Adaptive Control for Traffic Signals

Using a Stochastic Hybrid System Model

Sutarto

Promotoren : ereprof.dr.ir R. Boel, prof.dr.ir D. Aeyels
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Werktuigkunde-Elektrotechniek

Vakgroep Elektronica en Informatiesystemen
Voorzitter : prof.dr.ir.R.van de Walle
Faculteit Ingenieurswtenschappen en Architectuur
Academiejaar 2015-2016

ISBN 978-90-8578-891-1

NUR 950

Wettelijk depot: D/2016/10.500/23

Dit proefschrift is goedgekeurd door de promotoren:

Ereprof.dr.ir René Boel

Prof.dr.ir. Dirk Aeyels

Samenstelling examencommissie:

Prof. dr.ir. Gert de Cooman	voorzitter
Ereprof.dr.ir. René Boel	Universiteit Gent, promotor
Prof.dr.ir. Dirk Aeyels	Universiteit Gent, promotor
Prof.dr.ir. Stijn De Vuyst	Universiteit Gent
Prof.dr. Dries Benoit	Universiteit Gent
Prof.dr.ir Mia Loccufier	Universiteit Gent
Prof.dr.ir. Lyudmila Mihaylova	University of Sheffield, VK
Dr.ir. Ivana Semanjski	Universiteit Gent
Dr.ir. Jasper De Bock	Universiteit Gent

Affiliatie

Vakgroep Elektronica en Informatiesystemen

Faculteit Ingenieurswetenschappen en Architectuur

Universiteit Gent

Technologiepark Zwijnaarde 914 BE-9052, Zwijnaarde

Acknowledgments

Here it is….my PhD thesis. It took me 7 years to create, including only almost 3 years of full time work and more than 4 years of sharing time with other work: teaching, supervising and some consulting job from morning to evening. How this could have happened? Well, Indeed I enjoy the process and it is important that I got: determination to finish what I had started. This means that I owe a lot of thanks to my supervisor Prof René Boel for his deep understanding and helpful discussion and having eye for the details. He was able to use a few solid words for deflecting my focus on the things that are fundamental and essential. Likewise, I also want to thank to Prof Dirk Aeyels preserving the continuity of the administrative process, when René retired and Prof Endra Joelianto at Institut Teknologi Bandung, who gives me a freedom to attract some funding supporting for my research and having collaboration in his group.

I wish to acknowledge that thesis was supported by European Community's Seventh Framework Programmed under Project DISC (Grant Agreement n.INFSO-ICT-224498). Thanks you to all researchers who involved in DISC give me possibility to learn how the European researcher work and interact, especially to Renato Vazquez and Prof Manuel Silva from University of Zaragoza for having collaboration.

I am particularly very grateful to all jury members, for their rigorous reading and constructive comments to further improve my thesis. Thank you for Muriel Vervaeke for the administrative assistance preparing for my defense.

I would also like to thank to gratefully acknowledge Ir. Agoes Suwardi and Ir. Rosyad from P.T. Newtel, Bandung, Ir. Zulkarnaen Zakaria, M.Eng. from P.T. General Intelligent Technology, Bandung and also ir.Taufik Sumardi M.Eng from Institute of Road Engineering, Bandung for their helpful discussion and assistance of the measurement data. Special thanks goes to Aditya Nugroho at Surbana International Consultans Pte. Ltd., Singapore for help me running the VISSIM simulator and Singgih Satrio Wibowo for giving me help when I stuck with my code. My colleagues at Institut Teknologi Harapan Bangsa: Samuel Tarigan (Rector), Tunggul, Dina, Sinung, Mac, Dian, and Jusak for the support and a pleasant and enjoyable atmosphere for teaching.

For member of SYSTeMS research group, I would like to thank Frits, Filip (H), Gert, Keivan, Erik, Bram, Filip, Abhisek, Jonathan, Arthur, Jasper, Alain and Simon for being friendly. Margot for “the Oase” during the hard-time, Mia for guiding me to have ”a good structure of my thesis”. I would like to thank to Nicolae, my office roommate, for useful discussion and learn to say what to say and Mohammad for helping me bring my daughter to the hospital and learn to be brave.

During the time spent in Ghent, I would like to thank Father Charles de Hemptinne for giving me opportunity to live in his lovely and beautiful house and being friendly to discuss about our faith; Wa Ann Djuzman and her husband Francis van de Walle for their support and attention; Juliando Kusmanto and Ci Lanny and their family for being friendly and support; Prayer group “Regina”: for many attentions, pray and understanding; PERKI and of course Indonesian Student Association (PPI), and also Ministry of Education Indonesia for giving supports for some funding.

On more personal note, I would like to thank my dear mami Sri Hamurwani (+) *yang selalu mencintai apa adanya saya* and teaches me a lot about the willingness to suffer for the people we love. My father dr Poerwadi (+) *maturnuwun untuk semua contoh hidupmu mo*. My Brother Priyadi who gives me the example how to be spearhead, My Brother Sigit who always teaches me to reflect more deeply about anything, my sister Wiwid, Nining, Jaja, Bude Tina, Mas Is, Mas Yuli and Yani for their big support. Finally I would like to thank you very much for my beloved wife Nieta and my lovely daughter Tiara, forgive me who always has a limited time in the evening. I am proud of you, we are a great team! I dedicate this thesis to you both. Most importantly, I must thank my God, Jesus Christ and Bunda Maria who lead me beside still waters. His patience when I fall short and His comfort when I feel lost have never failed me.

Gent, April, 2016

Herman Y Sutarto

DEDICATED To:

Mami, Nieta and Tiara

Contents

List of Figures

List of Tables

List of Symbols

$\theta_\alpha\,(s_k)$	:	Parameters of stochastic hybrid model in mode-*s* at time t_k that consists of : (a) parameters of AR $\{\beta_\alpha(s_k), \gamma_\alpha(s_k), \sigma^2_\alpha(s_k)\}$ and (b) transition probability matrix $\prod_\alpha$
$\mathbf{y}_{t_k}$	:	time series of observed values $(y_{t_0}, y_{t_1}, ..., y_{t_k})$ up to time t_k
t_k	:	k-th sampling time
k	:	denotes cycle-index (sample time) t_k depends on the context
$\lambda_{1,t_{2k}}$	:	Arrival flow during Green period at movement/Link L1 in the interval $[t_{2k}, t_{2k+1})$
$\lambda_{1,t_{2k+1}}$	:	Arrival flow during Red period at movement/Link L1 in the interval $[t_{2k+1}, t_{2(k+1)})$
$\mu_{1,t_{2k}}$	:	Departure flow during Green period at movement/Link L1 in the interval $[t_{2k}, t_{2k+1})$
$q_{m,k+p}$	:	Queue length in movement L_m at index instant *k+p* on the corresponding traffic signal sequence at index k where *p*=0,1,…,N-1 and N indicates the N-step horizon.
q_{cr}	:	Critical queue length
$T_{g,k}$, $T_{r,k}$, C	:	Green period $t_{2k+1}-t_{2k} = T_{g,k}$, Red period $t_{2k+2}-t_{2k+1} = T_{r,k}$ at cycle (sample-time) k and Cycle length $C = T_{g,k} + T_{r,k}$
$\mathbf{T_{g,k+p}}$	:	$[\,T_{g,k}\ T_{g,k+1}\ \ldots\ T_{g,k+p}\,]^T$.

$p[q_{k+p}^{L1} \leq q_{cr}^{L1}] \geq 1-\delta$	:	Chance constraint is applied to queue-length at movement L_1 with $\delta \in (0,1]$ is user-specified probability chosen according to process requirements
$\psi(X,U)$	:	$(\hat{q}-q_{cr})$; scalar function related to chance probability
m	:	iteration index
Π	:	Transition probability matrix
π_{ij}	:	entries (i,j) of TPM Π
$s_k \in \{1,2,3\}$	:	Mode of the system at time t_k
$L(\theta)$	:	Log-likelihood in EM algorithm
$\{\chi_{tk}\}$	:	augmented state $\{\chi_{tk}\} = \{x_{tk}, \theta_{tk.}\}$ at time t_k
$\{x_{t_k}\}$	:	State $=\{q_{t_k},(\lambda_{1,t_{2k}},s_{t_{2k}}),(\lambda_{1,t_{2k+1}},s_{t_{2k+1}}),(\mu_{1,t_{2k}},s_{t_{2k}})\}$ at time time t_k
$D(\delta)$	:	Dirichlet distribution with parameter δ
$D(\delta_{i1}+n_{i1},\delta_{i2}+n_{i2})$	:	Diricihlet distribution with parameter δ and n_{ij} is the number of one-step transitions from i to j in sample s_K
$\Gamma(\delta_{ip}+n_{ip},\hbar)$	:	Gamma distribution, where $\hbar$ is the *scale parameter* and $(\delta_{ip}+n_{ip})$ is the *shape* parameter
K	:	Number of modes
X~F	:	denotes that random variable X has probability distribution F
$\{m_{k-1}^i\}_{i=1}^N$	:	shrinkage locations
$h_k \in [0,1]$	:	a kernel parameter
$\{\theta_k^i, w_k^i, i=1,...,N\}$	:	the N particles, i=1,…,N with their weights at cycle index k.
z_p	:	Gamma distributed random variables with p=1,2 $\in$ K, K is

		number of modes.
VISSIM	:	microscopic multi-modal traffic flow simulation software package
EM	:	Expectation Maximization
mpm	:	most probability mode
wedf	:	weighted empirical distributed function
MFD	:	Macroscopic fundamental diagram
PF	:	Particle Filter
OTPF	:	Observation and Transition-based most likely modes tracking Particle Filter
UTC	:	Urban Traffic Control
SCATS	:	The Sydney Coordinated Adaptive Traffic System
SCOOT	:	Split Offset Optimization Technique
AR	:	Autoregressive
E(•)	:	Expected values
FFM	:	Fluid Flow Model
PBM	:	Platoon Based Model
SHM	:	Stochastic Hybrid Model
TPM	:	Transition probability matrix
JMM	:	Jump Markov Model
MPC	:	Model Predictive Control

Abstract

A more efficient use of the existing road infrastructure, using advanced traffic control strategies, can lead to reduced congestion, reduced emissions, reduced fuel consumption and improved safety. A large urban network consists mainly of two elements link roads and signalized intersections. Modeling the traffic flow dynamics along each of these elements in a large urban network with a large number of links and intersections is a complex task. The model-based control strategies that are needed in order to achieve such an improvement depend strongly on the quality and the accuracy of the dynamic model of the system, and on the accuracy of the state estimates available for feedback.

In this thesis, we develop a stochastic hybrid model (SHM) to effectively describe the evolution over time of the queue-length and arrival/departure flow rates of vehicles in an urban traffic network as stochastic processes. Using this SHM framework, it is possible to model the queue-length evolution at a signalized intersection, describing the interaction between traffic light sequences and arrival/departure traffic flow in all the intersections. This interaction is a combination of event-driven dynamics and time-driven dynamics. The event-driven dynamics are dictated by the green-red light switches and by the events causing some queue lengths to switch from positive to zeros or vice versa. The continuous variables describing, for each mode of the traffic operations, the arrival and departure flow rates can be modeled by a first-order autoregressive (AR) model. The complete SHM is thus a jump Markov model.

The SHMs for which we need to estimate the model parameters can be used for filtering the raw traffic flow measurements and for predicting queue lengths at signalized intersections. The variability of the traffic flow during successive cycles of the traffic light must be estimated with sufficient accuracy in order to predict the expected queue length resulting from control decisions. Parameter estimation techniques are used to determine the unknown parameters of the SHM as a prerequisite for implementing a good real-time controllers of the traffic lights in an urban environment. The parameters of the AR process take different values depending on the mode of traffic operation making this a hybrid stochastic process. We assume that mode switching occurs according to a first-order Markov chain. This thesis proposes both offline techniques and online techniques for estimating the parameters of the AR models describing the flow rate for each mode and the entries of the transition matrix of

this Markovian model process. The offline technique uses expectation-maximization (EM) algorithm while the online algorithm utilizes a particle filtering approach. Both techniques have been validated using actual traffic flow from Belgium and Indonesia, as well as data generated a VISSIM traffic simulator.

The EM method is an iterative algorithm working in two steps. In the E-step the sufficient statistics of the complete data are estimated. This estimated is then used to obtain a new estimated of the parameters in the so called M-step which is minimizing the prediction error. This parameter estimate is then fed back to E-step, and so the method iterates until convergence. In this thesis we do reformulate the EM approach for jump Markov model (JMM) in order to get a simpler and easier algorithm, changing the cost-function in the M-step *by adding a weighted term with corresponding smoothed inferences about the currently active mode* taking into account the probability with which an observation at time t_k corresponds to modes s.

The performance of any state estimator, including *Observation and Transition-based most likely modes tracking Particle Filter* (OTPF) for hybrid systems, depends on the prior knowledge of a good model for the plant dynamics and the noise characteristics, including the knowledge of the transition probabilities between discrete modes. We check the accuracy of a PF estimator, that uses the parameter values estimated by the EM algorithm as obtained in one time window, to obtain state estimates in another time window that is very close to the original time window (time window shift method), so close that one can expect that the identified model parameters are still valid. We do find that this adaptive PF does lead to a sufficiently accurate estimator, validating the usefulness of the proposed method for state estimation and for online control applications.

However, this EM offline parameter estimation along with the time-window shift technique has at least two disadvantages: (a) the performance of state estimator strongly depends on the length of the time window shift W, (b) this offline approach needs significant memory requirements and processing power for storing and processing large datasets. In practice W is so large that the adaptation of this method is too slow for feedback control applications.

To address the disadvantage, we extend the particle filter approach to an algorithm that achieve online joint state and parameter estimation. The extension of particle filters to joint state and parameter estimation for SHM is non-trivial. The conventional strategy is to add a random walk to the parameters and then augment the state-space with the parameters for joint estimation. The use of a random walk however increases the covariance of the parameters (dispersion), making the posterior distributions too diffuse. In this thesis we improve the natural kernel smoothing approach for reducing the covariance by optimally tuning the kernel smoothing factor. The optimal selection of the kernel parameter $h_k \in [0,1]$ maximally reducing the over-dispersion in the PF remains a difficult problem. The current practice for tuning of the smoothing factor h_k use ad-hoc rules defining a constant h_k for which optimally cannot be established w.r.t the online data. In this thesis we will introduce a systematic approach for choosing this tuning parameter h_k by using an on-line optimization algorithm based on Kullback-Leibler (KL) divergence. Since the hybrid model describes the different traffic modes, the transition probabilities matrix (TPM) of the traffic modes must be estimated. We use the Dirichlet distribution to generate particles for the PF estimation of this TPM. The combination of adaptive OTPF with Dirichlet based kernel smoothing leads to an efficient joint estimation of the TPM, of the other parameters and of the state.

The traffic control is developed in this thesis by integrating an identification and estimation technique approach into a control policy avoiding long queues that would lead spillbacks. This thesis develops controllers using a chance constraint based feedback strategy. In this proposed controller, the risk of spillbacks is limited by using a chance constraint in order to keep the queue length less than threshold to avoid spillback. We thus achieve a state feedback control of the traffic lights, adjusting their red/green phases to the actual queue lengths and traffic flow rates, while adapting in real time to changes in the SHM parameters. The performance of this controller is defined by the objective function with constraints on the inputs and states that should be satisfied in the presence of uncertainties. Since objective functions of this type are not convex we use a convex bounding approximation method for the probability distribution, leading to feasible solutions.

This thesis proposes a new control algorithm, probabilistically constrained predictive control. The propagation of probabilistic parameter uncertainties and exogenous disturbances through the stochastic hybrid model and the reformulation of probabilistic constraints to computationally tractable expressions are key issues in this stochastic control algorithm. The proposed stochastic control is demonstrated using a simulated case study and give satisfactory results compared to both *fixed controller* and *adaptive stochastic with different cost function*. The proposed controller is able to fulfill all the constraints including the chance constraint which ensures that the probability of the queue length on specific road exceeding a threshold always remains sufficiently small.

Samenvatting

Geavanceerde strategieën voor verkeersregeling kunnen leiden tot een efficiënter gebruik van de bestaande infrastructuur, met minder verkeershinder, minder uitstoot van uitlaatgassen, minder brandstofverbruik, en veiliger verkeer tot gevolg. Om het dynamisch gedrag van het verkeer te beschrijven in een grote stad moet de evolutie van het verkeer beschreven worden in de vele verbindingsstraten en kruispunten van het verkeersnetwerk. Dit is een complex modelleringsprobleem dat moet opgelost worden voor het ontwerpen van model-gebaseerde regelstrategieën. De performantieverbetering die kan bekomen worden met behulp van terugkoppelregeling hangt sterk af van de kwaliteit en de nauwkeurigheid van het model dat gebruikt wordt bij het ontwerp, en van de nauwkeurigheid van de model-gebaseerde toestandsschattingen die voor de terugkoppeling beschikbaar zijn.

In dit proefschrift ontwikkelen we een stochastisch hybriede model (SHM) om de evolutie te beschrijven van het stochastische proces van wachtrijlengte, en van de aankomst- en vertrekintensiteiten van verkeer. Dit SHM model laat toe om de evolutie te beschrijven van de wachtrijlengte op een kruispunt met verkeerslichten, en van de uitstroom uit een dergelijk kruispunt en dus van de interactie van de verschillende verkeerslichten en de verschillende kruispunten in een netwerk. Deze interactie wordt gestuurd zowel door discrete gebeurtenissen als door tjidsgebonden modellen. Gebeurtenissen omvatten onder meer de groen/rood omschakeling van verkeerslichten, veranderingen in het verkeersgedrag (vlot verkeer, opstopping, incidenten, afhankelijk van allerlei interne en externe fenomenen) en de momenten waarop wachtrijen leeg worden of terug beginnen aan te groeien. De intensiteit van aankomst- en vertrekprocessen kan voorgesteld worden door autoregressieve processen (AR) waarbij de AR-model parameters afhangen van een werkingsmode. Er wordt verondersteld dat de werkingstoestand evolueert volgens een eerste-orde Markov proces.

De parameters van de SHModellen moeten op basis van verkeersmetingen geschat worden vooraleer die modellen gebruikt kunnen worden voor het wegfilteren van de ruis op de verkeersmetingen en voor het voorspellen van de wachtrijlengte aan kruispunten. De stochastische evolutie van de verkeersstromen, over verschillende opeenvolgende cycli van de verkeerslichten, moet geschat worden ten einde te kunnen bepalen wat de effecten zijn van regelbeslissingen voor de verkeerslichten op toekomstige wachtrijlengtes. Om goede regelaars

in reële tijd te bekomen is deze schatting van de parameters van het SHModel dus nodig, en deze schatting moet gebeuren aan de hand van meetgegevens uit een stedelijk verkeersnetwerk. In dit proefschrift worden zowel online als offline methodes ontwikkeld om de transitiematrix van het eerste-orde Markovproces van verkeersmodes te schatten, samen met de AR parameters inelke verkeersmode. Het offline schattingsalgoritme gebruikt de verwachtingsmaximisering (EM) methode, terwijl het online algoritme gebruik maakt van particle filtering (PF). Beide methodes worden in dit proefschrift gevalideerd aan de hand van reële verkeersmetingen (waarbij zowel metingen in Belgische als in Indonesische steden wordt gebruikt). Tevens werden verkeersdata gegenereerd door een VISSIM simulator gebruikt om de schattingen te valideren.

De EM methode is een iteratief algoritme. Bij elkee E-stap wordt een voldoende statistiek geschat die de metingen volledig beschrijft. Deze schatting van de voldoende statistiek wordt dan in de M-stap gebruikt om die modelparameters te schatten die de kleinste voorspellingsfout geven. Deze nieuwe parameterwaarde wordt dan gebruikt voor een nieuwe E-stap, die leidt tot een nieuwe M-stap, tot convergentie wordt bekomen. Ten einde deze EM-methode toepasbaar te maken voor de SHMmodellen van verkeer wordt in dit proefschrift de brekening van de kostenfunctie, die in de M-stap wordt geminimiseerd, vereenvoudigd door gebruik te maken van gewogen benadering van de afgevlakte schatting van de meest waarschijnlijke actieve mode. Dit levert eenvoudiger berekeningen op dan gebruik te maken van de geschatte kans dat elke mode voorkomt. Het is dan eenvoudig om via Lagrange-vermenigvuldigers de transitiematrix te schatten die de kost minimiseert. Deze schatter wordt de *Observation and Transition-based most likely modes tracking Particle Filter* (OTPF) genoemd.

Zoals voor alle toestandsschatters hangen de prestaties van OTPF sterk af van de a priori kennis van een goede benadering van het te schatten model, van de ruiskarakteristieken, en van de transitiematrix voor de modes. In dit proefschrift valideren we de offline EM-methode ook door schattingen, bekomen aan de hand van metingen in een tijdsvenster van lengte W, te gebruiken in een PF dat de toestand voorspelt in een later tijdsvenster, dicht genoeg bij het eerste venster zodat dat de parameters niet te veel zijn veranderd. Deze time-window shift methode blijkt voldoende nauwkeurig te zijn voor toestandsschatting, en kan dus in principe gebruikt worden voor online regelingtoepassingen. In praktijk vereist de EM offline methode in combinatie met de time-window shift methode echter een grote waarde W, en dus ook zeer veel geheugencapaciteit en rekencapaciteit, waardoor de methode te traag is voor online toepassingen.

Om dit probleem te verhelpen worden de PF toestandsschatters in dit proefschrift uitgebreid tot online schatters van zowel de modelparameters als van de toestand. De klassieke manier om dit te bekomen bestaat erin dat de parameters geïnterpreteerd worden als toestandsveranderlijken door er een stochastische wandeling bij op te tellen. Dit leidt echter tot erg diffuse a posteriori kansvedelingen, met een te grote covariantie van de geschatte parameters. Dit niet-triviale probleem wordt vaak opgelost door kernel smoothing toe te passen. Dit vereist een gepaste keuze van de kernel parameter $h_k \in [0,1]$ om de onzekerheid in de voorspellingen van het PF zo klein mogelijk te maken. In praktijk gebeurt dit tegenwoordig op een ad hoc manier, waarbij de kernel parameter een constante waarde aanneemt gekozen op basis van ervaring, zodanig dat geen optimaliteit kan aangetoond worden. In dit proefschrift gebruiken we een online optimiseringsalgoritme dat de Kullback-Leibler divergentie voor de schatters minimiseert. Aangezien onze modellen hybriede systeemmodellen zijn moeten we ook online de transitiematrix schatten van het Markovproces dat de werkingsmodes beschrijft. Dit vereist het frequent genereren van mode transities in de gegenereerde particles, wat in dit proefschrift gerealiseerd wordt door gebruik te maken van Dirichlet toevalsgetallen. In dit proefschrift wordt aangetoond dat de combinatie van een adaptieve OTPF techniek met optimaal kernel smoothing en Dirichlet variabelen leidt tot een efficiënte online schattingsmethode die tegelijkertijd de transitiematrix en de AR-parameters en de toestand schat.

Ten slotte combineert dit proefschrift de voorgestelde schattingstechnieken met een terugkoppelregeling die lange wachtrijen vermijdt vooraleer ze aanleiding kunnen geven tot blokkering van stroomopwaartse kruispunten. Hiertoe wordt een toestandsterugkoppeling ontworpen die de kans begrenst dat de wachtrij een bepaalde grenswaarde overschrijdt. Deze regelaar past de rood/groen fractie aan van de verkeerslichten op basis van de geschatte wachtrijlengte en van de geschatte verkeersintensiteiten, en schat tegelijkertijd de veranderingen in de parameters van het SHModel dat gebruikt wordt voor de toestandsschatting. Het ontwerp van dergelijke regelaars vereist de minimisering van een kostenfunctie die de kans bevat dat een grenswaarde wordt overschreden. Dergelijke kostenfuncties zijn niet-convex. In dit proefschrijft ontwikkelen we een convexe benadering ervan die de optimisering computationeel haalbaar maakt. Zo slagen we erin om een regelalgoritme te ontwerpen dat gebruik maakt van gezamenlijke schatting van toestand en modelparameters voor het SHModel, en dat een kostenfunctie minimiseert, rekening houdend met begrenzingen op de kans dat een wachtrij te lang wordt. Via een case study gebaseerd op een VISSIM simulatie wordt aangetoond dat de voorgestelde regelaar voldoet aan alle voorwaarden, en leidt tot beter verkeersgedrag dan een regelaar met vaste cyclus voor de verkeerslichten en beter dan een adaptieve regelaar met een niet-probabilistische kostenfunctie.

1 Introduction

The topic of this thesis is the design of an adaptive controller for a signalized intersection in urban traffic network, using joint state and model parameter estimation while optimizing traffic performance taking into account the risk of global network interactions causing performance deterioration. This introductory chapter 1 provides the motivation for the research presented in this thesis, and justifies the approach taken in this research for solving the traffic signal control problem. The first part of this chapter provides a motivation for the traffic control problem under investigation. Next this chapter provides an overview of the state of the art, discusses some open problems and explains briefly the objectives and contributions of this thesis. The next section presents some background knowledge and some notation useful for understanding the approach taken in this thesis, and explains the interrelationship between the different parts of the thesis, highlighting the contributions made in each chapter.

1.1 Problem statement/Motivation

The increasing economic and social activity leading to an increasing number of vehicles in metropolitan areas result in over-saturated and highly congested traffic conditions at some locations in the urban traffic network during the peak periods, or even during a large part of the day The service quality deteriorates drastically for the users of the network, increasing the average travel times. Increased level of pollution lead to deteriorating living conditions in the cities. Provision of new infrastructure is not deemed to be a sustainable solution. Thus, a more efficient utilization of the existing infrastructure, using advanced online controllers, is a crucial ingredient towards sustainable urban mobility.

Alleviating congestion and reducing delay in urban traffic networks by implementing feedback control in order to optimally utilize the existing infrastructure is one of the currently vital issues of traffic researcher and practitioners. Many studies have addressed this issue over the last decades, and many different types of feedback control have been proposed in order to deal with this issue of online management and control of large-scale urban network [1],[2],[3].

A large urban network consists mainly of two types of elements: link roads and signalized intersections. The most obvious feedback control action that can be used to influence the evolution of the traffic flow is the selection of the switching times of the traffic lights at the signalized intersections. Designing good feedback controllers requires a sufficiently accurate model of the dynamic behavior of the traffic flow in all elements of the network, link roads and intersections. Modeling the traffic flow dynamics for each element in an urban network, at the time scale relevant for online control, with a large number of links and intersections is a complex task.

Recently there has been a lot of interest in obtaining a reproducible relationship between traffic flow and density, known as the macroscopic fundamental diagram (MFD) occurring at the network level under certain conditions (e.g. homogeneous spatial distribution of the congestion). This work postulates that average traffic flow and average vehicle density are related by a well-defined unimodal curve provided that vehicles are approximately uniformly distributed across space in the network under consideration. This notion had been initially proposed by Godfrey [63] in the 1960s, and has recently been validated using real traffic data [100]. This MFD relationship has been used for the design of perimeter control controlling the flow of traffic allowed to enter the network under control. This MFD relationship is valid for average traffic flow and density, averaged at a fairly long time scale, covering several cycles of the traffic lights in the network under control. In the perspective of MFD, urban traffic networks are inherently unstable when congested/oversaturated. This instability causes a natural tendency towards spatially inhomogeneous vehicle distributions. Hysteresis patterns may also arise for which flows during the onset of congestion are significantly different from those during the dissipation of congestion [101] further leading to an additional form of instability.

This thesis works based on the control theoretic framework to design controllers selecting red/green fraction of the traffic lights, and hence models describing traffic evolution at the time scale of cycles of traffic light is needed to build. This faster time scale requires more detailed models than MFD which is based on the averages over slower time scales. The traffic model used for selecting red/green switching times should covers the variability of the traffic flow over successive cycles causes the dynamics of propagating of queues. The goal of traffic signal is ensuring that as few vehicles as possible have to stop, that the flow of vehicles through the intersection should be as "smooth", as uninterrupted, as possible. While at the same time guaranteeing safety. In order to keep the model computationally tractable to the research presented in this thesis considers one intersection at a time, both in the model and in the control design. However, the limitation of the model to one intersection can be relaxed by selecting local controllers that avoid spillbacks. Spillback are long queues at one intersection that block upstream intersections, causing bad local behavior to spread throughout the network. This thesis focuses on the behavior of traffic flows in one single intersection but considering traffic flows coming from the adjacent intersections and developing traffic signals that avoid spillbacks using a control theoretic approach. If these goals can be achieved it will be possible to achieve stable traffic behavior over a longer period of time. In a sense the local traffic control of the traffic lights can be used to indirectly modify the MFD in the surrounding area, reducing the risk of instability.

1.2 Some background material

Because this thesis has a cross-disciplinary aspect it is useful to introduce below some of the basic background material from the field of traffic control.

1.2.1 Basic Variables

In traffic engineering, traffic stream variables fall into two broad categories. *Macroscopic variables* describe the traffic stream as a whole; *microscopic variables* describe the behavior of individual vehicles of pairs of vehicles within the traffic stream.

The three principal macroscopic parameters that describe a traffic stream are

(1) *flow* or *flow rate* α is defined as ratio of the number of vehicles passing a location in one or more lanes of a road during a specified time interval divided by the length of that time interval, and it is often expressed as “vehicles per unit time” (in this thesis normally veh/sec).

(2) *speed* v is defined as a rate of motion in distance per unit time. Travel time is the time taken to traverse a defined section of roadway and it is expressed as m/s

(3) *density* ρ is defined as the number of vehicles occupying a given length of lane or a road, generally expressed as vehicles per m (or veh/km) or vehicles per km per lane. Density is difficult to measure directly and it is often computed from speed and flow rate measurements, according to the formula: density = flow/speed ($\alpha = \rho.v$). While density is difficult to measure directly, modern detectors can measure *occupancy* that is defined as the proportion of time that a detector is "occupied" or covered, by a vehicle in a defined time period.

In urban network with signalized intersections, the operation of the traffic light can be specified using the following variables [94],[96] :

(1) *Cycle length* : This is the time needed to complete one full running of all the active phases.

(2) *Approach:* is a single road, along which traffic arrives at an intersection; the flow of traffic along that approach lane, and heading towards a certain outflow direction is called a movement.

(3) *Phase*: This is the interval of time, part of a cycle, during which a certain set of "movements" are allowed coming from a subset of all the approaches, are allowed to cross the intersection. There are normally at least two phases per cycle.

When considering the interaction between neighboring signalized intersections it is important to specify the "synchronization" between the traffic lights. In many case the same cycle length is used for neighboring intersections. Then one needs to specify the shift in switching times between neighboring traffic lights

(4) *Offset* : this is the relationship between the start (or finish) of the green phases in successive sets of signals within a coordinated system.

This thesis uses only rate of flow as a traffic variable. The flow rates along the different approaches, along with information of traffic light (i.e. cycle length and phase) are sufficient to define the evolution of the queue-lengths (provided the initial queue length is known). The queue-lengths in turn specify the increment in delay caused by congestion and queueing behind red lights; this delay is used to determine the cost function to be minimized by a good traffic controller.

Note that the research presented in this thesis focuses on reducing the increment in the delay that is due to congestion, since there is an inevitable minimal travel time for all traffic.

