infty}$ norm	-19.3	-27.1
$l_{\infty,1}$ norm	-13.4	-15.9
$l_{2,\infty}$ norm	-13.1	-19.5
$l_{\infty,2}$ norm	-12.8	-9.5
Mean:	-16.5	-25.5

Table 8.6: Different norm results for the 4-arm junction

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-27.3	-25.7
$l_{2,2}$ norm	-40.3	-27.3
$l_{\infty,\infty}$ norm	-21.2	-23.7
Mean:	-29.6	-25.6

arm junctions. Measurements of vehicles entering and leaving each arm were simulated in Aimsun for 100 different realisations, each time recording the performance of the MPC controller on the queue lengths. The resulting total queue lengths over the whole day for the MPC are compared to those resulting from fixed time plans. Figures 8.4 and 8.5 show box plots representing the % reduction in total queue length over the whole day, for the 3 and 4-arm junctions respectively. In both cases, the MPC controller with the $l_{2,2}$ norm produced the largest % reductions in queue lengths. A % mean reduction in queue length is worked out by averaging the % reductions over all 100 realisations for each norm. For the 3-arm junction, a mean reduction of 25.2% was observed, compared to a % mean reduction of 22.3% with the $l_{1,1}$ norm and a % mean reduction of 18.1% with the $l_{\infty,\infty}$ norm for a 3-arm junction. Similarly, for the 4-arm junction, a % mean reduction of 45.7% with the $l_{2,2}$ norm compared to a % mean reduction of 38.7% with the $l_{1,1}$ norm and a % mean reduction of 25.8% with the $l_{\infty,\infty}$ norm.

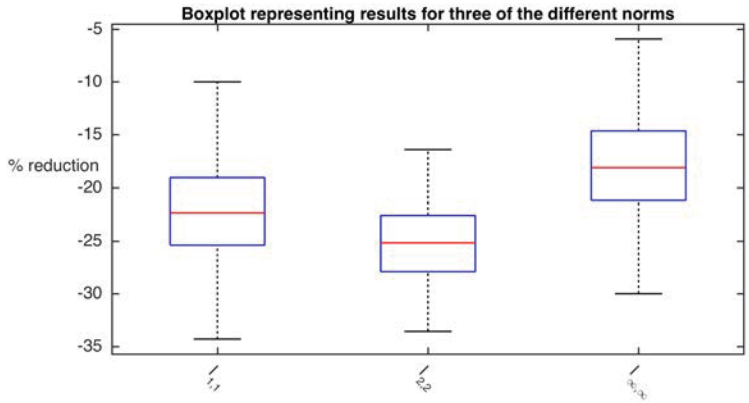

Figure 8.4: Box plot representing the results of 100 realisations using the MPC controller for the 3-arm junction

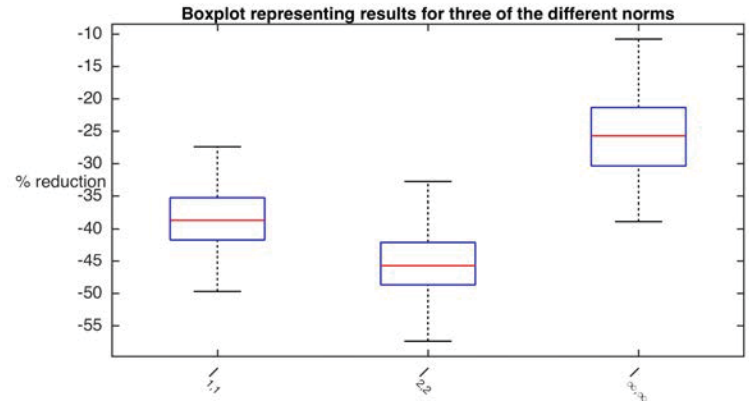

Figure 8.5: Box plot representing the results of 100 realisations using the MPC controller for the 4-arm junction

Furthermore, a one-way analysis of variance was used to determine whether there is any statistically significant differences between the means of three norms. There is a statistically significant difference between all 3 means with a p-value of 8.127×10^{-25} for the 3-arm junction and a p-value of 2.543×10^{-78} for the 4-arm junction at 95% confidence interval. Moreover, a paired sample t-test was carried out to determine whether the mean difference between pairwise combinations of all norms that were considered is zero. Table 8.7 and 8.8 show the results for the 3 paired sets of norms, where in all cases a statistically significant difference between the two respective means of the norms was observed at 95% confidence interval. Hence, these results show the statistical significant differ-

ence between the means of each pairwise combination with a p-value ranging between 2.564×10^{-21} and 4.903×10^{-6}.

Table 8.7: Paired sample t-test results for the 3-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	4.903×10^{-6}
$l_{2,2}$ and $l_{\infty,\infty}$	2.564×10^{-21}
$l_{1,1}$ and $l_{\infty,\infty}$	1.299×10^{-8}

Table 8.8: Paired sample t-test results for the 4-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	7.302×10^{-17}
$l_{2,2}$ and $l_{\infty,\infty}$	4.574×10^{-44}
$l_{1,1}$ and $l_{\infty,\infty}$	1.105×10^{-30}

The $l_{2,2}$ mixed norm minimizes the sum of the square of the queue lengths, while the $l_{1,1}$ mixed norm minimizes the sum of the queues. Hence, the $l_{2,2}$ mixed norm is more sensitive to large queue lengths than the $l_{1,1}$ mixed norm. On the other hand, in $l_{\infty,\infty}$ mixed norm resulted in inferior performance, because only the largest queue from all arms over the prediction horizon is minimized.

8.6 Conclusion

The results in this Chapter highlight the effectiveness of using MPC to control the traffic light timings. Different norms were tested to solve the optimization problem. A Monte Carlo run of 100 realisations was performed and it was found out that the $l_{2,2}$ mixed norm produced the largest % reductions in queue lengths when compared with fixed time plans. As detailed in the results section, a % mean reduction in queue length equal to 25.2% was observed for the 3-arm junction, while for the 4-arm junction, a % mean reduction of 45.7% was noted.

Chapter 9

Model Predictive Control with *Jump* Dynamics

9.1 Introduction

Chapter 8 presented a MPC algorithm for traffic flows operating under normal conditions. However, as discussed in Chapter 7, traffic conditions in signalized junctions may be subject to different modes of operation because of *jump* dynamics arising from say, traffic incidents or unanticipated network obstructions. Hence, under such circumstances, the dynamics of the traffic system may be represented by a set of different models that might not be identified *a priori*, with each model representing specific dynamic mode. In this Chapter, the two methodologies of Chapters 7 and 8 are combined, to yield a novel methodology for autonomic control of traffic junctions. The essence of MPC with multiple models is to have a set of models and controllers, each of which is associated with a given mode, so as to minimise optimally the queue length in the junction when that mode is detected to be active. When the dynamics *jump* to a different mode, this is detected and the controller is switched accordingly. The traffic models and the online joint estimation techniques discussed in Part I are required to be able to describe and predict the traffic flow dynamics in the future, as well as tune the controller parameters, once a mode becomes active. In addition to learning the various mode dynamics and detecting mode switches, the system will automatically configure itself and grow a new model within the set to learn the new traffic dynamics if novel conditions arise.

9.2 Model Predictive Control with Multiple Model Estimation for signalized junctions

Similar to the work presented in Chapter 7, suppose that H distinct modes of operation are postulated initially, each representing one regime (or mode) of operational conditions. Only one particular mode from these H could be active at time t, which is denoted as mode f, where $f \in [1,..,H]$. The dynamics of each mode f can be represented by model M^f with unknown state space matrices $\mathbf{A}^f$, $\mathbf{B}^f$, $\mathbf{C}^f$ and $\mathbf{D}^f$, noise covariances $\mathbf{Q}^f$ and $\mathbf{R}^f$ and model parameters $\mathbf{\Theta}^f$ which are online estimated as described in Chapter 7. In addition to the work presented in Chapter 7, now, each mode f has an associated controller with state space matrices $\mathbf{A}_c^f$, $\mathbf{B}_c^f$ and $\mathbf{C}_c^f$ and online estimated parameters as defined by Equation (8.2.1) in Chapter 8.

Similar to Chapter 7, individual state estimates $\hat{\mathbf{x}}_t^f$ are obtained by running a Kalman filter for each candidate model. The lower bounding approach together with Bayes' rule Equation (7.2.1) are applied to detect which mode is active at a given time instant. Hence, model M^f and its associated controller become active at time t.

In addition, the control process is similar to that described in Chapter 8. The online optimization problem, described by one of the mixed norms presented in Section 8.2.2 is solved at each time step and the optimal control input, $\mathbf{u}(t)$, is obtained. This control input is implemented.

The estimation and control algorithm is presented in Table 9.1. Similar to Chapter 8, the *linprog* function is used to solve the LP, while the *quadprog* function in Matlab [372] is used to solve the QP. The sequential relaxing of the upper limit of the queue lengths is still applied for one link at a time, when a feasible solution is not found.

Table 9.1: MPC controllers with multiple model estimation and control for signalized junctions

Initialise $\hat{\boldsymbol{\Theta}}^f$, $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{R}}^f$ estimates for f local models and $-2E\{G\}$
Iterate for every time step t.
Step 1:
Measure $\mathbf{y}(t)$.
E-step
Run Kalman-filter recursions for each local candidate models f to compute $\hat{\mathbf{x}}_t^f$, where $f = 1...H$.
M-step
For such candidate model f, $(f = 1...H)$, minimise -2E$\{G(\boldsymbol{\Theta}^f, \hat{\boldsymbol{\Theta}}_t^f)\}$ over $\boldsymbol{\Theta}^f$ for dynamic traffic conditions while satisfying Equations (4.2.12) and (4.2.13) (with projection if necessary).
Minimise -2E$\{G(\boldsymbol{\Theta}^f, \hat{\boldsymbol{\Theta}}_t^f)\}$ over $\hat{\mathbf{Q}}^f$ and similarly for $\hat{\mathbf{R}}^f$.
Calculate the posterior probability distribution for such candidate model f given in Equation (7.2.1).
Just after the H^{th} model has been selected, as a result of a new mode being detected active, introduce a new model with randomly selected $\hat{\boldsymbol{\Theta}}^f$, $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{R}}^f$ and let $H \rightarrow H+1$.
Calculate $\hat{\mathbf{x}}_t$ using Equation (7.3.1) or by $\hat{\mathbf{x}}_{t_{max:Pr(M^f|\mathbf{Y}^t)}}$ representation.
For the current active model, that is the model which corresponds to the $max : Pr(M^f|\mathbf{Y}^t)$ update $\hat{\mathbf{A}}^f$, $\hat{\mathbf{B}}^f$, $\hat{\mathbf{C}}^f$, $\hat{\mathbf{D}}^f$, with $\hat{\boldsymbol{\Theta}}_t^f$ to reflect the traffic conditions per arm.
Control step
Based on the prediction model with state space matrices $\mathbf{A}_c^f$, $\mathbf{B}_c^f$ and $\mathbf{C}_c^f$, minimise the control objective function J for the selected mixed norm, subject to the linear constraints and solve for $\mathbf{U}$.
Shift the whole time horizon one step forward.
Increment t and repeat from Step 1.

9.3 Simulation Experiment

The proposed control algorithm with online joint estimation of states, model parameters and process and measurement noise covariances, was tested on signalized 3-arm and 4-arm junctions, with geometry similar to Figures 4.2 and 4.3. Arm closures for arm 1 and 2 and lane closure in arm 1 were simulated with timings similar to those presented in Section 7.4.

9.4 Results

Several tests were carried out as will be discussed in detail in the next sections.

9.4.1 Multiple Model Estimation with MPC

9.4.1.1 Arm closures

Figures 9.1 and 9.2 show the estimated switching conditions for the 3-arm and 4-arm junctions respectively with the arm closures for such junctions occurring at the same timings described in Section 7.4. Initially, the number of modes H is unknown and hence modes and their models are learnt as they occur in real-time. Only one local model is known initially and this denotes 'normal' traffic conditions. The subplots on the right in Figure 9.1 zoom in on the situation between 6.58am and 8.10am and subplots on the right in Figure 9.2 zoom in on the situation between 7.05am and 8.13am so as to clearly show the transition before and after arm 1 closure.

As shown in Figure 9.1 *spare* models are introduced at the 44^{th} cycle (at 7.16am) and 417^{th} cycles (at 6.01pm) respectively when closures of the arms are detected for the 3-arm junction. Before the 44^{th} cycle, the estimated probability for 'blockage - arm 1' is unknown, because this model is introduced at 7.16am. Similarly, before the 417^{th} cycle, the estimated probability for 'blockage - arm 2' is unknown, because this model is introduced at 6.01pm. The system switches back to normal conditions at the 65^{th} cycle (at 7.53am) and at the 435^{th} cycle (at 6.32pm) respectively. This results in one cycle delay in detecting arm 1 and arm 2 closures and another cycle delay in switching back to normal conditions following the respective arm closures. Similarly, for a 4-arm junction, in Figure 9.2, *spare* models are introduced at the 44^{th} cycle (at 7.20am) and 417^{th} cycles (at 6.44pm) respectively when closures of the arms are detected. Before the 44^{th} cycle, the estimated probability for 'blockage - arm 1' is unknown, because this model is introduced at 7.20am. Similarly, before the 417^{th} cycle, the estimated probability for 'blockage - arm 2' is unknown, because this model is introduced at 6.44pm. The system switches back to normal conditions at the 65^{th} cycle (at

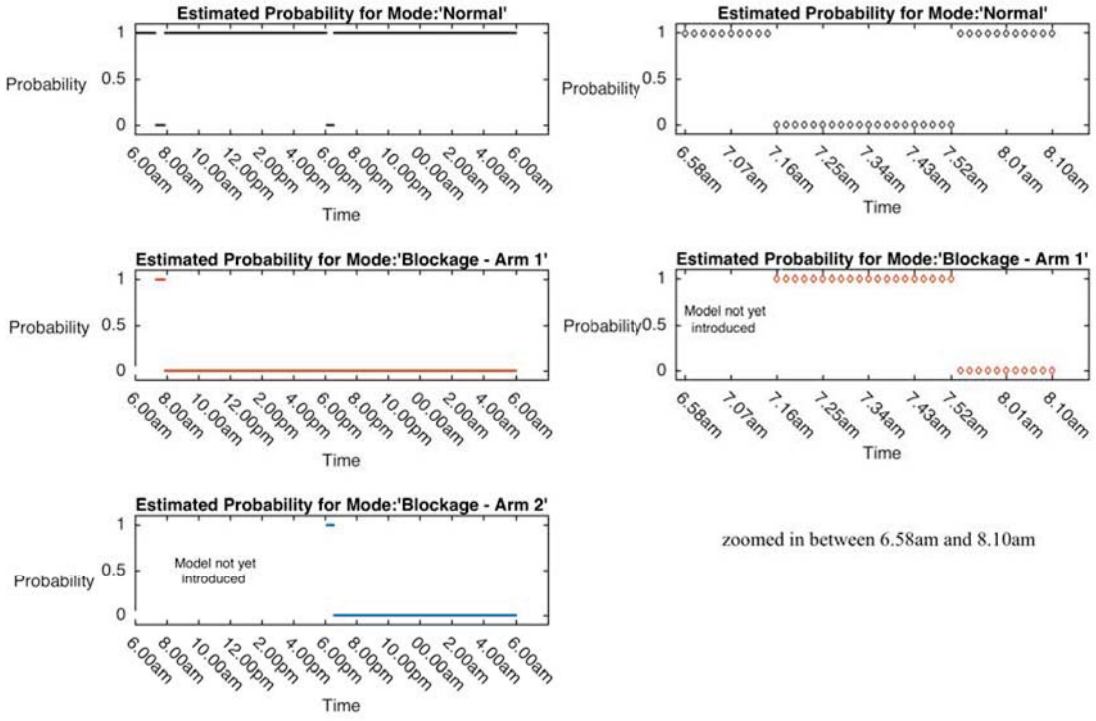

Figure 9.1: Estimated switching conditions for the 3-arm junction subject to arm closure

7.59am) and at the 435^{th} cycle (at 7.17pm) respectively. This results in one cycle delay in detecting arm 1 and arm 2 closures and another cycle delay in switching back to normal conditions following the respective arm closures in a 4-arm junction as in the previous case.

Figure 9.3 shows the estimated queue length for all 4 arms, with $l_{2,2}$ norm being implemented to solve the optimization problem depending on the order defined in Table 8.2 in Chapter 8. Similar to the previous chapter, during saturated traffic conditions, the MPC calculates the required control action, while during unsaturated traffic conditions, the fixed time plans are implemented. Comparisons are drawn with the fixed time plans applied to the junction as was performed in Chapter 8. In the morning, Υ of arm 3 is last to be relaxed if no solution is found. The

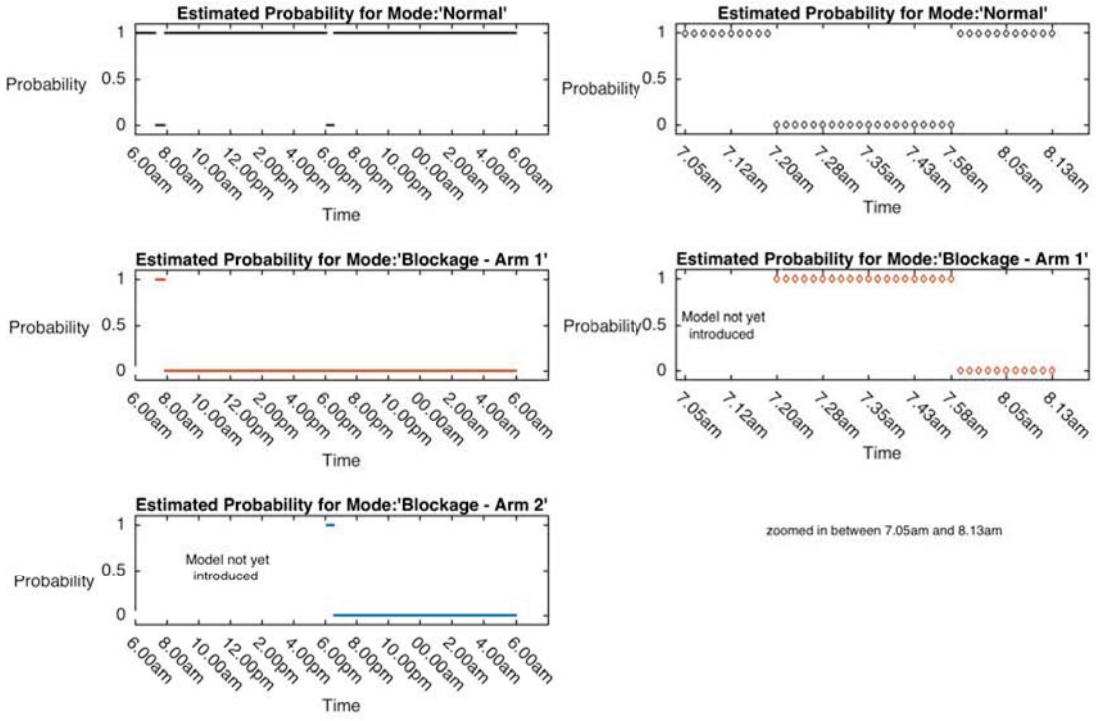

Figure 9.2: Estimated switching conditions for the 4-arm junction subject to arm closure

MPC assigns the large green times to this arm to keep the queue length bounded within Υ, while minimum green times are assigned to arms 2 and 4 respectively. Similarly in the evening, whereby Υ for arm 1 is last to be relaxed if no solution is found, compared to all other 3 arms. Hence significant green time is assigned to such arm while minimum green time is assigned to arm 4. Some green times are assigned to arms 2 and 3 respectively, so as to keep the queue lengths within the upper bound for each arm. Similarly for the 3-arm junction. Figure 9.4 shows the queue length for all 3 arms, with $l_{2,2}$ norm being implemented to solve the optimization problem. In the morning, the MPC assigns large green times to arm 3 and minimum green times to arm 2 during many cycles, at the cost of increasing queue lengths in arms 1 and 2. On the other hand, in the evening, Υ for arm 1 is last to be relaxed if no solution is found. Hence significant green time is

assigned to such arm at the expense of slightly increasing the queue lengths in arms 2 and 3 respectively.