.

1.2.2 Macroscopic Fundamental Diagram

The Macroscopic fundamental diagram (MFD) is fundamental tool for designing an adaptive control approach, to improve urban mobility, and to relieve congestion. MFDs can be used for estimation of the level of service on road networks, for designing perimeter control, and macroscopic traffic modeling [3]. In the past fundamental diagrams (FDs) were usually used for freeway traffic in order to express the flow rate at a given location along a freeway link to the density in a small area around that location. More recently Daganzo (see below) has extended this idea to MFDs describing the total outflow along all exit lanes leaving a specified region, as a function of the total number of vehicles inside that region.

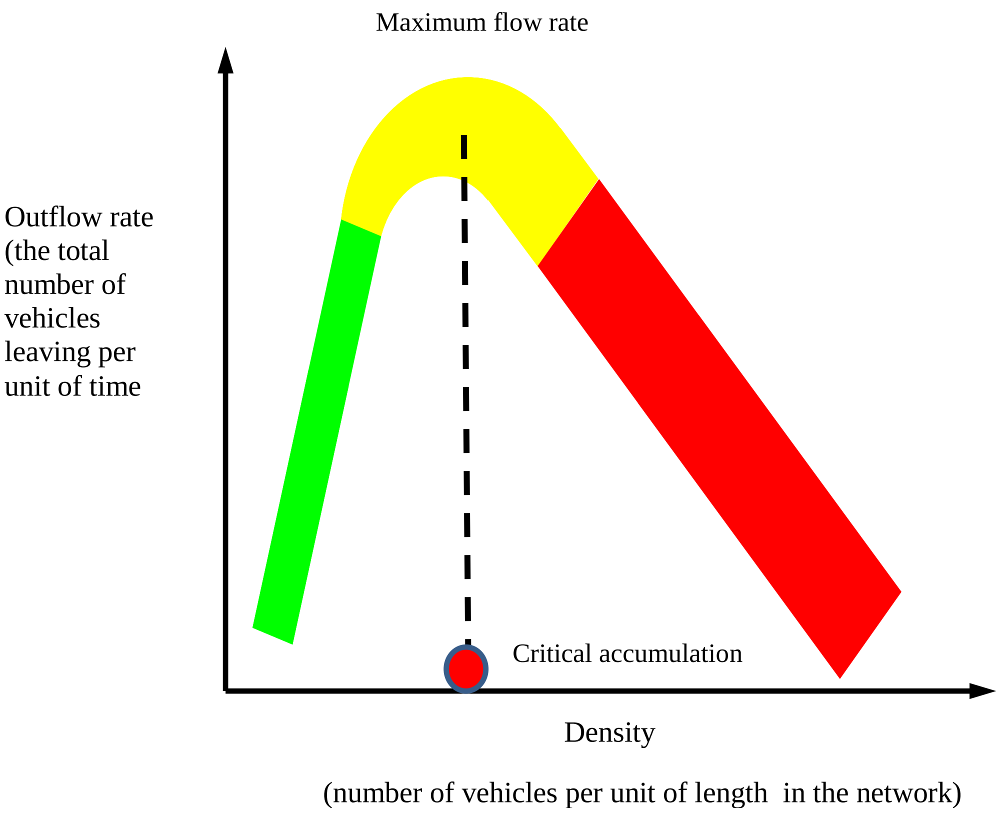

Figure.1.1 A Typical MFD of a network

The experiments and simulations made by Geroliminis and Daganzo [100] suggested that at least in some instances average flow and density were indeed related to a reproducible curve, which has come to be known as “the Macroscopic Fundamental Diagram” (MFD).

A typical MFD is illustrated in Fig 1.1. Like the critical density in a link fundamental diagram of a freeway, the MFD shows the value of the total number of vehicles in a region that leads to the maximal outflow from that region. This maximum is an important parameter called critical accumulation, marked as a red spot in Fig 1.1. The value of critical accumulation and maximum outflow rate can be directly calculated once the MFD for a given region has been derived from the observable traffic data of the network. Two traffic modes of operation can be simply distinguished by the point of critical accumulation, one mode corresponding to free flow condition (marked as a green curve, where vehicles travel at their own desired speed) and one corresponding to congestion (marked as a red curve, where interaction with other vehicles block causes significant speed reduction).

In traffic theory the location where the value of critical accumulation is reached corresponding to the maximum flow rate is also called state turning point (marked as a yellow curve). If the traffic variables are located on the left side of the state turning point, the road network is in the free flow condition. Otherwise the network is congested. That means, the rough traffic modes can be identified as long as the MFD is known.

Remark: It should be highlighted that in this thesis, traffic modes s (i.e. free flow and congestion condition) are defined, in line with the terminology of system and control theory, are defined only according to the values taken by traffic flow, since thesis considers only traffic flow as a variable without considering the density.

1.2.3 Adaptive Stochastic Control

This section introduces the adaptive stochastic control problem addressed in this thesis. This control design is based on the stochastic hybrid model (SHM) which will be developed for urban traffic in this thesis. This multi-mode SHM model describes the evolution of the queue-length at a signalized intersection using as states both continuous traffic flow variables (like flow rates) as well as a discrete variables like traffic operations mode and state of traffic lights. Continuous variable evolve according to difference equations (in the discrete time representation used in this thesis), and discrete variables which evolve according to a discrete event (automaton) model. Consider the following a generic hybrid model defined by:

State update equation is defined in one mode *s*; the full state model includes the evolution of the value of the modes *s*:

$$x_{t_{k+1}} = f_s(x_{t_k}, u_{t_k}, w_{t_k}), \qquad k=1,\ldots N-1 \tag{1.1}$$

where $x_{t_k} \in \Re^n$ is the unobserved n-dimensional state vector at time t_k, $u_{t_k} \in \Re^{n_u}$ is the control input at time t_k, $w_{t_k} \in \Re^{n_w}$ is the white process noise at time t_k, f_s is the state function vector in mode s and N is the time horizon. The initial state, x_0, is assumed to uncertain with a known probability distribution $p(x_0)$. Equation (1.1) implies that the state process $\{x_{t_k}\}$ is a Markov process of order one.

The mode evolution is defined as a first order Markov process with given transition matrix Π defining mode transitions.

At each time step, a noisy measurement of the state is taken, defined by:

$$y_{t_k} = g_s(x_{t_k}, v_{t_k}), \qquad k=1,\ldots N-1 \tag{1.2}$$

where $y_{t_k} \in \Re^p$ and $v_{t_k} \in \Re^{n_v}$ are the measurement output and noise of sensor at time t_k, g_s is the observation function vector in mode s. The measurement vector $\mathbf{y}_k = y_{t_k}, \ell = 1,\ldots,k$ is independent, conditional upon the state process $\mathbf{x}_k = \{x_{t_k}, \ell = 1,\ldots,k\}$. Thus, equation (1.1) and (1.2) can be equally described as

$$\begin{cases} p_s(x_{t_k} \mid x_{t_{k-1}}) \\ p_s(y_{t_k} \mid x_{t_k}) \end{cases} \tag{1.3}$$

where $p_s(.|.)$ is the conditional probability density function at mode s. The subscript s will be dropped whenever convenient for clarity presentation. Equation (1.3) is more suitable for representing in the uncertainty as a probability distribution which can be achieved using the joint state-parameter estimation algorithms, as a main contribution in this thesis. By integrating joint state-parameter estimation with stochastic control leads to the framework of adaptive stochastic controller.

In urban traffic case, the states vector x_{t_k} are the queue lengths, traffic flow, mode of operation and the measurement vector y_{t_k} are the traffic flows during green/red period, both in each links of a signalized intersection. In real urban application, the traffic flows system can be very complex and they can evolve according to different dynamics at different times. These dynamics can be either free flow and congestion condition and it is important to know (i.e. for efficient traffic signal) which dynamics the system is in and when a change of dynamics occurs, since for example, it might be important to know if a system is in a faulty condition or not. This topic will elaborate more detail in next chapter.

For the controlled system model defined by (1.1-1.2) one must select at time t_k a control value $u_{t_k} \in U$ as a function of the observations $\{y_n, n < t_k\}$ so that the objective function φ(X,U) is minimized. We assume that φ(X,U) is a convex function of X and U, where X and U are a compact representation for the trajectory of all states and inputs, generated by the closed loop system (1.1-1.2) with the selected control law u_{t_k} $\{y_n, n \leq t_k\}$. Please note that control value is a value of u at time t_k and control law refer to a specific algorithm to deliver control value. Since the process and measurement noise are random variables, the state trajectory and the sequence of selected control inputs are also random variables. Consequently the performance of the system must be defined based upon the expected value of the objective function φ, over X and U:

$$\text{Min E } [\varphi(X,U)] \tag{1.4}$$

In selecting the control values u_{t_k} one should also take into account that a set of constraint functions ψ_i on the state trajectory and control inputs must be satisfied at all time t_k with k=1,…N-1,and i is the number of constraint functions.

The final aim of traffic signal control is to keep the queue length less than a given threshold to avoid spillback via a feedback control with a chance constraint. It means that defining traffic signal control as only local control, but include chance constraints limiting the risk of long queues that could cause interactions between neighboring intersections, where these interactions might cause global performance deterioration even though the controllers without interaction would be well behaved.

1.3 State of the art and open problem

Many different real-time urban traffic control (UTC) strategies have been proposed up to now, but there is still space for further improvements. In this thesis, the measurement data in urban traffic area generated from The Sydney Coordinated Adaptive Traffic System

(SCATS) [82]. SCATS is an adaptive urban traffic management system that synchronizes traffic signals to optimize traffic flow across a whole city, region or corridor. SCATS controls traffic in a region consisting of several interacting intersections by controlling three parameters for traffic lights in its domain: cycle time, phase split and offset. By using three parameters, SCATS can be a distributed system, in the sense of “cooperating local controllers”. Various types of detectors at intersections can be used video cameras as is used for the Jakarta data used in this thesis. The SCATS algorithms uses the output of these detectors to calculate as controlled variables the following: Original Volume (OV), Degree of Saturation (DS) and Reconstituted Volume (VK). OV is the number of vehicles that have passed over the detector in one traffic cycle. DS is defined as the ratio of effectively used green time to the total available green time. VK is a measurement of how many cars should have passed over the detector at the stop line for the current DS. A DS value less than 1 will show that current traffic flow for the current cycle is not saturated, otherwise the current traffic flow is oversaturated. One popular alternative to SCATS is the Split Offset Optimization Technique (SCOOT) [83] is a system developed in the UK in 1970’s. The SCOOT architecture is organized into a fully centralized model, in which data is passed from the traffic lights directly to a regional data center where processing occurs. SCOOT also employs a second set of induction loops located anywhere from 50 to 300 meter before the stop-line (unlike SCATS which only utilizes the single set at the stop-line). The second set of detectors provide a count vehicles approaching to the stop line so that the queue length can be predicted more accurately.

The traditional UTC intends to minimize the traffic flow through the whole network, using as an the objective function the total average delay time experienced by the vehicles in all queues. Most other papers define the control strategies based on one single traffic model without taking into account that different control strategies are suitable for different traffic modes. As explained above that traffic behavior consists of at least two different traffic modes: free-flowing and congested condition, and possibly also modes corresponding to certain faults. In order to be able to represent different modes of traffic behavior and mode switches, a mode-dependent model is needed to design a controller that automatically adapts the red/green cycles to the current mode of traffic behavior. The adaptation can be realized using parameter adaptation of the models, using estimation of the current mode of operation, that are used to predict the effects of the control actions. This allows the generation of good control laws. Both mode estimation and adaptation are missing in most other papers up to now. Understanding the possibility of improvement control design by using mode estimation and adaptation is one of research goal in this thesis.

Remark: The term “traffic mode” or “mode” as used in this thesis should not be confused with the term ”mode” which sometimes refers in traffic engineering as the type of vehicle used by the participants in the mobility process that one studies (car or truck or bus or motorcycles or bicycle, etc). In this thesis we only consider vehicles as participants in the mobility process, even though in practice of course also other vehicles (like motorcycles in the traffic study for Jakarta) play an important role.

Over the last 10 years several model based approaches to urban traffic control have been proposed. Varaiya, in his paper [105], proposed the *max-pressures* feedback policy for the control of arbitrary network of signalized intersections. At the beginning of each cycle, a controller selects the duration of every stage at each intersection as a function of all queues in the network. A stage is a set of permissible (non-conflicting) phases along which vehicles may move at pre-specified saturation rates. Demand is modeled by vehicles entering the network at a constant average rate with an arbitrary burst size and moving with pre-specified average turn ratios. The movement of vehicles is modeled as a “store and forward” [102] queuing network. A controller is said to stabilize a demand if all queues remain bounded. It differs from other network controllers analyzed in the literature in three respects. First, max-pressure requires only local information: the stage durations selected at any intersection depends only on queues adjacent to that intersection. Second, max-pressure is provably stable: it stabilizes a demand whenever there exists any stabilizing controller. Third, max-pressure requires no knowledge of the demand, although it needs turn ratios. The max-pressure policy is *decentralized*: the decision at any intersection depends only on the queue adjacent to that intersection; the policies in the other studies are *centralized* [103].

Daganzo, in his paper [3], proposed perimeter control using MFDs to avoid performance deterioration due to congestion. The basic idea of perimeter control is to limit the inflow into a region, by setting the traffic lights at the inflow intersections, so that the number of vehicles inside the region always remains close to or lower than the critical accumulation, thus maximizing the network outflow. The idea is to hold traffic back (via prolonged red phases at traffic signals) upstream of the links to be protected from oversaturation. Following up on this idea, Geroliminis [86] pursued a model-predictive control (MPC) approach in deterministic setting and Haddad [87] used a robust control framework, in order to optimally select the allowed into inflow into region.

However, all the approaches work without addressing the congestion problem by keeping the queue length less than a threshold avoiding spillback (when demand exceeds capacity queues fail to clear during the allocated green times creating oversaturated traffic conditions and spillbacks occur when growing queues at the downstream link block the arrivals from the upstream link such that vehicle queues cannot discharge at capacity, although the signal phase is green). However, this spillback phenomenon needs to be estimated in real-time. The evolution of queue-length can be estimated (and predicted in a probabilistic sense) through the SHM hybrid model (the mode-dependent model) for traffic evolution. In order to be able to obtain good predictions it is necessary to estimate the model parameters. Using the estimated parameters allows the application of good adaptive local controllers. By ensuring that there is as little spillback as possible, this controller avoids network interaction that could deteriorate the performance of the overall network even though each local intersection seems to have a good controller. Our approach can be classified as a decentralized approach as max-pressure policy.

The research presented in this thesis uses the framework of system and control theoretic approach to design a model based feedback controller addressing this problem, trying to minimize some cost function. Several recent papers have considered the online optimal control of the red/green cycle for traffic lights. In [24], Wardi proposed a gradient-descent algorithm that adjusts a given mode schedule by changing multiple modes over time-sets of positive Lebesgue measures, thereby avoiding the inefficiencies inherent in existing techniques that change the modes one at a time. Vazquez et al [15] proposed a fluid flow petri net (PN) model to develop model predictive control in a deterministic setting. Ilya [9] and Haddad [12] proposed a Discrete Event Max-Plus approach with an assumption that the arrival and departure flow rate is a constant. This assumption differs from the assumption in this thesis, which uses a jump Markov model in order to be able to capture the time-varying behavior in traffic flow. This thesis address the congestion problem in one link of a signalized intersection by adaptive stochastic control with chance constraint. Compared to [24], our approach has a fixed cycle time and a fixed cyclic phases while Wardi proposed the algorithm that can change the cycle time and change a cyclic phases where the objective is to minimize a cost-performance functional defined on the state trajectory as a function of the sequence of modes and the switchover times between them which makes the computational load is much demanding.

1.4 Objectives and contributions

The research presented in this thesis follows the idea of Daganzo [3] but merely to act only on one exit intersection of the region under control trying to maximize the outflow from this intersection along a major road in an urban area, while avoiding spillbacks. The basic idea is to apply a local low level intersection controller that is compatible with perimeter control for the region to which the intersection belongs. In fact good local control designed according to the approach proposed here have the potential, intuitively, to modify the shape of the regional MFD leading to better performance of the perimeter controller. Using the mode-dependent model (or hybrid model) of the traffic flow, the queue length trajectories can be estimated, provided the parameters of mode-dependent model have been identified. By knowing the evolution of queue-length the risk of spillback can be estimated and monitored in real-time. The traffic control is developed by integrating an identification and estimation technique approach into a control policy of long queues prior to spillback occurrence. This thesis advances the state-of-the-art of traffic signal control in urban links by integrating a queue-length predicting approach into a chance constraint based feedback control strategy reducing the risk of spillbacks. The stochastic model used in this thesis averages traffic variables over the duration of one cycle of the traffic lights (i.e., green and red phases) and proposes a fluid flow model as an appropriate model at this time scale. By averaging traffic flow variables over the duration of one cycle then the dynamics of propagating of the maximal queue lengths per cycle can be properly captured as a function of the flow generated by the neighboring intersection. This dynamic model is a prerequisite for a model based feedback controller that achieves robust performance despite errors in our predictions (due to model inaccuracy, state noise, or to sensor noise).

As indicated by the title of the thesis, there are two main themes that are discussed in this thesis, namely stochastic and adaptive properties of the closed loop system.

Model predictive control (MPC) is widely used in many applications owing to its ability to deal with multivariable complex dynamics and to incorporate system constraints into the optimal control problem. However, parametric uncertainties and exogenous disturbances are ubiquitous in real-world systems, and the classical MPC framework is inherently limited in its ability to account for uncertainties [88]. This consideration has led to the development of numerous robust MPC formulations that deal with uncertainties. The robust MPC approaches can be broadly categorized as deterministic and stochastic approaches based on the representation of uncertainties and the handling of constraints. In deterministic robust MPC approaches (for a review see, e. g., [89]), uncertainties are often assumed to be bounded. The control law is determined such that the control objective is minimized with respect to worst-case uncertainty realizations, and/or such that the constraints are satisfied for all admissible values of uncertainties. Hence, robust MPC approaches discard statistical properties of uncertainties and are conservative [90] *if the worst-case uncertainty realizations have a small probability of occurrence.*

In stochastic MPC (SMPC) approaches (e. g., see early work [91],[92],[93]) uncertainties are described by probability distributions (instead of by describing these variables as belonging to bounded sets, as is done in robust control) This can be implemented by using the joint state-parameter estimation algorithms proposed in this thesis. The research presented in this thesis develops such a stochastic approach to MPC not only alleviating the conservatism of worst-case control, but also enabling proper tuning of the performance robustness by allowing pre-specified levels of risk during operation. The trade-off between control performance and robustness is achieved using chance (or probabilistic) constraints, which ensure the satisfaction of constraints with a desired probability level [5], [6].

Adaptive control means that the closed loop system has capabilities to do automatic parameter adaptation to changing system modes, provided we can estimate the model parameters “online” with sufficient accuracy. By putting the algorithm for parameter estimation and state prediction in the feedback control loop then the stochastic control achieves the desirable properties of an adaptive stochastic control. The research presented in this thesis has developed two approaches to do joint parameter estimation and state prediction, the essential parts of feedback loop of an adaptive controller. Firstly, the research reported in this thesis has developed an offline approach by using expectation maximization (EM) for maximizing the posterior probability $p(\theta \mid y_{i\ldots I})$ of the parameters θ of the stochastic hybrid model. Secondly, it has been developed an online approach which is an extension of adaptive particle filters for combined state and parameter estimation of the hybrid model (mode-dependent model)of the queue-length dynamic.

In summary, this thesis introduces the following contributions:

a. It develops a stochastic hybrid model for traffic flows and queue lengths in each links of a signalized intersection
b. It develops joint parameter and state estimation algorithm for the stochastic hybrid model of queue-length and traffic flow dynamic, both using the offline EM algorithm and the online particle filter technique.
c. It integrates the joint state-parameter estimation with stochastic control with chance constraint leading to adaptive stochastic control.

1.5 Layout

The material presented in this thesis is partly based on the author's paper [14,15,18,38,39,40]. The thesis is divided into seven chapters. The content of the remaining chapters is briefly summarized as follows:

Chapter 1. The present chapter has described the basic problem investigated in the research leading to the thesis.

Chapter.2 proposes a stochastic hybrid model (SHM) for urban traffic flow and queue length dynamics at a signalized intersection. This SHM is using only flow rates averaged per red/green phase of the traffic light as basic traffic variables, characterizing these flow rates using a mode-dependent first-order AR stochastic process. The parameters of the AR process take different values depending on the mode of traffic operation. Queue lengths can be predicted using the flow rates per cycle, together with information on the red/green phases of the traffic light.

Chapter.3 provides a compact review of the Bayesian recursive estimators both for states estimation and parameters estimation, and focusing on the particle filtering (PFs) approach for sequential estimation. The extension of PFs to state-parameter estimation is studied and developed for dynamics models, including hybrid models like SHM. The applicability of PF for these models has been studied by using kernel smoothing in order to reduce the dispersion. This chapter presents the existing technique of joint state-parameter estimation for two cases of state-parameter estimation of traffic flow with a different sampling time updates for two different cases of actual measurement data.

Chapter.4 proposes an offline parameter estimation technique for hybrid dynamic system model as a powerful approach for capturing the complicated dynamics of urban traffic flow, including many sources of uncertainty. The model parameters characterize traffic flow conditions that can be classified into three modes and the switch between these three modes is controlled by a first-order Markov chain. The study reported in this chapter investigates the proposed approach by using actual traffic flow data and confirms its validity by showing that the 'smoothed inferences' technique and a particle filter based on the identified model provide satisfactory state estimation and correctly capture the random variation of the traffic flow.

Chapter.5 presents an online technique for joint parameter and state estimation for a stochastic hybrid model of the queue-length dynamics and its application to queue-length estimation and prediction at signalized intersections in urban traffic networks. The main novel contributions of this chapter are:

(a) developing the OTPF algorithm along with an optimal kernel smoothing method (using online optimization) for estimating the AR parameters;

(b) using a Dirichlet distribution to generate the PF samples that allow reliable estimates of the transition probability matrix of the modes in the SHM. The proposed method is validated in terms of the prediction accuracy for traffic flow and queue length, for a case study generated by a VISSIM traffic simulator.

Chapter.6 describes an original approach to adaptive stochastic control by integrating it with the previously derived joint state-parameter estimation technique and with probabilistic risk constraints. The risk constraints are by themselves non-convex making the problem computationally intractable. The convex bounding method is used to replace the individual chance constraints by a conservative convex approximation, that is successfully implemented in the stochastic control for avoiding queue spillback in the critical intersection. The joint state-parameter estimation is used to detect in real-time the link with a queue larger than a critical queue length q_{cr}. The control that we propose has been compared to other existing feedback controllers for traffic lights, and found to achieve better performance. This risk constrained has the potential to decrease the likelihood of gridlock on the network.

Chapter.7 summarizes the important points in each chapter, mainly focusing on the potential benefits of the traffic feedback control developed in this thesis by integrating an identification and estimation technique approach into a control policy of long queues, reducing the risk of the occurrence of spillovers through a chance constraint. Some future extensions of the theoretical aspect of control problem studied here together its possible application in the city of Surabaya conclude this chapter.

2

A Stochastic Hybrid Model : Traffic flow and Queue-length dynamic

Urban traffic networks consist mainly of two types of elements: link roads, forming approaches to the signalized intersections, and signalized intersections. The first part of this chapter models the traffic flow dynamics using the fluid flow approach in each link road forming an approach to an intersection. A model of the interaction between the traffic flow along the approaches of a signalized intersection and the red/green cycles of the traffic lights at that intersection is described in the second part of this chapter. This interaction is the basic phenomenon determining the evolution of the queue lengths. Capturing this propagation of queue lengths into a well-defined dynamic model is an important part of the design of a model based feedback controller for the traffic lights. The last section of this chapter discusses the operation of a signalized intersection by defining a set of event describing the evolution of the discrete states, modeled as an automaton. The overall model of the intersection behavior is then a stochastic hybrid model (SHM).

2.1 Introduction : a fluid flow model approach

This section introduces a stochastic model that considers the traffic flow variables at locations along a link road, and at all entrance and exit locations of the signalized intersections. This traffic flow variable is defined (and can be measured) by dividing the number of vehicles crossing a given location during each red phase, viz. during each green phase of the traffic light. Reliable speed and density data are not available in the companies/institutes that cooperate with us in this research. and therefore we build models for the flow variables only . More specifically, a generic traffic flow is defined as the ratio $\alpha_{t_k} = {N_{t_k}}/{(t_{k+1} - t_k)}$ where N_{t_k} counts the number of vehicles that pass the given location in the interval $[t_k, t_{k+1})$. It means that traffic flow is a result of averaging over an interval which here is the duration of one phase of the signal. A fluid flow model (FFM) is proposed as an appropriate model at the time scale of successive phases of the traffic lights. This FFM describes the evolution over time of the traffic flow at a given location, in a given link or at the entrance or exit point of an intersection, by a continuous random variable which expresses the average rate α_{t_k}, expressed in vehicles per sec, at which vehicles pass a location at time t_k. This fluidization of traffic flow variables avoids working with large integers, approximating integer numbers of vehicles by real number as proposed in [26].

Paper [12] proposes FFM with discrete-event max-plus model while an FFM with stochastic hybrid model is proposed in [14]. The FFM in [15] and [12] do not consider random variation of the flow rates, whereas the FFM in [14] assumes that the evolution of arrival flow rate and departure flow rate are well defined by the parameters of the certain stochastic processes that has capability to describe varying of intensity flow.

Note that in the FFM, there is an implicit assumption that the vehicles travel approximately at equal distances from each other during the interval $[t_k, t_{k+1})$ since the flow rate is assumed constant during $[t_k, t_{k+1})$. This is an approximation that is only acceptable for sufficiently small values of the time increments t_{k+1}-t_k (and it may not really be true for the duration of a red or green phase of a traffic light) but it reduces the computational complexity of our algorithms a lot since we do not have to consider individual vehicles. This assumption implies that the flow rates α_{t_k} are approximately constant over the intervals $[t_k, t_{k+1})$. As previously mentioned our FFM operates at the time scale of observing variables each time the traffic signals switches from red to green or vice versa, and counts the number of passing vehicles during red or green phase. The implicit assumption in the FFM model is that vehicles are uniformly distributed over the time interval used for defining the flow rate, which in this case would mean that vehicle arrivals are approximately uniformly distributed over one red or green phase of the traffic light.

Most of the past work related to traffic signal control design are based on the assumption that traffic flow is deterministic [8],[9],[10],[11]. Several stochastic model for traffic flow have also been proposed such as platoon-based model (PBM) [13], fluid flow model (FFM) [12,14]. The PBM uses as basic unit a platoon of vehicles that travel at approximately the same speed, closely following each other. The PBM describes the state of an urban traffic network at a given time by specifying the location of the head of each platoon and the size of these platoons along link roads additionally specifying the queue sizes at each approaching lane of each intersection, and the red/green state of the traffic lights. Using the PBM, we can predict δ_L time unit ahead (where δ_L is the time that it takes to drive the entire length of link road L at the maximum legal speed) the arrival times of the platoons which already entered on the link road L upstream of the intersection. [54]. For predictions more than δ_L, the time units ahead the traffic flow coming from the upstream intersection can only be replaced with average platoon arrivals models unless information about the state of the upstream intersection would be available. This reduces the PBM information to the information available in a fluid flow model. The PBM considers as event times the arrival of the head or the tail of a platoon (as explained by N.Marinica in his thesis in [54], which allows efficient simulations and calculations. The information of a PBM representation is initially richer than that of FFM, but it may "diffuse" eventually after passing through several intersections. Therefore in this thesis we work with fluid flow rates everywhere.

The effectiveness of FFM will be elaborated below based on analysing some available traffic data that observe the passage time of each vehicle and

(1) Analysing the available measurement data

We are analysing the available measurement in the area of Dendermonde, Belgium as shown in Fig.2.1 (data courtesy of the "Vlaamse overheid" - Belgium). The data generated by tube sensors located at various locations along the road network, as indicated by the bars in Fig.2.2. Each time the axle of a vehicle crosses the sensor location this generates a pressure pulse that is detected by the sensor, and whose occurrence time (up to the msec) is stored for later analysis. This allows us define the traffic flow fairly accurately (though the errors due to missing detections and false detections are significant). The format of the available measurement data is as follows:

037303 13/05/2009 08:00:02 446 02009 037304 13/05/2009 08:00:02 705 02009 037305 13/05/2009 08:00:04 740 02009 037306 13/05/2009 08:00:05 018 02009 037307 13/05/2009 08:00:17 858 02009 037308 13/05/2009 08:00:18 061 02009 037309 13/05/2009 08:01:11 061 02009 037310 13/05/2009 08:01:11 345 02009	This program read text file has the following format: % 037303 13/05/2009 08:00:02 446 02009 % (1) (2) (3) (4) (5) % (1) number/counter % (2) date % (3) time in H:M:S format % (4) time in msec % (5) identifier

The format shows that we have available the almost exact passage times of successive axles of vehicles at the sensor location as indicated by third column (number 3) and thus could obtain the number of passing vehicles over very short time successive intervals.

Even though we did not have available the data on the switching times of the traffic lights at each intersection it was possible to approximately determine these switching times by observing the changes in number of vehicles passing different exit sensors of the traffic light. The sensors with number 02010/02012 as indicated by green box, in the link between intersection A and B, measured the number of vehicles during a cycle $[t_k, t_{k+1})$ defining the departure flows μ_{t_k} from intersection A in the direction of intersection B. Arrival flows λ_{t_k} of intersection B can be determined by using sensors 03004/03005 during a cycle $[t_k, t_{k+1})$ by assuming the constant delay δ determined as ratio (L/v), where L= 250 and speed $\approx$ 60 km/hours .

In order to determine the queue-length, it is important to classify the arrival and departure flow during green and red period (see below) .