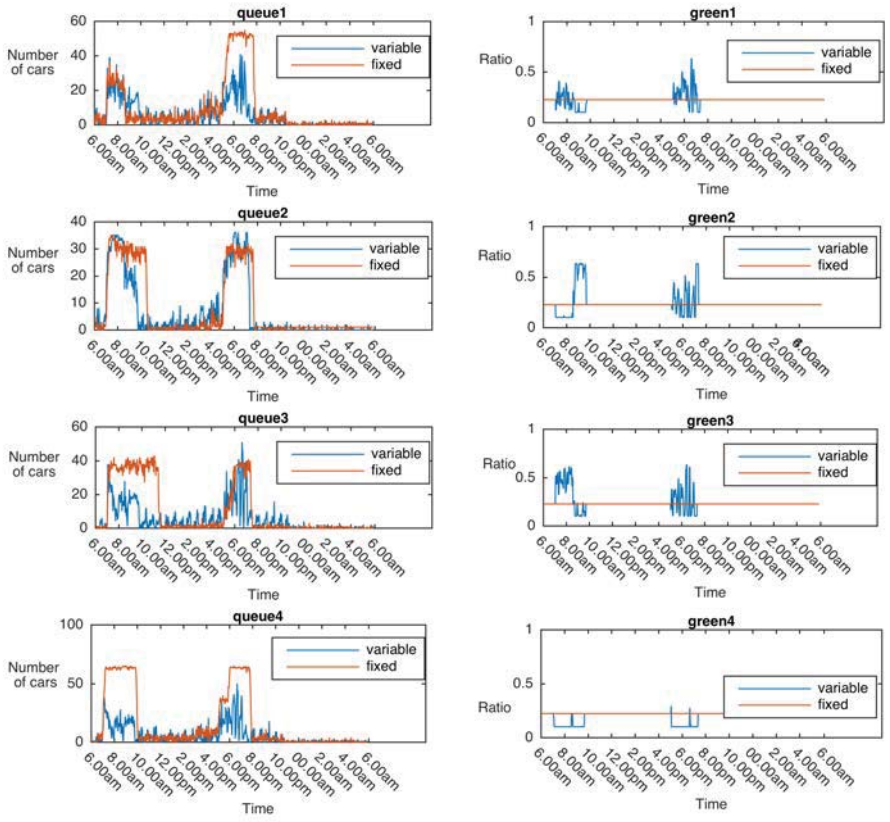

Figure 9.3: 4-arm $l_{2,2}$ norm with multiple model estimation with arms closure

Table 9.2 shows the performance of the different mixed norms when applied to solve the optimization problem for a 3-arm junction, subject to the same traffic data. The resulting total queue lengths and travel time, over the whole day, for the MPC are compared to those resulting from fixed time plans. The % difference in total queue lengths and the % difference in travel time between both implementations over the whole day are worked out as given in Table 9.2. A reduction in queue length was observed for all mixed norms ranging between 14.3% to 23.2% and a reduction in travel time ranging between 19.5% and 28.9%. For

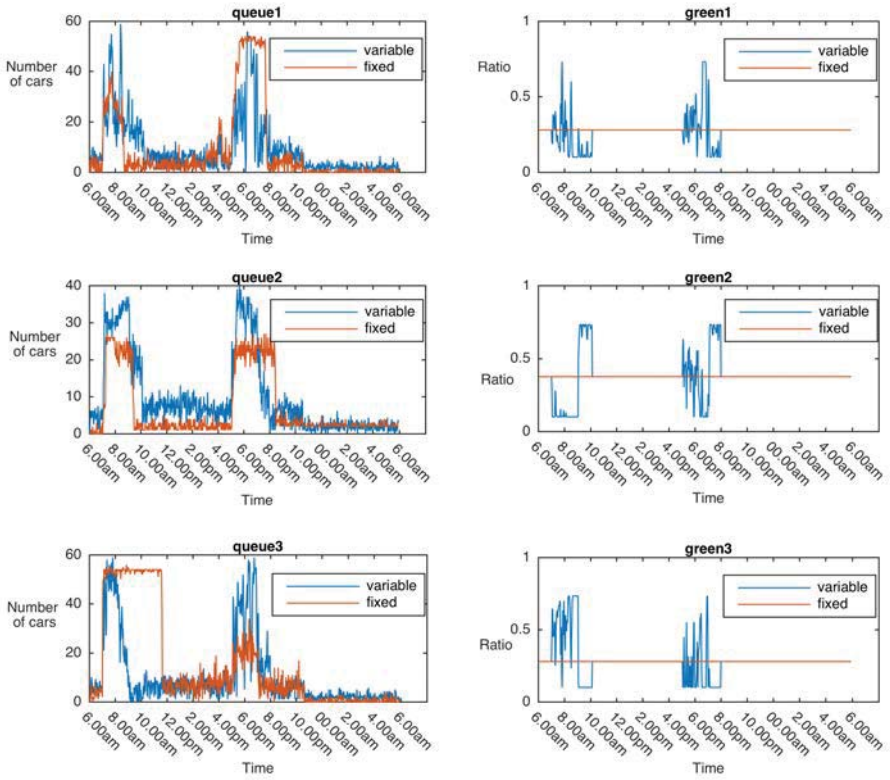

Figure 9.4: 3-arm $l_{2,2}$ norm with multiple model estimation with arms closure

the 4-arm junction the reduction in queue length for the mixed norms ranged between 31.5% to 42.2% and the reduction in travel time ranged between 27.1% to 30.4% as shown in Table 9.3.

Table 9.2: Results of different norm for the 3-arm junction with multiple model estimation and MPC controllers

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-17.6	-19.5
$l_{2,2}$ norm	-23.2	-28.9
$l_{\infty,\infty}$ norm	-14.3	-25.9

Table 9.3: Results of different norm for the 4-arm junction with multiple model estimation and MPC controllers

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-38.5	-30.3
$l_{2,2}$ norm	-42.2	-30.4
$l_{\infty,\infty}$ norm	-31.5	-27.1

To test the consistency of the performance obtained by the MPC controllers with multiple model estimation, a Monte Carlo run of 100 realisations was performed on the 3 and 4-arm junctions. Measurements of vehicles entering and leaving each arm were simulated for the whole day in Aimsun, for 100 different realisations subject to the same arm closures, each time recording the performance of the MPC controllers on the queue lengths. Figures 9.5 and 9.6 show box plots representing the % reduction in total queue length for the 3 and 4-arm junctions respectively. In both cases, the MPC controller with the $l_{2,2}$ norm produced the largest % reductions in queue lengths. A % mean reduction is noted by averaging the % reductions over all 100 realisations for each norm. For the 3-arm junction, a mean reduction of 30.2% was observed, compared to a % mean reduction of 16.9% with the $l_{1,1}$ norm and a % mean reduction of 19.1% with the $l_{\infty,\infty}$ norm for a 3-arm junction. Similarly, for the 4-arm junction, a % mean reduction of 41.0% with the $l_{2,2}$ norm compared to a % mean reduction of 34.9% with the $l_{1,1}$ norm and a % mean reduction of 21.2% with the $l_{\infty,\infty}$ norm.

Furthermore, a one-way analysis of variance was used to determine whether there is any statistically significant differences between the means of three norms. There is a statistically significant difference between all 3 means with a p-value of 5.549×10^{-100} for the 3-arm junction and a p-value of 1.209×10^{-98} for the 4-arm junction at 95% confidence interval. Moreover, a paired sample t-test was carried out to determine whether the mean difference between two sets of norms is zero. Tables 9.4 and 9.5 show the results for the 3 paired sets of norms, where in all cases a statistically significant difference between the two respective means of the norms was observed at 95% confidence interval.

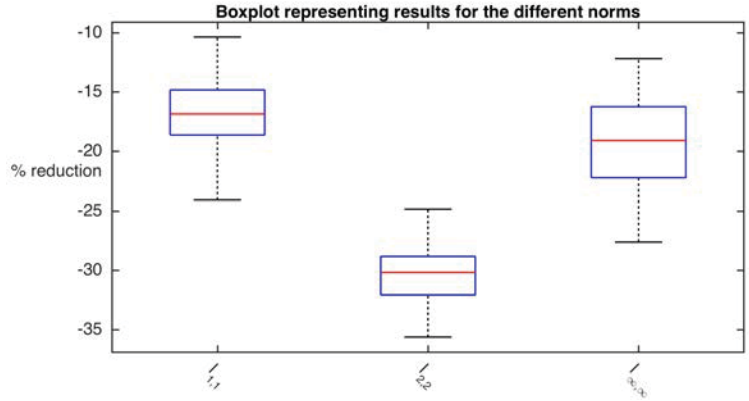

Figure 9.5: Box plot representing the results of 100 realisations using the multiple model estimation with MPC controller for the 3-arm junction with arm closures

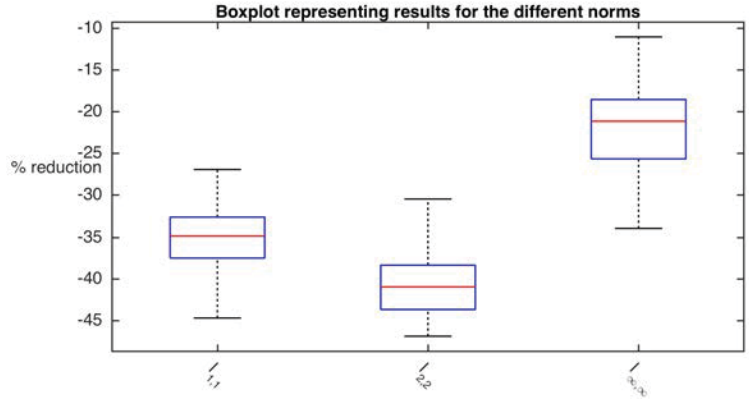

Figure 9.6: Box plot representing the results of 100 realisations using the multiple model estimation with MPC controller for the 4-arm junction with arm closures

Table 9.4: Paired sample t-test results for the 3-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	1.780×10^{-63}
$l_{2,2}$ and $l_{\infty,\infty}$	7.053×10^{-46}
$l_{1,1}$ and $l_{\infty,\infty}$	6.950×10^{-6}

9.4.1.2 Lane closure

The 3-arm and 4-arm junction were also subject to a lane closure in arm 1, with actual timings of such closure described in Section 7.4. Figure 9.7 shows the estimated switching conditions for the 3-arm and Figure 9.8 shows the estimated

Table 9.5: Paired sample t-test results for the 4-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	6.675×10^{-19}
$l_{2,2}$ and $l_{\infty,\infty}$	2.227×10^{-49}
$l_{1,1}$ and $l_{\infty,\infty}$	2.864×10^{-35}

switching conditions for the 4-arm. Initially, the number of modes H is unknown and only one local model is known initially and this denotes 'normal' traffic conditions. Hence modes and their models are learnt as they occur in real-time. The subplots on the right in Figures 9.7 and 9.8 zoom in on the situation between 6.45am and 9.01am so as to clearly show the transition before and after lane closure in arm 1.

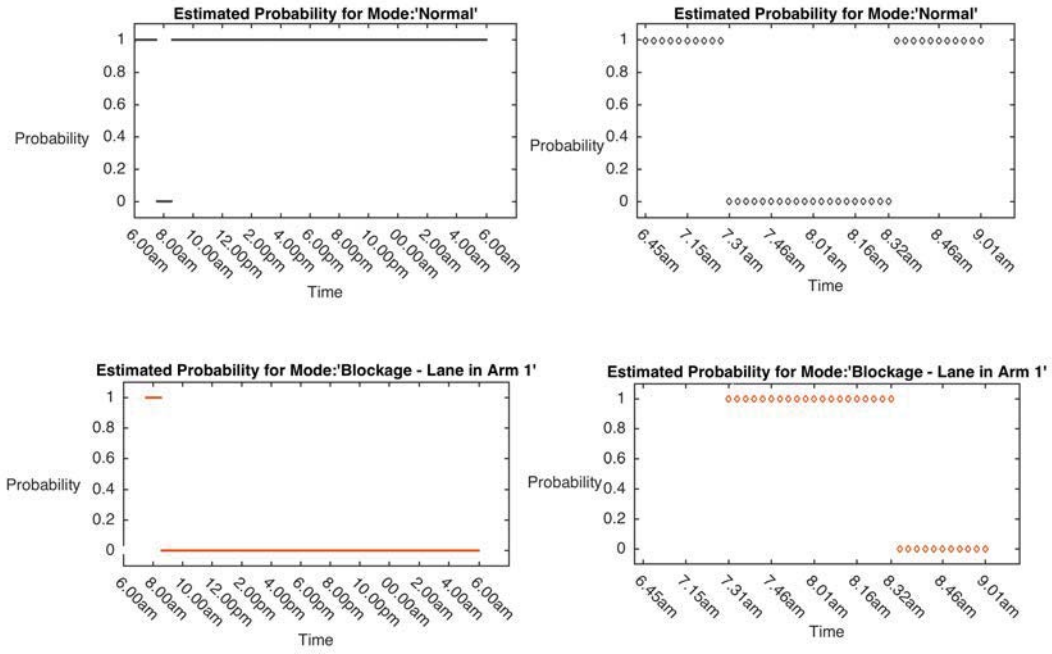

Figure 9.7: Estimated switching conditions for the 3-arm junction subject to lane closure and with multiple model estimation and MPC controllers

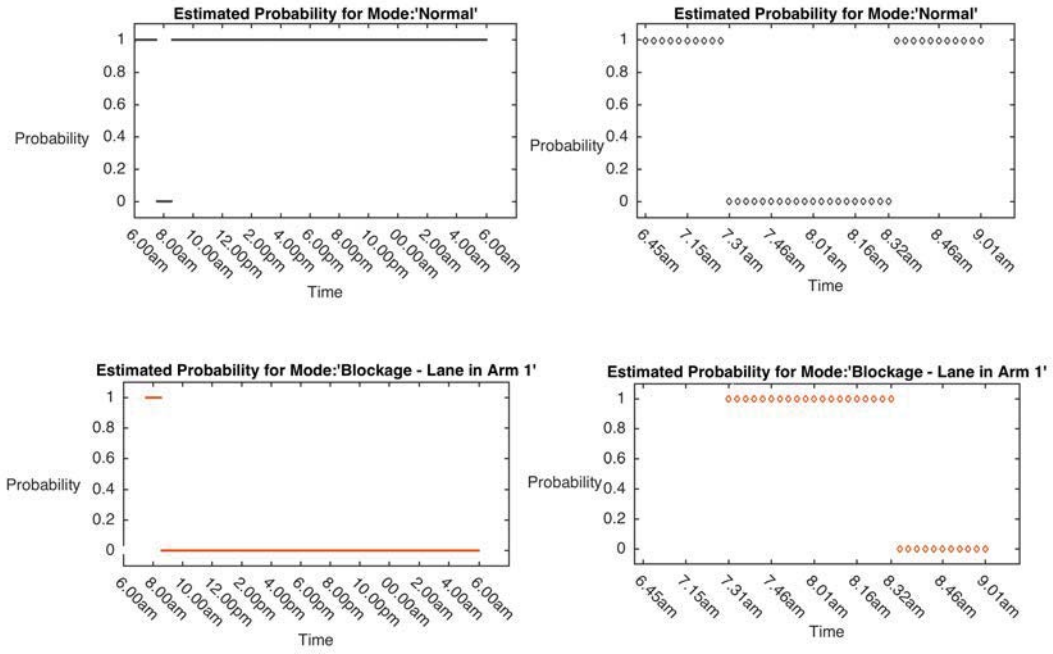

Figure 9.8: Estimated switching conditions for the 4-arm junction subject to lane closure and with multiple model estimation and MPC controllers

Similar to the results in Chapter 7, before the 53^{rd} cycle, there is no trace for the probability for 'blockage - lane in arm 1', because this model is introduced at 7.31am. The system switches back to normal conditions at the 88^{th} cycle (at 8.32am). This results in one cycle delay in detecting lane closure and another cycle delay in switching back to normal conditions following the lane closure in the 3-arm junction as in the previous case. Similarly, for a 4-arm junction, lane closure is detected at the 53^{rd} cycle (at 7.31am) and switches back to normal conditions at the 88^{th} cycle (at 8.32am).

Tables 9.6 and 9.7 show the performance of the different mixed norms when applied to solve the optimization problem for the 3-arm and 4-arm junctions respectively, subject to the same traffic data. The resulting total queue lengths and travel time, over the whole day, for the MPC are compared to those resulting

from fixed time plans. The % difference in total queue lengths and the % difference in travel time between both implementations over the whole day are worked out as given in Tables 9.6 and 9.7 . A reduction in queue length was observed for all mixed norms ranging between 27.1% to 34.6% and a reduction in travel time ranging between 19.9% and 32.7%, for the 3-arm junction and a reduction in queue length ranging between 33.7% to 48.1% and travel time between 20.8% to 28.2% for the 4-arm junction.

Table 9.6: Results of different norm for the 3-arm junction with multiple model estimation and MPC controllers

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-29.2	-24.4
$l_{2,2}$ norm	-34.6	-32.7
$l_{\infty,\infty}$ norm	-27.1	-19.9

Table 9.7: Results of different norm for the 4-arm junction with multiple model estimation and MPC controllers

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-42.6	-22.4
$l_{2,2}$ norm	-48.1	-28.2
$l_{\infty,\infty}$ norm	-33.7	-20.8

Figure 9.9 shows the estimated queue length for all 3 arms, with $l_{2,2}$ norm being implemented to solve the optimization problem. Similar to the previous results in this Chapter, comparisons are drawn with the fixed time plans applied to the junction. The queue lengths and green times generated from the MPC per cycle are denoted in the legend as *variable* in Figure 9.9, while the queue lengths and green times generated from the fixed time plans are denoted as *fixed*. In the morning, the MPC assigns large green times to arm 3 and minimum green times to arm 2 during many cycles, at the cost of increasing queue lengths in arms 1 and 2. Υ of arm 3 is last to be relaxed if no solution is found. The MPC assigns the large green times to this arm to keep the queue length bounded within Υ, while minimum green times are assigned to arms 1 and 2 respectively. During lane

closure between 7.30am and 8.30am, the controller assigns large green time to arm 1, while keeping the queue length in arm 3 bounded within Υ. On the other hand, in the evening, significant green time is assigned to arm 1 at the expense of slightly increasing the queue lengths in arms 2 and 3 respectively, with Υ for arm 1 last to be relaxed if no solution is found.

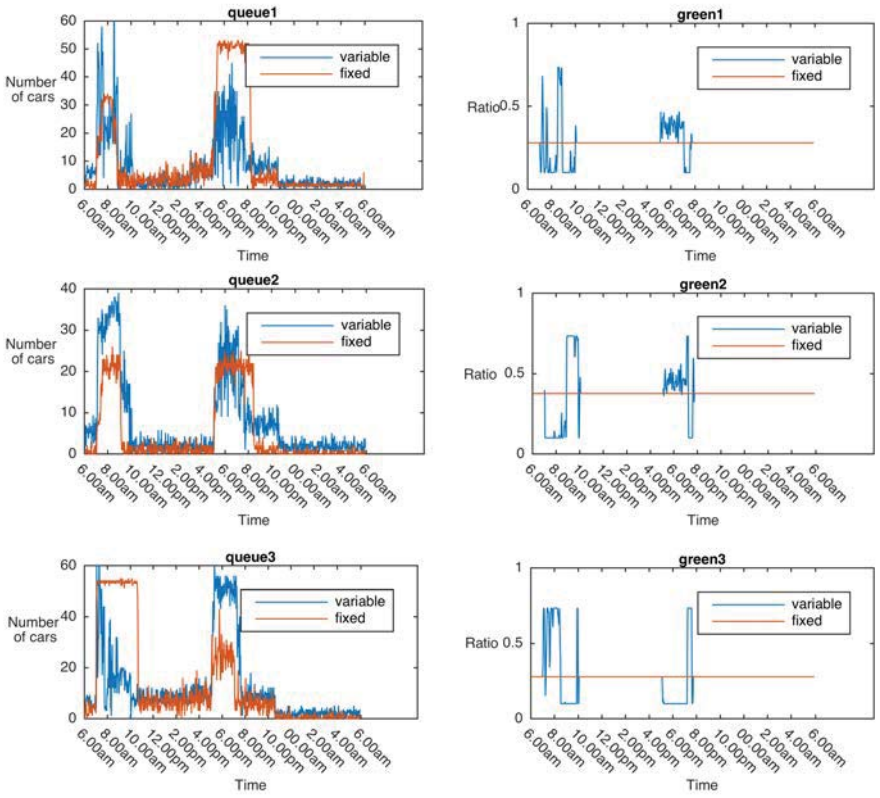

Figure 9.9: 3-arm $l_{2,2}$ norm with multiple model estimation with lane closure

Similar behaviour is observed for the 4-arm junction. Figure 9.10 shows the estimated queue length for all 4 arms, with $l_{2,2}$ norm being implemented to solve the optimization problem. During lane closure between 7.30am and 8.30am, the controller assigns large green time to arm 1, while keeping the queue length in all other 3 arms bounded within Υ. Similarly in the evening, whereby Υ for arm 1 is last to be relaxed if no solution is found, compared to all other three arms. Hence significant green time is assigned to this arm while minimum green time

is assigned to arm 4. Some green times are assigned to arms 2 and 3 respectively, so as to keep the queue lengths within the upper bound for each arm.

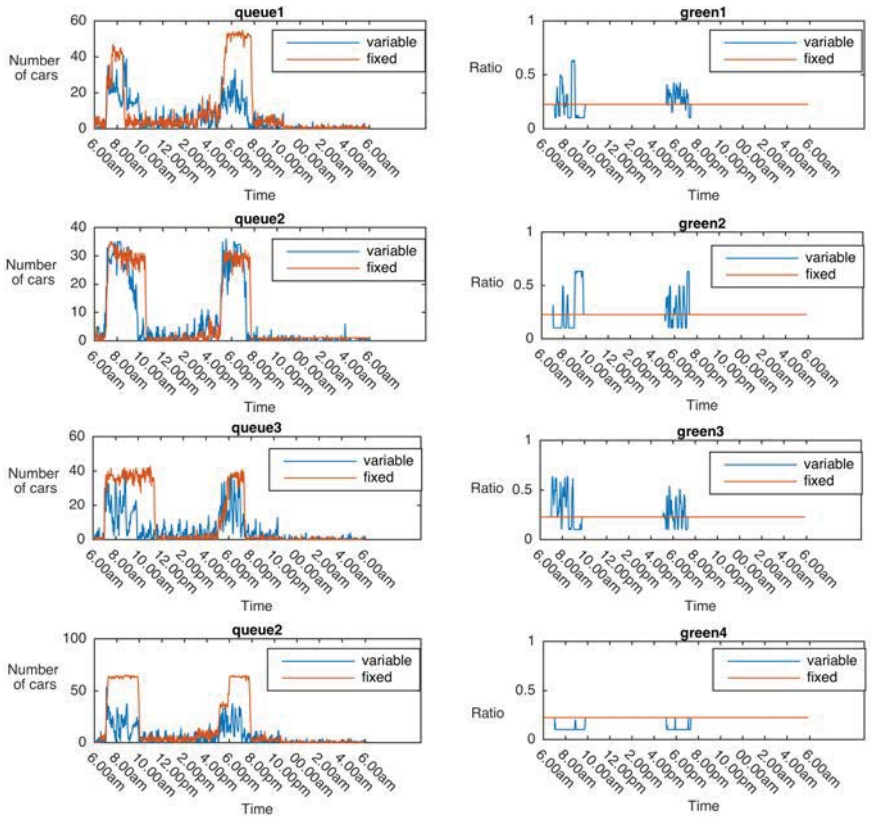

Figure 9.10: 4-arm $l_{2,2}$ norm with multiple model estimation with lane closure

To further test the consistency of the performance obtained by the MPC controllers with multiple model estimation, a Monte Carlo run of 100 realisations was performed on the 3-arm and 4-arm junctions. Measurements of vehicles entering and leaving each arm were simulated over the whole day, in Aimsun for 100 different realisations subject to the same lane closure in arm 1, each time recording the performance of the MPC controllers on the queue lengths. Figure 9.11 and 9.12 show box plots representing the % reduction in total queue length for the 3-arm and 4-arm junction respectively. Similar to the previous results presented earlier in the Chapter, the MPC controller with the $l_{2,2}$ norm produced

the largest % reductions in total queue lengths. A % mean reduction is noted by averaging the % reductions over all 100 realisations for each norm. For the 3-arm junction, a mean reduction of 33.3% was observed, compared to a % mean reduction of 29.9% with the $l_{1,1}$ norm and a % mean reduction of 27.6% with the $l_{\infty,\infty}$ norm for a 3-arm junction. For the 4-arm junction, a % mean reduction of 40.3% with the $l_{1,1}$ norm, a % mean reduction of 49.7% with the $l_{2,2}$ norm and a % mean reduction of 30.3% with the $l_{\infty,\infty}$ norm were noted.

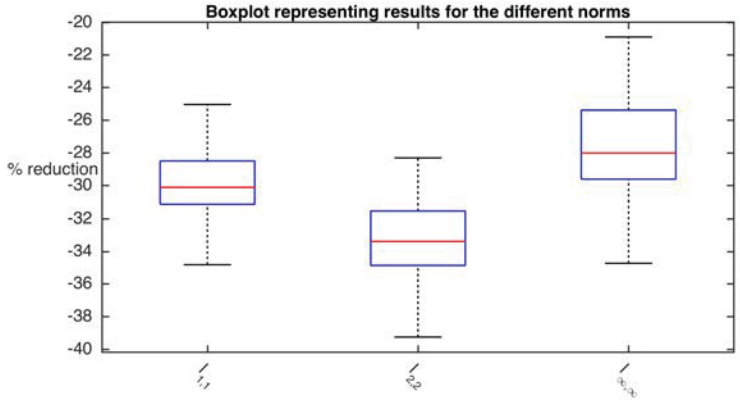

Figure 9.11: Box plot representing the results of 100 realisations using the multiple model estimation and the MPC controller for the 3-arm junction with lane closure

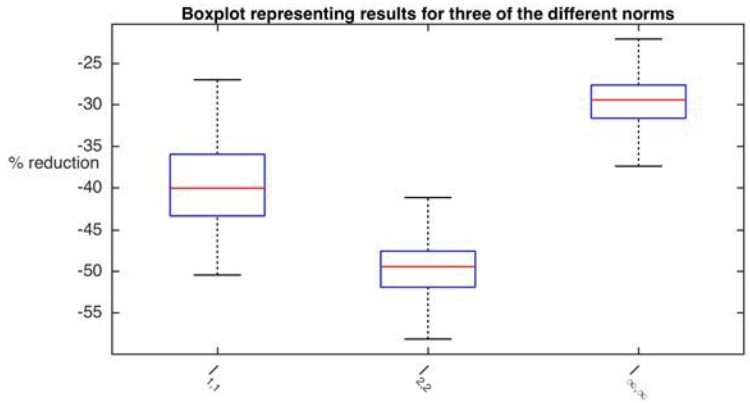

Figure 9.12: Box plot representing the results of 100 realisations using the multiple model estimation and the MPC controller for the 4-arm junction with lane closure

A one-way analysis of variance was used to determine whether there is any statistically significant differences between the means of three norms. There is a statistically significant difference between all 3 means with a p-value of 1.874×10^{-43} for the 3-arm junction and 1.884×10^{-104} for the 4-arm junction at 95% confidence interval. Moreover, a paired sample t-test was carried out to determine whether the mean difference between two sets of norms is zero. Tables 9.8 and 9.9 show the results for the 3 paired sets of norms, where in all cases a statistically significant difference between the two respective means of the norms was observed at 95% confidence interval.

Table 9.8: Paired sample t-test results for the 3-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	5.522×10^{-19}
$l_{2,2}$ and $l_{\infty,\infty}$	5.638×10^{-28}
$l_{1,1}$ and $l_{\infty,\infty}$	3.0695×10^{-9}

Table 9.9: Paired sample t-test results for the 4-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	9.905×10^{-30}
$l_{2,2}$ and $l_{\infty,\infty}$	2.263×10^{-68}
$l_{1,1}$ and $l_{\infty,\infty}$	3.156×10^{-30}

9.4.2 Multiple Model Estimation with one controller

The results of MPC controllers with multiple model estimation are further compared to the multiple model estimation but with one controller for all the considered regimes that is 'normal' and 'lane 1 blockage'. The controller which was applied to these regimes, was that designed for 'normal traffic'. Note that it is not relevant to compare with arm closure because arm closure will only affect the turning ratios, which in turn do not have an effect on the MPC prediction model Equation (8.2.1) whose objective is to minimise queue lengths only.

Tables 9.10 and 9.11 show the performance of all mixed norms when applied to

solve the optimization problem for a 3-arm junction and 4-arm junction, subject to the same lane closure in arm 1, with actual timings of such closure described in Section 7.4. The resulting total queue lengths and travel time for the MPC are compared to those resulting from fixed time plans. A reduction in total queue length was observed for all mixed norms ranging between 8.1% to 14.2% for the 3-arm and 11.8% to 25.5% for the 4-arm junction. A reduction in travel time ranging was also noted, ranging between 11.1% and 15.9% for the 3-arm junction and between 10.2% and 14.9% for the 4-arm junction.

Table 9.10: Results of different norm for the 3-arm junction with multiple model estimation and 1 controller for all traffic scenarios

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-10.7	-11.6
$l_{2,2}$ norm	-14.2	-15.9
$l_{\infty,\infty}$ norm	-8.1	-11.1

Table 9.11: Results of different norm for the 4-arm junction with multiple model estimation and 1 controller for all traffic scenarios

Norm	% Difference in total queue length	% Difference in travel time
$l_{1,1}$ norm	-23.4	-12.3
$l_{2,2}$ norm	-25.5	-14.9
$l_{\infty,\infty}$ norm	-11.8	-10.2

A Monte Carlo run of 100 realisations was performed on the 3-arm and 4-arm junctions subject to the same switching conditions. Figures 9.13 and 9.14 show box plots representing the % reduction in total queue length. A % mean reduction is noted, by averaging the % reductions over all 100 realisations for each norm. A mean reduction of 10.1% was observed with the $l_{1,1}$ norm, a % mean reduction of 13.9% with the $l_{2,2}$ norm and a % mean reduction of 8.5% with the $l_{\infty,\infty}$ norm for the 3-arm junction. Similarly, a mean reduction of 12.1% was observed with the $l_{1,1}$ norm, a % mean reduction of 14.2% with the $l_{2,2}$ norm and a % mean reduction of 10.9% with the $l_{\infty,\infty}$ norm for the 4-arm junction as shown in Figure

9.14.

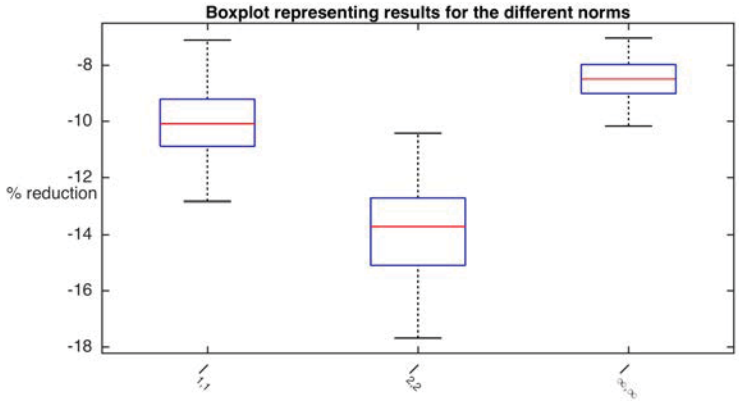

Figure 9.13: Box plot representing the results of 100 realisations using the MPC controller for the 3-arm junction with one controller

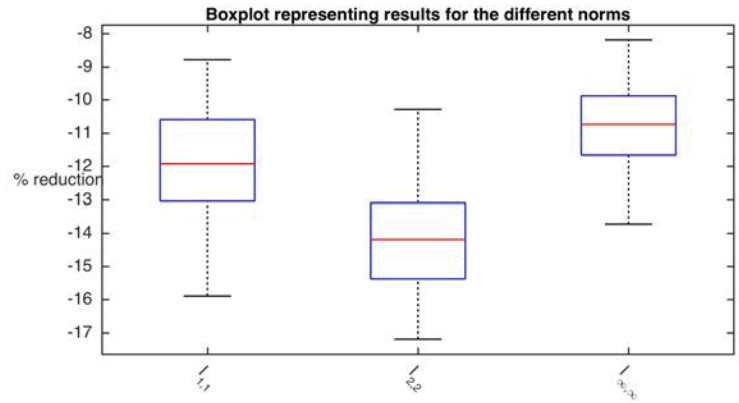

Figure 9.14: Box plot representing the results of 100 realisations using the MPC controller for the 4-arm junction with one controller

A one-way analysis of variance was used to determine whether there is any statistically significant differences between the means of three norms. There is a statistically significant difference between all 3 means with a p-value of 7.265×10^{-93} for the 3-arm junction and a p-value of 6.5208×10^{-42} for the 4-arm junction at 95% confidence interval. Moreover, a paired sample t-test was carried out to determine whether the mean difference between two sets of norms is zero. Table 9.12 and 9.13 show the results for the 3 paired sets of norms, where in all cases a statistically significant difference between the two respective means of the norms was observed at 95% confidence interval.

Table 9.12: Paired sample t-test results for the 3-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	3.770×10^{-33}
$l_{2,2}$ and $l_{\infty,\infty}$	3.949×10^{-54}
$l_{1,1}$ and $l_{\infty,\infty}$	4.392×10^{-19}

Table 9.13: Paired sample t-test results for the 4-arm junction

Pairs	p-value
$l_{1,1}$ and $l_{2,2}$	9.924×10^{-15}
$l_{2,2}$ and $l_{\infty,\infty}$	6.853×10^{-34}
$l_{1,1}$ and $l_{\infty,\infty}$	2.820×10^{-7}

The results obtained when applying the multiple model estimation with one controller, presented in Figures 9.13 and 9.14 are compared to the results obtained when applying the multiple model estimation with MPC controllers presented in Figures 9.11 and 9.12. The latter results further quantify the effectiveness of the MPC, whereby a mean reduction in queue length of 30.3% was observed for the 3-arm junction, by taking the average over all the 3 different norms when applying the MPC controllers, while a mean reduction in queue length of 10.8% was observed by taking the average overall the 3 different norms, when applying one controller. Similarly, a mean reduction in queue length of 40.1% was observed for the 4-arm junction, by taking the average over all the 3 different norms when applying the MPC controllers, while a mean reduction in queue length of 12.4% was observed by taking the average overall the 3 different norms, when applying one controller. A paired sample t-test was carried out to determine whether there is any significant difference in the mean of each norm for the multiple model estimation with one controller and the multiple model estimation with MPC controllers. Tables 9.14 and 9.15 show the results for the 3-arm and 4-arm junctions where in all cases a statistically significant difference between the two respective means of the norms was observed at 95% confidence interval.

Table 9.14: Paired sample t-test results for the 3-arm junction

Pairs	p-value
$l_{1,1}$ for the multiple model estimation with MPC controllers and $l_{1,1}$ for the multiple model estimation with one controller	1.568×10^{-94}
$l_{2,2}$ for the multiple model estimation with MPC controllers and $l_{2,2}$ for the multiple model estimation with one controller	1.764×10^{-95}
$l_{\infty,\infty}$ for the multiple model estimation with MPC controllers and $l_{\infty,\infty}$ for the multiple model estimation with one controller	8.047×10^{-90}

Table 9.15: Paired sample t-test results for the 4-arm junction

Pairs	p-value
$l_{1,1}$ for the multiple model estimation with MPC controllers and $l_{1,1}$ for the multiple model estimation with one controller	3.766×10^{-107}
$l_{2,2}$ for the multiple model estimation with MPC controllers and $l_{2,2}$ for the multiple model estimation with one controller	4.671×10^{-114}
$l_{\infty,\infty}$ for the multiple model estimation with MPC controllers and $l_{\infty,\infty}$ for the multiple model estimation with one controller	9.840×10^{-93}

9.5 Conclusion

The results in this Chapter highlight the effectiveness of using a multiple model adaptive control scheme for *jump* structural changes in junction dynamics. Several *jump* dynamics were simulated and tested on the 3-arm and 4-arm junction, such as arm closure and lane closure. The results showed the accuracy in the estimated switching conditions and the effectiveness of the MPC compared to fixed time plans. A mean reduction in queue length of 30.3% was observed for the 3-arm junction, by taking the average over all the 3 different norms when applying the MPC controllers and a mean of 40.1% for the 4-arm junction. These results are further compared to the results obtained from the multiple model estimation

with one controller for all traffic scenarios. A mean reduction in queue length of 10.8% was observed for the 3-arm junction, by taking the average over all the 3 different norms and a mean of 12.4% for the 4-arm junction. Statistical tests also showed significant difference between the mean results obtained for both cases, that is the multiple model estimation with MPC results and the multiple model estimation with one controller, at 95% confidence interval.