Herein after the period of the cycle is noted $[t_{2k}\ t_{2(k+1)})$. Because the k-th cycle consists of a green phase and a red phase then the cycle can be defined as $[t_{2k}\ t_{2(k+1)}) = [t_{2k}\ t_{2k+1}) \cup [\ t_{2k+1}\ [t_{2k}\ t_{2k+1}\ t_{2(k+1)})$, where k=1,2,3… is the index number of cycle. Denote moreover:

(a) during green phase t_{2k+1}-t_{2k} = $T_{g,k}$ the arrival flow is noted $\lambda_{i,t_{2k}}$ and departure flow is noted $\mu_{i,t_{2k}}$

(b) during red phase t_{2k+2}-t_{2k+1} = $T_{r,k}$, the arrival flow is noted $\lambda_{i,t_{2k+1}}$. The departure flow is of course 0 in this case.

where i is number of the lane. In next section this notation would be explained in more formally. Based on information $\lambda_{i,t_{2k}}$, $\lambda_{i,t_{2k+1}}$ and $\mu_{i,t_{2k}}$ along with green phase $T_{g,k}$ and red phase $T_{r,k}$ then the queue-lengths in that link in signalized intersection B can be estimated by assuming that the initial length of queue is known and assuming that there is no measurement noise:

$$q_{i,t_{2k+2}} = \max[\,q_{i,t_{2k}} + (\lambda_{i,t_{2k}} - \mu_{i,t_{2k}})T_{g,k}\,, 0] + \lambda_{i,t_{2k+1}}\,T_{r,k} \qquad (2.1)$$

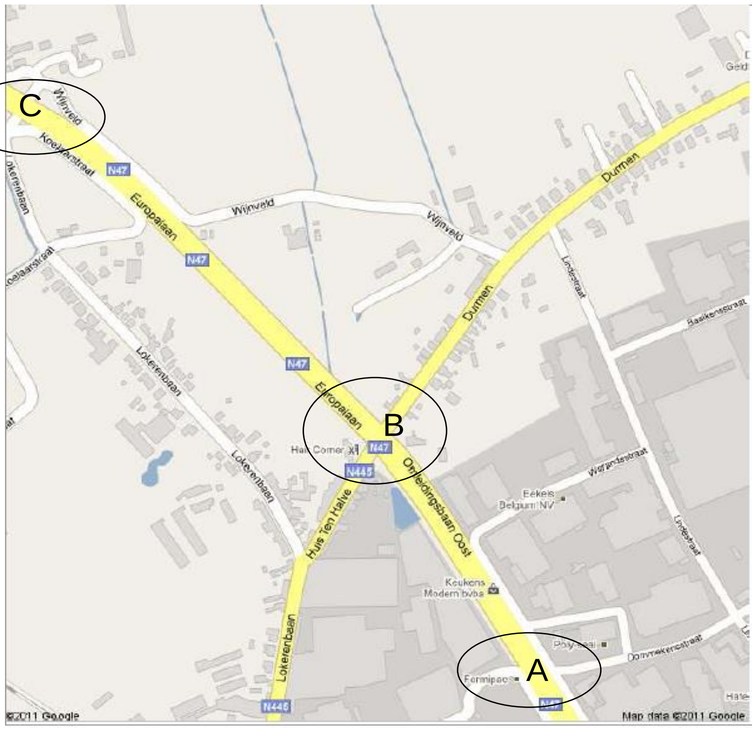

Figure 2.1. Area of Dendermonde where the measurements are taken

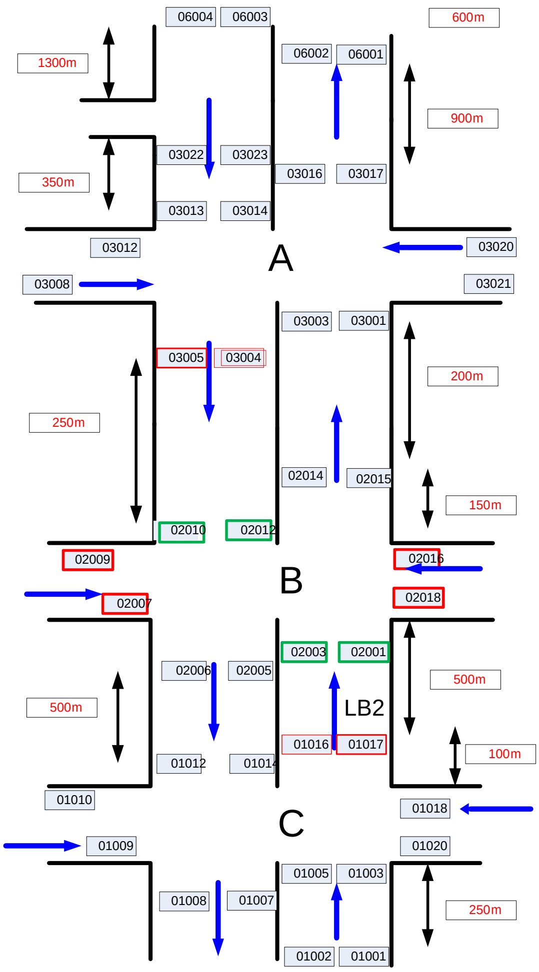

Figure 2.2 Sensors Location

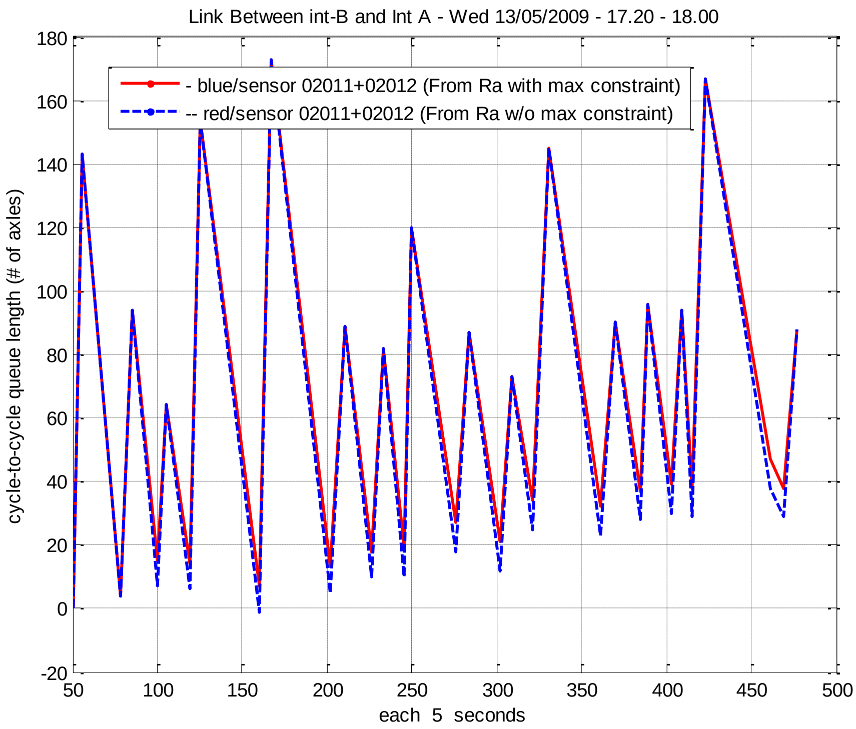

Figure 2.3. Evolution of queue length

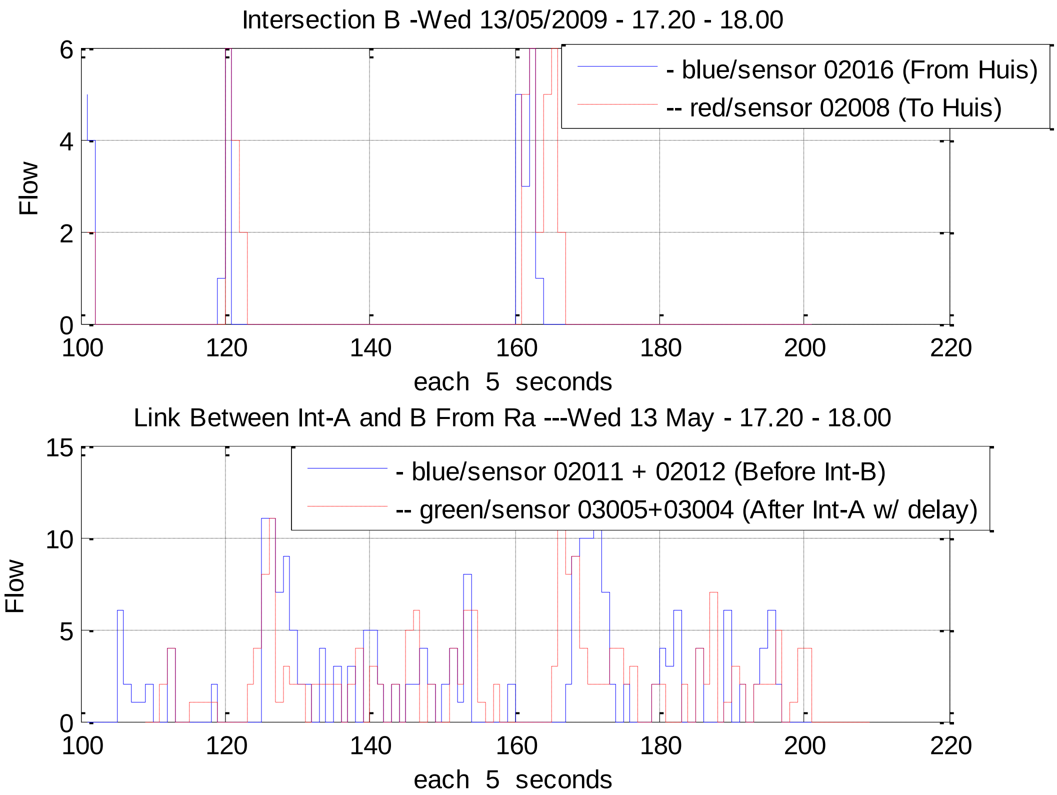

Figure.2.4 Arrival flow **λ** and departure flow **μ** at intersection B

Fig.2.3 shows the comparison the estimated queue-length between with and without *max-constraint*. In addition, from the data describing each departure flow at intersection B (as indicated by sensor numbers 02010/02012 in the major road from and 020016/02018 in the minor road), we are able to extract the red and green periods as shown in Fig.2.4. Careful analysis of these traffic data from the Dendermonde case study has shown that FFM is a useful and efficient model for calculating the important variables in traffic network such as queue-lengths, traffic flows, green/red phase and cycle [t_{2k} $t_{2(k+1)}$). This means that counting the number of vehicles during each green and red phase provides an efficient means for predicting traffic behavior at a reduced the computational cost. FFM reduces the computational complexity a lot. However, note that the approximation of piecewise constant flow rates $\lambda_{i,G,2k}$, $\lambda_{i,R,t2k+1}$ and $\mu_{i,G,t2k}$ over successive interval [t_{2k} $t_{2(k+1)}$) leads to piecewise linear trajectories $q_{i,t}$, which of course is only an approximation of the more complex true behavior. In other words the deviation from this uniformity can be seen as some random noise.

2.2 A Stochastic hybrid model

This section focuses on the framework of a stochastic hybrid model for urban traffic. The stochastic hybrid systems (SHM) considered here represent stochastic behavior both in the discrete event dynamics and in the continuous dynamics. Their broad modeling expressivity has enabled various researchers to use stochastic hybrid systems as models in various application domains such as system biology, traffic networks and smart grids. There are several modeling formalisms for stochastic hybrid systems. In [22], a general type of stochastic hybrid systems, whose continuous dynamics is described by diffusion stochastic differential equation is presented. Another framework that is also popular is the piecewise deterministic Markov processes [23]. This framework does not feature stochastic differential equations, but uses in between random jump at random times a deterministic continuous dynamics model described by ordinary differential equations. Autonomous switched-mode hybrid dynamical systems are proposed by Wardi in [24]. This switch-mode SHM can be used for analyzing the problem of controlling the switching times of traffic lights as shown in Sutarto in [25].

The behavior of the traffic flow in urban networks is characterized by stop-and-go phenomena resulting from green/red switching at signalized intersections, from irregular arrival stream of vehicles, and from complicated interactions between conflicting traffic streams, and from external disturbances like accidents or incidents that modify the traffic carrying capability of the road. In this thesis, the irregular arrival stream of vehicles is characterized only by flow rate as a traffic variable measured in “vehicles per time unit”. This is obtained by dividing the measured number of vehicles by the duration of the time unit. The time unit can be the (possibly variable) duration of green and red phase of traffic light of a signalized intersection. This flow rate variables can be modeled by mode dependent AR process. These mode represent traffic conditions remaining unchanged during a sufficiently long period of time, typically many time units. As a consequence, the term mode here refers only to the traffic flow with different levels of intensity that can be grouped into few categories (in this thesis we consider sometimes 2 modes, but sometimes the operational conditions are grouped into 3 modes). Classification into 2 modes refers to the traffic situation representing the free-flowing condition and the congestion condition. This can be somehow related to the MFD, considering the cases where the traffic density is to the left or the right of the maximal flow rate. Note however that our model only considers the flow rate as a traffic variable, while MFD is characterized by flow rate and density. In the offline approach in chapter 4 we also consider a 3^{rd} mode called by the faulty condition. The faulty condition can be caused by the *significant change* both in the number of vehicles passing through or in unexpected changes in the time unit (which can occur for traffic responsive signals). This significant change can be increasing/decreasing and shows up in the data set as a sudden increase in the variance of traffic data.

2.2.1 Queue Length Dynamic

The flow rate along with traffic light variables (i.e. cycle length and phase) are used to define the evolution of the queue-lengths. As discussed above (cf. equation (2.1)) the queue-length evolves as a piecewise linear functions, being the integral of the difference between arrival and departure rate; these arrival and departure rates are described by stochastic AR model with mode dependent parameters, which remain constant during each time interval $[t_k, t_{k+1})$; the mode changes are modeled by a first order 3-state Markov process. The overall traffic model is thus a jump Markov model. In this chapter we describe in detail this stochastic hybrid modelling for the traffic flow along one particular approach route to a signalized intersection, and indicate how this model is useful in controlling the operation of a signalized intersection since we want to control the signalized intersections so as to minimize the average delays which in turn depend on the queue length trajectories.

In this modeling, we assume the following simplification:

a) we do not consider the classification of vehicles;
b) ignore the yellow and all red
c) we ignore the problem of left turns, and the influence of pedestrian crossings.
d) we use observation on traffic flow only

A typical signalized intersection and a red/green sequence are shown in Fig.2.5. The intersection is controlled according to two phases. During phase A, the traffic signals T_1 and T_3 have a green light in the interval $[t_{2k}, t_{2k+1})$ while in phase B during the interval $[t_{2k+1}, t_{2(k+1)})$, traffic light T_2 and T_4 have a green light. In both phases, the cycle has two states: green and red (note that for simplicity we ignore in this thesis the yellow period; we also ignore the complications due to left turning traffic).

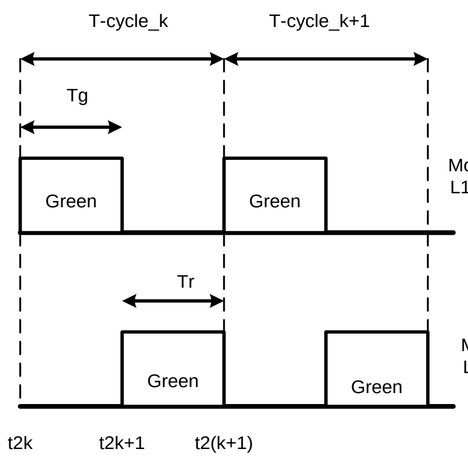

Figure.2.5.a Traffic Signal Sequence

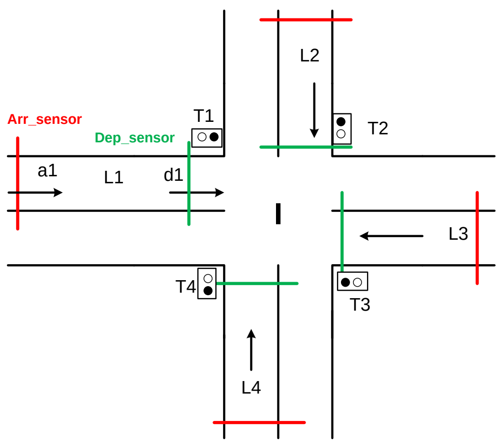

Figure.2.5.b Intersection with incoming lanes

The time instants when the traffic signals T_1 and T_3 initiates a green period and T_2 and T_4 begins red period are t_0, t_2, t_4….(or t_{2k} with k=0,1,2…). The time instants when the traffic signals T_1 and T_3 initiates a red period and T_2 and T_4 begins green period are t_1, t_3, t_5….(or t_{2k+1} with k=0,1,2…).

Notation:

Subscript i=1,2,3,4 indicates the number of the approach lane L_i, while the Greek character λ or μ indicates that it is an arrival or departure flow. Subscript *t* indicates the real time, while t_k denotes the starting time of red and green phases; remember that these flow rates remain constant during an interval $t \in [t_k, t_{k+1})$;

example-1:

The arrival rate of vehicles in traffic movements L_1 and L_3 in the interval $[t_{2k}, t_{2k+1})$ are $\lambda_{1,t_{2k}}$ and $\lambda_{3,t_{2k}}$ and in the interval $[t_{2k+1},t_{2(k+1)})$ are $\lambda_{1,t_{2k+1}}$ and $\lambda_{3,t_{2k+1}}$. The departure rate of vehicles in movements L_1 and L_3 in the interval $[t_{2k}, t_{2k+1})$ is $\mu_{1,t_{2k}}$ and $\mu_{3,t_{2k}}$.

example-2:

The arrival rate of vehicles in traffic movements L_2 and L_4 in the interval $[t_{2k}, t_{2k+1})$ are $\lambda_{2,t_{2k}}$ and $\lambda_{4,t_{2k}}$ and in the interval $[t_{2k+1},t_{2(k+1)})$ are $\lambda_{2,t_{2k+1}}$ and $\lambda_{4,t_{2k+1}}$. The departure rate of vehicles in movements L_2 and L_4 in the interval $[t_{2k+1}, t_{2k+2})$ is $\mu_{2,t_{2k+1}}$ and $\mu_{4,t_{2k+1}}$

Consequently, one can see that $t_{2k+1}-t_{2k} = T_{g,k}$ and $t_{2k+2}-t_{2k+1} = T_{r,k}$, where k is a cycle index. Therefore, $T_{g,k}$ represents the green time and T_r represents the red time in traffic signal T_1 and T_3. A cycle length C_k=C is equal to $T_{g,k}+T_{r,k}$, where in this thesis, cycle length is constant. Furthermore, $\lambda_{1,t_{2k}}, \lambda_{3,t_{2k}}, \lambda_{1,t_{2k+1}}, \lambda_{3,t_{2k+1}}, \mu_{1,t_{2k}}, \mu_{3,t_{2k}}, \lambda_{2,t_{2k}}, \lambda_{4,t_{2k}}, \lambda_{2,t_{2k+1}}, \lambda_{4,t_{2k+1}}$, $\mu 2_{R,t_{2k+1}}$, $\mu_{4,t_{2k+1}} \geq 0$ and $t_{2k} < t_{2k+1} < t_{2k+2}$, for all k.

Formulation of the queue length trajectories can be expressed equivalently as (a) a piecewise affine model and as (b) a max-plus model.

(a) Piecewise Affine model

The evolution of the queue length, for traffic signal T_1 and T_3, in movements L_1 and L_3 is obtained by:

$$\frac{dq_i}{dt} = \begin{cases} (\lambda_{i,t} - \mu_{i,t})1(q_{i,t\_}), & t \in [t_{2k}, t_{2k+1}] \\ \lambda i_{i,t}, & t \in [t_{2k+1}, t_{2k+2}] \end{cases} \tag{2.2}$$

for i=1,3 and k=0,1,2…

1(.) is the indicator function defined as [70]

$$1(z) = \begin{cases} 1 & if\ \ z > 0 \\ 0 & otherwise \end{cases} \tag{2.3}$$

where $z = q_{i,t\_}$ This indicates that queue-length in movement *i* is bigger than zero. Notation $t\_$ in $q_{i,t\_}$ means that we consider the value of the queue length just before time t_{2k+1}, something that is well defined because q_t is a piecewise continuous function, with left hand limits.

Similarly, for traffic signals T_2 and T_4, the evolution of the queue lengths in traffic movements L_2 and L_4 are obtained by

$$\frac{dq_i}{dt} = \begin{cases} \lambda_{i,t}, & t \in [t_{2k}, t_{2k+1}] \\ (\lambda_{i,t} - \mu_{i,t}) 1(q_{i,t\_}), & t \in [t_{2k+1}, t_{2k+2}] \end{cases} \tag{2.4}$$

for *i*=2,4 and *k*=0,1,2,…

The relation of the queue length between the time instants t_{2k} and t_{2k+2} are represented by the following equations:

$$\begin{aligned} q_{i,t_{2k+1}} &= q_{i,t_{2k+1}} + \int_{t_{2k+1}}^{t_{2k+2}} \lambda_{i,t} \, dt \\ q_{i,t_{2k+1}} &= q_{i,t_{2k}} + \int_{t_{2k}}^{t_{2k+1}} (\lambda_{i,t} - \mu_{i,t}) 1(q_{i,t\_}) > 0 \end{aligned} \tag{2.5}$$

for *i*=1,3 and *k*=0,1,2…

The equation describing the relationship of the queue length at traffic movement L_2 and L_4 are obtained in a similar manner.

Please note that in (2.4) and (2.5), the arrival and departure flow rates are nonnegative values and the queue-lengths will never become negative (but can be equal to 0), thanks to the inclusion of $1(q_{i,t\_})$ in the equations describing the evolution of queue lengths. Note also that these equations justify equation (2.1) as written down in the preceding section.

(b) Max-Plus Model

We may write the complete model of queue-lengths evolution in movement L_1 (or L_3) as a function of the occurrence times of discrete events describing switching of traffic lights, and zero crossing of queue length variables:

at the switching time instant t_{2k+1} for *i*=1,3 and *k*=0,1,2… are given by

$$q_{i,t_{2k+1}} = \max[q_{i,t_{2k}} + (\lambda_{i,,t_{2k}} - \mu_{i,t_{2k}}) T_{g,k}, 0] \tag{2.6}$$

and at the switching time instant t_{2k+2}

$$q_{i,t_{2k+2}} = q_{i,t_{2k+1}} + \lambda_{i,t_{2k+1}} T_{r,k} \tag{2.7}$$

Equation (2.8) is equivalent to equation (2.1)

The queue lengths for movements L_2 (or L_4) *i*=2,4 and *k*=0,1,2… are given by

$$q_{i,t_{2k+1}} = q_{i,t_{2k}} + \lambda_{i,t_{2k}} T_{g,k} \tag{2.8}$$

$$q_{i,t_{2k+2}} = \max[q_{i,t_{2k+1}} + (\lambda_{i,t_{2k+1}} - \mu_{i,t_{2k+1}}) T_{r,k}, 0] \tag{2.9}$$

Equations (2.6-2.9) are complete hybrid models of queue-length evolution in each movement and where arrival and departure flows are stochastic and are represented by a jump Markov model (JMM) that will be described in next section.

Discussion: The evolution of queue length is important to design feedback control, especially to define the possible cost functions. In this thesis, as for traditional UTC, the cost function represents the total average waiting time experienced by all vehicles in all queues:

$$J = \frac{1}{t_N - t_0} \sum_{i=1}^{4} \frac{1}{\bar{\lambda}} \int_{t_N}^{t_N} q_{i,t}\, dt$$

where q is the queue length at movement i, $\bar{\lambda}$ is the average arrival flow rate, N is the number of time instants and t_N-t_0 is the time interval considered. One of the advantages of using criteria based on time averaged values is that the objective function has a finite value even if N or t_N tend to infinity, provided that the queue lengths remain finite (which will with very high likelihood be guaranteed under our risk constrained controller developed in chapter 6). This cost function is equivalent to cost function that represents the total average queue length over all queues.

2.2.2 Jump Markov Model (JMM) Structure

Different from the most of the past work on traffic control where one assumed that traffic flow is deterministic, this thesis defines the traffic flow rates as stochastic variables. This section describes in detail the complete stochastic hybrid model for one signalized intersection, also called further on the jump Markov model (JMM). In order to express this we use a mode dependent AR model for a generic traffic flow rate α_{t_k} (which could represent the arrival flow rate λ_{t_k} or the departure flow rate μ_{t_k}). Remember that the period of the cycle is noted [t_{2k} $t_{2(k+1)}$) which consists of green phase and red phase during a period of time when the traffic conditions, also called the mode *s* of operation of the traffic, remains unchanged, this variable α_{t_k} is modelled by a first-order autoregressive (AR) model:

$$\alpha_{t_{k+1}} = \beta(s) + \gamma(s)\alpha_{t_k} + \eta_{t_k} \tag{2.10.a}$$

(where $\beta(s)$ and $\gamma(s)$ are mode-dependent parameters to be identified, and η_{t_k} , $k = 1, 2, ...$ is an independently identically distributed sequence of zero mean Gaussian random variables with variance $\sigma^2(s)$, with $\sigma^2(s)$ also a mode dependent parameter to be identified). The fluid flow assumption of this paper implies that the traffic flow is constant during each interval [t_k,t_{k+1}): $\alpha(t) = \alpha_{t_k}$ for $t \in [t_k, t_{k+1})$. The traffic flow implication is that during any interval [θ, r) $\subseteq$ $t \in [t_k, t_{k+1})$ the number of vehicles that cross the location where this model is valid is $\alpha_k.(r-\theta)$, and that these vehicles are approximately uniformly distributed over this interval. The drift parameter, $\beta(s)$, indicates the mode shifting effect. The elements in the set of mode switching are the labels for discrete states/modes. The transitions probabilities will be defined later in the end of this section.

In this thesis, we use state-variables representation in order to be consistent to the representation in filtering and control theory. The AR process of traffic flow in (2.10.a) can be convert to state-space form as follow.

Define state vector:

$x_{t_k} = \{x_{c,t_k}, x_d\} = \{\alpha_{t_k}, s\}$, where x_{c,t_k} is a continuous state and *s* is a discrete state/mode, where *s*= {1,2,3}.

Focus on the continuous state, we can define state-space form of AR process as an innovation model described in the book [104, pp 109] below:

State equation $$x_{c,t_{k+1}} = \beta(s) + \gamma(s) x_{c,t_k} + \eta_{t_k} \tag{2.10.b}$$

Measurement equation $$y_{t_k} = \beta(s) + \gamma(s) x_{c,t_k} + \eta_{t_k} \tag{2.10.c}$$

Notice that the noise η_{t_k} is acting both process and measurement noise, i.e., we have so called *innovations model*. Examination of the equations (2.13.b) and (2.13.c) show that the $x_{t_{k+1}}$ is precisely y_{t_k}, so that $x_{t_{k+1}}$ is known given past measurement. The noise η_{t_k} is a sequence of independent zero mean random variables with known distribution that depends on sensor.

Proof:

$$
\begin{aligned}
y_{t_k} &= \gamma(s)\, x_{t_k} + \beta(s) + \eta_{t_k} \\
&= \gamma(s)\{\gamma(s)\, x_{t_{k-1}} + \beta(s) + \eta_{t_{k-1}}\} + \beta(s) + \eta_{t_k} && [\mathit{from}\ (2.10.b)] \\
&= \gamma^2(s)\, x_{t_{k-1}} + \gamma(s)\,\beta(s) + \gamma(s)\,\eta_{t_{k-1}} + \beta(s) + \eta_{t_k} \\
&= \frac{\gamma^2(s)}{\gamma(s)}(y_{t_{k-1}} - \beta(s) - \eta_{t_{k-1}}) + \gamma(s)\beta(s) + \gamma(s)\eta_{t_{k-1}} + \beta(s) + \eta_{t_k} && [\mathit{from}\ (2.10.c)] \\
&= \gamma(s)\, y_{t_{k-1}} - \gamma(s)\beta(s) - \gamma(s)\,\eta_{t_{k-1}} + \gamma(s)\beta(s) + \gamma(s)\eta_{t_{k-1}} + \beta(s) + \eta_{t_k} \\
y_{t_k} &= \gamma(s)\, y_{t_{k-1}} + \beta(s) + \eta_{t_k}
\end{aligned}
$$

The last equation is equivalent to (2.13.c)

Q.E.D

From time to time the mode *s*(t) of operation of traffic will change, due to external changes of the inflow rate, due to incidents that make the operation more or less efficient, or due to randomness. As previously mentioned FFM works with time scale $(t_{k+1} - t_k)$ of observing variables each time the traffic signals switches from red to green or vice versa, and counts N_{t_k} the number of passing vehicles during red or green period which basically can be seen as a ratio, $\alpha_{t_k} = N_{t_k} / (t_{k+1} - t_k)$ then the value of traffic flow α_{t_k} strongly depends both on N_{t_k} and $(t_{k+1} - t_k)$. This implies that when the traffic flow is increasing then it could be happen because the interval $(t_{k+1} - t_k)$ is decreasing or because the number of vehicles is increasing. This note is important because the measurement data generated in the Jakarta case study, treated in this thesis come from an adaptive traffic signal (e.g. SCATS system) which means that the duration $T_{G,k}$ and $T_{R,k}$ of successive green and of red periods vary over time, depending on the traffic condition.

The mode variable *s* used above should also be considered as a time varying random process. We assume that the mode changes only occur at time instant when traffic signal change from green to red, or vice versa, as shown in Fig.2.5.a when the value of the traffic flow rate α_{t_k} is updated by the AR equations. Therefore, we denote the mode as s_{t_k} in the interval $[t_k, t_{k+1})$. In the traffic flow model introduced in this subsection we consider 3 different values for the mode of operation $s_{t_k} \in \{1, 2, 3\}$:

- $s_{t_k} = 1$ denotes the desirable mode of operation where traffic is flowing freely without too much interference between successive vehicle;
- $s_{t_k} = 2$ denotes the congested mode where vehicles hinder each other significantly, and the system operates inefficiently;
- $s_{t_k} = 3$ corresponds to a faulty state, describing outliers in the behaviour, possibly caused by the *significant change* either in the number of vehicles passing through or in the duration of the time unit. This significant change can be increasing/decreasing and shows up in the data set by increases or decreases in the variance of traffic data.

This thesis assumes that the mode process s_{t_k} can be modelled by a first order Markov chain, i.e. at each time t_k the mode variable $s_{t_{k-1}} = i$ changes randomly to the value $s_{t_k} = j$ with a probability Π_{ij}= Prob$(s_{t_k} = j | s_{t_{k-1}} = i, s_{t_{k-2}}, s_{t_{k-3}}, ...)$ which only depends on the most recent mode (or Markov state) $s_{t_{k-1}}$, not on states further in the past.

Equation (2.13) with its interpretation $\alpha(t) = \alpha_{t_k}$ for $t \in [t_k, t_{k+1})$, together with the Markov chain model for the mode s_{t_k}, and the queueing model of equation (2.9-2.12) provides us with a complete mathematical model of traffic flow. The parameters of this model will be estimated in the next section according to the parameter estimation both EM method and PF method. In total there are 15 parameters to be estimated for one single approach lane L_i:

- For each mode $s \in \{1, 2, 3\}$ the AR model has 3 parameters, $\beta(s)$, $\gamma(s)$ and $\sigma^2(s)$, for total of 9 parameters to be estimated

- The transition matrix Π_{ij} of the Markov chain describing the mode process has 3 rows of 3 elements, satisfying the normalization condition $\sum_{j=1,2,3} \Pi_{ij} = 1, \forall i$, for a total of 6 free parameters to be estimated.

2.3 Conclusions

This chapter proposed a stochastic hybrid model for urban traffic flow and queue length dynamics at a signalized intersection. SHM is characterized only by flow rates as traffic variables. These flow rate variables can be modeled using a mode-dependent first-order AR stochastic process. The parameters of the AR process take different values depending on the mode of traffic operation. Mode switching occurs according to first-order 3-state Markov process. Classification into 3 modes refers to the traffic situation, namely the free-flowing, the congestion condition, and faulty condition. The queue-lengths evolves as a piecewise linear functions, being the integral of the difference between arrival and departure rate.

3
Bayesian recursive estimation via particle filtering

This chapter provides a compact review of the Bayesian recursive estimators both for states estimation and parameters estimation, and focusing on the particle filtering (PFs) approach as sequential estimation. PFs are flexible simulation based techniques that have become popular for approximating the computationally intractable integration needed for Bayesian recursive optimal filtering. However, standard PF methods assume knowledge of the model parameters while in real applications, the parameters are unknown and should be estimated. How to deal with unknown parameter to perform optimal filtering, especially in a model with a reasonably large number of parameters (like a SHM model), remains a challenging problem.

The extension of PFs to state-parameter estimation is studied and developed for dynamics model. The applicability of PF for these models has been studied in the research reported in this thesis by using kernel smoothing in order to reduce the dispersion which is caused by the need to add of random walk to the parameter model, thus causing covariance increases over time. The current practice for tuning of the kernel smoothing parameter h_{t_k} is still ad-hoc. This chapter presents the existing technique of joint state-parameter estimation for two cases of state-parameter estimation of traffic flow with a different sampling time updates for two different cases of actual measurement data. The result shows that the there is a need to develop and to extend the existing technique for stochastic hybrid model. This topic will be elaborated and validated in chapter 5.

3.1 Bayesian Estimation

Particle filters are a tool to perform approximate Bayesian estimation. Consider a system with a hidden state *X* and an observation *Y* related to the hidden state *X*. In a probabilistic setting, any information about the value *x* taken by the random variable *X* before any observation is made can be described using its prior distribution $\wp(X \le x) = \int_{-\infty}^{x} p_X(x)\,dx$. The conditional probability density of *Y* depends on *x* and is given by *p(y|x)* (meaning that $\wp$(Y$\in$[y,y+dy)|X=x)=p(y|x).dy) and is called *the likelihood function*. The likelihood is the model for the measurement system (the sensor) because it models how *Y* depends on *X*. Bayes' theorem states that, given the observation *Y*=y, the conditional density of *X* given that *Y*=y can be calculated by:

$$
\begin{aligned}
p(x \mid y) &= \frac{p(y \mid x)\, p(x)}{p(y)} \\
&= \frac{p(y \mid x)\, p(x)}{\int p(y, x)\, dx} \\
&= \frac{p(y \mid x)\, p(x)}{\int p(y \mid x)\, p(x)\, dx}
\end{aligned}
\tag{3.1}
$$

where *p(y)* is the total probability of the observed data and actually plays the role of a normalization constant.

The posterior density represents what is known about *X* given the observation *Y=y*. The minimum-mean-squared-error (MMSE) estimate or the Bayes' estimate of the realized value *x* of random variable X given the observation *y* is the conditional mean of the random variable X given the observation *y*:

$$\hat{X} = E(X \mid y) = \int x\, p(x \mid y)\, dx \tag{3.2}$$

For simplicity, the presentation given herein assumes integration over the total range of the Euclidean space. Numerical integration deals with the problem of numerically valuating general integrals over complex ranges (but for simplicity of notation we will use Euclidean space $\Re^n$),

$$I = \int_{\Re^n} s(x)\, dx = \int_{\Re^n} g(x).h(x).dx \tag{3.3}$$

where *h(x)* is a probability density, i.e. positive function that integrates to unity,

$$h(x) \ge 0, \qquad \int_{\Re^n} h(x)\, dx = 1$$

Transforming the evaluation of an integral into the problem of evaluating an expectation one needs a suitable factorization of the integrand *s(x)= g(x) h(x)*. This shows that any expected value can be approximated by the law of large number by a sample mean, leading to a Monte Carlo evaluation. In the Bayesian estimation context, the density of interest is the posterior density of the state/parameters given the observed data, i.e., *h(x)= p(x|y).*

The Monte Carlo method relies on the assumption that is possible to draw $N >> 1$ samples $\{x_i\}_{i=1}^N$ distributed according to the probability density *h(x)*. The Monte Carlo estimate of the integral (3.3) is formed by taking the average over the set of samples also called particles:

$$g_N = \frac{1}{N} \sum_{i=1}^{N} g(x^i) \tag{3.4}$$

Where *N* is assumed to be large. If the samples in the set $\{x^i\}_{i=1}^N$ are independent, g_N will be an unbiased estimate and will almost surely converge, as N $\rightarrow \infty$, to *I*,

$$\Pr\left(\lim_{N \to \infty} g_N = I\right) = 1 \tag{3.5}$$

by the strong law of large numbers.

In practice it is often difficult to generate samples with density *h(x)*. Importance sampling deals resolves this difficulty by using a proposal distribution *q(x)* which is easy to generate samples from. The only general assumption on the *importance function q(x)* is that its support set covers the support of *h(x)*, i.e., that *h(x)* > 0 $\Rightarrow$ *q(x)* > 0 for all $x \in \Re^n$. Under this assumption, any integral on the form (3.4) can be rewritten

$$I = \int_{\Re^n} g(x)\, h(x)\, dx = \int_{\Re^n} g(x) \frac{h(x)}{q(x)} q(x)\, dx \tag{3.6}$$

A Monte Carlo estimate is computed by generating $N >> 1$ independent samples from *q(x)*, and forming the weighted sum

$$g_N = \frac{1}{N} \sum_{i=1}^{N} g(x^i)\, w(x^i), \qquad \text{where} \qquad w(x^i) = \frac{h(x^i)}{q(x^i)} \tag{3.7}$$

are the *importance weights*.

If the normalizing factor of the target density *h(x)* is unknown, the importance weights in (3.8) can only be evaluated up to a normalizing factor [75,79] . Then, the weights can be formed using a function proportional to the target density and then normalized afterwards, forming the estimate

$$g_N = \frac{\sum_{i=1}^{N} g(x^i)w(x^i)}{\sum_{j=1}^{N} w(x^j)} = \sum_{i=1}^{N} \bar{w}(x^i)\, g(x^i) \tag{3.8}$$

where $\bar{w}(x^i) = \dfrac{w(x^i)}{\sum_{j=1}^{N} w(x^j)}$ *,i=1,…,N*

It is important to note that the target density approximation is given by

$$\bar{h}_N(x) = \sum_{i=1}^{N} \bar{w}(x^j)\,\delta(x - x^i)$$

3.2 Particle filter

Let us consider the filtering problem in the perspective of the random sample generation. Hence, the concept of time and dynamic models will be combined with random number generators.

A rather general state-space model is given by equation (1.1) and (1.2) and an alternative and useful formulation is given by (1.3), repeated here for convenience without take into account *s* mode in the representation:

$$x_{t_{k+1}} = f(x_{t_k}, \eta_{t_k}) \tag{3.9}$$

$$y_{t_k} = g(x_{t_k}, n_{t_k}) \tag{3.10}$$

$$\begin{aligned} x_{t_{k+1}} &\sim p(x_{t_{k+1}} \mid x_{t_k}) \\ y_{t_k} &\sim p(y_{t_k} \mid x_{t_k}) \end{aligned} \tag{3.11}$$

In formulation (3.11), the model can be seen as a probability density function, which describe both the dynamics and measurement equations. In order to formulate equation (1.1)-(1.2) in the form (3.11) we make the following observations,

$$p_m(x_{t_{k+1}} \mid x_{t_k}) = p_{\eta_{t_k}}(x_{t_{k+1}} - f_s(x))$$
$$p(y_{t_k} \mid x_{t_k}) = p_{n_{t_k}}(y_{t_k} - g_s(x_{t_k})) \tag{3.12}$$

Equation (3.12) represents the probability density function of noise process η and noise η measurement.

In filtering problem, the target density is given by filtering density,

$$h(x_{t_k}) = p(x_{t_k} \mid \mathbf{y}_{t_k}) = p(x_{t_k} \mid y_{t_k}, y_{1:t_{k-1}}) \tag{3.13}$$

where $\mathbf{y}_{t_k} = (y_{t_0}, y_{t_1}, ..., y_{t_k})$ and using Bayes' theorem (3.1) and the Markov property can be written as

$$p(x_{t_k} \mid \mathbf{y}_{t_k}) = \frac{p(y_{t_k} \mid x_{t_k}, \mathbf{y}_{t_{k-1}})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})}{p(y_{t_k} \mid \mathbf{y}_{t_{k-1}})} = \frac{p(y_{t_k} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})}{p(y_{t_k} \mid \mathbf{y}_{t_{k-1}})} \tag{3.14}$$

In order to handle the denominator in the above equations it has to be expressed using known densities. This can be accomplished by marginalizing the following equation with respect to $x_{t,}$

$$p(y_{t_k}, x_{t_k} \mid \mathbf{y}_{t_{k-1}}) = p(y_{t_k} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}}) \tag{3.15}$$

which corresponds to integrating (3.15) with respect to $x_{t,}$, resulting in

$$p(y_{t_k} \mid \mathbf{y}_{t_{k-1}}) = \int_{\Re^n} p(y_{t_k} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})\, dx_{t_k} \tag{3.16}$$

Furthermore, in order to derive an expression for the one step ahead prediction density $p(x_{t_{k+1}} \mid \mathbf{y}_{t_{k-1}})$ we employ the marginalization trick once more by integrating the following equations with respect to x_t,

$$p(x_{t_{k+1}}, x_{t_k} \mid \mathbf{y}_{t_{k-1}}) = p(x_{t_{k+1}} \mid x_{t_k}, \mathbf{y}_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_k}) = p(x_{t_{k+1}} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_k}) \tag{3.17}$$

resulting in the following expression

$$p(x_{t_{k+1}} \mid \mathbf{y}_{t_{k-1}}) = \int_{\Re^n} p(x_{t_{k+1}} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_k})\, dx_{t_k} \tag{3.18}$$

This equation is sometimes referred to as the Chapman-Kolmogorov equation. These expressions are important, hence we summarize the conclusions in the following theorem.

Theorem 3.1 *For the dynamic model given by*

$$\begin{aligned} x_{t_{k+1}} &\sim p(x_{t_{k+1}} \mid x_{t_k}) \\ y_{t_k} &\sim p(y_{t_k} \mid x_{t_k}) \end{aligned} \tag{3.19}$$

The filtering density $p(x_{t_k} \mid \mathbf{y}_{t_k})$ *and the one step ahead prediction density* $p(x_{t_{k+1}} \mid \mathbf{y}_{t_k})$ *are given by:*

$$p(x_{t_k} \mid \mathbf{y}_{t_k}) = \frac{p(y_{t_k} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})}{p(y_{t_k} \mid \mathbf{y}_{t_{k-1}})} \tag{3.20}$$

$$p(x_{t_{k+1}} \mid \mathbf{y}_{t_k}) = \int_{\Re^n} p(x_{t_{k+1}} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_k})\, dx_{t_k} \tag{3.21}$$

where

$$p(y_{t_k} \mid \mathbf{y}_{t_{k-1}}) = \int_{\Re^n} p(y_{t_k} \mid x_{t_k})\, p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})\, dx_{t_k} \tag{3.22}$$

In the filtering problem, we use (3.13) and (3.14) which suggests the following choices :

$$\underbrace{p(x_{t_k} \mid \mathbf{y}_{t_k})}_{h(x_{t_{t_k}})} \propto \underbrace{p(y_{t_k} \mid x_{t_k})}_{w(x_{t_{t_k}})} \underbrace{p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})}_{q(x_{t_{t_k}})} \tag{3.23}$$

where $h(x_{t_k})$ is target density, $w(x_{t_k})$ is the importance weight and $q(x_{t_k})$ is the importance function as indicated in equation (3.7).

Here we follow Table 3.1 to illustrate sequential estimation PF step by step as follow:

Algorithm of particle filter starts at time t=0 by initializing the particle and their corresponding weights according to

$x_0^i \sim p(x_0)$ *i=1,…,N*

$w_0^i = \frac{1}{N}$ *i=1,…,N*

Resulting in the following approximation

$$\hat{p}_N(x_0) = \sum_{i=1}^{N} \frac{1}{N} \delta(x_0 - x_0^i)$$

Table 3.1 PF algorithm

1. Initialization t=0. For i=1,… , N, sample $x_0^{(i)}$ from an initial distribution and set t=1. 2. Prediction • For i=1,…,N, sample $\tilde{x}_{t_k}^{(i)} \sim p(x_{t_k} \mid x_{t_{k-1}}^{(i)})$ • For i=1,…,N, evaluate the importance weights $\tilde{w}_{t_k}^{(i)} = p(y_{t_k} \mid \tilde{x}_{t_k}^{(i)})$ • Normalize the weights $w_{t_k}^i = \dfrac{\tilde{w}_{t_k}^i}{\sum_{i=1}^{N} \tilde{w}_{t_k}^i}$ 3. Resampling • Resample N news particle $\{x_{t_k}^{(i)}$ for $i = 1,\ldots, N\}$ with replacement from the set $\{\tilde{x}_{t_k}^{(i)}$ for $i = 1,\ldots, N\}$ according to the importance weights 4. Set t= t+1 and repeat from step 2

At time t_k , assume that the following approximation

$$\hat{p}_N(x_{t_{k-1}} \mid \mathbf{y}_{t_{k-1}}) = \sum_{i=1}^{N} \frac{1}{N} \delta(x_{t_{k-1}} - x_{t_{k-1}}^i) \tag{3.24}$$

is available from t_{k-1}. According to Table 3.1 we should now generate N i.i.d. samples $\{\tilde{x}_t^i\}_{t=1}^N$ from the importance density $q(x_{t_k})$. From (3.23) we have that

$$q(x_{t_k}) = p(x_{t_k} \mid \mathbf{y}_{t_{k-1}}) \tag{3.25}$$

In order to generate samples from $p(x_{t_k} \mid \mathbf{y}_{t_{k-1}})$ we will make use of the time update (3.20) in Theorem 3.1 in the following way,

$$
\begin{aligned}
q(x_{t_k}) = p(x_{t_k} \mid \mathbf{y}_{t_{k-1}}) &= \int p(x_{t_k} \mid x_{t_{k-1}})\, p(x_{t_{k-1}} \mid \mathbf{y}_{t_{k-1}})\, dx_{t_{k-1}} \\
&\approx \int p(x_{t_k} \mid x_{t_{k-1}}) \sum_{i=1}^{N} \frac{1}{N} \delta(x_{t_{k-1}} - x_{t_{k-1}}^{i})\, dx_{t_{k-1}} \\
&= \sum_{i=1}^{N} \frac{1}{N} \int p(x_{t_k} \mid x_{t_{k-1}})\, \delta(x_{t_{k-1}} - x_{t_{k-1}}^{i})\, dx_{t_{k-1}} \\
&= \sum_{i=1}^{N} \frac{1}{N}\, p(x_{t_k} \mid x_{t_{k-1}})
\end{aligned}
\tag{3.26}
$$

This implies that the proposal density can be chosen as

$$q(x_{t_k}^{i}) = p(x_{t_k} \mid x_{t_{k-1}}^{i}) \tag{3.27}$$

Selecting the proposal density as (3.27) is known as SIR (Sequential Importance Resampling). Hence, according to (3.27) the predicted particles $\{\tilde{x}_t^i\}_{t=1}^{N}$ are obtained simply by passing the filtered particles from the previous time instance $\{x_t^i\}_{t=1}^{N}$ through the process dynamics (1.3)

$$\tilde{x}_{t_k}^{i} \sim p(x_{t_k} \mid x_{t_{k-1}}^{i}) \qquad i = 1,\ldots,N \tag{3.28}$$

Or alternatively using the notation of dynamic model and this can be formulated as

$$\tilde{x}_{t_k}^{i} = f(x_{t_{k-1}}^{i}) + \eta_{t_{k-1}}^{i} \qquad i = 1,\ldots,N \tag{3.29}$$

where $\eta_{t_{k-1}}^{i}$ is a realization from the process noise $p_{\tau_{t_{k-1}}}(\eta_{t_{k-1}})$. The next step in algorithm PF is to compute the importance weights, which according to (3.23) are given by

$$\tilde{w}_{t_k}^{i} = p(y_{t_k} \mid \tilde{x}_{t_k}^{i}) \quad i = 1,..,N \tag{3.30}$$

The acceptance probabilities are then found simply by normalizing the importance weights, step 2 in Table.3.1

$$w_{t_k}^{i} = \frac{\tilde{w}_{t_k}^{i}}{\sum_{i=1}^{N} \tilde{w}_{t_k}^{i}} = \frac{p(y_{t_k} \mid \tilde{x}_{t_k}^{i})}{\sum_{j=1}^{N} p(y_{t_k} \mid \tilde{x}_{t_k}^{j})} \tag{3.31}$$

Then the following approximation for the filtering density is obtained:

$$\hat{p}_N(x_{t_k} \mid \mathbf{y}_{t_{k-1}}) = \sum_{i=1}^{N} w_{t_k}^i \delta(x_{t_k} - \tilde{x}_{t_k}^i) \tag{3.32}$$

where $w_{t_k}^i$ is given in (3.31). According to (3.31) $\{w_{t_k}^i\}_{i=1}^N$ are affected by the likelihood function $p(y_{t_k} \mid x_{t_k})$. This makes sense, since the this conditional density reveals how likely the obtained measurement y_{t_k} is, in case the present state would take the value x_{t_k}. The better a certain particle explains the received measurement, the higher the probability is that this particle is in fact close to the true current value of the state. The normalized importance weight $w_{t_k}^i$ may be interpreted as the probability of occurrence for each particle. In other words, the probability of particle $x_{t_k}^i$ being chosen at a single sample is approximately $w_{t_k}^i$ and after N samples $x_{t_k}^i$ will be multiplied approximately N $w_{t_k}^i$ times.

3.2.1 Resampling

The next step in the algorithm of Table.3.1 (step 3), generates a new set of particles $\{x_{t_k}^i\}_{i=1}^N$ approximating $p(x_{t_k} \mid \mathbf{y}_{t_k})$ by resampling with replacement among the predicted particles $\{\tilde{x}_{t_k}^i\}_{i=1}^N$, distributed according to the importance density. The goal is to maintain diversity among the samples, avoiding that the variance of the importance weights increases over time [69]. This increase in variance leads to degeneracy of the particle filter, when used over a long time with many recursions. In practice, after a certain number of recursions, all but one particle will have very small normalized weights. The set of samples then no longer provides a proper approximate representation of the conditional distribution. A suitable measure of variation for the importance weights or degeneracy of an algorithm is the effective sample size N_{eff} which is estimated as:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_{t_k}^i)^2}$$

where $w_{t_k}^i$ is the normalized weight obtained using (3.31). The value of N_{eff} lies in the interval $1 \leq N_{eff} \leq N$, so a small N_{eff} indicates degeneracy. N_{eff} is a measure of variation of the normalized importance weights. If only very few particles have significant weight while others are negligible, the N_{eff} is small; if all the particles are nearly equally weighted, then $N_{eff} \approx N$.

Resampling is a strategy to overcome this degeneracy of samples in SIS and is a crucial step in particle filtering algorithms when the effective samples size N_{eff} falls below a predetermined threshold [69]. Resampling eliminates particles with low importance weights and replicates samples with high importance weights. It involves mapping the set of weighted particles { $x_{t_k}^i$, $w_{t_k}^i$ } into a new set of particles $\left\{\hat{x}_{t_k}^i, 1/N\right\}$ with uniform weights. The resampling algorithm is a kind of "*black box algorithm*" that takes as input the normalized importance weights and particle indices and outputs new indices. It has nothing to do with the particles' dimension, values, and so on. There are four types of basic resampling approaches: 1) multinomial resampling, 2) systematic resampling, 3) stratified resampling, and 4) residual resampling. In this thesis we use systematic resampling, which is often preferred over the others thanks to its simplicity [66].

Table 3.2 Systematic Resampling

The steps are as follows:

1. Denote z^j as the j^{th} cumulative sum element of the weights: $z^j = \sum_{i=1}^{j} w^i$

   Note that $z^N=1$. Draw a single uniform sample, v, on the interval(0,1]. For i=1,2,…,N compute

   $$u^i = \frac{(i-1)+v}{N}$$

2. Set j=1. Perform the next steps for i=1,2,…,N.

```
if u^i<z^j
        x^i ← x^j
        i ← i+1
else
        j ← j+1
end if
```

Discussion

The performance of the PF heavily depends on whether the particles are located in the significant regions of the state space. The particles are propagated through the dynamic model and then weighted according to the likelihood function, which determines how closely the particles match the measurements. This problem becomes even worse when the initial estimation errors are large, for example, a few orders of magnitude larger than the sensor accuracy. Consequently, the basic PF quickly suffers the problem of severe particle degeneracy (the loss of diversity of the particles).

3.2.2 Joint State-Parameter estimation

Since a model parameter can be considered as a state that remains constant (or that varies only very slowly) it is possible in principle to use recursive Bayesian estimators, and hence also particle filters, for jointly estimating the current state of a stochastic system and its parameters. However some tricky computational problems must be overcome in order to successfully apply this idea.

To simplify the discussion, here we only treat the model with one single mode *s*, so that the parameters to be estimated include just one group of AR parameter $\theta=(\beta, \gamma, \sigma)$. The complete model for the dynamics evolution of traffic flow and queue-length evolution in (2.6-2.10), depends explicitly on model parameters only written in traffic flow equation (2.10) where α_k is the generic notation for arrival flow in movements L_i during green period $\lambda_{i,2k}$ and red period $\lambda_{i,2k+1}$ and also for departure flow $\mu_{i,2k}$. In this section, we use *k* as index rather than t_k to simplify the notation. The equation (2.10) can be rewritten as follow where for clarity the parameter θ is explicitly included:

$$x_k = f(x_{k-}, \theta, \eta_k) \ , \qquad k = 0,..., N-1; \tag{3.33.a}$$

$$y_k = g(x_{k-1}, \theta, n_k) \ , \qquad k = 0,..., N-1; \tag{3.33.b}$$

where f is the state update equation, g is the measurement equation. In the traffic model introduced in chapter 2 these are viz. for f the AR update equation and the queue length update equation (assuming red and green switching times known) for g as shown in (2.10.c).

PF based estimation algorithms are often used to estimate the unknown parameters θ by adding a random walk to the parameters and augmenting the state-space with the parameters for joint estimation as expressed by *(x, θ)*. Model (3.33.a), assuming random walk noise added to parameter values, can be written in compact form as follows:

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, \theta_{k-1}) \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \eta_k \\ np_{k-1} \end{bmatrix} \tag{3.34.a}$$

$$(x_{k+1}, \theta_{k+1}) \sim \ p_x(x_{k-1} \mid x_k)\, p(\theta_{k+1} \mid \theta_k) \tag{3.34.b}$$

where np_k is a random variable with Gaussian distribution $N(np_k;0,W_{k-1})$ (with zero mean and variance W_{t-1}) independent of all other random variables, of appropriate strength. This random walk is introduced to generate parameter values moving around in parameter space, in order to allow the proper exploration of the parameter space. However, the use of a random walk implies an increase in the magnitude of the variance, resulting in posterior distribution that are more diffuse than they should be. This issue will be demonstrated under the framework of particle filters in the next paragraph.

By assuming state update, parameter random walk and observation noise are independence and if both states and parameters are to be estimated, joint posterior distribution can be defined by using Bayes' rule :

$$p(x_k, \theta_k | \mathbf{y}_k) \propto p(y_k | x_k, \theta_k) p(x_k | \mathbf{y}_{k-1}, \theta_{k-1}) p(\theta_k | \mathbf{y}_{k-1}) \tag{3.35}$$

Proof:

Based on the Bayes' theorem in equation (3.1):

$$
\begin{aligned}
p(x_k, \theta_k | \mathbf{y}_k) &= \frac{p(\mathbf{y}_k \mid x_k, \theta_k)\, p(x_k, \theta_k)}{p(\mathbf{y}_k)} \\
&= \frac{p(y_k, \mathbf{y}_{k-1} \mid x_k, \theta_k)\, p(x_k, \theta_k)}{p(\mathbf{y}_k)} \\
&= \frac{p(y_k \mid \mathbf{y}_{k-1}, x_k, \theta_k)\, p(\mathbf{y}_{k-1} \mid x_k, \theta_k)\, p(x_k, \theta_k)}{p(y_k)} \\
&= \frac{p(y_k \mid \mathbf{y}_{k-1}, x_k, \theta_k)\, \dfrac{p(x_k, \theta_k \mid \mathbf{y}_{k-1})\, p(y_{k-1})}{p(x_k, \theta_k)}\, p(x_k, \theta_k)}{p(y_k)} \\
&\propto p(y_k \mid \mathbf{y}_{k-1}, x_k, \theta_k)\, p(x_k, \theta_k \mid \mathbf{y}_{k-1}) \\
&\propto p(y_k \mid \mathbf{y}_{k-1}, x_k, \theta_k)\, p(x_k \mid \theta_k, \mathbf{y}_{k-1})\, p(\theta_k \mid \mathbf{y}_{k-1})
\end{aligned}
$$

Q.E.D

It is clear that we need to deal with the problem of not knowing the form of the density $p(\theta_k \mid \mathbf{y}_{k-1})$ to obtain the joint posterior $p(x_k, \theta_k | \mathbf{y}_k)$. This issue can be demonstrated in the framework of particle filters, where in this thesis, we use kernel based density estimation to approximate $p(\theta_k \mid \mathbf{y}_{k-1})$.

For a general set of a training instance parameter points $\{\theta^i, i=1,\dots,N\}$, N is the number of particles and θ is a test instance parameter point or possible value for the true parameter value. We use a kernel-based density estimate of the underlying distribution of the parameter $z(\theta)$, expressed as a linear mixture of kernels:

$$z(\theta) = \sum_{i=1}^{N} w^i K(\theta, \theta^i) \tag{3.36}$$

where the w^i are weights and nonnegative constant that sum to one, and K is a nonnegative kernel function which integrates to one. We define K is the density of a multivariate normal. The position of the kernel function would be defined by $\{\theta^i, i=1,\dots,N\}$, N is a number of a training points and in particle filter case.

Kernel density estimation estimates the probability density function by imposing a model function on every data point and then adding them together. The function applied to each training points is called a kernel function. For example, a Gaussian function can be imposed on every single data point, making the center of each Gaussian kernel function the data point that it is based on. The standard deviation on the Gaussian kernel function adjusts the dispersion of function of kernel and is called a bandwidth of the kernel function.

The approach used in this thesis follows the method introduced by Liu and West in the paper [20] where $p(\theta_k \mid \mathbf{y}_{k-1})$ is approximated by Gaussian kernel function according to:

$$p(\theta_k \mid \mathbf{y}_{k-1}) \approx \sum_{i=1}^{N} w_k^i \; N(\theta_k, \theta_{k-1}^i, W_{k-1}) \tag{3.37}$$

Let $\bar{\theta}$ and V be the mean value and the variance value of the weighted particles θ, that are calculated as follows:

$$\bar{\theta}_{k-1} = \sum_{i=1}^{N} w_{k-1}^i \theta_{k-1}^i$$

$$V_{k-1} = \sum_{i=1}^{N} w_{k-1}^i (\theta_{k-1}^i - \bar{\theta}_{k-1})^2 \tag{3.38}$$

The derivation in [20] shows that the distribution in Eq.(3.37) has a mean of $\bar{\theta}_{k-1}$ and covariance $V_{\mathrm{k}} = V_{k-1} + W_{k-1}$, where V_0 is specified by generating θ_0 according to Gaussian random generator. Thus it is clear that, as a consequence of the random walk (the addition of independent zero mean noise with covariance W_{k-1}), the covariance V_{k} increases over time. This called *dispersion*.

A natural approach to overcome the issue of dispersion, is the use of kernel smoothing with *a properly tuned smoothing factor* as proposed in [20] :

$$p(\theta_k | \mathbf{y}_{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^i N(\theta; m_{k-1}^i, h^2 \, V_{k-1}) \tag{3.39}$$

The idea behind this approach is the shrinkage of the kernel width forcing the particles to be closer to their prior mean according to:

$$m_{k-1}^i = \sqrt{1 - h_k^2} \, \theta_{k-1}^{\,i} + \left(1 - \sqrt{1 - h_k^2}\right) \bar{\theta}_{k-1} \tag{3.40}$$

where $\{m^i_{t_{k-1}}\}_{i=1}^N$ are the shrinkage locations, $h_t \in [0,1]$ is a kernel parameter, $\bar{\theta}_{k-1}$ and V_{k-1} are the Monte Carlo mean and covariance computed using all the particles, with their values and weights $\{\theta^i_{k-1}, w^i_{k-1}, i = 1,..., N\}$ as shown in (3.38).

The optimal selection of the kernel parameter $h_k \in [0,1]$ maximally reducing the over-dispersion in the PF remains a difficult problem. The current practice for tuning of h_k is ad-hoc. Liu and West in [20] suggest choosing h_k=0.1, whereas Chen in [21] selects h_k as the optimum for a historical data-set, and then applies this choice for future batches. These ad-hoc rules define a constant h_k, for which optimality cannot be established w.r.t the online data to be analyzed. In chapter 5 we will introduce a better method for selecting the tuning parameter h_k.

3.3 Example of traffic state estimation using PF

The aim of this example is to show how the proposed approach for particle filtering estimation explained in section 3.2 can be applied to dynamic evolution equations.

The model represents traffic flow rate α_k where *k* is a short hand for the time index t_k of the sampling time update. We assume that the flow rate α_k is generated according to one single autoregressive model (AR), not a SHM. The state equation and measurement equation are assumed to follow equation (2.10.b) and (2.10.c) by dropping mode *s*. We examine how to jointly estimate the current state and the AR parameters for this model.

In this example, the joint state-parameter estimation of the state α_k and the parameter $\theta = \{\beta, \gamma, \sigma\}$ is studied for the two different experimental set-ups:

a. **Case-1**: uses data from an arterial road in city of Bandung with sampling time update every 15 minutes

In this case, we will use the discrete time-series model of traffic flow recorded every 15 minutes during the period of Monday,11st of June 2012 until Friday,15th of June 2012. The data is obtained from 00-24 pm each day. Video sensor measurement data is used to validate the traffic flow estimator/predictor. Figure.3.1 shows that the traffic flow during the different workdays follows a similar pattern. One exception is Friday between 12 am-14pm, when the flow pattern is low, reflecting the fact that in during that period many people go to Mosque for Friday prayer. But, everywhere else the daily flow is very similar.

Please note that in this example we use online joint state and parameter estimation. We use measurement data only on Friday to know the performance of state and parameter particle filter (PF) based estimator/prediction. In this case, we use the autoregressive (AR) process as shown in equation (3.46) and (3.47). It should be noted that t_k is sampling time update every 15 minute.