Chapter 10

Discussion and Conclusions

10.1 Introduction

Modern urban areas are witnessing increasing traffic congestion. Signalized junctions often pose the main bottleneck in urban traffic networks resulting in decreased traffic flow on the network links and excessive delays, with detrimental effects to our health, economy and standard of living. Expansion of the current road infrastructure is a possible option, yet not always viable due to land restrictions and financial limitations. Moreover, expansion of the road infrastructure could lead to an increased use of private vehicles, leading back to bottleneck situations. An alternative option is to use more efficiently the current infrastructure. Hence, there is an incentive for further development of traffic signal control systems. The configurations of traffic signal systems range from simple time-based signal systems that only utilize predetermined timing sequences, to traffic-responsive systems that heavily rely on surveillance and communications to optimize traffic flows, as discussed in Chapters 1, 2 and 3.

Despite the great advances in ITS and in the response of traffic light systems to traffic demand, the controller parameters of such systems are not tuned to changing traffic behaviour. Hence such systems can fail when networks are subject to irregularities, such as roadworks, accidents or flooding [23]. Human intervention is required with the current systems to provide route diversion recommendations that might not always guarantee the optimal use of the available network capacity.

Realizing the limitations of the above mentioned current practice, an adaptive

system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is required. The need to design self-managing systems, which self-handle the complexity and uncertainties and thus reduce human intervention to a minimum is of utmost importance. As the complexity of traffic systems grows, the need to develop systems which can manage and maintain themselves becomes necessary. Autonomic traffic light systems are required to self-manage the traffic network.

To obtain traffic flow measurements, sensors are installed at strategic locations inside the signalized intersection. Online data given from the sensors can be used to control the traffic light timings in real-time, thus making the system traffic-responsive. However such infrastructural installation and maintenance costs are a burden and should be kept to a minimum. To mitigate such problems, autonomic traffic light systems should not ignore these limitations and constraints.

10.2 Contributions

This thesis presented a set of novel techniques for the development of more autonomic signalized junctions in urban environments. The main contributions and the novelty in this work are summarized below:

I The development of real-time joint state and parameter estimation algorithms to simultaneously estimate the model states, parameters and noise covariance of the state-space model describing the traffic flow dynamics in a signalized junction. In the previous literature publications discussed in Chapter 2, the model parameters are typically assumed known apriori from past traffic measurements or by applying nonlinear recursive estimation algorithms such as the EKF. However, difficulties with nonlinear estimation algorithms such as divergence issues, motivated the development of joint state and parameter estimation algorithms from a different perspective. The proposed algorithms make use of a data-driven, macroscopic signalized junction model in state-space form as presented in Chapter 4 and a novel variation on the EM algorithm. Classical EM is designed to operate in batch

mode, making it impractical for real-time estimation.

Thus, a quasi real-time joint estimation of model states, parameters and noise is first proposed in Chapter 5, whereby the standard EM algorithm is modified to carry out estimation in quasi real-time. An iterative algorithm is proposed making use of a short uniform window of sensor measurements of fixed time length, which looks back in time. Joint parameter and state estimation is carried out for those particular time points falling inside the window, and at every time iteration the window slides forward by one time instant and the procedure is repeated again. However, this proposed estimation cannot be considered to represent a truly real-time methodology because it still requires continuous storage of a reduced set of measurement data inside a moving window. Hence, traffic information from sensors needs to be available and stored as a batch of window samples during all time instants.

Therefore, a more efficient joint estimation algorithm is next presented in Section 6.1 which makes use of Robbins-Monro stochastic approximation. This estimation algorithm uses the measurement data only once and avoids batch storage, leading to a full real-time algorithm exhibiting shorter temporal delays and reduced storage requirements. Given the large number of model parameters to be estimated and the high level of dependency among these parameters, this algorithm was posed first as a single variant estimation algorithm in Section 6.1 followed by a multivariate estimation algorithm as detailed in Section 6.2 in Chapter 6.

All the estimation algorithms previously discussed in this work, required the partial derivative of the likelihood function with respect to each parameter to be worked out analytically. In practice, this approach is impractical for larger and complex junctions due to the complexity of the derivatives involved in deriving such equations. Hence a derivative-free approach is proposed, as presented in Section 6.3, allowing for easier generalisation to other junctions and scalability to more complex junctions.

II The development of real-time multiple model adaptive estimation methods to estimate states and parameters of signalized traffic junctions suitable for structurally diverse dynamical regimes arising abruptly from unexpected traffic congestions or network irregularities due to, for example, road works, blockages due to accidents and extreme weather conditions. Previously proposed adaptive systems discussed in literature, relied on traffic surveillance technologies to warn the commuters of any detected irregularities and in most cases rely on human experience to evaluate the impact on the network performance and to provide route diversion recommendations that might not guarantee the optimal use of the available network capacity. The autonomic systems that were introduced to urban traffic networks heavily rely on a communication and cooperative agent infrastructure that shares or exchanges information between vehicles leading to a complicated timely process. Hence a less infrastructure demanding multiple model adaptive estimation method is proposed in Chapter 7 to estimate states and parameters of signalized traffic junctions subject to time-varying parameters, *jump* dynamics and unpredictable disturbances. This approach is able to detect such abrupt changes in the dynamic structure of signalized traffic junctions, learns the dynamics and model parameters of the different active regimes, and estimates which regime is active at any given time. This approach is also able to grow a new model set to represent traffic behaviour under novel different conditions.

III The development of an autonomic control scheme. Many so-called "adaptive" systems were proposed in literature. Although such systems are responsive to traffic conditions, the controller parameters are not autonomously tuned to changing traffic behaviour so that the controller is able to adapt itself to changing traffic conditions and maintain optimal levels of performance. A truly adaptive system can self-tune and adjust the controller parameters to adapt to changing traffic conditions. Such a system is presented in Chapter 8. The proposed autonomic controller for signalized junctions is based upon MPC using linear and quadratic programming optimization techniques as discussed in Chapter 8. This scheme is validated by simu-

lating typical signalized 3-arm and 4-arm junctions, with the ability to autonomously adjust to changing traffic conditions.

IV The integration of the latter two novelties (multiple model adaptive estimation and autonomic control) to obtain a multiple model adaptive control scheme for *jump* structural changes in junction dynamics as presented in Chapter 9.

10.3 Interpretation of Results

Results show that the proposed state-space model with online self-estimation of model states and parameters is able to capture the complex traffic dynamics of urban junctions. Validation of the macroscopic results with a microscopic simulator, Aimsun, for a typical 3-arm and 4-arm junction, during a typical working day from 6.00am of one day to 6.00am of the next day show significant similarities. The average % RMSE in the estimated states for the 3-arm junction was found to be 0.725%, and 0.6% for the 4-arm junction, when implementing the derivative-free approach presented in Chapter 6. Aimsun, being a microscopic simulator, implements a car-following model known as Gipps model [375] where vehicles accelerate to achieve the desired speed and decelerate when drivers have to avoid a collision, whilst maintaining the desired speed. On the other hand, the macroscopic model used in this study provides a general evaluation of the traffic flow behaviour, rather than individual driver behaviour with its acceleration and deceleration instances. Hence the minor discrepancies in such results could be due to these modelling differences. Moreover, the measurement noise could also contribute to such minor discrepancies.

The self-estimation algorithm was further extended to include self-detection of *jump* dynamics in real-time. Results show that the proposed algorithm was able to not only detect changes between different operating regimes, but also to learn potential dynamical regimes not anticipated *a priori*. Typical arm closure scenarios were simulated for the 3-arm and 4-arm junctions, with the system being able to detect in real-time the switching among the multiple models as well as learning

simultaneously any new regime while estimating the state variables and parameters of each active model. The *jump* structural changes in junction dynamics are not limited to arm closures but could include network changes such as closure of only a few lanes within an arm, arising from traffic accidents, flooding in the network or maintenance of roads. Variable message signals could be integrated with such a self-detection algorithm to autonomously warn commuters of any detected irregularities in the network.

Traffic signal timing plays an important role in ensuring efficient flow and reduction of traffic congestions. Fixed signal times work well when traffic conditions are consistent. However performance is degraded when traffic conditions are subject to high demands or during unusual occurrences such as traffic incidents or unanticipated network obstructions, causing significant changes to the normal traffic conditions. To address these situations, several traffic- responsive systems as discussed in Chapter 1 were developed. However in these cases, the controller parameters are not adaptively tuned to changing traffic behaviour. Hence, a novel adaptive system which can self-tune and adjust the controller parameters to adapt to changing traffic conditions is developed, leading towards an autonomic system. This system makes use of a MPC based approach which can tune the controller parameters to reflect changing traffic conditions. Different norms were tested to solve the optimization problem such as the $l_{1,1}$, $l_{2,2}$, $l_{\infty,\infty}$ mixed norms. Results in Chapter 8 highlight the effectiveness of using a MPC to control the traffic light timings. The major links receive more green time whilst minor once receive less. This typically benefits the heaviest movements through the intersection, as well as the transit of emergency vehicles which frequently use the major roads. A Monte Carlo run of 100 realisations was performed and it was found out that the $l_{2,2}$ mixed norm produced the largest % reductions in queue lengths compared with the $l_{1,1}$, $l_{\infty,\infty}$ mixed norms. For the 3-arm junction, a mean reduction in queue length of 25.2% was observed, while for the 4-arm junction, a % mean reduction of 45.7% was noted.

As previously indicated, all the above algorithms are subsequently combined to obtain a multiple model adaptive control scheme for *jump* structural changes

in junction dynamics. Results in Chapter 9 highlight the effectiveness of MPC with multiple models, having a controller associated with each mode so as to minimise the queue length in the junction when that mode is detected to be active. The traffic models and the online joint estimation techniques discussed in Part I are included to be able to describe and predict the traffic flow dynamics in the future, as well as tune the controller parameters, once a mode becomes active. In addition to learning the various mode dynamics and detecting mode switches, the system automatically configures itself and grows a new model set to represent traffic behaviour under novel different conditions. Typical arm closure scenarios were simulated for the 3-arm and 4-arm junctions. Results show that the system autonomously makes signal adjustment based on estimated and predicted traffic conditions in real-time. Significant reductions in queue length were observed when compared with queues arising from fixed time plans. For the 3-arm junction, a mean reduction of 30.2% was observed, while for the 4-arm junction, a mean reduction of 40.2% was noted when compared with queues arising from fixed time plans. These results are further compared to the results obtained from the multiple model estimation but with one controller for all traffic scenarios. A mean reduction in queue length of 10.8% was observed for the 3-arm junction and a mean of 12.4% for the 4-arm junction. Statistical tests also showed significant difference between the mean results obtained for both cases, that is the multiple model estimation with MPC and the multiple model estimation with one controller, at 95% confidence interval.

Results point towards the introduction of autonomic control of urban traffic junctions to efficiently improve traffic congestions in junctions, where:

- Traffic conditions fluctuate randomly on a day-to-day basis;
- Traffic conditions change rapidly due to new or changing circumstances in land use;
- Incidents, crashes, or other events that are frequent and result in sudden unexpected changes to traffic demand.

This system can in turn result in several benefits which include:

- reductions in travel times and queues;
- opportunities to improve performance for transit, pedestrians and freight movement.

Furthermore, in order to achieve improvements in efficiency and meet the ever-increasing demands, this system does not affect the safety of the drivers in the sense that the yellow change interval, also known as clearance interval, is left intact to facilitate safe transfer or right-of-way from one movement to another.

10.4 Future Directions

In this work, an isolated traffic junction has been considered. Additional work can include control of cycle duration in the case of isolated junctions, where this approach is relevant. Furthermore, the research can be extended to take into consideration nearby signalized junctions and consider sequence and coordination between junctions to ensure efficient traffic behaviour in the network, consisting of several intersections. In such cases, a hierarchical control structure can be implemented to facilitate coordination on the same lines as in [228] whereby, MPC controllers for each junction can act as local controllers and a global optimum solution is found whilst reducing spill back queueing between closely spaced intersections. In addition, the developed algorithms can be implemented and evaluated on real physical traffic junctions. Further modifications of practical concern include how to handle non Gaussian noise and biased measurements in the sensors. Different control laws can be considered as an alternative to MPC such as adaptive fuzzy control based on human expert knowledge or other intelligent control schemes. These proposed schemes can also be extended to include V2V or V2I communication.

Moreover, the principles of autonomic control can be extended to include preemption or signal priority. Preemption interrupts normal signal operations to transfer right of way to the direction of an approaching emergency vehicle by shortening or extending the green timings to allow a priority vehicle such as ambulances or fire engines to pass through an intersection.

Appendix

$l_{1,2}$ Norm - 1-norm with respect to time and 2-norm with respect to space

The cost function based on the $l_{1,2}$ norm is given by:

$$J_{1,2} = \sum_{t}^{N} \sqrt{\sum_{i}^{n} |y_{c_i}(t)|^2} \tag{A.1}$$

For simplification, the square root in (A.1) is removed. Note that by removing the square root function, the minimum value is not changed. Hence, let:

$$J_{1,2} = \sum_{t}^{N} \sum_{i}^{n} |y_{c_i}(t)|^2 \tag{A.2}$$

The evaluation of Equation (A.2) is similar to that of Section 8.2.2.2.

$l_{2,1}$ Norm - 2-norm with respect to time and 1-norm with respect to space

The cost function based on the $l_{2,1}$ norm is given by:

$$J_{2,1} = \sqrt{(\sum_{t}^{N} (\sum_{i}^{n} |y_{c_i}(t)|)^2)} \tag{A.3}$$

For simplification, the square root in (A.3) is removed. Note that by removing the square root function, the minimum value is not changed. Hence, let:

$$J_{2,1} = \sum_{t}^{N} (\sum_{i}^{n} |y_{c_i}(t)|)^2 \tag{A.4}$$

and let:

$$\mathbf{Y}_s = \sum_{i}^{n} |y_{c_i}(t)| \tag{A.5}$$

$$= \mathbf{T}_{trsf}\mathbf{Y} \tag{A.6}$$

$$= \mathbf{T}_{trsf}(\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t) + \mathbf{R}_c\mathbf{B}_y\mathbf{u}(t)) \tag{A.7}$$

where $\mathbf{T}_{trsf}$ is an $n \cdot N \times n \cdot N$ square matrix, given by:

$$\mathbf{T}_{trsf} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & \cdots \\ \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 & \cdots & 1 & \cdots \end{pmatrix} \tag{A.8}$$

Hence,

$$J_{2,1} = \sum_{t}^{N} (\mathbf{T}_{trsf}\mathbf{Y})^T \mathbf{T}_{trsf}\mathbf{Y} \tag{A.9}$$

$$= \left\| \mathbf{T}_{trsf}(\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t) + \mathbf{R}_c\mathbf{B}_y\mathbf{u}(t)) \right\|_2 \tag{A.10}$$

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_2\mathbf{U} + \mathbf{g}_2\|_2 \tag{A.11}$$

where $\mathbf{F}_2 = \mathbf{T}_{trsf}\mathbf{R}_c\mathbf{B}_y$ and $\mathbf{g}_2 = \mathbf{T}_{trsf}\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t)$. This minimization problem can be cast as a quadratic program with linear constraints:

$$\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^T[2\mathbf{F}_2^T\mathbf{F}_2]\mathbf{U} + [2\mathbf{F}_2^T\mathbf{g}_2]^T\mathbf{U} \tag{A.12}$$

such that the queue lengths are upper and lower bounded, where:

$$0 \leq \mathbf{Y}_s \leq \mathbf{c}_2 \tag{A.13}$$

Hence,

$$-\mathbf{g}_2 \leq \mathbf{F}_2\mathbf{U} \leq \mathbf{c}_2 - \mathbf{g}_2 \tag{A.14}$$

Similar to Section 8.2.2.2, the linear constraints can be expressed in matrix form, where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{F}_2 \\ -\mathbf{F}_2 \end{bmatrix} \tag{A.15}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 - \mathbf{g}_2 \\ -\mathbf{g}_2 \end{bmatrix} \tag{A.16}$$

$$\mathbf{A}_{constraint}\mathbf{U} \leq \mathbf{b}_{constraint} \tag{A.17}$$

and

$$\mathbf{A}_{eq}\mathbf{U} = \mathbf{b}_{eq} \tag{A.18}$$

$l_{1,\infty}$ Norm - 1-norm with respect to time and ∞-norm with respect to space

The cost function based on the $l_{1,\infty}$ norm is given by:

$$J_{1,\infty} = \sum_{t}^{N} max_i |y_{c_i}(t)| \tag{A.19}$$

Let:

$$\mathbf{Y}_s = \begin{bmatrix} max_i\{y_{c_i}(t)\} \\ max_i\{y_{c_i}(t+1)\} \\ \vdots \\ max_i\{y_{c_i}(t+N)\} \end{bmatrix} \tag{A.20}$$

Hence,

$$J_{1,\infty} = \sum_t \mathbf{Y}_s \tag{A.21}$$

$$= \|\mathbf{Y}_s\|_1 \tag{A.22}$$

$$= \left\|(\mathbf{Q}_c\mathbf{A}_{y_{max}}\mathbf{x}_c(t)_{max} + \mathbf{R}_c\mathbf{B}_{y_{max}}\mathbf{u}(t)_{max})\right\|_1 \tag{A.23}$$

$$= \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\|_1 \tag{A.24}$$

where $\mathbf{A}_{y_{max}}$ correspond to the matrix $\mathbf{A}_y$ for arms with maximum queue lengths and $\mathbf{x}_c(t)_{max}$ correspond to the states for arms with maximum queue lengths. Similarly, $\mathbf{B}_{y_{max}}$ correspond to the matrix $\mathbf{B}_y$ for arms with maximum queue lengths and $\mathbf{u}(t)_{max}$ correspond to the input $\mathbf{u}(t)$ for arms with maximum queue lengths. Moreover, $\mathbf{F}_1 = \mathbf{Q}_c\mathbf{A}_{y_{max}}\mathbf{x}_c(t)_{max}$ and $\mathbf{g}_1 = \mathbf{R}_c\mathbf{B}_{y_{max}}\mathbf{u}(t)_{max}$.

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\|_{1,1} \tag{A.25}$$

where

$$0 \leq \mathbf{F}_1\mathbf{U} + \mathbf{g}_1 \leq \boldsymbol{\Upsilon} \tag{A.26}$$

This minimization problem can be cast as a linear program with linear constraints:

$$\min_{\mathbf{U},\boldsymbol{\Upsilon}} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} \tag{A.27}$$

The constraints for the queue lengths and green times still apply. Such constraints can be expressed in matrix form, similar to Section 8.2.2.1 where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 & \mathbf{0} \\ \mathbf{R}_c\mathbf{B}_y & -\mathbf{I} \\ -\mathbf{R}_c\mathbf{B}_y & -\mathbf{I} \\ \mathbf{0} & \mathbf{\Omega}_2 \end{bmatrix} \tag{A.28}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ -\mathbf{g}_1 \\ \mathbf{g}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{A.29}$$

$$\mathbf{A}_{constraint} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} \leq \mathbf{b}_{constraint} \tag{A.30}$$

and

$$\mathbf{A}_{eq} \begin{bmatrix} \mathbf{U} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \mathbf{b}_{eq} \tag{A.31}$$

$l_{\infty,1}$ Norm - ∞-norm with respect to time and 1-norm with respect to space

The cost function based on the $l_{\infty,1}$ norm is given by:

$$J_{\infty,1} = max_t \sum_i^n |y_{c_i}(t)| \tag{A.32}$$

Let:

$$\mathbf{Y}_s = \sum_i^n \|y_{c_i}(t)\|_1 \tag{A.33}$$

$$= \mathbf{T}_{trsf}\mathbf{Y} \tag{A.34}$$

where $\mathbf{T}_{trsf}$ is similarly defined by Equation (A.8).