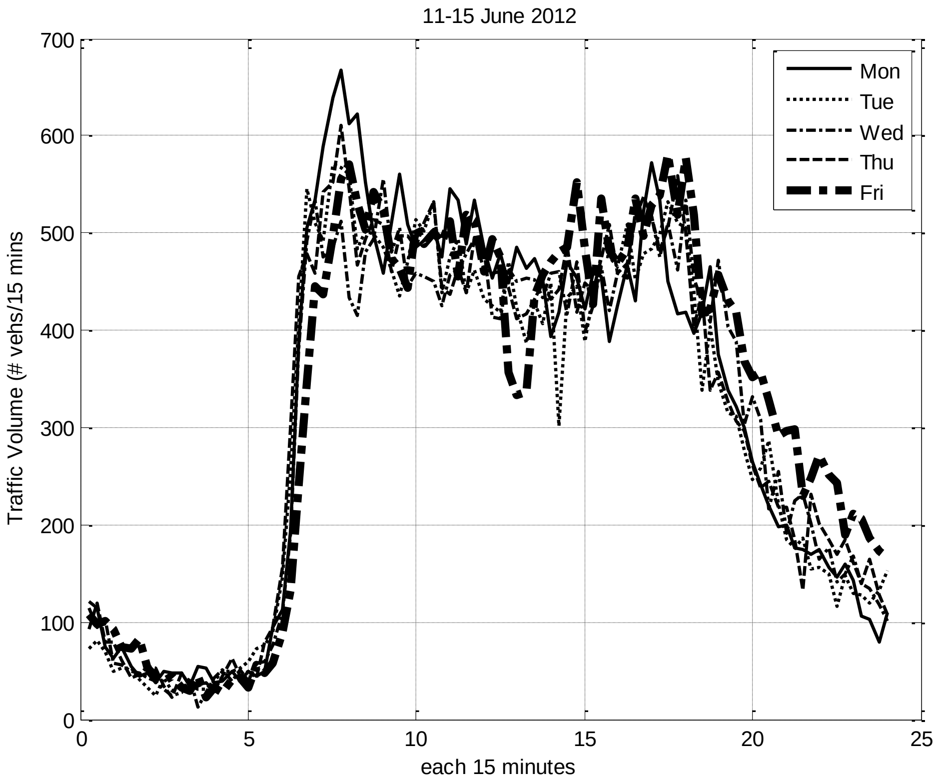

Figure.3.1 The pattern of traffic flow

In this simulation, we use *kernel smoothing* with smoothing factor $(1 > h > 0)$ to reduce the covariance by using kernel smoothing in (3.40) . Based on past experience we choose h=0.15.

In this simulation we use $\eta \sim N(0,1000)$, $n \sim N(0,1600)$, initial conditions $\alpha_0 = 0$,and $\theta_0 = \{\beta, \gamma, \sigma\} = \{0.5, 0.5, 0.5\}$ with particle number N=500;

Fig.3.2 shows that the PF predictor gives results close to the actual traffic flow, both for prediction 1-cycle ahead or 2-cycle ahead, except for the period 06-07 am.

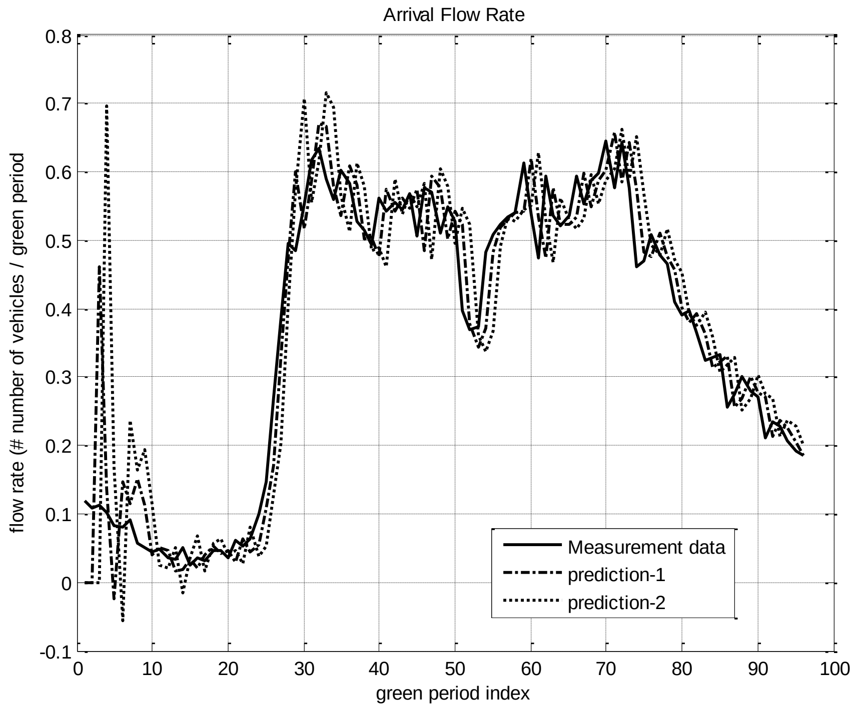

Figure 3.2. Prediction of 15’ ahead (prediction 1) and 30’ ahead (prediction 2)

b. Case 2 : uses data on the arrival flow at an approach link of a signalized intersection in city of Jakarta

The experiment **case-2** uses the same model (2.10.b-2.10.c) as in the **case-1**, but it uses shorter time sampling update intervals, with two different cases being compared with:

(**case-2.a**) average traffic flow over each green period: interval= 50 sec – 200 sec;

(**case-2.b**) average traffic flow over intervals corresponding to 5 successive green periods: 250 sec – 1000 sec (summing 5 values as used in cases 2.a)

The available data consists of the traffic volume in each a green period, as observed by the sensors installed for the SCATS traffic control system implemented in that area. Because the SCATS system is counting the number of vehicles during the green/red period and the fact that SCATS system is an adaptive system then duration of the green/red period is varying depending on the intensity of traffic flow in that area. The joint state and parameter estimation technique described in section 3.2.3 is applied to these cases. Figure 3.3 shows that PF predictor gives an unsatisfactory result for the shorter time update as indicated in the **case-2.a**. The lower graph clearly indicates that the prediction results, during index 60-120, are unsatisfactory for the shorter time update interval of **case-2.a**. It seems that one single AR model does not allows for occasional discrete shifts in the parameter determining the level of 'the volatility'. We need the model to explain some form of persistent 'volatility' persistence.

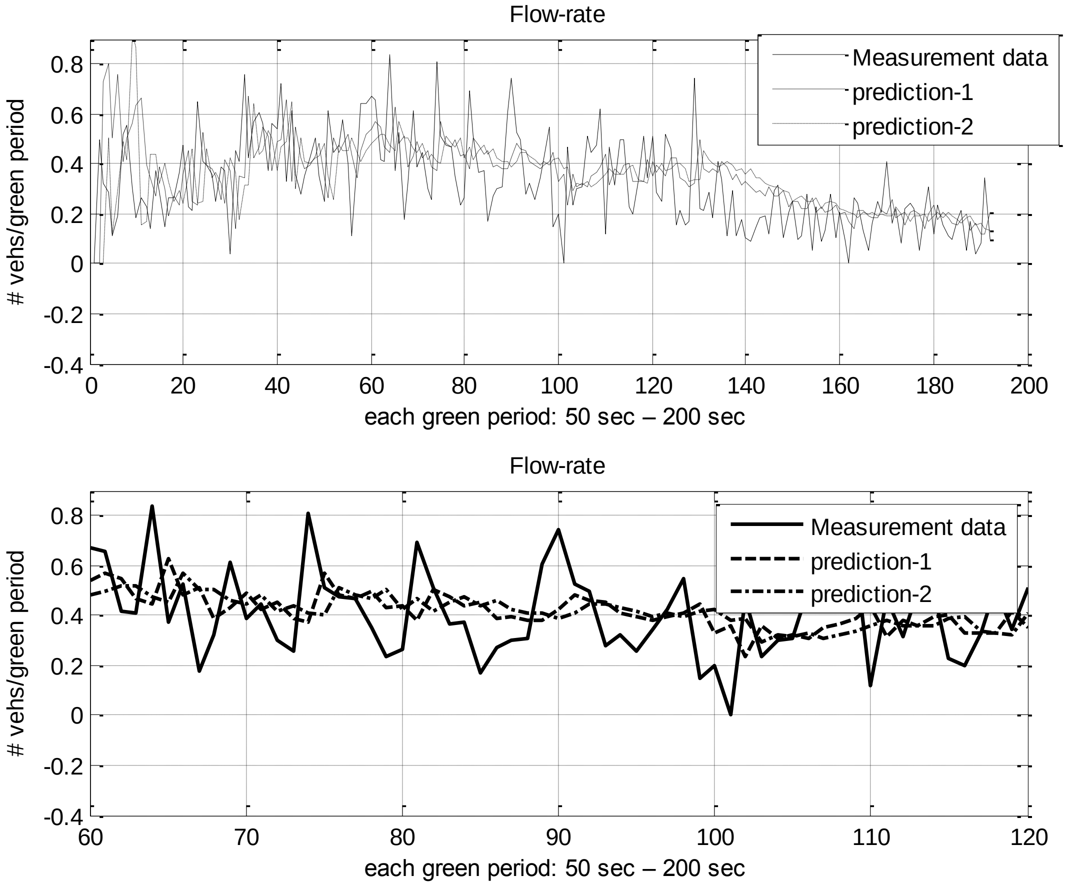

Figure.3.3 Performance of the state-parameter estimation for **case-2.a**

However, if we use AR model with the longer time update interval, corresponding to the summation of 5-summation of green period as in **case-2.b**, the performance of state-parameter estimation becomes much better as shown in Fig.3.4. It is reasonable, because the summation at longer intervals makes the variability of the data becomes smaller, so the data becomes less volatile. It is comparable to the results of such as the **case-1.**

These two experiments above show that if we count the number of vehicles passing within a given time update in successive green/red phases, the existing joint state-parameter estimation technique needs more sophisticated models to improve the performance of prediction. One single AR model cannot sufficiently describe the persistence of the volatility of the arrival flow. One of the possible models that we are looking at in order to remedy this limitation is jump Markov model, because looking at the data in Figure.3.3 one sees from time to time that the measurement data jumps to a different mean value and/or a different variances. This suggested the need to develop an advanced PF based joint state-parameter estimation for stochastic hybrid model as will be discussed in Chapter 5.

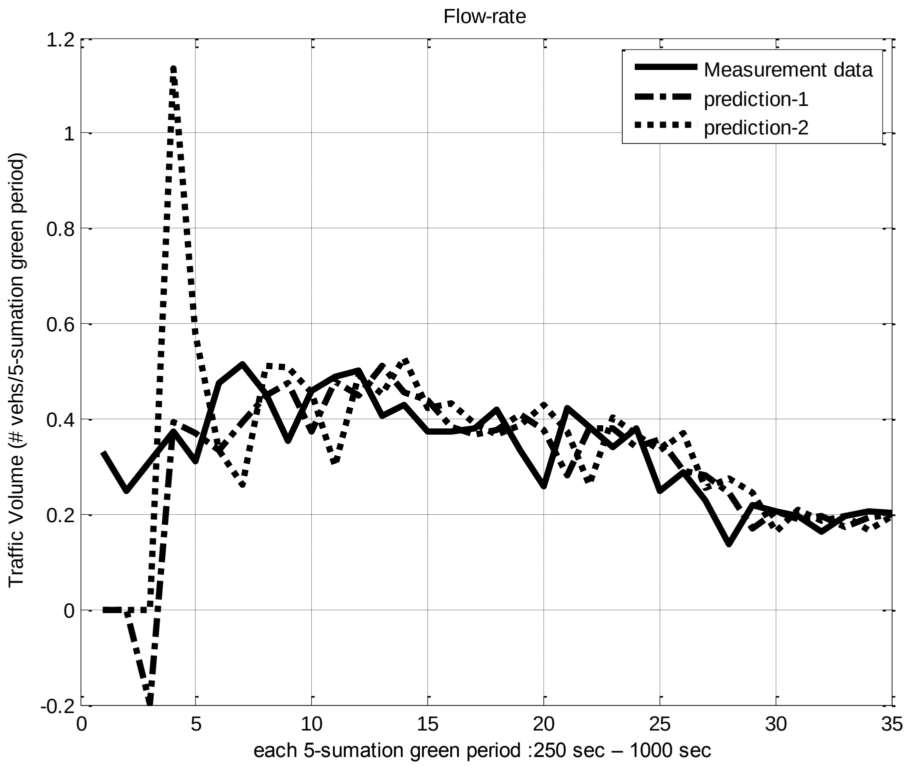

Figure.3.4 Performance of the state-parameter estimation for the **case-2.b**

3.4 Conclusions

In this chapter, the particle filtering (PFs) based approach for joint states estimation and parameters estimation is explored for traffic flow model. The standard PF methods assume knowledge of the model parameters while in real applications, the parameters are unknown and must be estimated. Joint state-parameter estimation is achieved by using kernel smoothing in order to reduce the dispersion which is caused by the need to add of random walk to the parameter model.

By using actual measurement data of traffic flow along an arterial road and at an approach link to a signalized intersection, the joint state-parameter estimation with a simple model can be potentially applied to estimate traffic flow. However, with shorter sampling time updates (e.g. updating for each green phase) the results show that one single AR model does not allows for the occasional discrete shifts in the parameters determining the level of ‘the volatility’. This suggests the use of a jump Markov model (JMM) for improved prediction performance. This requires us to develop an advanced PF based joint state-parameter estimation for JMM will be discussed in Chapter 4 and 5.

4
Parameter Estimation: offline approach

4.1 Introduction

A fundamental and widely-applicable approach to the problem of obtaining parametric models from observed data involves adopting a statistical framework and then selecting as estimated model, that model which maximizes the likelihood of the observed data. Schemes guided by this principle are known as Maximum Likelihood (ML) methods and, due to the fact that they have been studied for almost a century, they benefit from a very large and sophisticated body of supporting theory. This theoretical underpinning allows, for example, important practical issues such as error analysis and performance trade-offs to be addressed. However, despite their theoretical advantages, the practical deployment of ML methods is not always straightforward. This is largely due to the non-convex optimization problems that are often implied. Since these cannot be solved in closed-form, they are typically attacked via a gradient based search strategy based on Newton's method or one of its derivatives [27]. However, Newton's method is never guaranteed to converge to the global optimum for a non-convex problem. So a Newton method will only guarantee that you find a local optimum.

This chapter explores a different approach to the problem of finding ML estimates of fully-parametrized state-space models from single/multivariable observations. More specifically, the work here employs the Expectation Maximization (EM) algorithm as a means of computing ML estimates. The EM algorithm enjoys wide popularity and acceptance in a broad variety of fields of applied statistics [28], [29]. However, despite this acceptance and success in other fields, it could be argued that in systems and control settings, the EM algorithm is not as well understood, accepted and utilized as it may deserve to be.

In this chapter, we will develop the parameter estimation of the hybrid model such as jump Markov model as described in the previous section. In total - for a single traffic flow α_{t_k} where α is a generic flow (it can be $\lambda_{1,t_{2k}}, \lambda_{1,t_{2k+1}}, \mu_{1,t_{2k}}$ for movement L_1) - there are 15 parameters to be estimated:

$$\theta(s) = \{\beta(s), \gamma(s), \sigma^2(s), \Pi\} \tag{4.1}$$

- For each mode s ϵ {1,2,3} the AR model in equation (2.13) has 3 parameters, β(s), γ(s) and σ^2(s).
- The entries of transition probability matrix Π are π_{ij} of the Markov chain has 3 rows of 3 elements, each summing to 1.

Estimation of the parameters $((\beta(s), \gamma(s), \sigma^2(s), s = 1,2,3)$ of the AR-models, and of the transition probabilities (π_{ij} $, i, j = 1, 2, 3)$ of the JMM can be performed using an iterative two-step EM procedure, where π_{ij} are entries of matrix TPM. Therefore the EM algorithm allows us to completely identify the JMM stochastic hybrid model proposed in the previous sections as a model for traffic flow. In this chapter, we develop the application of the EM approach, originally proposed by Dempster et al [30] and further extended in [31,32,14], to switching systems as our JMM, based on forward-backward recursion or 'smoothing'. A good introduction and survey paper on EM can be found in [33]. This EM approach is formulated in batch or offline form, i.e. it uses a given number of observations obtained over a time interval [0,T) to iteratively find better and better estimates of the unknown parameters of a model that is valid over the period [0,T). This offline approach needs significant memory requirements and processing power for storing and processing large datasets, but this approach is shown to be useful and applicable further on in this chapter.

4.2 Expectation Maximization Algorithm

Let us give a brief review of the mathematical background to EM algorithm. Let **y** denote a data vector with the associated probability density $f_{\mathbf{y}}(y|\theta)$, where the parameter vector θ ϵ Θ is the unknown parameter vector to be estimated. The maximum likelihood (ML) estimate given an observation y of **y** is

$$\hat{\theta}_{ML} = \arg\max_{\theta \in \Theta} \log f_{\mathrm{y}}(y \mid \theta) \tag{4.2}$$

The problem is that often the maximization problem (4.2) is complicated.

Suppose that we can specify some data **x** related to **y**, such that if **x** is observed, an observation **y** is available too. Remember that our state variables consists of continuous state $x_{c,t_k} = \alpha_{t_k}$ and discrete state/mode *s*, where *s*={1,2,3}, as described in section 2.2.2. The probability density function of the complete data **x** is $f_{\mathbf{x}}(x,\theta)$. The idea is that the complete data **x** is chosen such that solution of

$$\hat{\theta}_{ML} = \arg\max_{\theta\in\Theta} \log f_x(x \mid \theta) \tag{4.3}$$

can be found.

The EM method is an iterative algorithm working in two steps. In the E-step the sufficient statistics of the complete data are estimated, i.e., the expected value of $f_{\mathbf{x}}(x,\theta)$ conditioned on **y** is computed. This estimated is then used to obtain a new estimated of θ in the so called M-step. This parameter estimate is then fed back to E-step, and so the method iterates until convergence. In mathematical form the method is summarized below:

- Start, m=0: guess an initial value θ^m
- Iterate m → m+1 (until convergence),
  - **The E step:** Calculate
    $Q(\theta \mid \theta^m) = E\{\log f_{\mathbf{x}}(\mathbf{x} \mid \theta) \mid \mathbf{y}=y; \theta^m\}$
  - **The M step:** solve
    $$\theta^{m+1} = \arg\max_{\theta\in\Theta} Q(\theta \mid \theta^m)$$

In this section we do reformulate the EM approach proposed by Hamilton [31] in order to get a simpler and easier algorithm, changing the cost-function in the M-step *by adding a weighted term with corresponding smoothed inferences about the currently active mode*. The smoothed inferences are the probabilities $p\left(s(t_k) = s \middle| \mathbf{y}_{t_{k-1}}; \theta\right)$ that the *k*-th observation comes from mode s given all the information $y^{t_{k-1}}$ available prior to time t_k, where we use the notation $\mathbf{y}_{t_{k-1}} = \left(y_1, y_2, \ldots, y_{t_{k-1}}\right)$.

Remark: In this chapter to simplify the notation, we use *k* in order to denote the sampling times t_k.

The reason for this modification is that each observation y_k belongs to the s-th mode with probability $p(s_k = s | y_k, \theta)$. By using these smoothed inferences we find a better maximum likelihood estimate. If we transform our AR model to state-space (SS) form, as explained in equation (2.10), then we notice that in the SS model, process noise and measurement noise is equivalent and the assumption that the mode s_k depends on past observation $\mathbf{y}_{k-1}$ only through the value of s_{k-1}, implies that the calculation of the distribution $p(s_k = s | \mathbf{y}_{k-1}; \theta)$ is a problem of hidden state estimation of hybrid systems, where $\theta = \{\beta_0, \varphi_0, \sigma_0^2, ..., \beta_3, \varphi_3, \sigma_3^2, \Pi\}$. The key idea of this calculation is to perform a forward-backward or 'smoothing' filter recursion for each possible mode sequence.

The algorithm starts with an arbitrarily chosen vector of 15 independent parameters, chosen randomly but using prior information in order to speed up convergence of the algorithm and to increase the likelihood of converging to the global maximum. The EM technique for hybrid systems requires a method for recursively approximating the conditional probability $p(s_k = s | \mathbf{y}_{k-1}; \theta)$, for each index m counting the number of EM iterations. Moreover an algorithm for finding the parameter values that achieves the maximum likelihood must be available. This differs from the approach proposed in [28] for a general non-linear system, where the objective is to find the best estimate of the conditional density $p_\theta(y_k | \mathbf{y}_k)$. In the case of hybrid systems, as in this thesis, the E-step, indirectly infers about the discrete states/modes. Since the mode s_k, is unobservable, only the conditional probability of the modes at successive time t, given the observation vector, can be calculated.

Since the observed trajectory $\{y_1, \ldots, y_k\}$ of the measurement depends on the trajectory $\{s_1, \ldots, s_k\}$ of the past and present modes, and on the parameters, as specified by the probability $p(s_k = s | \mathbf{y}_k; \theta)$ (given all the data y_k, k = 1, 2….,T), it is reasonable if each component on the right-hand side of equation (4.4) below has to be weighted with the corresponding smoothed inferences. Therefore, the weighted log-likelihood function is given by assuming that the noise terms η_k have a normal distribution as shown in equation (2.10b); similar formulae can be written down for other noise distributions):

$$L(\theta) = -\log \ell(\theta) = -\log \ell(\beta_s, \gamma_s, \sigma_s, \pi)$$

$$= -\sum_{t=2}^{T} p(s_k = s \mid y_k; \theta^m) \; * \left[-\log\left(\sigma_s \sqrt{2\pi}\right) + \frac{(y_k - \beta_s - \gamma_s \, y_{k-1})^2}{2\sigma^2} \right] \tag{4.4}$$

The conversion to the (log-)likelihood function is an important step for obtaining easier explicit formulas by solving the nonlinear equation obtained by setting the partial derivatives of the log-likelihood function to zero in the M-step.

A detailed explanation of the EM-algorithm [38],[39] is given below:

The E-Step

The E-step consists of forward filtering and backward filtering/smoothing and this step aims to calculate the conditional probabilities $p\left(s_k = s \middle| y_k; \theta^m \right)$. Assume that θ^m is the parameter vector calculated in the M-step during the *m*-th iteration and $\rho_i^{(0)} \equiv p\left(s(1) = s\right)$. The algorithm starts with an arbitrarily (but sensibly, as explained above) chosen vector of initial parameters

$$\theta^0 = \left\{ \beta_{1,0}, \gamma_{1,0}, \sigma_{1,0}^2 \;, \beta_{2,0}, \gamma_{2,0}, \sigma_{2,0}^2 \;, \beta_{3,0}, \gamma_{3,0}, \sigma_{3,0}^2 \;, \Pi^0 \right\}$$

i) **Step E-1:** *Forward Filtering* : for $k = 1,2\ldots,T$ iterate:

$$p\left(s_k = s \middle| \mathbf{y}_k; \theta^m\right) = \frac{p\left(s_k = s \middle| \mathbf{y}_{k-1}; \theta^m\right) g_s\left(y_k \mid s_k; \mathbf{y}_{k-1}; \theta^m\right)}{\sum_s^3 p\left(s_k = s \middle| \mathbf{y}_{k-1}; \theta^m\right) g_s\left(y_k \mid s_k; \mathbf{y}_{k-1}; \theta^m\right)} \tag{4.5.a}$$

where $\mathbf{y}_k = \left(y_1, y_2, \ldots, y_k\right)$ is the vector of traffic flow measurements available at time *k* and $g_s\left(y_k \mid s_k; \mathbf{y}_{k-1}; \theta^m\right)$ is the conditional probability density function at of y_k time *k* assuming that the mode *s* is active at time *k*:

$$g_s\left(y_k \mid s_k; \mathbf{y}_{k-1}; \theta^m\right) = \frac{1}{\sigma_s \sqrt{2\pi}} \exp\left(-\frac{\left(y_k - \beta_s - \gamma_s y_{k-1}\right)^2}{2\sigma_s^2} \right)$$

and

$$p\left(s_{k+1} = s \middle| \mathbf{y}_k;\theta^m\right) = \sum_{j=1}^{3} \Pi_{js}^m \ p\left(s_k = j \middle| \mathbf{y}_k;\theta^m\right), \qquad j\text{=1,2,3 is the number of modes} \tag{4.5.b}$$

until $p\left(s_T = s \middle| y_T;\theta^m\right)$ is calculated.

The starting point for the iteration is chosen as : $p\left(s(1) = s \middle| y(1);\theta^m\right) = \rho_s^m$ where ρ_s^m is the likelihood of being in mode s obtained in the previous iteration m and Π is transition probability matrix (TPM).

Proof:

To simplify the notation in this proof, we drop the symbol θ.

The conditional probability density function for the observations y_k given the modes s_k, s_{k-1} and the previous observations $\mathbf{y}_{k-1}$ is

$$g_s\left(y_k \mid s_k = s, s_{k-1}; \mathbf{y}_{k-1}\right) = \frac{1}{\sigma_s\sqrt{2\pi}} \exp\left(-\frac{\left(y_k - \beta_s - \gamma_s y_{k-1}\right)^2}{2\sigma_s^2}\right)$$

The chain rule for conditional probabilities provides then for the joint probability density function for the variables y_k,, s_k, s_{k-1} given previous observations $\mathbf{y}_{k-1}$

$$g_s\left(y_k, s_k = s, s_{k-1} \mid \mathbf{y}_{k-1}\right) = g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1})\, p(s_k = s, s_{k-1} \mid \mathbf{y}_{k-1})$$

In order to find the conditional joint probabilities $p(s_k, s_{k-1} \mid \mathbf{y}_{k-1})$ we use again the chain rule for conditional probabilities:

$$\begin{aligned} p(s_k = s, s_{k-1} \mid \mathbf{y}_{k-1}) &= p(s_k = s \mid s_{k-1}, \mathbf{y}_{k-1})\, p(s_{k-1} \mid \mathbf{y}_{k-1}) \\ &= p(s_k = s \mid s_{k-1})\ p(s_{k-1} \mid \mathbf{y}_{k-1}) \end{aligned}$$

where we use the Markov property.

The probabilities $p(s_{k-1} \mid \mathbf{y}_{k-1})$ and the joint probabilities $p(s_k, s_{k-1} \mid \mathbf{y}_{k-1})$ are obtained using the following two steps :

1. Given $p(s_{k-1} \mid \mathbf{y}_{k-1})$, i=0,1, at the beginning of time k,

$$p(s_k, s_{k-1} \mid \mathbf{y}_{k-1}) = p(s_k = s \mid s_{k-1})\, p(s_{k-1} \mid \mathbf{y}_{k-1})$$

2. Once y_k is observed, we update the information set $\mathbf{y}_k=\{\mathbf{y}_{k-1},y_k\}$ and the probabilities by backwards application of the chain rule and using the law of total probability

$$\begin{aligned} p(s_k = s, s_{k-1} \mid \mathbf{y}_k) &= p(s_k = s, s_{k-1} \mid \mathbf{y}_{k-1}, y_k) \\ &= \frac{g_s(s_k = s, s_{k-1}, y_k \mid \mathbf{y}_{k-1})}{g_s(y_k \mid \mathbf{y}_{k-1})} \\ &= \frac{g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1})\, p(s_k = s, s_{k-1} \mid \mathbf{y}_{k-1})}{\sum_{s=1}^{3} g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1})\, p(s_k = s, s_{k-1} \mid \mathbf{y}_{k-1})} \\ &= \frac{g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1})\, p(s_{k-1} = s \mid \mathbf{y}_{k-1})\, p(s_k \mid s_{k-1})}{\sum_{s=1}^{3} g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1})\, p(s_k = s \mid \mathbf{y}_{k-1})\, p(s_k \mid s_{k-1})} \end{aligned}$$

$$p(s_k = s \mid \mathbf{y}_k) = \sum_{j=1}^{3} p(s_k = s, s_{k-1} = j \mid \mathbf{y}_k)$$

$$\begin{aligned} p(s_k = s \mid \mathbf{y}_k) &= \frac{g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1}) \sum_{j=1}^{3} p(s_k \mid s_{k-1} = j) p(s_{k-1} = s \mid \mathbf{y}_{k-1})}{\sum_{s=1}^{3} g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1}) \sum_{j=1}^{3} p(s_k \mid s_{k-1} = j) p(s_k = s \mid \mathbf{y}_{k-1})} \\ &= \frac{g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1}) p(s_k = s \mid y_{k-1})}{\sum_{s=1}^{3} g_s(y_k \mid s_k = s, s_{k-1}, \mathbf{y}_{k-1}) p(s_k = s \mid y_{k-1})} \qquad (\textit{from } 4.5.b) \end{aligned}$$

Q.E.D

ii) **Step E-2:** *Backward filtering* : for $k = T-1, T-2, \ldots, 1$ iterate on

$$p(s_k = s \mid y_T; \theta^m) = \sum_{j=1}^{3} \frac{p(s_k = s \mid \mathbf{y}_k; \theta^m)\, p(s_{k+1} = j \mid y_T; \theta^m)\, \pi_{sj}^m}{p(s_{k+1} = j \mid \mathbf{y}_k; \theta^m)} \tag{4.6}$$

where π_{ij} are entries of Π.

See [31] for the proof (taking into account that the structure and terminology used there is completely different from what is used in this thesis).

The M-Step

In the M-step of the EM algorithm, new and hopefully more exact maximum likelihood estimates θ^{m+1} for all model parameters are calculated. As mentioned previously each component of the log-likelihood function has to be weighted with the corresponding smoothed inference. In particular, for the model defined by equation (2.10.b) explicit formulas for the estimates can be derived by setting the partial derivatives of the log-likelihood function (4.4) to zero and solving the resulting system of non-linear equations by a substitution technique:

$$\alpha_s = \frac{\sum_{k=2}^{T} p\left(s_k = s \mid y_T, \theta^m\right)\left(y_k - \varphi_s y_{k-1}\right)}{\sum_{k=2}^{T} p\left(s_k = s \mid y_T, \theta^m\right)} \tag{4.7}$$

$$\varphi_s = \frac{\sum_{k=2}^{T}[p\left(s_k = s \mid y_T, \theta^m\right) y_k y_{k-1}] - \alpha_i [\sum_{k=2}^{T} p\left(s_k = s \mid y_T, \theta^m\right) y_{k-1}]}{\sum_{k=2}^{T} p\left(s_k = s \mid y_T, \theta^m\right) {y_{k-1}}^2} \tag{4.8}$$

$$\sigma_s^2 = \frac{\sum_{k=2}^{T}[p\left(s_k = s \mid y_T, \theta^m\right)\left(y_k - \alpha_s - \varphi_s y_{k-1}\right)^2]}{\sum_{k=2}^{T} p\left(s_k = s \mid y_T, \theta^m\right)} \tag{4.9}$$

Because equations (4.7),(4.8) and (4.9) are nonlinear, it is not possible to solve analytically for θ^m as a function of $\{y_1, ..., y_T\}$. However, these equations do suggest an appealing iterative gradient ascent algorithm for finding the maximum likelihood estimate (or in practice a gradient descent for the log-likelihood since the relation between likelihood and log-likelihood is monotonely decreasing).

Iterating back from T to 1, as shown in E-step E-2, we obtain $\rho_s^{m+1} = p\left(s(1) = s \,\middle|\, \mathbf{y}_T ; \theta^m\right)$ and the transition probabilities are estimated by using equation (4.10) (for a proof see [31]). Basically, in the proof, the maximum likelihood estimates are obtained by forming the Lagrangean in order to estimate the transition probabilities. Following Hamilton in [31] and this fits to our traffic case by considering that our jump Markov model, then evaluating the derivative in this expression is made relaxing by the independence of mode transitions on the continuous state. The transition probabilities are restricted only by the condition that $\pi_{ij} \geq 0$, and $(\pi_{i1} + \pi_{i2} + \pi_{i3}) = 1, i = 1,2,3$

$$\pi_{ij}^{m+1} = \frac{\sum_{k=2}^{T} p\left(s_k = j \mid y_T; \theta^m\right) \frac{\pi_{ij}^m p\left(s_{k-1} = i \mid \mathbf{y}_k; \theta^m\right)}{p\left(s_k = j \mid \mathbf{y}_{k-1}; \theta^m\right)}}{\sum_{k=2}^{T} p\left(s_{k-1} = i \mid y_T; \theta^m\right)} \tag{4.10}$$

where π_{ij}^m is a entry of the transition probability matrix Π obtained during the m-th iteration.