Hence,

$$J_{\infty,1} = \|\mathbf{Y}_s\|_\infty \tag{A.35}$$
$$= \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\|_\infty \tag{A.36}$$

where $\mathbf{F}_1 = \mathbf{T}_{trsf}\mathrm{R}_c\mathbf{B}_y$ and $\mathbf{g}_1 = \mathbf{T}_{trsf}\mathrm{Q}_c\mathbf{A}_y\mathbf{x}_c(t)$, such that:

$$0 \leq max_t \|\mathbf{F}_1\mathbf{U} + \mathbf{g}_1\| \leq \Upsilon \tag{A.37}$$

This minimization problem can be cast as a linear program with linear constraints:

$$\min_{\mathbf{U},\Upsilon} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} \tag{A.38}$$

The same constraints for the queue length and green time as expressed in Section 8.2.2.1 still apply. Such constraints can be expressed in matrix form, where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \boldsymbol{\Omega}_1 & 0 \\ \mathbf{T}_{trsf}\mathbf{R}_c\mathbf{B}_y & -I \\ -\mathbf{T}_{trsf}\mathbf{R}_c\mathbf{B}_y & 0 \\ 0 & \boldsymbol{\Omega}_2 \end{bmatrix} \tag{A.39}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ -\mathbf{g}_1 \\ \mathbf{g}_1 \\ \mathbf{c}_2 \end{bmatrix} \tag{A.40}$$

$$\mathbf{A}_{constraint} \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} \leq \mathbf{b}_{constraint} \tag{A.41}$$

and

$$\mathbf{A}_{eq} \begin{bmatrix} \mathbf{U} \\ \Upsilon \end{bmatrix} = \mathbf{b}_{eq} \tag{A.42}$$

$l_{2,\infty}$ Norm - 2-norm with respect to time and ∞-norm with respect to space

The cost function based on the $l_{2,\infty}$ norm is given by:

$$J_{2,\infty} = \sqrt{(\sum_{t}^{N}(max_i|y_{c_i}(t)|)^2)} \tag{A.43}$$

For simplification, the square root in (A.43) is removed. Note that by removing the square root function, the minimum value is not changed. Hence, let:

$$J_{2,\infty} = \sum_{t}^{N}(max_i|y_{c_i}(t)|)^2 \tag{A.44}$$

and let:

$$\mathbf{Y}_s = \begin{bmatrix} max_i\{y_{c_i}(t+1)\} \\ max_i\{y_{c_i}(t+2)\} \\ \vdots \\ max_i\{y_{c_i}(t+N)\} \end{bmatrix} \tag{A.45}$$

Hence,

$$J_{2,\infty} = \|\mathbf{Y}_s\|_2 \tag{A.46}$$
$$= \left\|(\mathbf{Q}_c\mathbf{A}_{y_{max}}\mathbf{x}_c(t)_{max} + \mathbf{R}_c\mathbf{B}_{y_{max}}\mathbf{u}(t)_{max})\right\|_2 \tag{A.47}$$
$$= \|\mathbf{F}_2\mathbf{U} + \mathbf{g}_2\|_2 \tag{A.48}$$

where $\mathbf{A}_{y_{max}}$ correspond to the matrix $\mathbf{A}_y$ for arms with maximum queue lengths and $\mathbf{x}_c(t)_{max}$ correspond to the states for arms with maximum queue lengths. Similarly, $\mathbf{B}_{y_{max}}$ correspond to the matrix $\mathbf{B}_y$ for arms with maximum queue lengths and $\mathbf{u}(t)_{max}$ correspond to the input $\mathbf{u}(t)$ for arms with maximum queue lengths. Moreover, $\mathbf{F}_1 = \mathbf{Q}_c\mathbf{A}_{y_{max}}\mathbf{x}_c(t)_{max}$ and $\mathbf{g}_1 = \mathbf{R}_c\mathbf{B}_{y_{max}}\mathbf{u}(t)_{max}$.

The optimization problem leads to:

$$\min_{\mathbf{U}} \|\mathbf{F}_2\mathbf{U} + \mathbf{g}_2\|_2 \tag{A.49}$$

where $\mathbf{F}_2$=$\mathbf{R}_c\mathbf{B}_y$ and $\mathbf{g}_2$=$\mathbf{Q}_c\mathbf{A}_y\mathbf{x}_c(t)$. This minimization problem can be cast as a quadratic program with linear constraints:

$$\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^T[2\mathbf{F}_2^T\mathbf{F}_2]\mathbf{U} + [2\mathbf{F}_2^T\mathbf{g}_2]^T\mathbf{U} \tag{A.50}$$

such that the queue lengths are upper and lower bounded, where:

$$0 \leq \mathbf{Y}_s \leq \mathbf{c}_2 \tag{A.51}$$

Hence,

$$-\mathbf{g}_2 \leq \mathbf{F}_2\mathbf{U} \leq \mathbf{c}_2 - \mathbf{g}_2 \tag{A.52}$$

Moreover, the constraints for the green time as discussed in the previous sections still apply. Similar to Section 8.2.2.2, such constraints can be expressed in matrix form, where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{F}_2 \\ -\mathbf{F}_2 \end{bmatrix} \tag{A.53}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 - \mathbf{g}_2 \\ -\mathbf{g}_2 \end{bmatrix} \tag{A.54}$$

$$\mathbf{A}_{constraint}\mathbf{U} \leq \mathbf{b}_{constraint} \tag{A.55}$$

and

$$\mathbf{A}_{eq}\mathbf{U} = \mathbf{b}_{eq} \tag{A.56}$$

$l_{\infty,2}$ Norm - ∞-norm with respect to time and 2-norm with respect to space

The cost function based on the $l_{\infty,2}$ norm is given by:

$$J_{\infty,2} = max_t \sqrt{\sum_i^n (|y_{c_i}(t)|)^2} \tag{A.57}$$

For simplification, the square root in (A.57) is removed. Note that by removing the square root function, the minimum value is not changed. Hence, let:

$$J_{\infty,2} = max_t (\sum_i^n (|y_{c_i}(t)|)^2) \tag{A.58}$$

$$= max_t \begin{bmatrix} \|\mathbf{y}_c(t+1)\|_2 \\ \|\mathbf{y}_c(t+2)\|_2 \\ \vdots \\ \|\mathbf{y}_c(t+N)\|_2 \end{bmatrix} \tag{A.59}$$

The optimization problem leads to:

$$\min_{\mathbf{U}} \{\|\mathbf{F}_2\mathbf{U} + \mathbf{g}_2\|_2\} \tag{A.60}$$

where

- if $J_{\infty,2} = \{\|\mathbf{y}_c(t+1)\|_2\}$, $\mathbf{F}_2 = \mathbf{R}_c[\mathbf{C}_c\mathbf{B}_c \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]$ and $\mathbf{g}_2 = \mathbf{Q}_c\mathbf{C}_c\mathbf{A}_c\mathbf{x}_c(t)$.
- if $J_{\infty,2} = \{\|\mathbf{y}_c(t+2)\|_2\}$, $\mathbf{F}_2 = \mathbf{R}_c[\mathbf{C}_c\mathbf{A}_c\mathbf{B}_c \quad \mathbf{C}_c\mathbf{B}_c \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]$ and $\mathbf{g}_2 = \mathbf{Q}_c\mathbf{C}_c\mathbf{A}_c^2\mathbf{x}_c(t)$.
- if $J_{\infty,2} = \{\|\mathbf{y}_c(t+3)\|_2\}$, $\mathbf{F}_2 = \mathbf{R}_c[\mathbf{C}_c\mathbf{A}_c^2\mathbf{B}_c \quad \mathbf{C}_c\mathbf{A}_c\mathbf{B}_c \quad \mathbf{C}_c\mathbf{B}_c \quad \mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]$ and $\mathbf{g}_2 = \mathbf{Q}_c\mathbf{C}_c\mathbf{A}_c^3\mathbf{x}_c(t)$.

This minimization problem can be cast as a quadratic program with linear constraints:

$$\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^T[2\mathbf{F}_2^T\mathbf{F}_2]\mathbf{U} + [2\mathbf{F}_2^T\mathbf{g}_2]^T\mathbf{U} \tag{A.61}$$

such that the queue lengths are upper and lower bounded, where:

$$0 \leq \mathbf{Y} \leq \mathbf{c}_2 \tag{A.62}$$

Hence,

$$-\mathbf{g}_2 \leq \mathbf{F}_2\mathbf{U} \leq \mathbf{c}_2 - \mathbf{g}_2 \tag{A.63}$$

Moreover, the constraints for the green time as expressed in Section 8.2.2.2 still apply. Such constraints can be expressed in matrix form, where:

$$\mathbf{A}_{constraint} = \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{F}_2 \\ -\mathbf{F}_2 \end{bmatrix} \tag{A.64}$$

$$\mathbf{b}_{constraint} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 - \mathbf{g}_2 \\ -\mathbf{g}_2 \end{bmatrix} \tag{A.65}$$

$$\mathbf{A}_{constraint}\mathbf{U} \leq \mathbf{b}_{constraint} \tag{A.66}$$

and

$$\mathbf{A}_{eq}\mathbf{U} = \mathbf{b}_{eq} \tag{A.67}$$

Bibliography

[1] C. Cai. *Adaptive Traffic Signal Control Using Approximate Dynamic Programming*. PhD thesis, University College of London, 2009a.

[2] Y. Li, X. Guo, J. Yang, Y. Liu, and S. He. Multi-Objective Optimization of Traffic Signal Timing for Oversaturated Intersection. *Journal of Mathematical Problems in Engineering, 2013:1-9*, 2013.

[3] T. Tettamanti. *Advanced Methods for Measurement and Control in Urban Road Traffic Networks*. PhD thesis, Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, 2013.

[4] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C, 17:2, 163-174*, 2009.

[5] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C, 24, 122-140*, 2012.

[6] P. Pecherkova, J. Dunik, and M. Flidr. *Automation and Control*, chapter Modelling and Simultaneous Estimation of State and Parameters of Traffic System. InTech, 2008.

[7] P. Pecherkova and J. Dunik. Modelling of traffic system with time-variant saturation flow. *9th International PhD Workshop on System and Control: Young Generation Veiwpoint*, 2008.

[8] D. Magri. Autonomic Control of Traffic Networks, Dissertation submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering (Hons.), University of Malta, Department of Systems and Control Engineering, 2013.

[9] M. Papageorgiou. Overview of road and motorway traffic control strategies. *IFAC Proceedings Volumes, 37:19*, 2004.

[10] European Commission. Roadmap to a Single European Transport Area - Towards a Competitive and Resource-Efficient Transport System. Technical report, Directorate-General for Mobility and Transport, 2011.

[11] European Union. Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. Technical report, European Union, 2010.

[12] CIVITAS: Cleaner and better transport in cities 2020. Intelligent Transport Systems and traffic management in urban areas. *Available online at: https://civitas.eu/content/civitas-policy-note-intelligent-transport-systems-and-traffic-management-urban-areas-0 [Last accessed 28/01/2019]*.

[13] D. I. Robertson. TRANSYT: a traffic network study tool Report LR253. Technical report, Road Research Laboratory, Ministry of Transport, Crowthorne, Berkshire, U.K., 1969.

[14] R. A. Vincent, A. I. Mitchell, and D. I. Robertson. User guide to TRANSYT version 8, Report LR888. Technical report, Transport and Road Research Laboratory, Crowthorne, Berkshire, U.K., 1980.

[15] P. B. Hunt, D. I. Robertson, and R. D. Bretherton. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering and Control, 23, 190-192*, 1982.

[16] J. Y. K. Luk. Two traffic-responsive area traffic control methods: SCAT and SCOOT. *Traffic Engineering and Control, 25, 14-22*, 1984.

[17] N. H. Gartner. Demand-responsive Decentralized Urban Traffic Control, Part I: Single-Intersection Policies. Technical report, U.S. Department of Transportation, 1982.

[18] N. H. Gartner. Demand-responsive Decentralized Urban Traffic Control, Part 2: Network Extensions. Technical report, U.S. Department of Transportation, 1983a.

[19] N. H. Gartner. OPAC: A demand-responsive strategy for traffic signal control. Technical report, Transportation Research Record 906, 75-81, 1983.

[20] J. J. Henry, J. L. Farges, and J. Tuffal. The PRODYN real time traffic algorithm. In *Proceedings of the forth IFAC-IFIP-IFORS Conference on Control in Transportation Systems, 307-311*, 1983.

[21] V. Mauro and C. Di Taranto. UPTOPIA, CCCT'89. In *6th IFAC-IFIP-IFORS Conference on Control, Computers and Communication in Transport, Paris, September*, 1989.

[22] P. Mirchandani and L. Head. RHODES: a real-time traffic signal control system: architecture, algorithms and analysis. *Transportation Research C, 9:6, 415-432*, 2001.

[23] J. Falilat, T. L. McCluskey, L. Chrpa, and P. Gregory. Enabling Autonomic Properties in Road Transport System. *30th Workshop of the UK Planning And Scheduling Special Interest Group PLANSIG 2012*, 2012.

[24] L. Chrpa, D. Magazzeni, K. McCabe, T. L. McCluskey, and M. Vallati. Automated Planning for Urban Traffic Control: Strategic Vehicle Routing to Respect Air Quality Limitations. *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2013), October, 6-9, The Hague Netherlands*, 2013.

[25] H. Etemadnia, K. Abdelghany, and S. Hariri. Toward an Automatic Architecture for Real-Time Traffic Network Management. *Journal of In-*

telligent Transportation Systems: Technology, Planning and Operations, 16:2, 45-59, 2012.

[26] K. Stoilova, T. Stoilov, and H. Abouaissa. Traffic Lights Optimization with Measurements of Noise Levels. *1st IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, 2013.

[27] A. Tizghadam and A. Leon-Garcia. Autonomic traffic engineering for network robustness. *IEEE Journal on Selected Areas in Communications, 28:1, 39-50*, 2010.

[28] T. L. McCluskey. A New COST Action: Autonomic Road Transport Support (ARTS) Systems. *Quarterly, The Newsletter of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, 4-5*, 2011.

[29] K. A. Ullas and S. Nayak. Smart Traffic Signals using Autonomic Computing. *IJIRST - International Journal for Innovative Research in Science & Technology, 1:1*, 2014.

[30] A. Kotsialos and A. Poole. Autonomic Systems Design for ITS Applications. *Proceedings of the 16th International IEEE Annual Conference on-Intelligent Transportation Systems (ITSC 2013), The Hague, TheNetherlands, October 6-9*, 2013.

[31] R. Sterritt and M. Hinchey. Towards Self-Managing Real-Time Systems. *Proceedings of the International Multiconference on Computer Science and Information Technology, 751-756, October 18-20, Wisla, Poland*, 2008.

[32] T. L. McCluskey and A. Kotsialos and J. P. Muller and F. Klugl and O. Rana and R. Schumann. *Autonomic Road Transport Support Systems*. Birkhauser, Cham, 2016.

[33] S. Goel, S. F. Bush, and C. Gershenson. Self-Organization in Traffic Lights: Evolution of Signal Control with Advances in Sensors and Communication. *Cornell University, 1708.07188, Available online at: https://arxiv.org/abs/1708.07188 [Last accessed 28/01/2019]*, 2017.

[34] C. Gershenson. Self-Organizing Traffic Lights. *Complex Syst. 16, 29-53*, 2005.

[35] C. Gershenson and D. A. Rosenblueth. Self-Organizing Traffic Lights at Multiple-Street Intersections. *Complexity, 17,4*, 2012.

[36] B. Cesme and P. G. Furth. Self-organizing traffic signals using secondary extension and dynamic coordination . *Transportation Research Part C,48,1-15*, 2014.

[37] S. Tomforde, H. Prothmann, F. Rochner, J. Branke, J. Hahner, C. Muller-Schloer, and H. Schmeck. Decentralised Progressive Signal Systems for Organic Traffic Control. *Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, October 20-24, Venezia, Italy*, 2008.

[38] H. Prothmann, J. Branke, and H. Schmeck. Organic traffic light control for urban road networks. *Int. J. Autonomous and Adaptive Communications Systems, 2:3*, 2009.

[39] J. Kephart and D. Chess. The vision of autonomic computing. *IEEE Computer Society, 36:1, 41-50*, 2003.

[40] H. Schmeck. Organic Computing - a new vision for distributed embedded systems. *Proceedings of the 8th International Symposium on Object-Oriented Real-time Distributed Computing (ISORC'05), 201-203*, 2005.

[41] H. Prothmann. *Organic Traffic Control*. KIT Scientific Publishing, 2011.

[42] S. B. Cools, C. Gershenson, and B. D'Hooghe. Self-organizing Systems. *Advanced Information and Knowledge Processing, Springer, London, 41-50*, 2008.

[43] C. Gershenson and D. A. Rosenblueth. Self-organizing traffic lights at multiple-street intersections. *Complexity, 17:4, 23-39*, 2012.

[44] B. Placzek. A self-organizing system for urban traffic control based on predictive interval microscopic model. *Engineering Applications of Artificial Intelligence 34, 75-84*, 2014.

[45] D. de Oliveira and A. L. C. Bazzan. Emergence of Traffic Lights Synchronization. *Proceedings 20th European Conference on Modeling and Simulation, Bonn, May 28-31*, 2006.

[46] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis. Multiagent Reinforcement Learning for Urban Traffic Control using Coordination Graphs. *ECML 2008: Proceedings of the Nineteenth European Conference on Machine Learning, 656-671, September 15-19, Antwerp, Belgium*, 2008.

[47] S. Shamshirband. A Distributed Approach for Coordination Between Traffic Lights Based on Game Theory. *The International Arab Journal of Information Technology, 9:2*, 2012.

[48] J. C. Medina and R. F. Benekohal. Agent-based Traffic Management and Reinforcement Learning in Congested Intersection Network. Technical report, University of Illinois at Urbana-Champaign, 2012.

[49] F. Chen, L. Xiang, W. Lan, and G. Chen. Coordinated Tracking in Mean Square for a Multi-Agent System With Noisy Channels and Switching Directed Network Topologies. *IEEE Transactions on Circuits and Systems - II: Express Briefs, 59:11*, 2012.

[50] D. Oliveira, J. R. Ferreira, A. L. C. Bazzan, and F. Kluegi. A Swarm-based Approach for Selection of Signal Plans in Urban Scenarios. *IV International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS), Brussels*, 2004.

[51] S. G. Fabri and V. Kadirkamanathan. *Functional Adaptive Control An Intelligent Systems Approach*. Springer-Verlag London, 2001.

[52] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, LLC, 2009.

[53] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. In *Proceedings of the IEEE, 91(12), 2043-2067*, 2003.

[54] D. Manolis, T. Pappa, C. Diakaki, I. Papamichail, and M. Papageorgiou. Simulation study of centralized vs decentralized approaches to the signal control of large-scale urban networks. *Proceedings of the 96th Annual Meeting of the Transportation Research Board (TRB) at Washington D.C., U.S.A.*, 2017.

[55] S. P. Hoogendoorn and V. L. Knoop. *The Transport System and Transport Policy*, chapter Traffic flow theory and modelling. Edward Elgar Publishing Limited, 2012.

[56] F. L. Hall. Traffic stream characteristics. In *Traffic Flow Theory: A State-of-the-Art Report*. US Department of Transportation, Turner-Fairbank Highway Research Center, 2001.

[57] B. S. Kerner. *Introduction to Modern Traffic Flow Theory and Control*. Springer-Verlag Berlin Heidelberg, 2009.

[58] A. Spiliopoulou, M. Kontorinaki, M. Papageorgiou, and P. Kopelias. Macroscopic traffic flow model validation at congested freeway off-ramp areas. *Transportation Research Part C, 41:18-29*, 2014.

[59] S. P. Hoogendoorn and P. H. L. Bovy. State-of-the-art of Vehicular Traffic Flow Modelling. In *Proceedings of the I MECH E Part I Journal of Systems & Control Engineering, 215:4, 283-303*, 2001.

[60] Dunn Engineering Associates and Siemens Intelligent Transporation Systems. Traffic Control Systems Handbook. Technical report, Federal Highway Administration, 2005.

[61] European Commission. Understanding and Monitoring the Cost-Determining Factors of Infrastructure Projects. *Directorate-General for Regional Policy and Cohesion, Available Online:*

https://ec.europa.eu/regional_policy/sources/docgener/evaluation /pdf/5_full_en.pdf, [Last accessed: 28/01/2019], 1998.

[62] S. Hancock. City Congestion: Why congestion charges alone are only a short term fix. *MobileDOCK White Paper, Available Online: http://www.mobiledock.com/wp-content/uploads/2017/07/White-Paper-City-Congestion-Why-congestion-charges-are-only-a-short-term-fix-for-congestion.pdf, [Last accessed: 28/01/2019]*, 2017.

[63] K. Scerri. *A Systems Approach to Spatio-Temporal Modelling*. PhD thesis, University of Sheffield, 2009.

[64] A. P. Akgungor, A. Bullen, and R. Graham. A New Delay Parameter for Variable Traffic Flows at Signalized Intersections. *Turkish Journal of Engineering & Environmental Sciences, 31(1), 61*, 2007.

[65] M. S. Chaudhry and P. Ranjitkar. Capacity and Signal Timing Analysis of Signalized Intersections with increasing saturation flow rate. Technical report, TRB 2013 Annual Meeting, January 13-17, Washington D.C., 2013.

[66] R. Mukwaya and G. Mwesige. Saturation Flow Rate for Through-Traffic at Signalized Junctions in Kampala. In *Second International Conference on Advances in Engineering and Technology, October 12-13, Sangrur, Punjab, India*, 2011.

[67] D. Gazis, R. Herman, and A. Maradudin. The Problem of the Amber Signal Light in Traffic Flow. *Operations Research, 8(1), 112-132*, 1960.

[68] C. Liu, R. Herman, and D. Gazis. A review of the yellow interval dilemma. *Transport Research Part A, 30(5), 333-348*, 1996.

[69] B. Peter, L. Ronghui, and Y. William. Modelling safety-related driving behaviour - impact of parameter values. *Transportation Research Part A, 39, 425-444*, 2005.

[70] P. Papaioannou. Driver behaviour, dilemma zone and safety effects at urban signalised intersections in Greece. *Accident Analysis and Prevention, 39, 147-158*, 2007.

[71] E. Kometani and T. Sasaki. A safety index for traffic with linear spacing. *Operations Research, 7:6, 704-720*, 1959.

[72] W. Helly. Simulation of bottlenecks in single lane traffic flow. In *Proceedings of the Symposium on Theory of Traffic Flow, Research Laboratories, General Motors, 207-238*, 1959.

[73] C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B, 28B(4), 269-287*, 1994.

[74] C. F. Daganzo. The cell transmission model, part II: Network traffic. *Transportation Research Part B, 29, 79-93*, 1995.

[75] G. C. Gazis and R. B. Potts. The oversaturated intersection. *Proceedings of the Second International Symposium on Traffic Theory, London, UK, 221-237*, 1963.

[76] G. C. D'Ans and D. C. Gazis. Optimal Control of Oversaturated Store-and-Forward Transportation Networks. *Transportation Science,10(1), 1-19*, 1976.

[77] C. Diakaki, M. Papageorgiou, and K. Aboudolas. Traffic-responsive urban network control using multivariable regulators. *Proceedings of the International Conference on Modelling and Management in Transportation, October 12-16, Poznan/Cracow, Poland, 2, 11-16*, 1999.

[78] C. Diakaki, V. Dinopoulou, K. Aboudolas, M. Papageorgiou, E. Ben-Shabat, E. Seider, and A. Leibov. Extensions and New Applications of the Traffic Signal Control Strategy TUC. *82nd Annual Meeting of the Transportation Research Board, January 12-16, Washington D.C.*, 2003.

[79] T. Tettamanti, I. Varga, and T. Peni. *Model Predictive Control*, chapter MPC in Urban Traffic Management. IntechOpen, 2010.

[80] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. Integrated traffic control for mixed urban and freeway networks: A model predictive control approach. *European Journal of Transport and Infrastructure Research, 7:3, 223-250*, 2007.

[81] J. Kratochvilova and I. Nagy. Local traffic control of a microregion. *Ministry of Transportation, Czech Republic, National Programme of Research 2004-2009, Projectnum. 1F43A/003/120*, 2009.

[82] J. Homolova. Traffic Flow Control. *Process Control' 05*, 2005.

[83] J. Homolova and I. Nagy. Traffic Model of a Microregion. *Ministry of Transportation, Czech Republic, National Programme of Research 2004-2009, Projectnum. 1F43A/003/120*, 2005.

[84] J. Dunik, P. Pecherkova, and M. Flidr. State space model of traffic system and its estimation using derivative-free methods. *International Journal of Adaptive Control and Signal Processing*, 2006.

[85] P. Dohnal. State Space Model of a Traffic Microregion. *Proceedings of the 10th International Student Conference on Electrical Engineering - Poster 2006, May 18, Prague*, 2006.

[86] V. Dinopoulou, C. Diakaki, and M. Papageorgiou. Application and Evaluation of the Signal Traffic Control Strategy TUC in Chania. *Journal of Intelligent Transportation Systems 9:3:133-143*, 2005.

[87] 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, 229, 317-345*, 1955.

[88] M. Kontorinaki, A. Spiliopoulou, C. Roncolic, and M. Papageorgiou. First-order traffic flow models incorporating capacity drop: Overview and real-data validation. *Transportation Research Part B 106, 52-75*, 2017.

[89] M. Treiber and A. Kesting. *Traffic Flow Dynamics: Data, Models and Simulation*. Springer-Verlag Berlin Heidelberg, 2013.

[90] C. Diakaki, M. Papageorgiou, and K. Aboudolas. Traffic-Responsive Urban Network Control Using Multivariable Regulators. *The Archives of Transport, XII:4*, 2000.

[91] L. Kleinrock. *Queuing Systems*, chapter Theory, pages 3–8. Wiley, New York, 1975.

[92] N. Vandaele, T. Van Woensel, and A. Verbruggen. A queueing based traffic flow model. *Transportation Research Part D: Transport and Environment, 5:2, 121-135*, 2000.

[93] D. Heidemann and H. Wegmann. Queueing at unsignalized intersections. *Transportation Research Part B: Methodological, 31(3):239-263*, 1997.

[94] D. Heidemann. Queue length and waiting-time distributions at priority intersections. *Transportation Research Part B: Methodological, 25(4):163-174*, 1991.

[95] A. Hegyi. Model predictive control for integrating traffic control measures. *TRAIL thesis series. Netherlands TRAIL Research School*, 2004.

[96] O. Cappe. *Mixtures: Estimation and Applications*, chapter Online Expectation-Maximisation, pages 1–53. John Wiley & Sons, Ltd, Chichester, 2011.

[97] P. Liang and D. Klein. Online EM for Unsupervised Models. *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), June 2-7, Minneapolis, Minnesota*, 2009.

[98] R. E. Kopp and R. J. Orford. Linear regression applied to system identification for adaptive control systems. *AIAA, 1:10, 2300-2306*, 1963.

[99] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo. An adaptive freeway traffic state estimator. *Automatica 45, 10-24*, 2009.

[100] G. C. Goodwin and K. S. Sin. *Adaptive Filtering Prediction and Control*. Prentice-Hall, USA, 1984.

[101] S. F. Schmidt. Compensation for modeling errors inorbit determination problems. *Analytical Mechanics Ass., Westbury, NY, 67-16*, 1967.

[102] S. R. Neal. Linear estimation in the presence of errors in assumed plant dynamics. *IEEE Transaction on Automatic Control, AC-IZ, 582-594*, 1967.

[103] D. T. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transaction on Automatatic Control, AC:10, 434-439*, 1965.

[104] C. G. Hilborn and D. G. Lainiotis. Optimal estimation in the presence of unknown parameters. *IEEE Trans. Syst., Sci., Cybern., SSC:5, 38-43*, 1969.

[105] J. A. Quinn, C. K. I. Williams, and N. McIntosh. Factorial Switching Linear Dynamical Systems Applied to Physiological Condition Monitoring. *IEEE Transaction of Pattern Analysis and Machine Intelligence, 31:9, 1537-1551*, 2009.

[106] E. A. Wan and A. T. Nelson. Dual Kalman Filtering Methods for Nonlinear Prediction, Smoothing, and Estimation. *In Advances in Neural Information Processing Systems 9*, 1997.

[107] A. UmaMageswari, J. J. Ignatious, and R. Vinodha. A Comparitive Study Of Kalman Filter, Extended Kalman Filter And Unscented Kalman Filter For Harmonic Analysis Of The Non-Stationary Signals. *International Journal of Scientific & Engineering Research, 3:7, 1-9*, 2012.

[108] O. Cappe. Online Sequential Monte Carlo EM Algorithm. *IEEE/SP 15th Workshop on Statistical Signal Processing, 37-40, Cardiff University and City Hall Cardiff, United Kingdom*, 2009.

[109] R. Wang and D. B. Work. *Joint parameter and state estimation algorithms for real-time traffic monitoring*. PhD thesis, USDOT Region V Regional University Transportation Center, University of Illinois, Urbana Champaign, 2013.

[110] L. W. Nelson and E. Stear. The Simultaneous On-Line Estimation of Parameter and States in Linear Systems. *IEEE Transactions on Automatic Control, 21:1, 94-98*, 1976.

[111] Y. Bar-Shalom. Optimal Simultaneous State Estimation and Parameter Identification in Linear Discrete-Time Systems. *IEEE Transactions on Automatic Control, 17:3, 308-319*, 1972.

[112] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, 2008.

[113] B. Ninnes and S. Gibson. Robust and Simple Algorithms for Maximum Likelihood Estimation of multivariable. Technical report, Dept. of Elec. & Comp. Eng, Uni. Newcastle, Australia, 2002.

[114] A. Dempster, N. M. Laird, and D. B. Rubin. ML from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39, 1-38*, 1977.

[115] D. Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society, Series B 46:2, 257-267*, 1984.

[116] S. Chen. The Application of the Expectation-Maximization Algorithm to the Identification of Biological Models. Master's thesis, Faculty of the Virginia Polytechnic Institute and State University, 2006.

[117] J. L. Starck, F. Murtagh, and A. Bijaoui. *Image processing and data analysis*. Cambridge University Press, 1998.

[118] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77:2, 257-285*, 1989.

[119] O. E. Barndorff-Nielsen, D. R. Cox, and C. Kluppelberg. *Complex Stochastic Systems*. Chapman & Hall/CRC, Boca Raton, London, New York, Washington D.C., 1999.

[120] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics, 22:3, 400-407*, 1951.

[121] J. R. Blum. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics, 25:4, 737-744*, 1954.

[122] J. C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control, 37:3, 332-341*, 1992.

[123] J. C. Spall. Adaptive Stochastic Approximation by the Simultaneous Perturbation Method. *IEEE Transactions on Automatic Control, 45:10, 1839-1853*, 2000.

[124] M. U. Altaf, A. W. Heemink, M. Verlaan, and I. Hoteit. Simultaneous perturbation stochastic approximation for tidal models. *Ocean Dynamics, 61:8, 1093-1105*, 2011.

[125] Yue and Christopher J. C. Burges. On using simultaneous perturbation stochastic approximation for learning to rank , and the empirical optimality of lambdarank. In *Microsoft Research Technical Report MST-TR-2007-115*, 2007.

[126] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C, 17, 163-174*, 2009.

[127] D. Simon and D. L. Simon. Aircraft Turbofan Engine Health Estimation Using Constrained Kalman Filtering. *Proceedings of the ASME Turbo Expo, 1-8, Atlanta, USA, June 16-19*, 2003.

[128] N. Santitissadeekorn and C. Jones. Two-stage filtering for joint state-parameter estimation. *Journal of the Monthly Weather Review, 143:6, 2028-2042*, 2015.

[129] H. E. Soken and S. Sakai. A multiple model adaptive estimation algorithm for systems with parameter change. *Proceeding of the 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Japan, September 20-23*, 2016.

[130] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi. Multi-Model Adaptive Switching Control with Fine Controller Tuning. *Proceedings of the 18th IFAC World Congress, Italy, August 28 - September 2*, 2011.

[131] S. Baldi, G. Battistelli, E. Mosca, and P. Tesi. Multi-model unfalsified adaptive switching control: Test functionals for stability and performance. *International Journal of Adaptive Control Signal Process, 25, 593-612*, 2011.

[132] J. P. Hespanha, D. Liberzon, and A. S. Morse. Hysteresis-based switching algorithms for supervisory control of uncertain systems. *Automatica, 39, 263-272*, 2003.

[133] D. Liberzon. *Switching in Systems and Control*. Springer Science + Business Media Inc., USA, 2003.

[134] A. S. Willsky, E. Y. Chow, S. B. Gershwin, C. S. Greene, P. K. Houpt, and A. L. Kurkjian. Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control, 25:3, 347-360*, 1980.

[135] R. Wang and D. B. Work. Interactive multiple model ensemble Kalman filter for traffic estimation and incident detection. *Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), China, October 8-11*, 2014.

[136] R. Wang, D. B. Work, and R. Sowers. Multiple Model Particle Filter for Traffic Estimation and Incident Detection. *IEEE Transactions on Intelligent Transportation Systems, 17:12, 3461-3470*, 2016.

[137] R. Wang. *Joint parameter and state estimation algorithms for real-time traffic monitoring*. PhD thesis, University of Illinois at Urbana Champaign, 2013.

[138] R. Wang, S. Fan, and D. B. Work. Efficient multiple model particle filtering for joint traffic state estimation and incident detection. *Transportation Research Part C, 71, 521-537*, 2016.

[139] L. Chetcuti Zammit, S. G. Fabri, and K. Scerri. Real-Time Parametric Modeling and Estimation of Urban Traffic Junctions. *IEEE Transactions on Intelligent Transportation Systems, 99:1-11*, 2019.

[140] K. Wood. Urban traffic control, systems review. *Transport and Road Research Laboratory Report PR 41. Crowthorne: TRL.*, 1993.

[141] B. G. Heydecker. Objectives, stimulus and feedback in signal control of road traffic. *Intelligent Transportation Systems, 8, 63-76*, 2004.

[142] Department of Transport. MCE 0141: Microprocessor based traffic signal controller for isolated linked and urban traffic control installations. 1984.

[143] H. J. Van Zuylen. Acyclic traffic controllers. Technical report, Note from the Centre for Transport Studies, UCL, 1976.

[144] F. V. Webster. Traffic Signal Settings. Technical report, Road Research Technical Paper, No. 39, Road Research Laboratory, London, 1957.

[145] R. E. Allsop. Delay-minimising settings for fixed-time traffic signals at a single road junction. *Journal of the Institute of Mathematics and its Applications, 8:2, 164-185*, 1971a.

[146] R. E. Allsop. SIGSET: a computer program for calculating traffic signal settings. *Traffic Engineering and Control, 13, 58-60*, 1971b.

[147] J. D. C. Little. The synchronisation of traffic signals by mixed-integer linear programming. *Operations Research, 14, 568-594*, 1964.

[148] G. Improta and G. E. Cantarella. Control system design for an individual signalized junction. *Transportation Research, 18B (2), 147-167*, 1984.

[149] B. G. Heydecker and I. W. Dudgeon. Calculation of signal settings to minimise delay at a junction. *Proceedings of 10th International Symposium on Transportation and Traffic Theory, MIT, 159-178*, 1987.

[150] P. C. M. Ribeiro. Handling Traffic Fluctuation with Fixed-Time Plans Calculated by TRANSYT. *Traffic Engineering and Control, 35, 365-366*, 1994.

[151] B. L. Smith, W. T. Scherer, A. T. Hauser, and B. B. Park. Data-Driven Methodology for Signal timing plan development: A Computational Approach. *Computer-Aided Civil and Infrastructure Engineering, 17:6, 387-395*, 2002.

[152] Y. Yafeng. Robust optimal traffic signal timing. *Transportation Research Part B, 42, 911-924*, 2008.

[153] Y. K. Wong and W. L. Woon. An iterative approach to enhanced traffic signal optimization. *Expert Systems with Applications, 34, 2885-2890*, 2008.

[154] M. Bell and R. Bretherton. Ageing of fixed-time signal plans. *Proceedings of the Second IEEE Conference on Road Traffic Control*, 1986.

[155] J. Haddad, B. D. Schutter, D. Mahalel, I. Ioslovich, and P. O. Gutman. Optimal Steady-State Control for Isolated Traffic Intersections. *IEEE Transactions on Automatic Control, 55(11), 2612-2617*, 2010.

[156] I. Ioslovich, P. O. Gutman, and M. Borshchevsky. Time-Optimal Traffic Control Sysnthesis for a Signalized Isolated Intersection. *Proceedings of the 9th IEEE International Conference on Control and Automation (ICCA), December 19-21, Santiago, Chile*, 2011.

[157] I. Ioslovich, P. O. Gutman, D. Mahalel, and J. Haddad. Design of Optimal Traffic Flow Control at Intersection with Regard for Queue Length Constraints. *Automation and Remote Control, 72:9, 1833-1840*, 2011b.

[158] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. The mathematical theory of optimal processes. *NY: Wiley-Interscience*, 1962.

[159] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Control and optimization methods for traffic signal control in large-scale congested urban road networks. *Proceedings of the American Control Conference, July 9-13, New York*, 2007.

[160] F. B. Lin. Use of binary choice decision process for adaptive signal control. *Journal of Transportation Engineering, 115:3*, 1989a.