All values obtained in the M-step are then used as a new parameter vector $\theta^{m+1} = \left(\beta_i^{m+1}, \varphi_i^{(m+1)}, \sigma_i^{2\,m+1}, \pi^{m+1} \right)$ and ρ_i^{m+1}, $i = 1,2,3$ in the next iteration of the E-step. The algorithm is terminated when $\left|\theta^m - \theta^{m-1}\right| \leq \delta$ for some preset accuracyδ. It means that one continues iterating in this fashion until the change between θ^m and θ^{m-1} is smaller than some specified convergence criterion. In the algorithm that we implemented we used the L_2 norm, but other norms could also be used.

The EM-algorithm can be implemented in 2 different forms, EM and GEM (Generalized EM) [30]. The EM algorithm maximizes the conditional expectation at every iteration, while the GEM only ensures that the likelihood increases at each step. Our implementation follows the GEM form. We avoid oscillations between different modes with identical likelihoods by updating the model only when the likelihood increases. The algorithm implementing this Generalized (G) EM-iteration procedure is shown in Table 5.1 where the equations used in the both E-step and the M-step have been explained above.

Related to the algorithm, the two main problems with its implementation are:

(1) the conditional expectation is difficult to compute, and therefore we replace it by smoothed inferences. Thus we use the whole data (batch) obtaining more accurate results, especially for the estimate of the mode of operation at each time t. The price we pay for more accurate evaluation is that in the E-step we need a backward filtering iteration as shown in equation (4.6). This by the way automatically makes the algorithm unsuitable for online applications, apart from memory and computation time requirements.

(2) since the convergence may be to a local maximum that is not the global maximum, a good choice of initial conditions is necessary. In our traffic case, we simply choose the initial conditions based on intuitive guesses corresponding to similar historical real measurement data. In practice we found that the parameter estimates always converged to good estimates (global optimality cannot be proven simply because the true parameters are not know, and may not even exist in reality if the proposed AR model is an approximation to a much more complex model).

Table.4.1 : GEM Algorithm

1). **Initialization**	Set m=1. Select ε ϵ **R** arbitrarily small Initialize θ^1 to initial guess
2). **Expectation (E) Step**	Set m = m+1 Given θ^m calculate $L(\theta^m)$ using equation (4.5) and (4.6)
3).**Maximization (M) Step**	Set θ^{m+1} to value of θ by using equation (4.7), (4.8), (4.9) and (4.10) that maximize $L(\theta^{m+1})$
4). **Convergence check**	Evaluate $L(\theta^{m+1})$. If $\lvert L(\theta^{m+1})$- $L(\theta^m)\rvert<$ ε, , stop. Otherwise go to 2.

Since we use a stochastic model with Gaussian noise, the likelihood functions are bounded and the Hessian is always negative definite [27]. Therefore, the sequence converges to a local maximum (of the likelihood, a local minimum of the log-likelihood), proving convergence of our parameter estimation, assuming identifiability of the model (see [27,34]). In practice we found that the urban traffic model propose in this paper always leads to an identifiable system.

4.3 Experimental layout

This section focuses on estimating parameter values for the urban traffic flow model introduced in chapter 2, using the GEM-algorithm of section 4.2. The case study uses the data over a time window [0,T] for the road layout shown in Fig.4.1 which represents a traffic network, in the area of Thamrin Street, Jakarta, Indonesia. The time-window size is important to define the model that will be used for estimating traffic flow for the next time-window. We will study in the next section the effect of varying time-window size and its practical implication on the performance of the estimator. The accuracy of the estimated traffic flow model is crucial for achieving good performance of the model based state estimators and of feedback traffic control strategies.

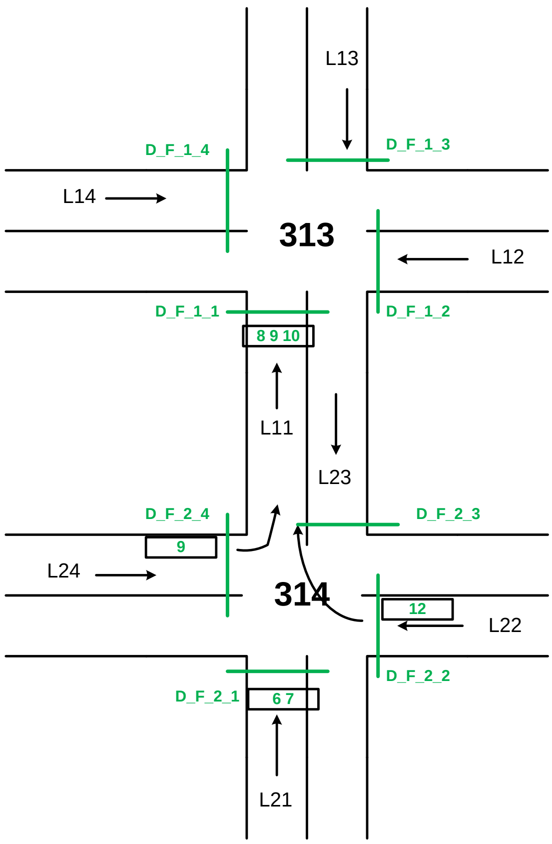

Figure.4.1 Configuration of area under investigation

In this section, we validate our traffic flow model, and the parameter estimation algorithm based on this model, using data collected at 2 neighboring intersections, called 313 and 314 as shown in Fig.4.1. At each of the locations indicated in Fig.4.1 by a box, a video camera, identified by its numbers, tries to detect all passing vehicles using a computer based counting algorithm. While false detections and missed vehicles cause significant errors in the data, the output of the sensors approximately counts the number of vehicles that cross an entrance location during one cycle of the traffic light.

The data set obtained from these video cameras lists the number of vehicles that cross a location per cycle of the local traffic light, as well as the duration of each cycle (between 120-200 seconds). This provides us with a data set of traffic flow rates averaged per cycle. By using these measurement data we build a model describing the dynamics of the traffic flow α_k

veh/sec passing the location observed by the video camera, and we apply parameter estimation techniques to estimate the model parameters, and finally to validate the model thus obtained by using smoothed inference for predicting future values of α_k.

The SCATS system updates the traffic flow measurements on a cycle-by-cycle basis.

Remark: Please note that the SCATS system is an adaptive system, therefore the cycle-length is varying and so the variation in the value of the denominator in the definition of the flow rate may cause sudden changes in the measured flows. Note that a good feedback control system for traffic lights should increase the cycle length when the vehicle flows increase, something that should in fact reduce the variability of the measured flow rates in the data set used in this section.

Fig.4.1 shows that intersection I-313 has arrival flow λ_{L11} which is determined by summing the number of passing vehicles detected at sensor location 9,6,7 and 12 divided by the cycle length of the intersection I-314. The value of this arrival flow rate as measured during the k-th cycle of the traffic lights is further on called $\lambda_{L11,k.}$ The departure rate for intersection 313, called $\mu_{L11,k}$, is defined by counting the vehicles that pass the sensor location 8,9,10 at downstream intersection I-313, divided by the k-th green period of lane L11 at intersection I-313 (the time delay corresponding to the travel time between intersection 313 and 314 must of course be taken into account when using the flow rate λ_{L11} as inflow rate at the downstream intersection I-313). Of course this is under the assumption that drivers of vehicles that pass the sensor locations 9,6,7 and 12 follow the traffic rules.

Let $N_{s,k}$ be the number of vehicles passing sensor location s (s=8,9,10 at intersection n=313) during the time intervals $[t_{n,k},t_{n+1,k})$, n=313, then $\lambda_{L11,k} = (N_{7,314,k} + N_{8,314,k} + N_{9,314,k} + N_{12,314,k})/(t_{314,k+1} - t_{314,k})$, while $\mu_{L11,k} = (N_{8,313,k} + N_{9,313,k} + N_{10,313,k})/(_{313,k+1} - _{313,k})$. Note that the k-th sample for arrival and for departure flows does not in general correspond to the same physical time t. Due to this limitation in our experimental setup, for the time being, we will only focus on the study of the development and validation of a model of one single traffic flow (not a queue-length) as a hybrid system. The study of queue-length will be discussed in the chapter 5 using online Bayesian joint state-parameter estimation and while online control applications will be treated in the chapter 6.

Aim of the current experiment is to check the practical implementability of our algorithm by identifying the parameters $\theta = \{\beta_1, \gamma_1, \sigma_1^2, \beta_2, \gamma_2, \sigma_2^2, \beta_3, \gamma_3, \sigma_3^2, \Pi\}$ of the JMM model for arrival and departure traffic flows. For this purposes we use as input for the EM-algorithm described above the data from the case study shown in fig.4.1. Table 4.2 shows the 18 parameters of the model for traffic flow thus obtained (where of course the normalization of the transition probability matrix Π is satisfied; this explains why we go from 15 parameters to be estimated to 18 parameters). It turns out to be feasible to identify all possible modes: **free flow mode, congested mode and faulty mode**. By using these parameters as estimated we can characterize the AR model of the traffic flow in each of the 3 modes, and the transition probability matrix Π with entries π_{ij},(i,j = 1,2,3) describing the Markovian mode transition process. Since for all estimated values we find that $|\varphi_s| < 1$ ensuring stability of the AR model for each of the modes, the stationary value $E_s(y_k)$ can be defined $E_s(y_k) = \frac{\beta_s}{1-\varphi_s}$, where s=1, 2 and 3. The values of stationary value $E_s(y_k)$ can be found in Table.4.2.

It is clear from the results of parameter estimation in Table 4.2, that the EM technique is able to identify the modes. In the case of traffic flow μ_{L11}, the first and third modes have an average value $E_i(y_k)$ that is almost the same but with very different values of the variance. This might make the results sensitive to how outliers –unlikely or abnormal events or observations - in the data set are treated. In estimation techniques, a standard approach to deal with this outlier issue is to reject any measurement that is at least three standard deviations away from ‘the normal’ measurement. This means that the definition of outlier depends on what we consider the normal variation. A value for the variance that is almost tripled, is a strong indication that the third mode is an outlier mode. This is also obvious from Fig.4.2 when considering the results for the estimated probabilities for those measurement data y_k marked with a circle, indicating that they belong to the third mode. These points in time correspond to events like traffic incidents, persistent counting errors, night time counting error. It is important for the control application to know the system dynamics at each time, including at these rare events, and to detect when a change in the AR model parameters occurs. Fig.4.2 shows both forward and backward filtering results and it seems that both approaches give similar results in terms of predicting the outliers. Overall, the EM algorithm is able to correctly identify clusters of mode-1, mode-2 or mode-3 operation, meaning that the

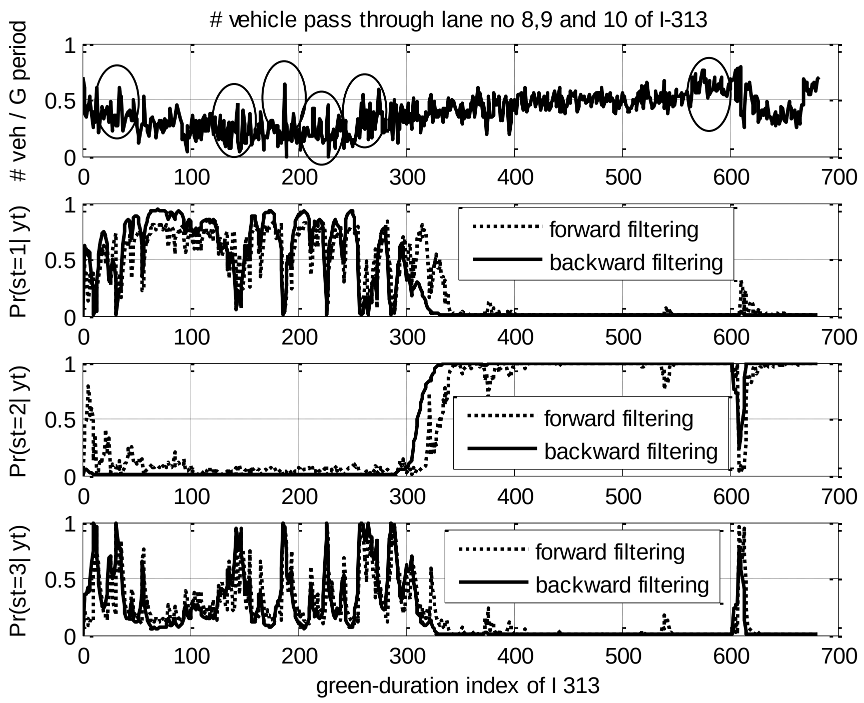

Figure 4.2 One-day (0 pm – 24 am) measurement data of μ_{L11} (top graph), JMM based estimated probability of mode s = 1,2,3 (second, third and bottom graph)

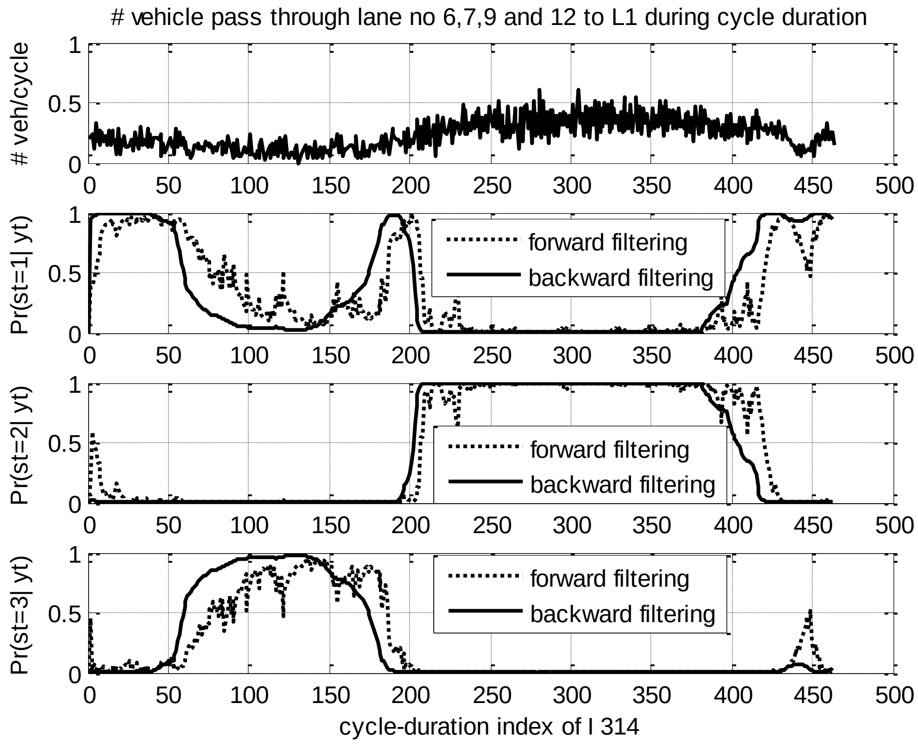

Figure 4.3. One-day (0 pm – 24 am) measurement data of λ_{L11} (top graph), JMM based estimated probability of mode s = 1,2,3 (second, third and bottom graph)

EM technique is applicable for modelling free flow and the congested modes as well as the faulty/outlier modes.

Analysis of Fig.4.2 and Table 4.2 shows that the transition probability to the outlier mode for traffic flow μ_{L11} is very high. By analyzing the green duration data (not shown here) at times when the probability of an outlier is high (mode-3), it becomes clear that this is due to the fact that when the traffic volume increases, the green duration is decreasing compared to the previous one. This is an undesirable effect of the control system that can be classified as an error in the timing of the green switches. This implies that traffic flow at those times increases drastically, as shown in Fig.4.2 with a circle marking. Remember, as described in chapter 2 that traffic flow is a ratio between N_{t_k} and $(t_{k+1} - t_k)$.

Discussions: It is interesting to analyze this “undesirable phenomena”. It looks like the opposite of what we discuss earlier. However, important to note that SCATS works based on “coordination” between the intersections and this involves changing the cycle, offsets and also the phase split. After discussing it carefully with the traffic engineers who deal with this area, author analyzed that, this "undesirable phenomena" may come from when SCATS change the phase split, because the phase split is determined in SCATS by attempting to equalize the DS on critical approaches. If DS (degree of saturation) value over 1 then SCATS will therefore vary cycle length and sub-systems share the same cycle length and use the offset as a way of maximizing traffic throughput. When SCATS trying to update itself on a cycle-by-cycle basis and to overcome this saturation, SCATS employs a decreasing weighted averaging mechanism which will take the values from the last three cycles. This mechanism itself takes time to updated the correct ones. Therefore we can say that this “undesirable phenomena” can be seen as a “transition time” for updating the cycle and phase split which sometimes does not deliver a proper phase split for certain critical approach.

Table 4.2. Parameter estimation results

Traffic flow μ_{L11}			Traffic flow λ_{L11}		
s_n =1	s_n =2	s_n =3	s_n =1	s_n =2	s_n =3
$\gamma_1 =$ 0.6130	$\gamma_2 =$ 0.9205	$\gamma_3 =$ 0.4579	$\gamma_1 =$ 0.6123	$\gamma_2 =$ 0.8163	$\gamma_3 =$ 0.5033
$\beta_1 =$ 0.0950	$\beta_2 =$ 0.0330	$\beta_3 =$ 0.1459	$\beta_1 =$ 0.0608	$\beta_2 =$ 0.0180	$\beta_3 =$ 0.0373
$\sigma_1^2 =$ 0.0070	$\sigma_2^2 =$ 0.0078	$\sigma_3^2 =$ 0.0222	$\sigma_1^2 =$ 0.0068	$\sigma_2^2 =$ 0.0292	$\sigma_3^2 =$ 0.0045
$E_1(y_t) =$ 0.245	$E_2(y_t) =$ 0.415	$E_3(y_t) =$ 0.269	$E_1(y_t) =$ 0.157	$E_2(y_t) =$ 0.098	$E_3(y_t) =$ 0.075
Transition Probabilities $\pi = \begin{bmatrix} 0.8836 & 0 & 0.1164 \\ 0 & 0.9979 & 0.0021 \\ 0.1788 & 0.0142 & 0.8071 \end{bmatrix}$			Transition Probabilities $\pi = \begin{bmatrix} 0.9843 & 0.0071 & 0.0086 \\ 0.0054 & 0.9946 & 0 \\ 0.0122 & 0 & 0.9878 \end{bmatrix}$		

From the TPM in table.4.2, it seems that the index of the free flowing mode is 2 in the case of μ, while it is 1 in the case of λ. It can also be seen from the estimates that the likelihood of remaining in the free flowing mode is very high, in both cases; for the congested mode the probability of remaining in it is very high also for λ but not so high for μ. Note also that there is never a direct transition from the free flowing mode to the faulty mode, it always happens via the congested mode. The explanation for these zeros in the TPM is in-line with the operation of SCATS as discusses previous page, that the “faulty condition” rises when SCATS faces the saturated condition (equivalent to congested mode).

The values of σ_i^2, with i=1,2,3, are related to the level of uncertainty of continuous state in certain mode. From the table.4.2, it seems that congested mode has a bigger uncertainty than free flowing.

Discussion: In the JMM based EM approach that we proposed, the assumption that the mode s_k depends on past observation $\mathbf{y}_{k-1}$ only through the value of s_{k-1} makes the evolution of continuous state only depends on the discrete state/modes, which can be classified as an unguarded mode transition. Mode transition is only characterized by transition probability matrix (TPM). One of the possible model that can reduce the uncertainty about discrete mode transitions at any given point in time is by conditioning discrete mode transition on the activation of guard conditions. In this perspective, the "potential guard conditions" that can be applied to our traffic flow cases, are the parameters of AR process each mode. One possible model including a guard mode transition is a Probabilistic Hybrid Automata (PHA) proposed by Santana in [76]. It is an interesting topic for further research.

In the case of traffic flow λ_{L11}, the EM parameter estimation is also able to identify the modes as shown in Fig.4.3 and Table 4.2. It should be noted that the horizontal axis of Fig.4.3 is cycle duration index of I-314 which is different from the horizontal axis of Fig.4.2 that is green-duration index of I-313. This explains the limitation of our experiment, and why we focus on the development and validation of a model of traffic flow, not on the queue-length yet.

The performance of this EM algorithm against the weighted log-likelihood function $L(\theta^m)$ can be evaluated and does provide insight in how close the convergence to the true maximum likelihood is feasible. We first analyse the convergence of the algorithm in terms of the evaluation of $L(\theta^m)$. Fig.4.4.d shows that, after a short transient period (< 40 iterations) , the EM-algorithm decreases the value of the weighted log-likelihood function $L(\theta^m)$ - thus also increases the likelihood - as the iterations of the parameter estimation proceed, indicating convergence of the estimated parameters (assuming identifiability). The convergence of the AR parameters ($\beta_1, \gamma_1, \sigma_1^2, \beta_2, \gamma_2, \sigma_2^2, \beta_3, \gamma_3, \sigma_3^2$) for the 3 different modes is shown in Fig.4.5. Both Fig.4.4 and Fig.4.5 show that around m=40 the value of $L(\theta^{m+1})$- $L(\theta^m)$ converges to an interval of width 2.ε that can be made arbitrarily small by making δ arbitrarily small; this corresponds to the convergence check step in Table 4.1. However, the analysis does not exclude the possibility that the algorithm converges to a local minimum.

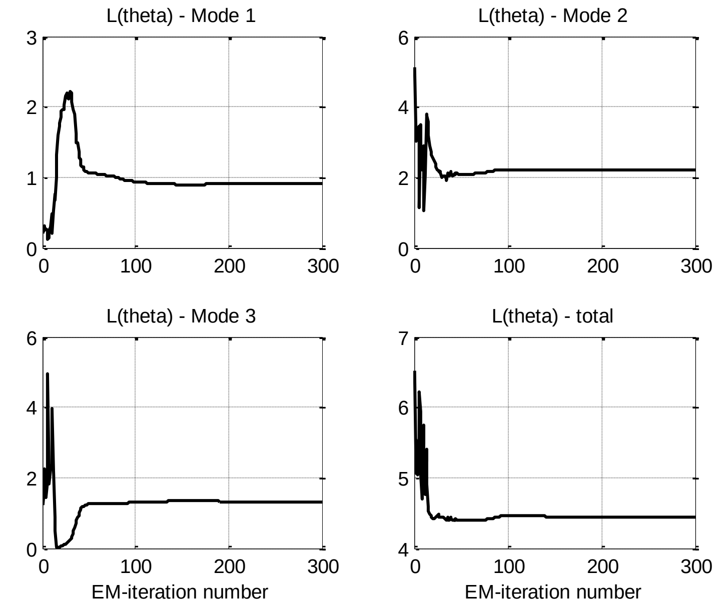

Figure.4.4 Convergence $L(\theta^k)$ of the EM algorithm for μ_{L11} : (a) $L(\theta^k)$ of mode 1 (b) $L(\theta^k)$ of mode 2 (c) $L(\theta^k)$ of mode 3 (d) $L(\theta^k)$

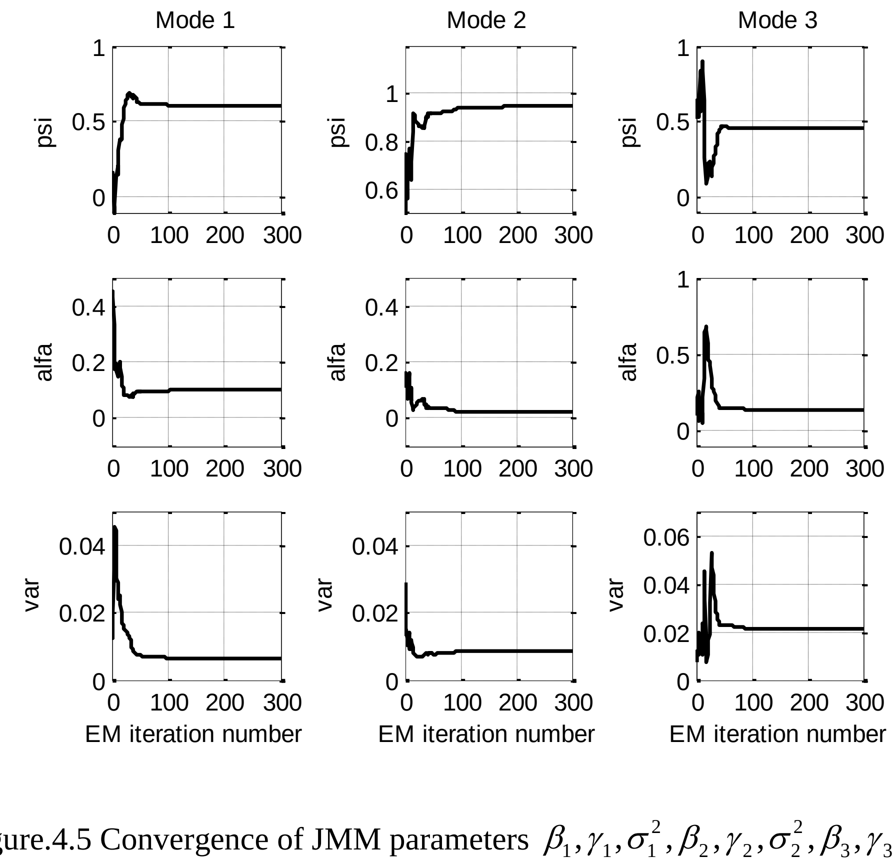

Figure.4.5 Convergence of JMM parameters $\beta_1, \gamma_1, \sigma_1^2, \beta_2, \gamma_2, \sigma_2^2, \beta_3, \gamma_3, \sigma_3^2$

4.4 Model Validation and Estimation

In the previous section we analysed the convergence of the log-likelihood parameter estimation to an stationary value. To validate whether this stationary value is useful for model based state estimation and for online control we compare in this section the evolution of measurement y_k and the estimated values $\{x_{c,k}, s\}$, based on a simulation that implements the JMM model of equation (2.10), with parameter values shown in Table 4.2, identified by the EM-algorithm. In other word, model validation is performed by using the model based estimator using the identified model (obtained via EM parameter estimation) and checking how accurately this estimator predicts the traffic flow. As mentioned in the previous section, modes are not directly observable and hence, *'the smoothed inferences'* about the mode state process are used.

There are two different ways that we propose in which the smoothed inferences can be used for the model based prediction:

(a) most probable mode (*mpm)* ,

(b) weighted empirical distribution function (*wedf*).

Below we compare the performance of the *mpm* and the *wedf* approach as part of the algorithm for the prediction of the traffic flow rates. For the *mpm* approach, we use the natural choice of relating each observation with the most probable mode. The evolution of the modes is shown in Fig.4.6. The first graph of Fig.4.7, which shows more detailed results of *mpm* that has a good predictive tendency. The second approach, we developed the w*edf* approach where we define weights $wf_i = p\left(s_k = s \middle| \mathbf{y}_{k-1}; \theta \right)$, and based on these weights, we predict the measurement by multiplying the weights *wf*$_i$ and the right-side of equation (2.10.b): $\beta_i + \gamma_i \, \alpha_{k-1} + \eta_k$ and then summing the 3 products (as symbolized by $\sum$) which amounts to averaging over modes:

$$\hat{\alpha}_k = \sum_i^3 wf_i \, (\beta_i + \gamma_i \, \alpha_{k-1} + \eta_k)$$

The second graph of Fig.4.7 shows that *wedf* gives slightly better prediction results than *mpm*.

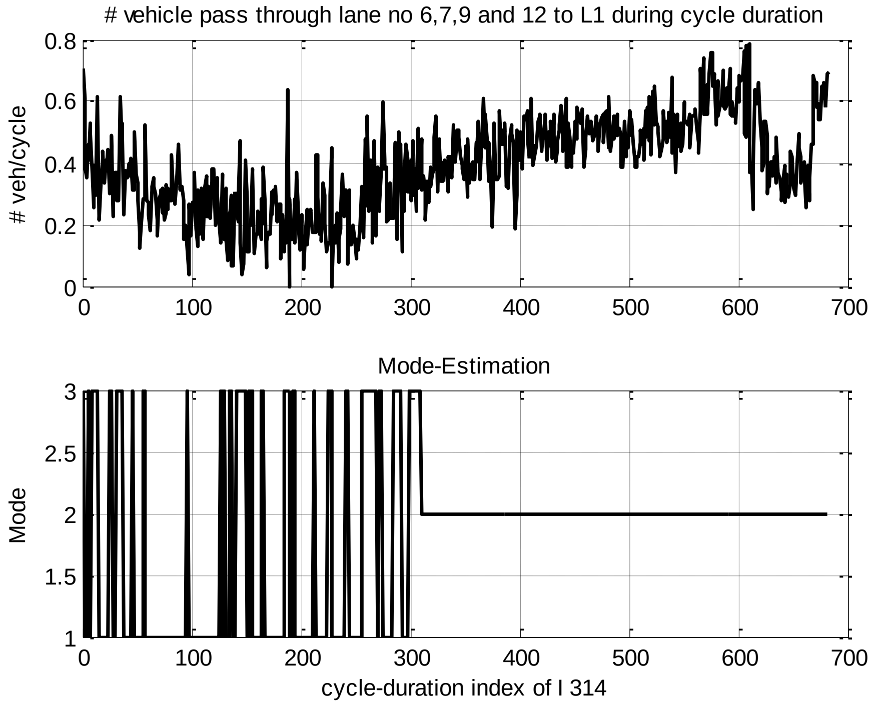

Figure.4.6 Mode-evolution

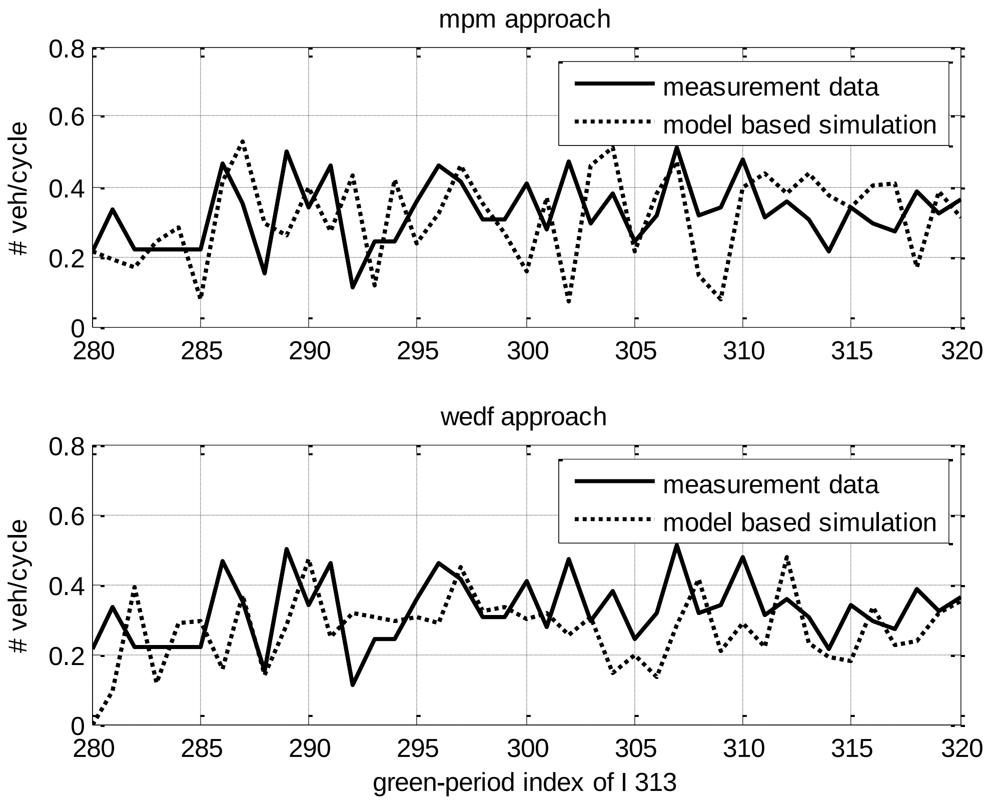

Figure.4.7 Measurement data vs model based simulation : mpm approach (first graph) and wedf (second graph)

It should be emphasized that both approaches *mpm* and *wedf* use *the smoothed inference* that was generated by the iterated EM algorithm. These approaches are not easily implementable in real time due to the need for a bigger memory to save the complete vector of values for $p\left(s_k = s \middle| \mathbf{y}_{k-1};\theta \right)$, but the approach is quite useful for validating the model.