[161] F. B. Lin. and S. Vijayakumar. Adaptive signal control at isolated intersections. *Journal of Transportation Engineering, 114:5*, 1989b.

[162] W. Recker, X. Zheng, and L. Chu. Optimization of Control Parameter for Adaptive Traffic Actuated Signal Control. *Proceedings of the ITS World Congress, November 16-20, New York*, 2008.

[163] A. J. Miller. A computer control system for traffic networks. *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow, London, 200-220*, 1963.

[164] K. L. Bang. Optimal control of isolated traffic signals. *Traffic Engineering and Control 17:7, 288-292*, 1976.

[165] R. A. Vincent and C. P. Young. Self-optimizing traffic signal control using microprocessors - the TRRL MOVA strategy for isolated intersections. *Traffic Engineering Control 27 (7/8), 385-387*, 1986.

[166] F. B. Lin, D. J. Cooke, and S. Vijayakumar. Use of predicted vehicle arrival information for adaptive signal control - an assessment. *Transportation Research Record 1112, 89-98*, 1987.

[167] X. Zheng and W. Recker. An adaptive control algorithm for traffic actuated signals. *Transportation Research Part C, 30, 93-115*, 2013.

[168] R. Bellman and S. Dreyfus. Functional approximations and dynamic programming. *Mathematic Tables and Other Aids to Computation, 13:68, 247-251*, 1959.

[169] D. I. Robertson and R. D. Bertherton. Optimum control of an intersection for any known sequence of vehicular arrivals. *Proceedings of the 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation System, Monte Carlo, September 16-21*, 1974.

[170] R. A. Vincent and J. R. Pierce. MOVA: traffic responsive, self-optimising signal control for isolated intersections. Technical report, Transport and Road Research Laboratory Report RR170, Crowthorne: TRRL, 1988.

[171] H. Liu, K. N. Balke, and W. H. Lin. A reverse causal-effect modelling approach for signal control of an oversaturated intersection. *Transportation Research Part C, 16, 742-754*, 2008.

[172] S. Baldi, I. Michailidis, V. Ntampasi, E. B. Kosmatopoulos, I. Papamichail, and M. Papageorgiou. Simulation-based Synthesis for ApproximatelyOptimal Urban Traffic Light Management. *Proceedings of the American Control Conference, July 1-3, Chicago, USA*, 2015.

[173] M. G. H. Bell, M. P. H. Cowell, and B. G. Heydecker. *Traffic Control Methods*, chapter Traffic-responsive signal control at isolated junctions, pages 273–294. New York: Engineering Foundation, 1990.

[174] C. Cai. An approximate dynamic programming strategy for responsive traffic signal control. *Proceeding of IEEE international Symposium on Approximate Dynamic Programming and Reinforcement Learning, Hawaii, April 1-5*, 2007.

[175] W. W. Recker, B. V. Ramanathan, X. H. Yu, and M. G. McNally. Markovian Real-Time Adaptive Control of Signal Systems. *Mathl. Comput. Modelling, 22:4-7), 355-375.*, 1995.

[176] X. H. Yu and W. W. Recker. Stochastic adaptive control model for traffic signal systems. *Transportation Research Part C, 14:4, 263-282*, 2006.

[177] S. P. Bradley, A. C. Hax, and T. L. Magnanti. *Applied Mathematical Programming*, chapter 11, pages 320–362. Addison-Wesley, 1977.

[178] W. B. Powell. *Approximate dynamic programming: solving the curses of dimensionality*. Wiley, New York, 2007.

[179] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, 834-846*, 1983.

[180] E. Bingham. Reinforcement Learning in Neurofuzzy Traffic Signal Control. *European Journal of Operational Research, 131:2, 232-241*, 2001.

[181] S. Mikami and Y. Kakazu. Genetic reinforcement learning for cooperative traffic singal control. *Proceedings of the First IEEE Conference on Evolutionary Computation IEEE World Congress on Computatinal Intelligence, June 27-29, Orlando*, 1994.

[182] J. C. Spall and D. C. Chin. Traffic-Responsive Signal Timing for system-wide traffic control. *Transportation Research Part C, 5, 153-163*, 1997.

[183] M. Wiering, J. Van Veenan, J. Vreeken, and A. Koopman. Intelligent traffic light control. *ERCIMM News. Eur. Res. Consrtium Inf. Math. 53, 40-41*, 2003.

[184] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman. Simulation and optimization of traffic in a city. *IEEE Intelligent Vehicles Symposium, 453-458*, 2004.

[185] M. Steingrover, R. Schouten, S. Peelen, E. Nijhuis, and B. Bakker. Reinforcement learning of traffic light controllers adapting to traffic congestion. *Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC), October 17-18, Brussels*, 2005.

[186] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning, 3:1, 9-44*, 1988.

[187] C. Watkins. *Learning from dedelay rewards*. PhD thesis, King's College, 1989.

[188] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning a survey. *Journal of Artifical Intelligence Research, 4, 237-285*, 1996.

[189] B. Abdulhai, R. Pringle, and G. J. Karakoulas. Reinforcement learning for true adaptive trafic signal control. *Journal of Transportation Engineering, 129:3, 278-285*, 2003.

[190] T. Thorpe. Vehicle traffic light control using SARSA. Technical report, Colardo State University, 1997.

[191] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad. Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems, 18:3, 227-245*, 2014.

[192] K. J. Prabuchandran, A. N. Hemanth Kumar, and B. Shalabh. Multi-agent reinforcement learning for traffic signal control. *Proceedings of the 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), October 8-11, Qingdao, China*, 2014.

[193] P. Mannion, J. Duggan, and E. Howley. Parallel Reinforcement Learning for Traffic Signal Control. *Procedia Computer Science, 52, 956-961*, 2015.

[194] C. Ozan, O. Baskan, S. Haldenbilen, and H. Ceylan. A modified reinforcement learning algorithm for solving coordinated signalized networks. *Transportation Research Part C: Emerging Technologies, 54, 40-55*, 2015.

[195] P. Mannion, J. Duggan, and E. Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. *Autonomic Road Transport Support Systems. Springer, 47-66*, 2016.

[196] K. Yang, I. Tan, and M. Menendez. A reinforcement learning based traffic signal control algorithm in a connected vehicle environment. *Proceedings of the 17th Swiss Transport Research Conference, May 17-19, Monte Verita, Ascona*, 2017.

[197] E. van der Pol and F. A. Oliehoek. Coordinated Deep Reinforcement Learners for Traffic Light Control. *In NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*, 2016.

[198] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and et al. Human-level control through deep reinforcement learning. *Nature, 518:7540, 529-533*, 2015.

[199] W. Genders and S. Razavi. Using a Deep Reinforcement Learning Agent for Traffic Signal Control. *CoRR, 1611.01142*, 2016.

[200] L. Li, L. Yisheng, and F. Y. Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica, 3:3*, 2016.

[201] S. S. Mousavi, M. Schukat, P. Corcoran, and E. Howley. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *CoRR, 1704.08883*, 2017.

[202] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori. Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network. *CoRR, 1705.02755*, 2017.

[203] W. R. Gilks, S. Richardson, and D. J. Speigelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1996.

[204] B. G. Heydecker, C. Cai, and C. K. Wong. Adaptive control for road traffic signals. *Proceedings of IEEE International Conference on Networking, Sensing and Control, London, UK, 193-198*, 2007.

[205] T. Li, D. B. Zhao, and J. Q. Yi. Adaptive dynamic programming for multi-intersections traffic signal intelligent control. *Proceedings of the 11th International IEEE Conference on Intelligent Transport Systems, Beijing, October 12-15, Beijing, China*, 2008.

[206] P. J. Werbos. Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent control: Neural, Fuzzy, and Adaptive Approaches, 493-515, Van Nostrand Reinhold, New York*, 1992.

[207] C. Cai, C. K. Wong, and B. G. Heydecker. Adaptive traffic Singnal control using approximate dynamic programming. *Transportation Research Part C, 17:5, 456-474*, 2009b.

[208] R. M. Kimber and E. M. Hollis. Traffic queues and delays at road junctions. *Transport and Road Research Laboratory Report LB909. Crowthorne: TRL*, 1979.

[209] D. P. de Farias. *The linear programming approach to approximate dynamic programming: theory and application*. PhD thesis, Stanford University, Department of Management Science and Engineering, 2002.

[210] C. Diakaki, M. Papageorgiou, and K. Aboudolas. A multivariable regulator approach to traffic-responsive network. *Proceedings of the IFAC Control in Transportation Systems, June 14-16, Budapest*, 2000.

[211] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Control and optimization methods for traffic signal control in large-scale congested urban road networks. *Proceedings of the American Control Conference, July 9-13, New York*, 2007.

[212] K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos. A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies, 18:5, 680-694*, 2009.

[213] A. Kouvelas, K. Aboudolas, M. Papageorgiou, and E. B. Kosmatopoulos. A Hybrid Strategy for Real-Time Traffic Signal Control of Urban Road Networks. *IEEE Transactions on Intelligent Transportation Systems, 12:3, 884-894*, 2011.

[214] S. Lin. *Efficient model predictive control for large-scale urban traffic networks*. PhD thesis, Netherlands TRAIL Research School, 2011.

[215] J. A. Rossiter. *Model-based Predictive Control*. CRC Press, New York, 2004.

[216] F. Basile, P. Chiacchio, and D. Teta. Hybrid Model for Model Predictive Control of Urban Traffic. *Proceedings of the 9th European Control Conference (ECC'07), Kos, July 2-5*, 2007.

[217] T. Tettamanti, I. Varga, and B. Kulcsar. Model predictive control in urban traffic network management. *Proceedings of the 16th Mediterranean Conference on Control and Automation, Ajaccio, June 25-27, France*, 2008.

[218] T. Le, H. L. Vu, Y. Nazarathy, B. Vo, and S. Hoogendoorn. Linear-Quadratic Model Predictive Control for Urban Traffic Networks. *Procedia - Social and Behavioral Sciences, 80:7, 512-530*, 2013.

[219] C. Guo, X. Gang, and M. Zhang. Model predictive control implementation and simulation for urban traffic networks. *Proceedings of the 2014 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), October 8-10, Qingdao, China*, 2014.

[220] X. Zhou and Y. Lu. Coordinate model predictive control with neighbourhood optimisation for a signal split in urban traffic networks. *IET Intelligent Transport Systems*, 2012.

[221] A. F. de Souza, B. V. Peccin, and E. Camponogara. Distributed Model Predictive Control Applied to Urban Traffic Networks: Implementation, Experimentation, and Analysis. *Proceedings of the 6th annual IEEE Conference on Automation Science and Engineering, Toronto, August 21-24*, 2010.

[222] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Fast Model Predictive Control for Urban Road Networks via MILP. *IEEE Transactions on Intelligent Transportation Systems, 12:3, 846-856*, 2011.

[223] T. Le, H. L. Vu, Y. Nazarathy, Q. B. Vo, and S. Hoogendoorn. Linear-quadratic model predictive control for urban traffic networks. *Transportation Research Part C, 36, 498-512*, 2013.

[224] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Model Predictive Control for Urban Traffic Networks via MILP. *2010 American Control Conference, Marriott Waterfront, Baltimore, MD, June 30 - July 2*, 2010.

[225] L. B. de Oliveira and E. Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C, 18, 120-130*, 2010.

[226] W. Kooistra. Simulation of coordinated traffic light control using model predictive control. *Research paper Business Analytics*, 2012.

[227] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Study on fast model predictive controllers for large urban traffic networks. Technical report, Delft Center for Systems and Control, Delft University of Technology, 2009.

[228] D.L. Borg and K. Scerri. Efficient Traffic Modelling and Dynamic Control of an Urban Region. *Transportation Research Procedia 6, 224-238*, 2015.

[229] K. Yu, J. Yang, and D. Yamaguchi. Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information. *Control Theory and Technology, 13:1, 17-28*, 2015.

[230] Z. Zhou, B. De Schutter, S. Lin, and Y. Xi. Multi-agent model-based predictive control for large-scale urban traffic networks using a serial scheme. *IET Control Theory & Applications, 9:3, 475-484*, 2015.

[231] J. Falilat. *A Synthesis of Automated Planning and Model Predictive Control Techniques and its Use in Solving Urban Traffic Control Problem*. PhD thesis, University of Huddersfield, 2015.

[232] J. Falilat and T. L. McCluskey. Towards The Integration of Model Predictive Control into an AI Planning Framework. *Proceedings of the 34th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG). PlanSIG, Huddersfield*, 2016.

[233] M. Hajiahmadi, B. De Schutter, and H. Hellendoorn. Robust H? switching control techniques for switched nonlinear systems with application to urban traffic control. *International Journal of Robust and Nonlinear Control, 26, 1286-1306*, 2016.

[234] E. Bradford and L. Imsland. Stochastic Nonlinear Model Predictive Control with State Estimation by Incorporation of the Unscented Kalman Filter. *CoRR, 1709.01201*, 2017.

[235] Z. Zhou, B. De Schutter, S. Lin, and Y. Xi. Two-level hierarchical model-based predictive control for large-scale urban traffic networks. *IEEE Transactions on Control Systems Technology, 25:2, 496-508*, 2017.

[236] S. Liu, H. Hellendoorn, and B. De Schutter. Model predictive control for freeway networks based on multi-class traffic flow and emission models. *IEEE Transactions on Intelligent Transportation Systems, 18:2, 306-320*, 2017.

[237] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. DeSchutter. Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods. *IEEE Transactions on Control Systems Technology, 26:3, 813-827*, 2018.

[238] A. Dotlinger and R. M. Kennel. Near Time-Optimal Model Predictive Control using an L1-Norm based Cost Functional. *Proceedings of the 2014 IEEE Energy Conversion Congress and Exposition (ECCE), September 14-18, Pittsburgh, USA*, 2014.

[239] K. Dresner and P. Stone. Traffic Intersections of the future. *AAAI'06, Proceedings of the 21st National Conference on Artificial Intelligence, July 16-20, Boston*, 2006.

[240] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence, 31, 591-653*, 2008.

[241] K. Dresner and P. Stone. Multiagent traffic management: a reservation-based intersection control mechanism. *AAMAS'04, Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2, IEEE Computer Society, July 19-23, New York*, 2004.

[242] K. Dresner and P. Stone. Multiagent traffic management: an improved intersection control mechanism. *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, 471-477*, 2005.

[243] G. Balan and S. Luke. History-based traffic control. *Proceedings of the fifth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, 616-621*, 2006.

[244] M. Vasirani and S. Ossowski. A market-inspired approach to reservation-based urban road traffic management. *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, May 10-15, Budapest*, 2009.

[245] R. Tachet, P. Santi, S. Sobolevsky, L. Reyes-Castro, E. Frazzoli, D. Helbing, and et al. Revisiting Street Intersections Using Slot-Based Systems. *PLoS ONE, 11:3*, 2016.

[246] J. Lee and B. Park. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems, 13, 81-90*, 2012.

[247] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence, 31, 591-653*, 2008.

[248] M. Hausknecht, T. C. Au, and P. Stone. Autonomous intersection management: multi-intersection optimization. *Proceedings of IROS 2011-IEEE/RSJ International Conference on Intelligent Robots and Systems, 4581-4586*, 2011.

[249] A. Giridhar and P. Kumar. Scheduling automated traffic on a network of roads. *IEEE Transactions on Vehicular Technology, 55, 1467-1474*, 2006.

[250] D. McKenney and T. White. Distributed and adaptive traffic signal control within a realistic traffic simulation. *Engineering Applications of Artificial Intelligence 26, 574-583*, 2013.

[251] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode. Adaptive Traffic light using car-to-car communication. *Proceedings of the 65th IEEE Vehicular Technology Conference, 21-25, April 22-25, Dublin*, 2007.

[252] Q. Huang and R. Miller. Reliable wireless traffic signal protocols for smart intersections. *14th Annual Meeting of ITS, America*, 2004.

[253] K. Pandit and D. Ghosal. Adaptive Traffic Signal Control with Vehicular Ad hoc Networks. *IEEE Transactions on Vehicular Technology*, 2013.

[254] N. Maslekar, J. Mouzna, M. Boussedjra, and H. Labiod. CATS: An adaptive traffic signal system based on car-to-car communication. *Journal of Network and Computer Applications*, 2012.

[255] H. J. Chang and G. T. Park. A study on traffic signal control at signalized intersections in vehicular ad hoc networks. *Ad Hoc Networks, 11:7, 2115-2124*, 2012.

[256] P. Gregoire, C. Desjardins, J. Laumonier, and B. Chaibdraa. Urban traffic control based on learning agents. *Proceedings of the 10th International IEEE conference on intelligent transporation systems, September 30-October 3, Seattle*, 2007.

[257] M. Amin and S. Jalili. Urban signal control using intelligent agents. *Proceedings of the 7th International FLINS conference on applied artificial intelligence, August 29-31, Genova*, 2006.

[258] M. A. Kafi, Y. Challal, D. Djenouri, A. Bouabdallah, L. Khelladi, and N. Badache. A study of Wireless Sensor Network Architectures and Projects for Traffic Light Monitoring. *Proceedings of the 3rd International Conference on Ambient Systems, Networks and Technologies (ANT), Procedia Computer Science, 10, 543-552*, 2012.

[259] N. J. Goodall, B. L. Smith, and B. B. Park. Traffic Signal Control with Connected Vehicles. *Transportation Research Record: Journal of the Transportation Research Board, No. 2381, Transportation Research Board of the National Academies, Washington, 65-72*, 2013.

[260] N. Al-Ostath, F. Selityn, Z. Al-Roudhan, and M. El-Abd. Implementation of an emergency vehicle to traffic lights communication system. *Proceedings of the 7th International Conference on New Technologies, Mobility and Security (NTMS), Paris, July 27-29*, 2015.

[261] V. Astarita, V. P. Giofre, G. Guido, and A. Vitale. The Use of Adaptive Traffic Signal Systems Based on Floating Car Data. *Wireless Communications and Mobile Computing 2017*, 2017.

[262] P. Jing, H. Huang, and L. Chen. An Adaptive Traffic Signal Control in a Connected Vehicle Environment: A Systematic Review. *MDPI, 8:3, 2-24*, 2017.

[263] C. N. Van Phu, N. Farhi, H. Haj-Salem, and J. P. Lebacque. A vehicle-to-infrastructure communication based algorithm for urban traffic control. *Proceeding of the 5th International Conference on Models and Technologies for Intelligent Transportation Systems, June 26-28, Naples*, 2017.

[264] M. H. Eiza and Q. Ni. Driving with Sharks: Rethinking Connected Vehicles with Vehicle Cyber Security. *IEEE Vehicular Technology Magazine, 12:2, 45-51*, 2017.

[265] S. Tbatou, A. Ramrami, and Y. Tabii. Security of communications in connected cars Modeling and safety assessment. *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, March 29-30, Morocco*, 2017.

[266] Z. Kai, Y. J. Gong, and J. Zhang. Real-Time Traffic Signal Control with Dynamic Evolutionary Computation. *Proceedings of 2014 IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAAI), August 31- September 4, Kitakyushu, Japan*, 2014.

[267] R. Armas, H. Aguirre, and F. Daolio. Traffic signal optimization and coordination using neighborhood mutation. *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), July 24-29, Vancouver*, 2016.