To overcome the limitation of the offline approach, we developed a method that can be used both for validating the model and for estimating traffic flow in an online algorithm. For the online state-estimation method, we only use the parameter θ of the JMM in order to estimate the traffic flow for the next time-window. In this chapter we use a type of particle filter technique called *observation and transition-based most likely modes tracking particle filter* (OTPF) technique for hybrid systems as developed by Tafazoli and Sun [35].

The performance of any state estimator, including estimators for hybrid systems, depends on the prior knowledge of a good model for the plant dynamics and the noise characteristics, including the knowledge of the transition probabilities between discrete modes. The identification of such a model has been done in section 4.2 for the parameters of the hybrid JMM model in equation (4.8). Below we check the accuracy of a Particle Filter (PF) estimator, that uses the parameter values estimated by the EM algorithm as obtained in one time window to obtain state estimates in another time window that is very close to the original time window, so close that one can expect that the identified model parameters are still valid. We do find that this adaptive PF does indeed lead to a sufficiently accurate estimator, validating the usefulness of the proposed method for state estimation and for online control applications. By sufficiently accurate we mean that the state and mode estimates obtained by the adaptive time window shift method are close to those obtained by the offline backward-forward estimation algorithm.

In this chapter, we will not discuss in detail the OTPF algorithm that has been implemented in this experiment, the interested reader can find details in paper [35], but we provide the OTPF based state-parameter estimation algorithm in Chapter.5. Our analysis starts by using the parameters estimated by the EM algorithm in order to generate a simulations programme implementing the JMM model identified in the preceding time window, and with samples for the initial values of traffic flows and traffic modes generated according to some probability distribution. The particles are propagated through the dynamic model in each mode.

This is done recursively as follows [35]: first collect all the next modes to which the current mode has non-zero transition probability, then use the dynamics in each next mode to simulate the particles of continuous state/traffic flow separately. Next for each of these modes, using the available observations, calculate the average weight of all particles of traffic flow in the mode, multiply the average weight with the corresponding transition probability, obtaining a compound weight, compare the compound weights from all the modes and then select the most likely mode. The next step is to resample particles from the most likely mode. It is important to note, that the number of particles N needed is not very high and the filtering calculation is still fast. In this chapter we use N=500

In this paper we use N=500. However the OTPF algorithm still uses a prior distribution (3.27) as an importance density that is independent of the current measurement, the state space is explored without knowledge of the measurement and hence the filter is sensitive to outliers.

We will use three different simulation scenarios with different time-window sizes (and corresponding time shift in the estimation of the modes) in order to study the performance of the JMM based estimators :

(a). Simulation is tested in the period of index 350-450 with a hybrid model, using the EM-estimated parameters based on measurement data from the period with index 1: 350 (corresponding in real time to midnight till approximately 10.30 am).

(b). Simulation is tested in the period of index 450-550 with a hybrid model, using the EM-estimated parameters based on measurement data from the period with index 300: 450.

(c). Simulation is tested in the period of index 575-675 with a hybrid model, using the EM-estimated parameters based on measurement data from the period with index 475-575.

Figure.4.8-4.10 shows the simulation of model validation w.r.t the three scenario above.

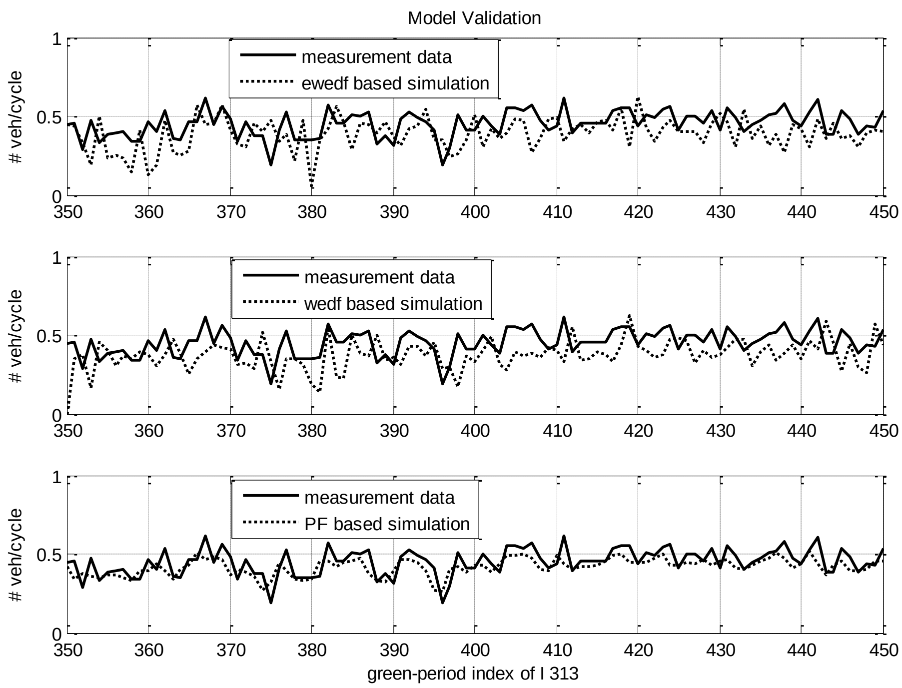

Figure.4.8 Scenario (a) based on JMM model

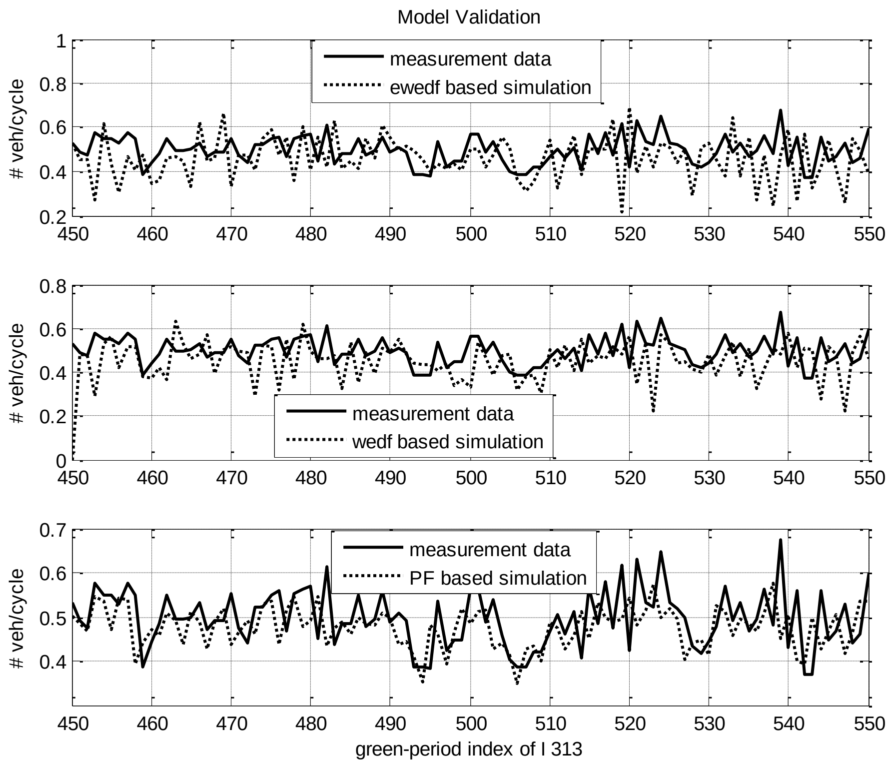

Figure.4.9 Scenario (b) based on JMM model

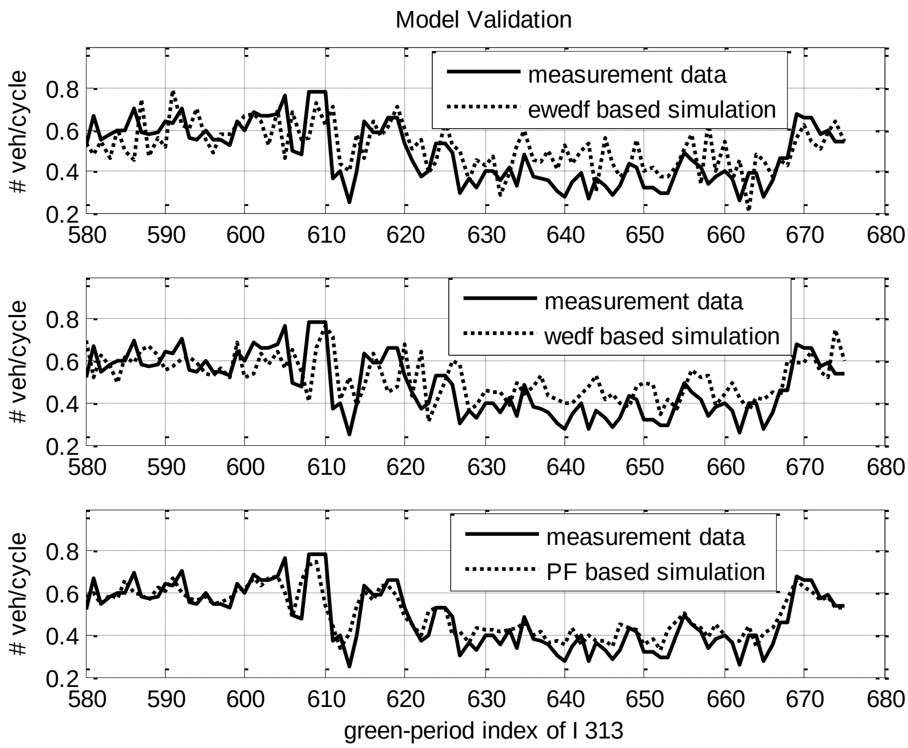

Figure.4.10. Scenario (c) based on JMM model

The results for the 3 scenarios above show that:

(1) EM offline parameter estimation along with a time-window shift technique can be useful and practical for updating the parameters of the JMM hybrid model. The results shows that what choice of window size W is best for estimating and for prediction in the sense that it leads to the most useful results for online traffic light control. One approach to consider is a sliding window method, where the JMM model parameters are updated at times m.W, using measured data y_k over the interval k $\in$ [(m - n).W, m.W-1] to identify the model parameters, and these model parameters are then used over the interval [m.W, (m+1).W) for predicting the traffic flow rate. By choosing the time-window W is long enough ,W=100, the EM technique is able to provides a good model that along with the PF gives a good estimation results.

(2) Based on the update of JMM hybrid model, an online PF estimator for the next time-window can be applied and may be a good candidate as a traffic flow state estimator for coordinating traffic light control in small network. This may of course be combined with the sliding window approach.

The results above confirm that a good model is pre-requisite for predicting traffic flow. In the context of designing feedback control at a signalized intersection, the prediction is an important factor to anticipate the variability of traffic flow coming from neighbouring intersections. This variability should be anticipated by traffic control in terms of green/red switching in order to reduce the congestion and to reduce the delay time. Many works in traffic engineering consider the prediction of traffic flow in designing traffic light control. Interested readers may refer to paper [36,37].

However, this EM offline parameter estimation along with the time-window shift technique has at least two disadvantages: (a) that the performance of state estimator strongly depends on the length of the time window shift, (b) that this offline approach needs significant memory requirements and processing power for storing and processing large datasets.

Certainly, the optimal choice of W is a compromise. An EM algorithm will estimate the parameters more accurately if the time-window W is long enough. This is reasonable since the EM algorithm needs a certain amount of measurement data for accurate parameter identification. But of course this will affect the ability of adaptive filters to detect a change of the model parameters sufficiently quickly. In the context of the coordinated control, the online parameter estimation approach is a critical part of an adaptive filter and finding the best window size W becomes an critical issue.

In order to cope with these two disadvantages of this offline algorithm, in the next chapter we develop the online algorithm by simultaneously performing parameter estimation and state estimation and prediction.

4.5 Conclusions

This chapter proposes a parameter estimation technique for hybrid dynamic system model as a powerful approach for capturing the complicated dynamics of urban traffic flow, including many sources of uncertainty. The model parameters characterize traffic flow conditions that can be classified into three modes and the switch between these three modes is controlled by a first-order Markov chain. The model is characterized by a set of parameters to be estimated using measured data (e.g. from a video camera overlooking traffic) and it is shown that a time-window shift technique may lead to a useful real-time state estimation algorithm that can be part of a feedback control loop.

The study reported in this chapter investigated the proposed approach by using actual traffic flow data and confirmed its validity by showing that the ‘smoothed inferences’ technique and a particle filter based on the identified model provide satisfactory state estimation and correctly capture the random variation of the traffic flow.

However, at least there are two disadvantages are to be taken into account:

(a) the length of the time window shift need carefully to be given in order to deliver a good estimator,

(b) this offline approach needs a good management in terms of the memory requirements and processing power for storing and processing large datasets.

5

Joint state and parameter estimation: online Bayesian approach

5.1 Introduction

Implementing advanced traffic signal control strategies requires real-time data processing for online estimation of the key state variables as well as of the model parameters, neither of which can in practice be measured directly with sufficient accuracy. An extensive literature is available on online traffic flow state estimation for freeway traffic and for arterial roads, often using sub-optimal Bayesian filters and particle filters [42],[43]. Online joint state-parameter estimation for traffic flows in a signalized intersection on the other hand has not received much attention yet. Traffic in urban networks is characterized by stop-and-go phenomena resulting from green/red switching, irregular arrival stream of vehicles, and from complicated interactions between conflicting traffic streams. In [14] and in Chapter 2 and 4, we showed that the variability of urban traffic flow at near traffic lights in the time scale of green/red period can be approximated by an autoregressive process (AR), whose model parameters change over time depending on the mode of traffic operation (free-flowing, congested and faulty); the mode of operation can be represented by a first-order Markov-chain.

Coordination between traffic lights at neighboring intersections plays a major role in reducing congestion and avoiding grid-lock in an urban network. Coordination requires anticipation of queue sizes at neighboring intersections as quickly as possible, and this anticipation is only possible if a good model is available for predicting future traffic behavior, and in particular future queue sizes. In order to use a dynamic queue length model one must estimate the parameters of this model through online state-parameter estimation. Automatic adaptation to changing system condition becomes possible, provided we can estimate the model parameters online with sufficient accuracy. In the spirit of delivering a basic step to the coordination, we have to deal with the computational complexity and accuracy for the estimation. Can we propose one of these feasible solution to deal with those problems? This is a question that we have to answer. *However, note that in this thesis we are not studying the coordination itself.*

In the previous chapter, we used an offline expectation-maximization (EM) algorithm for parameter identification. In combination with a time window shift technique this was shown in section 5.4 to leads to an adaptive PF traffic flow state estimator. However, in practice W is so large that the adaptation of the offline method in section 4.4 is too slow. This is the main reason why in this chapter we improve this algorithm, simultaneously performing online parameter estimation and online traffic flow and queue length estimation and prediction.

In this chapter, in terms of modeling, we use the complete model of queue lengths evolution and of traffic flow as discussed in chapter 2. As opposed to chapter 4 that focused on the modeling of traffic flow by using 3 modes, this chapter focuses on the queue-length dynamic by defining a traffic flow model consisting of 2 modes. The reason to use only 2 modes is to reduce the computational complexity and hence the time constraint of the algorithm. Based on the queue length dynamics with 2 modes of traffic flows, we develop PF algorithms that can reliably and efficiently predict the queue length using joint state and parameter estimation for our stochastic hybrid model (SHM).

The extension of particle filters to joint state and parameter estimation for hybrid SHM models is non-trivial. The conventional strategy is to add a random walk to the parameters and then augment the state-space with the parameters for joint estimation as shown in section 3.2.2. The use of a random walk however increases the covariance of the parameters (dispersion), making the posterior distributions too diffuse. As such, the precision of the resulting estimates are inevitably limited and the estimates may not converge.

The key to yielding converging parameter estimates is to make the variance of the random walk decay with time. To achieve this, as explained in chapter.3, Liu and West [20] suggested using kernel smoothing with shrinkage for parameter evolution. Hence, the unwanted information loss effect or over-dispersion of the samples for the fixed parameter caused by the independent random shock np_k of (3.34) is corrected by the use of (3.40) which introduces negative correlations between θ_k and the random walk np_k. The optimal selection of the kernel parameter $h_k \in [0,1]$ maximally reducing the over-dispersion in the PF remains a difficult problem. The current practice for tuning of h_k is ad-hoc. Liu and West in paper [20] suggested choosing h_k=0.1, whereas Chen in [21] selects h_k as the optimum for a historical data-set, and then applies this choice for future batches. These ad-hoc rules define a constant h_k, for which optimality cannot be established w.r.t the online data.

A rule, based on the Kullback-Leibler (KL) divergence optimization, for selecting the optimal value for the kernel smoothing parameter h_{t_k} has been proposed by Tulsyan in [45] for nonlinear model dynamic. In this chapter, we extend this KL divergence optimization to the SHM to estimate the AR parameters.

In order to validate the performance of our algorithm we need data that simultaneously record traffic flow rates and queue sizes. It was impossible to obtain reliable real data of sufficient detail due to the technological limitations of the traffic sensors of the partners with whom we collaborated. Therefore we used synthetic data for the validation of our online joint state and parameter estimation algorithms, as developed in this chapter. We generated synthetic traffic flow and queue-length data using a VISSIM traffic micro-simulator, including realistic levels of state and sensor noise, and used these to simulate realistic values for the sensor output data of a system. By comparing the estimated and predicted queue sizes obtained using our PF estimator with the synthetic data we show that the joint state and parameter estimation tools proposed in this paper exhibits good performance for realistic scenarios. Note that VISSIM data were obtained using a microscopic traffic flow model, that is completely different from the SHM model that we proposed for performing joint state and parameter estimation. Hence there is no undue bias in favor of our algorithms in this validation.

The main novel contributions of this chapter are: (a) extending the *observation and transition-based most likely modes tracking particle filter* (OTPF) along with kernel smoothing method via online optimization for online joint state-parameter estimation of SHM, leading to an adaptive filter; (b) The use of Dirichlet distributed random samples in order to better estimate the transition probability matrix (TPM), describing the transition probability matrix Π between discrete states.

5.2 Problem formulation

In this chapter we follow the notation and model formulation as explained in chapter 2. By using online measurement data of the existing traffic flow for online parameter estimation, we build a dynamic traffic flow model and use this model for adaptively predicting future values of the arrival flow rate and the departure flow rate properly describing the variability of these traffic flow rates; this ultimately allows prediction the queue-length evolution.

Let us focus on the traffic flow of a particular interested movement L1. The description in chapter 2.2 implies that one treats $\lambda_{1,t_{2k}}, \lambda_{1,t_{2k+1}}$ and $\mu_{1,t_{2k}}$ as three independent processes: the mode of operation for the arrival flow rates in the red $\lambda_{1,t_{2k+1}}$ and in the green period $\lambda_{1,t_{2k}}$ of the cycle may be different e.g. due to different conditions at adjacent intersections. Surely these processes are in practice strongly correlated, but it will be very difficult to obtain sufficiently detailed statistics on this correlation and hence, we simplify the model by ignoring this correlation. Therefore, the fact that there are 3 flow rates to be estimated, each with 8 parameters (2 modes, each with 3 AR parameters), and 2 entries for Π. Since we treat the 3 flows as independent, we can do the estimation for each one separately; and so we only need 8 parameters to be estimated by the online algorithm. This is almost half of what we needed with 3 modes, explaining the reduction in computational cost. Remember that we assume each traffic flow has 2 mode of operation.

We split up the vehicle counts in a red and a green period as a measurement period because: (a) the queue length model in equation (2.6)-(2.9) has two periods, one covering the start of the cycle to the end of green, the other between the end of green and the end of the cycle, (b) if we look at the values of mean and variance of the traffic flow generated by the VISSIM simulator (simulating a signalized intersection during one hour with a constant arrival rate; for details see section 6.4) with a cycle length 80 s and green period 35s we see in Table 5.1 that selecting update intervals of around 35sec (typical green period) provide low variance. The variance is reduced only marginally by considering a full cycle (80sec).

Table 5.1 Mean and variance of traffic flow

Update period	Mean (veh/s)	Var (veh/s)^2
5s	0.1738	0.0354
20s	0.1738	0.0086
35s	0.1745	0.0049
80s	0.1748	0.0038

The complete hybrid model of the queue length (2.6)-(2.9) and of traffic flow (2.10.b)-(2.10c) can be presented in one single form by considering for multi-mode *s* and written as follows:

State equation:

$$x_{t_k} = f_s(x_{t_{k-1}}, \theta, \eta_{t_k}), \qquad k = 0, \ldots, N-1; \tag{5.1}$$

Measurement equation:

$$y_k = g_s(x_{k-1}, \theta, n_k), \qquad k = 0, \ldots, N-1; \tag{5.2}$$

where :

$\{x_{t_k}\} = \{q_{t_k}, (\lambda_{1,t_{2k}}, s_{t_{2k}}), (\lambda_{1,t_{2k+1}}, s_{t_{2k+1}}), (\mu_{1,t_{2k}}, s_{t_{2k}})\}$ and $s_{(\bullet)}$ is a mode of the system according to the flow, where α_{t_k} is the generic notation for arrival flow in movements L_i during green period $\lambda_{i,t_{2k}}$ and red period $\lambda_{i,t_{2k+1}}$ and also for departure flow $\mu_{i,t_{2k}}$.

5.3 Online Joint State-Parameter Estimation

In this section we develop a complete approach to on-line Bayesian joint state and parameter estimation for hybrid stochastic systems $\{x_{t_k}\} = \{q_{t_k}, (\lambda_{1,t_{2k}}, s_{t_{2k}}), (\lambda_{1,t_{2k+1}}, s_{t_{2k+1}}), (\mu_{1,t_{2k}}, s_{t_{2k}})\}$ as defined by equation (5.1) and (5.2), using an extended state vector representation with its parameters θ_k as one group of AR parameter β, γ, σ as shown in (2.10). We discuss the Bayesian framework for particle filtering (PF) and its extension to a hybrid system using the OTPF technique. The combined state-parameter estimation will be discussed using the joint posterior distribution. Reducing over dispersion in the PF for a hybrid system needs developing two techniques, which are the main contribution of this chapter:

(a) to select the optimal value of the kernel parameter via online optimization,

(b) to estimate the transition probabilities using samples of mode transitions generated according to a Dirichlet distribution.

These two techniques leads to the automatic adaptation of the estimator to changing system conditions. Carvalho in his paper [24] proposed the multinomial distribution to estimate the transition probabilities of Markov switching stochastic volatility model and simply setting h_k=0.1.

The *state space mode* (5.1.a) combined with a random walk added to the parameters, for the augmented state process $\{\chi_{t_k}\} = \{x_{t_k}, \theta_{t_k}\}$ and combined with *the measurement model* (5.2) specifies the probability distribution of the data y_{t_k}.

$$\begin{bmatrix} x_{t_k} \\ \theta_{t_k} \end{bmatrix} = \begin{bmatrix} f_s\left(x_{t_{k-1}}, \theta_{t_{k-1}}\right) \\ \theta_{t_{k-1}} \end{bmatrix} + \begin{bmatrix} \eta_{t_k} \\ np_{t_{k-1}} \end{bmatrix}$$

Remember that η_{t_k}, $k = 1, 2, ...$ is an independently identically distributed sequence of zero mean Gaussian random variables with variance $\sigma^2(s)$, with $\sigma^2(s)$ also a mode dependent parameter to be identified.

This description of the evolution of the augmented state, together with the distribution of the initial conditions, provides the necessary information to predict the queue sizes a few cycles ahead.

5.3.1 State estimation of hybrid system

A number of suggestions have been proposed in order to make the standard PF applicable to the state estimation problem of hybrid processes: *interacting multiple model particle filtering* (IMMPF) [49] and *observation and transition-based most likely modes tracking particle filter* (OTPF) [35]. IMMPF consists of three steps: (1) mixing/interaction of the mode-conditioned estimates at the beginning of the estimation cycle; (2) mode-conditioned state estimation (prediction and measurement update of the state), done independently for each mode by appropriate (mode-matched) filter module; (3) mode probability update and estimation done using the outputs of all the mode-conditioned filters.

In this subsection, we will review the OTPF applied to stochastic hybrid system model (the jump Markov model of section 2.2). In this case, one needs to calculate the probability density function (pdf) of the hybrid system $p(x_{t_k}, s_{t_k} | \mathbf{y}_{t_k})$, where x_{t_k} still denotes the state of the AR models, while s_{t_k} indicates the mode of the system at time t_k where s=1,2,…,K. Important to note is that in this thesis, the problem is much more complicated not just because we need to estimate the mode(or the likelihood of the different modes), but also because we need to estimate the transition probability matrix (TPM) for the Markov process describing the mode evolution. For estimating the AR parameters and TPM we will discuss in the next section on state-parameter estimation. In this subsection we focus on the state estimation assuming that AR parameter and TPM are known a priori, which is a standard a requirement in the usual applications of the OTPF.

OTPF, unlike IMMPF, does not combine the discrete state (modes) with the continuous states to construct the system states. Instead, the modes are considered as unknown system parameters which need to be estimated, something more akin to the joint state-parameter estimation explained in the preceding section. After the estimation of the most likely mode (we estimate the likelihood of each mode at each time step), the estimation of the hybrid system is reduced to an ordinary system and a particle filter can be properly applied. Contrary to IMMPF, which takes all modes into account and assigns a fixed number of particles in each mode *s*, OTPF chooses the most likely mode $\hat{s}_{t_k}$ at each time step and the evolution of all particles at the next time step is evaluated according to the model corresponding to this most likely mode $\hat{s}_{t_k}$. This ensures that OTPF has a lower computational load at the price of a less accurate result. In this thesis we use further on OTPF. This method is explained detail below.

State estimation based on OTPF calculates the conditional pdf of hybrid system : $p(x_{t_k}, s_{t_k} | \mathbf{y}_{t_k})$, $\mathbf{y}_{t_k} = \{y_j, j = 0, \ldots, t_k\}$. Since at each time step the system only follows one mode, it is reasonable to assume that it is actually only following the most-likely mode $\hat{s}_{t_k}$:

$$\begin{aligned} p(x_{t_k}, s_{t_k} \big| \mathbf{y}_{t_k}) &= p(s_{t_k} | \mathbf{y}_{t_k})\, p(x_{t_k} | s_{t_k}, \mathbf{y}_{t_k}) \\ &\approx p(\hat{s}_{t_k} | \mathbf{y}_{t_k})\, p(x_{t_k} | \hat{s}_{t_k}, \mathbf{y}_{t_k}) \\ &= p(\hat{s}_{t_k} | \mathbf{y}_{t_k})\, p_{\hat{s}_{tk}}(x_{t_k} | \mathbf{y}_{t_k}) \end{aligned} \tag{5.3}$$

Since $p(\hat{s}_{t_k} \mid \mathbf{y}_{t_k})$ is a constant in (5.3) its effect will be absorbed in the normalization constant and will not affect the basic PF algorithm.

The details of the OTPF implementation are described for a given hybrid system with *K* modes where the system and observation models in each mode are known. The initial mode is assumed to be known, otherwise, the most likely mode is chosen, based on *a priori* knowledge of the process.

Remark: Herein after, in this section, *k* denotes the sampling time t_k.

First, all the weights w_k^i of particles for the continuous state of the system in the given mode s_k^j are sampled. This follows by the following recurrence procedure: first step collect all the modes for which the current mode has *non-zero transition probability*. Use the dynamics in each mode *s* with non-zero likelihood to predict the next value for each of the N particles separately. For each mode *s*, using the available observations, calculate the average weight of all the particles in the mode *s*, multiply the average weight $\sum_{i=1}^{N} w_k^i / N$ with the corresponding transition probability, resulting in a compound weight $\overline{w}$ as shown in equation (5.4) table 5.2. Equation (5.5) compare the compound weights $\overline{w}$ from all the modes and *select the most likely mode*.

Table 5.2 Standard OTPF Algorithm

Step 1. Initialization The mode at time k=0 is given as s_0. For i=1,…,N, from an initial Gaussian distribution in mode s_0 and set k=1. **Step 2. Prediction** - For any mode s_k^j such that transition probability $p_{s_{k-1}s_k^j}$ from mode s_{k-1} to s_{k-1}^j is not zero (j=1,…,K) where K is the number of such modes), sample $\tilde{x}_k^i \sim p_{s_k^j}(x_k \mid x_{k-1}^i)$, for i=1,…,N (Note that the same notation is used for the particles in different mode for simplicity) - For each mode s_k^j, evaluate the importance weights $w_k^i = p_{s_k^j}(y_k \mid \tilde{x}_k^i)$ for i=1,…,N (again note that the same notation is used for the weights in different modes for simplicity). **Step 3. Mode Selection** a. Average the total particle weights in each mode $s^* = s_k^j$ and multiply by the transition probability:

$$\overline{w}_k^{s*} = p_{m_{k-1}m_k^j} \sum_{i=1}^{N} w_k^i / N \tag{5.4}$$

b. Find the most likely mode :

$$s_k = \arg\max_{m^*} \{\overline{w}_k^{m^*} \text{ for all } s_k^j, j=1,\ldots,K \} \text{ with } m^* = m_k^j \tag{5.5}$$

Normalize the weights of particles in mode s_{tk}

Step 4. Resampling

a. Resample N new particle $\{x_k^i$, for $i=1,\ldots,N\}$ with replacement from the particle in mode s_{tk} $\{x_k^i$, for $i=1,\ldots,N\}$, according to the importance weights.

b. Set for k=k+1 and go to step 2

The next step is to resample particles from the most likely mode. Particles in other modes are discarded. It can be argued that this approach may take a long time, waste calculations, and does not scale to a large number of modes. However, in this chapter, the number of modes is only 2 and the calculation turns out to be quite efficient. The complete algorithm is described and shown in table 5.2

As shown in [48] OTPF has a lower computational load, compared to IMMPF, but at the expense of a loss of accuracy since OTPF is biased and sensitive to observation outliers. In this thesis we use OTPF because of its lower computational load. This reduction in computational cost is important since we are not only considering the state estimation but joint state-parameter estimation where the computational load is demanding. Moreover we want to obtain an online implementation, which makes the computational issue even more critical.