[268] S. S. Leal, P. E. M. de Almeida, and E. Chungb. Active control for traffic lights in regions and corridors: an approach based on evolutionary computation based on evolutionary computation. *Proceedings of the World Conference on Transport Research, WCTR 2016, Shanghai, July 10-15, Transportation Research Procedia 25:2017, 1769-1780*, 2016.

[269] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[270] K.T.K. Teo, W.Y. Kow, and Y.K. Chin. Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm. *Proceedings of the 2010 Second International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM), September 28-30, Tuban, Indonesia*, 2010.

[271] H. Yang and D. Luo. Acyclic Real-Time Traffic Signal Control Based on a Genetic Algorithm. *The Journal of Institute of Information and Communication Technologies of Bulgarian Academy of Sciences, 13:3*, 2013.

[272] B. Zhao, C. Zhang, and L. Zhang. Real-Time Traffic Light Scheduling Algorithm Based on Genetic Algorithm and Machine Learning. *International Conference on Internet of Vehicles, September 1-3, Beijing, China*, 2015.

[273] E. Lienert. Simulation of Genetic Algorithm: Traffic Light Efficiency. *Neural and Evolutionary Computing, Available Online:https://arxiv.org/abs/1503.04475, [Last accessed: 28/01/2019]*, 2015.

[274] N. Sharma and S. Sahu. Review of Traffic Signal Control based on Fuzzy Logic. *International Journal of Computer Applications (0975-8887), 145:13, 18-22*, 2016.

[275] C. Papis and E. Mamdani. A fuzzy logic controller for a traffic junction. *IEEE Transportation System, Man and Cybern, 7, 707-717*, 1977.

[276] M. Nakatsuyama, H. Nagahashi, and N. Nishizara. Fuzzy logic controller for a traffic junction in the one-way arterial road. *Proceedings of the 9th IFAC-World Congress, July 2-6, Budapest*, 1984.

[277] R. Kelsey, K. Bisset, and M. Jamshidi. A simulation environment for fuzzy control of traffic systems. *Proceedings of the 12th IFAC-World Congress Sydney, July 18-23*, 1993.

[278] S. Chiu and S. Chand. Adaptive traffic signal control using fuzzy logic. *In Second IEEE International Conference on Fuzzy Systems, 1371-1376*, 1993.

[279] R. Hoyer and U. Jumar. Fuzzy control of traffic lights. *Proceedings of the third IEEE International Conference on Fuzzy Systems, June 26-29, Orlando*, 1994.

[280] K. K. Tan, M. Khalid, and R. Yusof. Intelligent traffic lights control by fuzzy logic. *Malaysian Journal of Computer Science, 9:2, 29-35*, 1996.

[281] J. H. Lee, K. M. Lee, and H. Lee-Kwang. Fuzzy controller for intersection group. *Proceedings of the International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies, May 22-27, Taipei*, 1995.

[282] G. Beauchamp-Baez, E. Rodriguez-Morales, and E. Muniz-Marrero. A fuzzy logic based phase controller for traffic control. *Proceeding of the Sixth IEEE Int. Conf. Fuzzy Systems, 1533-1539*, 1997.

[283] J. Niittymaki and S. Kikuchi. Application of fuzzy logic to the control of a pedestrian crossing signal. *J. Transp. Res. Bd., 1651, 30-38*, 1998.

[284] J. Anderson, T. Sayers, and M. Bell. The objectives of traffic signal control. *Traffic Engineering Control, 39:3, 167-170*, 1998.

[285] J. Niittymaki and M. Pursula. Signal control using fuzzy logic. *Fuzzy Sets and Systems, 116:1, 11-22*, 2000.

[286] W. Wei, Y. Zhang, J. B. Mbede, Z. Zhang, and J. Song. Traffic Signal control using fuzzy logic and MOGA. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 7-10, Tucson*, 2001.

[287] A. Hegyi, B. De Schutter, and S. Hoogendoorn. A fuzzy decision support system for traffic control centers. *Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference (ITSC'01), August 25-29, Oakland*, 2001.

[288] J. Niitymaki and M. Maenpaa. The role of fuzzy logic public transport priority in traffic signal control. *Traffic Eng. Control, 42:1, 22-26*, 2001.

[289] I. Kosonen. Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C, 11:5, 389-403*, 2003.

[290] J. Niittymaki and E. Turunen. Traffic signal control on similarity logic reasoning. *Fuzzy Sets and Systems, 133:1, 109-131*, 2003.

[291] Y. Chiou and L. W. Lan. Adaptive traffic signal control with iterative genetic fuzzy logic controller (GFLC). *Proceedings of the IEEE International Conference on Networking, Sensing and Control, March 21-23, Taipei*, 2004.

[292] Y. S. Murat and E. Gedizlioglu. A new approach for fuzzy traffic signal control. *Proceeding of 13th Mini-EURO Conference on Artificial Intelligence in Transportation Systems and Science, June 10-13, Bari*, 2002.

[293] M. A. P. Jacques, M. Pursula, and J. Niittymaki. The impact of different approximate reasoning methods on fuzzy signal controllers. *Proceedings of the 13th Mini-EURO Conferene on Handling Uncertainty in the Analysis of Traffic and Transportation Systems, Bari, June 10-13*, 2002.

[294] L. Zhang, H. Li, and P. Prevedouros. Signal control for oversaturated intersections using fuzzy logic. *Transportation Research Record 84th TRB Annual Meeting, January 9-13, Washington D.C.*, 2005.

[295] T. Akiyama and M. Okushima. Advanced fuzzy traffic controller for urban expressways. *International Journal of Innovative Computing, Information and Control, 2:2, 339-355*, 2006.

[296] C. H. Chou and J. C. Teng. A fuzzy logic controller for traffic junction signals. *Information Sciences, 143:1, 73-97*, 2002.

[297] J. Kim. A fuzzy logic control simulator for adaptive traffic management. *Proceeding of the sixth IEEE Int. Conf. Fuzzy Systems, 1519-1524, July 5, Barcelona, Spain,*, 1997.

[298] G. Nakamiti and R. Freitas. Adaptive real-time traffic control management. *International Journal of Automotive Technology, 3:3, 89-94*, 2002.

[299] M. A. Taha and L. Ibrahim. Traffic Simulation System based on Fuzzy Logic, Complex Adaptive Systems. *Conference organized by Missouri University of Science and Technology, Washington D.C.*, 2012.

[300] S. F. Chen and L. Mao. Fuzzy line control and simulation for urban traffic lights. *Journal of System Simulation, 12:6, 668-670*, 2000.

[301] J. H. Lee and K. H. Lee. Distributed and cooperative fuzzy controllers for traffic intersection group. *IEEE Transactions on Systems, Man and Cybernetics: Applications and Reviews, 29:2, 263-271*, 1999.

[302] J. D. Schmocker, S. Ahuja, and G. H. M. Bell. Multi-Objective signal control of urban junctions - Framework and a London case study. *Transportation Research Part C, 16, 454-470*, 2008.

[303] C. Karakuzu and O. Demirci. Fuzzy Logic based smart traffic light simulator design and hardware implementation. *Applied Soft Computing, 10, 66-73*, 2010.

[304] Y. Zheng, Y. Zhang, and J. Hu. Iterative Learning Based Adaptive Traffic Signal Control. *Journal of Transporation Systems Engineering & IT, 10:6, 34-40*, 2010.

[305] X. Yisheng, S. Goujiang, and C. Xiang. Fuzzy Neural Network Control Technique and its Application in a complex intersection. *Energy Procedia, 16, 1408-1415*, 2012.

[306] R. S. Masiur and T. R. Nedal. Review of the Fuzzy Logic Based Approach in Traffic Signal Control: Prospects in Saudi Arabia. *Journal of Transportation Systems Engineering and Information Technology, 9:5*, 2009.

[307] O. U. Chinyere, O. O. Francisca, and O. E. Amano. Design and Simulation of an Intelligent Traffic Control System. *International Journal of Advances in Engineering and Technology*, 2011.

[308] M. B. Trabia, M. S. Kaseka, and M. Ande. A two stage fuzzy logic controller for traffic signals. *Transportation Research Part C, 7:6, 353-367*, 1999.

[309] E. Onieva, V. Milanes, J. Villagra, J. Perez, and J. Goday. Genetic optimizatin of a vehicle fuzzy decision system for intersections. *Expert Systems with Applications, 39:18, 13148-13157*, 2012.

[310] Y. Ge. A Two-Stage Fuzzy Logic Control Method of Traffic Signal Based on Traffic Urgency Degree. *Modelling and Simulation in Engineering, 2014*, 2014.

[311] J. Alam and M. K. Pandey. Design and Analysis of a Two Stage Traffic Light System Using Fuzzy Logic. *Journal of Information Technology & Software Engineering, 5:3, 1000162*, 2015.

[312] M. J. Moghaddam, M. Hosseini, and R. Safabakhsh. Traffic light control based on fuzzy Q-leaming. *Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing (AISP), March 3-5, Mashhad, Iran*, 2015.

[313] B. Poletajew and A. Slowik. An Application of Fuzzy Logic to Traffic Lights Control and Simulation in Real Time. *Proceedings of the International Conference on Artificial Intelligence and Soft Computing, June 12-16, Zakopane, Poland*, 2016.

[314] F. Taher, A. El-Sayed, A. Shouman, and A. El-Mahalawy. Comparing Different Techniques for Controlling Traffic Signals. *International Journal on Power Engineering and Energy (IJPEE), 7:3, 680-686*, 2016.

[315] R. Mohebifard and A. Hajbabaie. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. *Transportation Research Part B, 121, 252-274*, 2019.

[316] S. V. Anfilets and V. N. Shuts. Artificial Neural Network for adaptive management traffic light objects at the intersection. *Proceedings of the*

10th International Conference Reliability and Statistics in Transportation and Communication (RelStat'10), October 20-23, Riga, Latvia, 2010.

[317] M. S. Yasar and M. T. Rashid. Implementation of Dynamic Traffic Light Controllers Using Artificial Neural Networks to Diminish Traffic Ordeals. *Proceedings of the 2015 IEEE European Modelling Symposium, October 6-8, Madrid*, 2015.

[318] J. Aboutaam, N. Amaneddine, and M. Al-Akaidi. A Comparitive approach to model traffic light controler based on artificial neural network. *Proceedings of the 14th Annual Industrial Simulation Conference, ISC'2016, June 6-8, Romania*, 2016.

[319] D. Teodorovi, V. Varadarajan, J. Popovi, M. R. Chinnaswamy, and S. Ramaraj. Dynamic programming-neural network real-time traffic adaptive signal control algorithm. *Annals of Operation Research, 143:1, 123-131*, 2006.

[320] D. Strinivasan. Neural Networks for Real-Time Traffic Signal Control. *IEEE Transations on Intelligent Transportation Systems, 7:3*, 2006.

[321] L. Zhiyong and W. Baiwu. Hierarchical Fuzzy Neural Network Control for Large Scale Urban Traffic Systems. *Information and Control, 26:6, 441-448*, 1997.

[322] T. Royani, J. Haddadnia, and M. Alipoor. Control of Traffic Light in Isolated Intersections Using Fuzzy Neural Network and Genetic Algorithm. *International Journal of Computer and Electrical Engineering, 5:1, 142-146*, 2013.

[323] D. de Oliveira and A. Bazzan. Traffic lights control with adaptive roup formaion based on swarm intelligence. *Ant Colony Optimization Swarm Intelligence 4150, 520-521*, 2006.

[324] J. Garcia-Nietoa, E. Alba, and A. Carolina Olivera. Swarm intelligence for traffic light scheduling: Application to real urban areas. *Engineering Applications of Artificial Intelligence, 25:2, 274-283*, 2012.

[325] J. Garcia-Nietoa, E. Alba, and A. Carolina Olivera. Swarm intelligence for traffic light scheduling: Application to real urban areas. *Engineering Applications of Artificial Intelligence, 25:2, 274-283*, 2012.

[326] F. Caselli, A. Bonfietti, and M. Milano. Swarm-Based Controller for Traffic Lights Management. *Proceedings of the XIVth International Conference of the Italian Association for Artificial Intelligence, September, 23-25, Ferrara, Italy*, 2015.

[327] M. Tahifa, J. Boumhidi, and A. Yahyaouy. Swarm reinforcement learning for traffic signal control based on cooperative multi-agent framework. *Proceedings of the Intelligent Systems and Computer Vision (ISCV), March 25-26, Fez, Morocco*, 2015.

[328] K. Gao, Y. Zhang, and A. Sadollah. Improved artificial bee colony algorithm for solving urban traffic light scheduling problem. *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), June 5-8, San Sebastian*, 2017.

[329] J. D. C. Little, M. D. Kelson, and N. H. Gartner. MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks. *U.S. Dept. Transp.,Washington, DC, Transp. Res. Record 795*, 1981.

[330] A. G. Sims and A. B. Finlay. SCATS. Splits and offsets simplified (SOS). *ARRB Proceedings 12, Part 4*, 1984.

[331] R. Boettger. On-line optimisation of the offset in signalised street networks. *IEE International Conference. Road Traffic Signalling, Publication No 207*, 1987.

[332] F. Boillot, J. M. Blosseville, J. B. Lesort, V. Motyka, M. Papageorgiou, and S. Sellam. Optimal signal control of urban traffic networks. *Proc. 6th IEE Int. Conf. Road Traffic Monitoring and Control, 75-79*, 1992.

[333] D. Renfrew and X. H. Yu. Traffic Signal Control with Swarm Intelligence. *Proceedings of the 2009 Fifth International Conference on Natural Computation*, 2009.

[334] S. Rangaswamy and G. Shobha. Techniques and theories of self-optimization in autonomic systems. *International Journal of Advanced Engineering Technology, 2, 73-80*, 2011.

[335] A. Ganek and T. Corbi. The dawning of the autonomic computing era. *IBM Systems Journal, 42:1, 5-18*, 2003.

[336] R. Foroughi, G. A. Montazer, and R. Sabzevari. Design of a new urban traffic control system using modified Ant colony optimization approach. *Iranian Journal of Science & Technology, Transaction B, Engineering, 32, 167-173*, 2008.

[337] Y. C. Chiu and H. Mahmassani. Hybrid real-time dynamic traffic assignment for robust network performance. *Transportation Research Record, 1783, 89-97*, 2002.

[338] O. Brukman, S. Dolev, Y. Haviv, and R. Yagel. Self-Stabilization as a Foundation for Autonomic Computing. *ARES 2007, The Second International Conference on Availability, Reliability and Security, 991-998*, 2007.

[339] A. Bazan. A game-theoretic approach to distributed control of traffic signals. *Proceedings of the International Conference on Multi-Agent Systems, June 12-14, San Francisco*, 1995.

[340] S. Lammer and D. Helbing. Self-control of traffic lights and vehicle flow in urban road networks. *J. Stat. Mech.: Theory Exp. 4*, 2008.

[341] Transport Malta. Transport Master Plan 2025. *European Regional Development Fund Malta 2007-2013*, 2016.

[342] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.

[343] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2009.

[344] S. Gibson and B. Ninnes. Robust Maximum-Likelihood Estimation of Multivariable Dynamic Systems. *Automatica*, 2005.

[345] A. Wills and B. Ninnes. System Identification of Linear Parameter Varying State-space Models. *School of Electrical Engineering and Computer Science, University of Newcastle*.

[346] B. Ninnes and S. Gibson. The EM algorithm for Multivariable Dynamic System Estimation. *Technical Report EE200101, Department of Electrical and Computer Engineering, University of Newcastle*.

[347] Z. Ghahramani and G. E. Hinton. Parameter Estimation for Linear Dynamical Systems. Technical report, Department of Computer Science, University of Toronto, 1996.

[348] R. H. Shumway and D. S. Stoffer. An Approach to Time Series Smoothing and Forecasting using the EM Algorithm. *Journal of Time Series Analysis 3:4*, 1982.

[349] T. B. Schon. An Explanation of the Expectation Maximization Algorithm. Technical report, Division of Automatic Control, Department of Electrical Engineering, Sweden, 2009.

[350] S. Sarkka. *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge, 2013.

[351] K. B. Peterson and M. S. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, 2012.

[352] F. R. Gantmacher. *The Theory of Matrices*. Chelsea, New York, 1959.

[353] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, USA, 1999.

[354] Transport Simulation Systems. Aimsun Software. Available online at: https://www.aimsun.com [Last accessed 28/01/2019], 2019.

[355] S. K. Mitter and I. C. Schick. *Systems, Models and Feedback: Theory and Applications*, chapter Point Estimation, Stochastic Approximation, and Robust Kalman Filtering. Birkhauser, Boston, MA, 1992.

[356] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, 1978.

[357] E. M. Airoldi and P. Toulis. Scalable estimation strategies based on stochastic approximations: Classical results and new insights. *Stat. Comput., 25:4, 781-795*, 2015.

[358] A. Howard. *Elementary Linear Algebra*. John Wiley & Sons, 1994.

[359] W. Cheney and D. Kincaid. *Linear Algebra: Theory and Applications*. Jones & Bartlett Publishers, New Dehli, 2010.

[360] L. Chetcuti Zammit, S. G. Fabri, and K. Scerri. Online state and parameter estimation for a 4-arm junction. *Proceedings of the 25th Mediterranean Conference on Control and Automation, July 3-6, Malta*, 2017.

[361] L. Chetcuti Zammit, S. G. Fabri, and K. Scerri. Online State and Multi-dimensional Parameter Estimation for a Macroscopic Model of a Traffic Junction. *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems, October 16-19, Japan*, 2017.

[362] A. Bjorck and S. Hammerling. A Schur method for the square root of a matrix. *Linear algebra and its applications, 52-53, 127-140*, 1983.

[363] G. A. Ackerson and K. S. Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control, AC-15:1,10-17*, 1970.

[364] C. B. Chang and M. Athans. State estimation for discrete system with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems, AES-14:3, 418-425*, 1978.

[365] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control, 33:8, 780-784*, 1988.

[366] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting Multiple Model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems, 34:1, 103-123*, 1998.

[367] D. L. Pogoda and P. S. Maybeck. Reconfigurable Flight Controller for the STOL F-15 with sensor/actuator failures. *Proceedings of the 28th Conference on Decision and Control, December 13-15, Tampa, FL, USA*, 1989.

[368] C. V. Rao and J. B. Rawlings. Linear programming and model predictive control. *Journal of Process Control 10, 283-289*, 2000.

[369] A. Bemporad, F. Borrelli, and M. Morari. Model Predictive Control Based on Linear Programming - The Explicit Solution. *IEEE Transactions on Automatic Control 47:12, 1974-1985*, 2002.

[370] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[371] Z. Szabo and M. Kovacs. On interior-point methods and simplex method in linear programming. *An. St. Univ. Ovidius Constanta, 11:2, 155-162*, 2003.

[372] The MathWorks, Inc. Matlab Software, 2019.

[373] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2007.

[374] Transport Simulation Systems. Aimsun 8 API Manual. 2019.

[375] P. G. Gipps. A behavioural car following model for computer simulation. *Transport Research Part B, 15, 403-414*, 1981.