5.3.2 State-Parameter estimation for stochastic hybrid system

In hybrid system, the problem becomes much more complicated because we need to estimate: (a) the mode(or likelihood of the modes); (b) parameter of AR process; (c) the transition probability matrix (TPM) for the Markov process describing the mode evolution. Can we estimate all of them by online estimation? The aim of this section is to developed standard OTPF to joint state-parameter estimation by combining kernel smoothing and Dirichlet distribution to address this question.

Originally OTPF was proposed solely for state estimation of hybrid systems, relying on prior knowledge of the parameters $\theta(s_k)$, and also of the transition probabilities Π. Since we do not know the parameters of a model then we must also identify all the parameters of the hybrid system model, using the online data observed during the plant operation, thus achieving automatic adaptation of the state predictors to the changing system condition. It can be achieved by finding a good compromise between computational efficiency and accuracy, for estimating the states-parameters and for estimating *the most likely mode*. The parameters of the AR model will be estimated using optimal parameter tuning for kernel smoothing, improving on the approach described in chapter 3, while the estimation of the transition probability matrix Π will be achieved by generating random mode switches according to a Dirichlet distribution by updating the parameter n_{ij} that are calculated based on the most likely mode in OTPF. Here n_{ij} is the number of one-step transitions from i to j generate by the random sample generation algorithm.

5.3.2.1 Parameter tuning for Kernel smoothing

Please see Chapter 3 for introductory issues about kernel smoothing. The goal of kernel smoothing is to reduce the over-dispersion and optimal selection of kernel parameter $h_k \in [0,1]$ is the crucial part for the successful application of this technique. We will extend the technique proposed by Tulsyan in his paper [45] to stochastic hybrid systems by combining it with Dirichlet distribution in the OTPF loop. The novel OTPF for Joint state-parameter estimation thus obtained is shown in Table 5.3.

The technique determines an optimal rule for h_k based on an on-line optimization is proposed in [45] . There are two reasons why here one needs an optimal rule. First, because this thesis uses SIR PF which is inefficient in handling the situation that *supp* $p(\chi_k \mid \mathbf{y}_{k-1})$ is larger or smaller compared to *supp* $p(\chi_k \mid \mathbf{y}_k)$ then only a few particles are assigned significant weights, where $\chi_k = \{x_k, \theta_k\}$. $p(\chi_k \mid \mathbf{y}_k)$ can be calculated using Bayes' rule in (3.35). This is because in a SIR PF $q(x_k \mid x_{k-1}^i, y_k) = p(x_k \mid x_{k-1}^i)$, then the particles from (3.27) are generated without taking the current measurement into consideration. This inefficiency is much more of a disadvantage for the multi-model case as SHM. Second, in state-parameter estimation as indicated by Liu and West, selecting kernel parameter h_k in order to reduce the over-dispersion in the PF is a critical step. This optimal tuning rule proposed in [45] minimizes the Kullback-Leibler (KL) divergence $D(h_k)$ between $p(\chi_k \mid \mathbf{y}_{k-1})$ and the target posterior density $p(\chi_k \mid \mathbf{y}_k)$ at each sampling time. This enables adaptation of the SIR PF for combined state-parameter estimation. Proposition 1 provides an optimal tuning rule for controlling the kernel width and for making an SIR PF adaptive.

Remarks: Remember that $x_k = \{x_{c,k}, x_d\}$ consist of continuous state and discrete state/modes. Hence we have to deal with part density (giving a conditional density for the continuous state variable) and part conditional probability (for the discrete mode). However, OTPF calculates the conditional pdf of hybrid system as given in (5.3) that it will not affect the basic SIR PF algorithm.

Proposition 1. The optimal value h_k^* of the parameter h_k at time k that minimizes the KL divergence $D(h_k)$ between the $p(\chi_k \mid \mathbf{y}_{k-1})$ and target posterior density $p(\chi_k \mid \mathbf{y}_k)$, is:

$$\begin{aligned} h_k^* &= \underset{h_k \in [0,1]}{\arg\min} \left[-\sum_{i=1}^{N} w_k^i \log\left[\omega_k^i\right] \right] \\ &= \underset{h_k \in [0,1]}{\arg\min}\ \hat{D}(h_k) \end{aligned} \tag{5.6}$$

where: $\hat{D}(h_k)$ is a Sequential Monte-Carlo (SMC) estimate of $D(h_k)$. Note that the dependence of $\hat{D}(h_k)$ on h_k can be established from equation (3.40) and (3.34). $\{w_{t_{k-1}}^i\}_{i=1}^N$ and $\{\omega_{k-1}^i\}_{i=1}^N$ are the particle weights given in (5.7.a) and (5.7.b), respectively :

$$\omega_k^i = \frac{w_k^i \ p(y_k \mid \chi_{k|k-1}^i)}{\sum_{i=1}^{N} w_k^i \ (y_k \mid \chi_{k|k-1}^i)} \tag{5.7.a}$$

$$w_k^i \quad \propto \quad w_{k-1}^i \ p(y_k \mid x_k^i) \tag{5.7.b}$$

where:

$$\{\chi_{k|k-1}^i\}_{i=1}^N \sim \tilde{p}(\chi_k \mid \mathbf{y}_{k-1}) \ \text{ and } \ \chi_k \overset{\Delta}{=} \{x_k, \theta_k\}$$

Proof: See [45].

5.3.2.2 *Dirichlet distribution*

As mentioned in the introduction section, this chapter focuses on defining SHM traffic flow model with 2 discrete modes, free flowing and congested. For estimating the transition probabilities of the mode process s_{t_k}, we propose an ongoing updated Dirichlet distribution in the OTPF. The Dirichlet distribution assigns probabilities to a vector of K integers. The distribution depends on K continuous parameters [δ_1,δ_2] where $\delta_k \in [0,1]$ and. $\sum_{i=1}^{K} \delta_k = 1$. The Dirichlet distribution is suitable for defining a distribution over the random values of the categorial distribution, i.e. for specifying the outcome of observing one of K possible outcomes. In our case traffic flow model, we set K=2.

The i-th row {π_{i1},π_{i2}} of an estimate of the TPM Π_i always has row sum equal to 1. Let $\hat{\Pi}_i$ be random variables: $\Pi_i \sim D(\delta_{i1},\delta_{i2})$, where D denotes a Dirichlet distribution. The reason we choose a Dirichlet distribution is that (see [52]); the updated distribution of $\hat{\Pi}$ given s_{t_k} or written as $\Pi | s_{t_k}$, after updating based on new observations (or new particles that are generated) is again a Dirichlet distribution, where s={s_1,s_2}:

$$\Pi | s_{t_k} \sim D(\delta_{i1} + n_{i1}, \delta_{i2} + n_{i2}) \tag{5.8}$$

where n_{ij} is the number of one-step transitions from current mode i to next mode j, where i,j={1,2}. The next mode j is determined by the most likelihood mode in OTPF. By calculating n_{ij} one can sequentially update the TPM.

A natural question to ask regarding the Dirichlet distribution is how to sample from it. We use a method based on transforming Gamma-distributed random variables. We will argue that generating samples from the Dirichlet distribution using Gamma random variables is more computationally efficient than both the *urn-drawing* method and the *stick-breaking* methods [53]. This method has two steps:

Step.1: Generate gamma realizations: for p=1,2, draw a number z_p from the Gamma distribution $\Gamma(\delta_{ip} + n_{ip}, \hbar)$ where $\hbar$ is the *scale parameter* and $(\delta_{ip} + n_{ip})$ is the *shape parameter.*

Step 2: Normalize them to perform a probability mass function (pmf). Then $\hat{\pi}$ is a realization of $D(\delta_{i1}+n_{i1},\delta_{i2}+n_{i2})$ and the estimated vector $\hat{\Pi}_i$ ($1 \le i \le j$), for i=1,2, can now be simulated from (5.9) by letting:

$$\hat{\pi}_{i1} = \frac{z_1}{\sum_{p=1}^{j} z_p} \ldots\ldots \hat{\pi}_{i2} = \frac{z_2}{\sum_{p=1}^{j} z_p} \qquad (5.9)$$

where $\hat{\pi}_{(\bullet)}$ are entries of the estimated vector $\hat{\Pi}_i$. Important to note that $\hat{\Pi}_i$ is a vector notation to represent Dirichlet distribution over a finite number of possible values.

The most important reason why we use OTPF filters is the ability of the OTPF to provide information about the most-likely mode $\hat{s}_k$ which in turn allows us to estimate the transition probabilities Π using the Dirichlet distribution.

Table.5.3. OTPF Procedure for Joint state-parameter estimation

Step 1. Initialization

The mode at time k=0 is given as s_0. For i=1,…,N, from an initial distribution in mode s_0 from an initial distribution in mode s_0 and set k=1.

Step 2. Prediction

- For any mode s_k^j such that transition probability $p_{s_{k-1}s_k^j}$ from mode s_{k-1} to s_{k-1}^j is not zero (j=1,…,K) where K is the number of such modes), sample $\tilde{x}_k^i \sim p_{s_k^j}(x_k \mid x_{k-1}^i)$, for i=1,…,N (Note that the same notation is used for the particles in different mode for simplicity).
- For each mode s_k^j, evaluate the importance weights $w_k^i = p_{s_k^j}(y_k \mid \tilde{x}_k^i)$ for i=1,…,N (again note that the same notation is used for the weights in different modes for simplicity).

Step 3. Kernel smoothing

- Doing parameter estimation using Kernel Smoothing (3.40) and then find the optimal value h^*_k of the parameter h_k at time t_k that minimizes the KL divergence $D(h_k)$ using (6.6)

Step 4. Mode Selection

a. Average the total particle weights in each mode $s^*=s_k^j$ and multiply by the transition probability:

$$\overline{w}_k^{s^*} = p_{m_{k-1}m_k^j} \sum_{i=1}^{N} w_k^i / N \qquad (5.4)$$

b. Find the most likely mode :

$$s_k = \arg\max_{m^*} \{\overline{w}_k^{m^*} \text{ for all } s_k^j, j=1,\ldots,K \} \text{ with } m^* = m_k^j \tag{5.5}$$

c. Normalize the weights of particles in mode s_{tk}

d. Estimate transition probabilities matrices using Dirichlet (6.9)

Step 5. Resampling

a. Resample N new particle $\{x_k^i$, for $i = 1,....,N\}$ with replacement from the particle in mode s_{tk} $\{x_k^i$,for $i = 1,....,N\}$, according to the importance weights.

b. Set for k=k+1 and go to step 2

Implementation of the OTPF based joint state-parameter estimation for stochastic hybrid system as shown in Table 5.3 below, which is the extension of Table 5.2 by adding step 3. kernel smoothing and step 4.d estimation TPM.

5.4 Online Bayesian Performance Evaluation

To test the performance of the proposed estimation and prediction algorithm for the queue length in a field experiment is difficult. We have not been able to find data where simultaneously traffic flow and queue length where recorded with sufficient accuracy over the successive cycles of the traffic light, as considered in our model. Fortunately the advanced computational power and the flexibility that state-of-the-art computer-based simulation software offers, makes it possible to validate our algorithm using the VISSIM traffic micro-simulator.

We use VISSIM as a microscopic traffic simulator generating synthetic traffic data implementing a detailed model with the same traffic flow rates as for the hybrid dynamic SHM model introduced in Chapter 2. The simulated output can be made more realistic by generating the noise as required for realistic representation of traffic irregularity, mode changes. Moreover VISSIM allows the user to read out data like current traffic flow, and queue length at different locations, data that after adding measurement noise, can be used as simulated output of real traffic sensors. These synthetic output data y_k simulate the sensor output available for online analysis. The output of such a simulation run provides the noisy data about traffic flow rates under various conditions, for the time intervals corresponding to the cycle of the traffic lights, used as input for our joint state and parameter PF-estimator as described in section 5.3. In order to achieve fast results applicable to online implementation our PF particles are generated by our hybrid model of section 5.2, which is very different from the microscopic VISSIM model. Hence comparing the queue length obtained via microsimulation with the estimations and predictions obtained via the PF estimator provides a fair and honest way of validating the correctness of the proposed joint state and parameter estimator of Table 5.3. Please note that in the OTPF procedure for joint state and parameter estimation, step 3 and step 4.d are the additional procedure to parameter estimation.

Our simulation experiment represents the typical situation at one intersection, with 2-way approach roads of length 300 m each, the distances between the source generating the traffic flow and the first measurement station, resp. the second station being 0m, resp.300m; the second station is located at the stop-line. The position of each vehicle is recalculated by the VISSIM simulator every 1 second and the distance between vehicles is exponentially distributed, with parameters selected so that there are on average 576 vehicles per hour.

In VISSIM, the author didn’t find the way to change the generated traffic flow during in the loop simulation. This makes it difficult to simulate the transition between two different mean of traffic flows. The uniform distribution of speed in this validation experiment is 35 to 65 kph, chosen on the basis of prior experience with VISSIM. The average distance between stopped cars and also between cars and stoplines, signal heads, and so forth is 2.0 m, uniformly distributed in the interval [1.0m 3.0m]. The lateral behavior parameters allow overtaking wherever legal and when traffic flow conditions allow this. We assume that one vehicle occupies the full width of one lane in VISSIM. The type of vehicles is mixed traffic consisting of cars and motor-cycles. The probabilities of generating cars (resp. motor-cycles) is 70% (resp. 30%). The minimum (maximum) length of a car is 4.1 m (4.7 m) and we take 4.4 m as the average of length of a car. The length of a motor-cycle is 1.4 m. For the sake of simplicity, the traffic model implemented in the PF algorithm does not distinguish between cars and motorcycles and the length of all vehicles is approximated by the average (0.7*4.4 m + 0.3*1.4 m) = 3.5 m. Based on these parameters setting, in VISSIM, where the experimental synthetic data are generated, one treats $\lambda_{1,t_{2k}}, \lambda_{1,t_{2k+1}}$ and $\mu_{1,t_{2k}}$ as strongly correlated processes due to the intricate dynamics of traffic flow and queue-length in space-time of the topology of network. But in the SHM model, we simplify the model of $\lambda_{1,t_{2k}}, \lambda_{1,t_{2k+1}}$ and $\mu_{1,t_{2k}}$ as three independent processes as mentioned previously. We show below that despite these differences between the VISSIM model and the model used in the PF, and many other simplifications implemented in the SHM model underlying the joint state and parameter estimation, good estimation and prediction results can be obtained.

The traffic signals in our simulation operate according to a fixed cycle. The cycle length was set to 80 second with a green duration of T_g =35 second, amber duration is 3 second (ignored in the SHM) and all-to-red is 2 second. The vehicle flow generated at the source location is 0.16 veh/sec and the simulation time interval is 3600 sec (1hr). In VISSIM, it is noted that the number of vehicles is counted by meter according to the classification of the vehicle whether it is a car or a motorcycle. In Table.1, the average traffic flow is ≈ 0.17 veh/sec and this value is computed based on the average length of vehicles 3.5 m, meaning that one does not distinguish between cars and motorcycles. This approach is then used in PF algorithm. The difference on the value of traffic flow can be assumed to be the result of the a measurement noise in our model.

For performance evaluation purposes, we consider the VISSIM microscopic approach for generating the actual queue-length. The microscopic simulation approach, allows the current queue length is to be measured at every time step, including at the switching times, which is when we need “the true” (albeit synthetic) queue size in order to compare to what our model predicts. The actual VISSIM queue length is recorded by putting the queue counter in the front of stop-line *at the end of every red light* (cycle-by-cycle).

In the PF approach, the queue-length is defined based upon the cumulative numbers of vehicles traversing sensor locations (see fig.2.5.b with red line for arrival flow and green line for departure flow in the lane L_1) according to (2.9)-(2.12). The number of vehicles crossing sensor locations in the VISSIM model are recorded during successive phases of the traffic signal, providing data for calculating time series data $\lambda_{1,t_{2k}}, \lambda_{1,t_{2k+1}}$ and $\mu_{1,t_{2k}}$ as shown in Fig.5.1.

Prediction of traffic flow over one or two cycle ahead is performed by using JMM equation (2.10) . Let assume at time t_k , the parameter $\theta=\{\beta,\gamma,\sigma^2\}$ and also mode s_{t_k} has been identified by joint state-parameter estimation and based on the equation (2.13) then the traffic flow rate $\alpha_{t_{k+1}}$ can be predicted and so on for the $\alpha_{t_{k+2}}$.

Fig.5.1 show that the PF queue length estimator and predictor gives results close to the ”synthetic” VISSIM queue length. However, since *the VISSIM model was not itself validated* and hence, it would be more justified to interpret the results as comparison between developed model and theoretical model implemented using the VISSIM tool. The biggest difference between synthetic and estimated queue length are at index 23; detailed analysis of the measurement data indicates that around that time almost all the vehicles that pass the detector are cars, causing the PF simplification of ignoring the difference between cars and motorcycles to become significant.

The implementation of the joint state and parameter estimation with optimal tuning kernel smoothing was carried out under the following initial conditions: parameters of AR model γ=0.5 variance=0.05, π_{11}=0.9,π_{22}=0.9, where $\pi_{(\bullet)}$ are entries of TPM Π, certain $\{\lambda_0\}$=30 . It turns out that using a particle filter with N=600 samples give a good result in terms of queue length and traffic flow prediction (1-cycle/ 2-cycle ahead) as shown in Fig.5.1-Fig.5.2.

We show in table 5.3 the root mean square (RMS) error as a measure of performance, comparing ”real” (albeit in our experiment synthetic data) against the predicted values generated by PF and also against the an average of traffic volume = 21.78 obtained from historical data. RMS error is defined as follows:

$$RMS = \sqrt{\sum \frac{(data - pred)^2}{M}}$$

where *pred* indicate the predicted values by PF and ’average method’ and M is the size of data vector. Table 5.4 shows that the performance of the OTPF algorithm of table 5.3 is much better than the traditional approach that defines predictions using *the average* of historical approach.

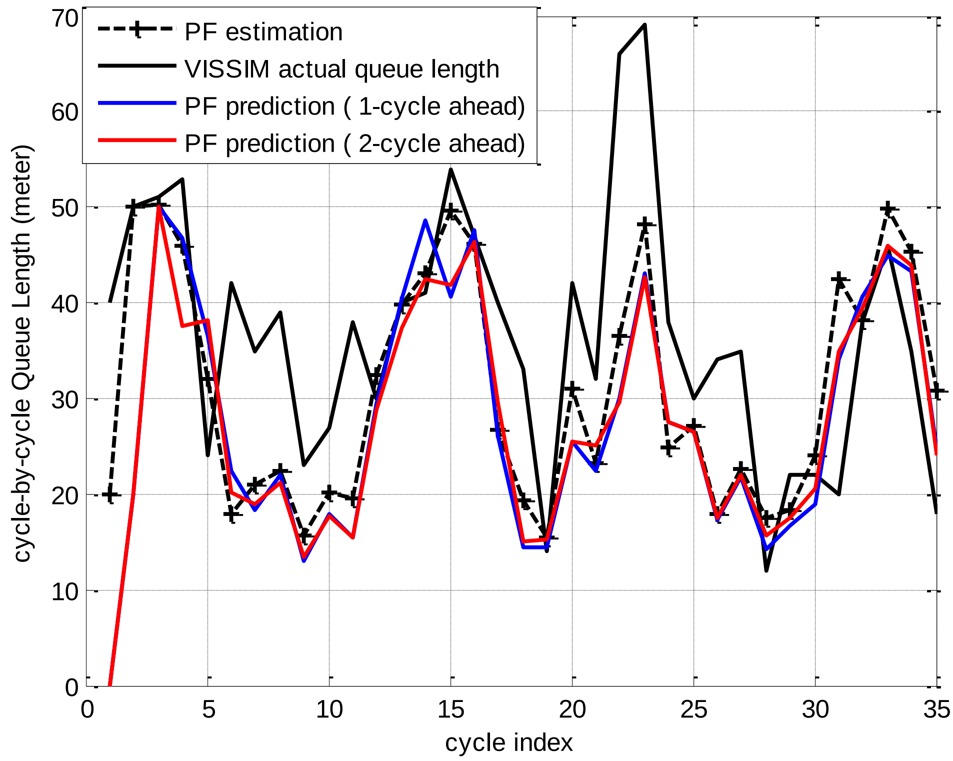

Figure.5.1 Queue-length prediction: N=500

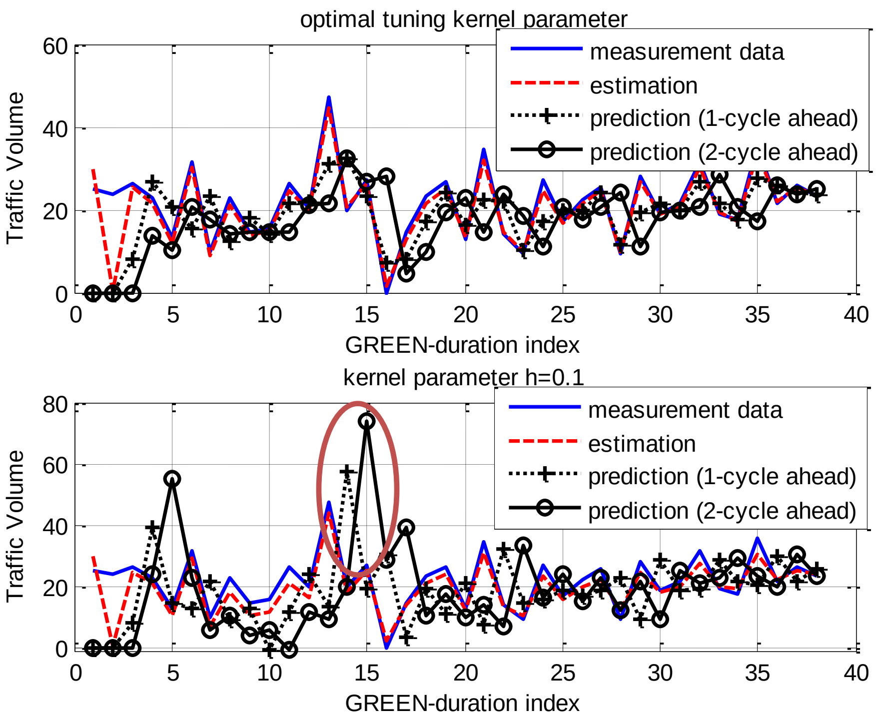

Figure.5.2 Arrival flow estimation and prediction (upper) with optimal tuning h and (lower) with h=0.1

Table 5.4 RMS performance measure comparison

Type\N	500		5000		20000		Average
	Pred-1	Pred-2	Pred-1	Pred-2	Pred-1	Pred-2	
VISSIM	296.5	310.5	252.3	253.3	239.6	244.5	890.25

Selecting optimal parameters, like choosing the appropriate number *N* of particles, or the optimal tuning of the kernel parameter are important issues in this joint state-parameter estimation, given that the algorithm is computationally much more demanding than for state estimation only. The optimal compromise between performance and computational load for online applications is a topic for further research. Figure.5.2 compares the PF performance as a predictor, when using kernel parameter h=0.1 as suggested by Liu in [20] (lower part) versus the better prediction results using the optimal smoothing parameter, using the approach explained in section 5.3

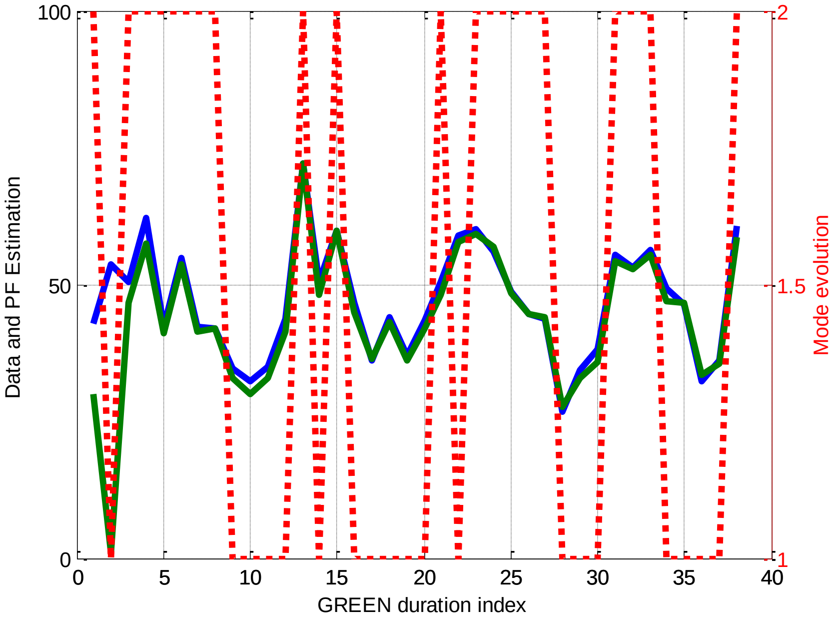

Figure.5.3 Mode evolution of the departure flow

In order to obtain a good prediction it is necessary to have an accurate estimation, and an accurate model describing the future evolution of the predicted variable. Fig.5.3 shows the evolution of mode changes of the departure flow data by choosing *scale* parameter $\hbar$ =0.5 of

Gamma distribution $\Gamma(\delta_{ip} + n_{ip}, \hbar)$ (see step.1). This Gamma distribution is used to generate samples from the Dirichlet distribution by equation (5.9). Implicitly, the online PF with optimal tuning kernel smoothing is able to estimate the parameters of the AR process as well as transition probabilities Π, where we found π_{11}=0.49 and π_{22}=0.58. These results are consistent to the results in Fig.5.3 that the number of jump to another mode is almost equivalent to the number of time the mode remains unchanged, which means that the mode change frequently but with almost single stationary rate. This a consequence of the setting a traffic flow data with a single constant flow rate P=576 veh/hour in VISSIM. While our algorithm detects most of the time the most likely mode it is clear that further improvement is needed for accurate mode estimation.

The experiment uses the MATLAB platform for implementing OTPF procedure in table 5.3 in one single loop iteration. The execution of one loop of the algorithm (i.e. updating from t_k to t_{k+1}) takes less than 1 second meaning that OTPF based joint state-parameter estimation can be implemented in real time.

In the next chapter in the context of implementation of state and parameter estimation as part of feedback control, we do simulation cases with generating multi mean traffic flow rate and it looks that this online estimation works properly.

5.5 Conclusions

This chapter has presented an online technique for joint parameter and state estimation for a stochastic hybrid model of the queue-length dynamics and its application to queue-length estimation and prediction at signalized intersections in urban traffic networks. The key idea of this method is to look at the mode as an unknown system parameter. The system is assumed to follow the dynamics of this most likely mode.

The main novel contributions of this chapter are:

(a) developing the OTPF along with an optimal kernel smoothing method via online minimization of a Kullback-Leibler distance, for estimating the AR parameters; (b) using a Dirichlet distribution to generate the PF samples that allow reliable estimates of the transition probability matrix (TPM) of SHM.

The proposed method is validated in terms of the prediction accuracy for traffic flow and queue length by considering a case study using synthetic data of queue-length generated by a VISSIM traffic micro-simulator. This validation shows that the proposed online joint state-parameter estimation method provides satisfactory queue-length estimation and prediction, and correctly captures the random variation of the traffic flow. The prediction results of traffic flows and queue-length and their computation times shows that this proposed technique can be used to develop good anticipating traffic controllers and this will be discussed in Chapter 6.

6

Adaptive Stochastic Predictive Control with Chance Constraint

Stochastic uncertainties are ubiquitous in complex system (such as the urban traffic networks studied in this thesis) and can lead to unacceptable variability of system outputs causing significant degradation of closed loop performance. A stochastic system is defined as a dynamical system model explicitly representing the source of uncertainties, both process uncertainty and sensing uncertainties. The presence of these uncertainties means that the exact system state is never known exactly, and that predictions of the effects of control action on future behavior are always uncertain. Robust control for linear system [55], linear parameter varying system [56], switched linear system [57] have been proposed as a tool for taking these system uncertainties into account considering worst case bounds on the system state, formulated based upon bounds in the disturbances.

But in applications that can tolerate some failures the robust control solution tends to be too conservative if the worst case realizations in the uncertainty set have a small probability of occurrence. In stochastic control approaches, uncertainties are described by probability distributions (instead of considering every possible occurrence in some bounded sets as being possible). This requires of course that one can obtain the relevant probability distributions without excessive computational cost. In this thesis we use the particles generated by PF method of chapter 3 to represent these distributions. Such a stochastic control not only alleviates the conservatism of worst-case control, but also enables tuning robustness against performance by allowing pre-specified levels of risk during operation. The trade-off between control performance and robustness is achieved using chance (or probabilistic) constraints, which ensure the satisfaction of constraints with a desired probability level.

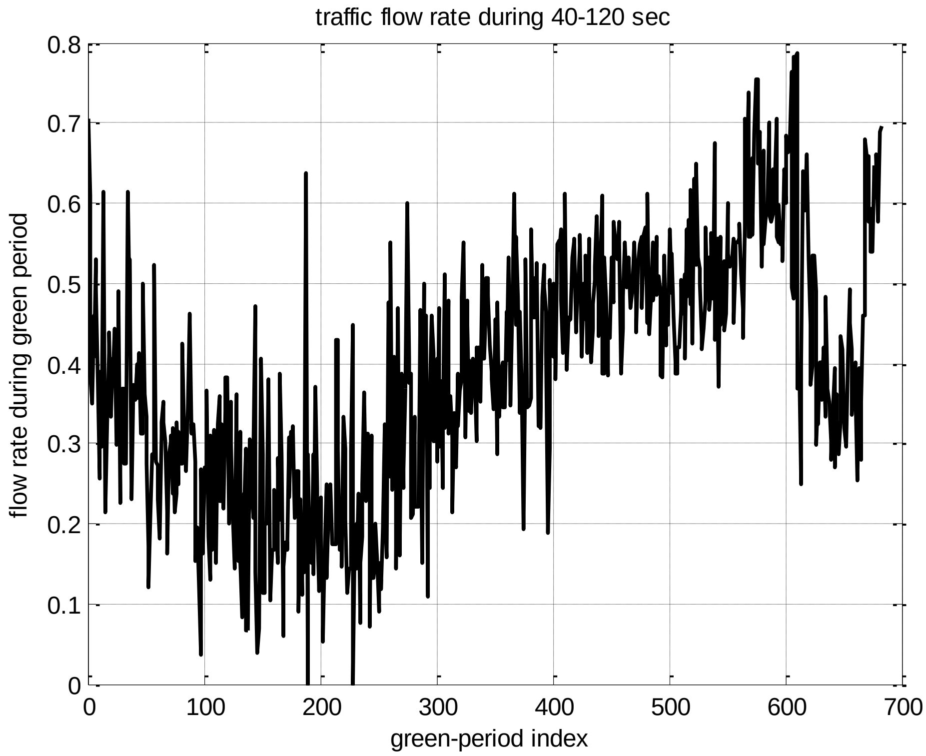

Figure.6.1. Short update interval (green~ 40-120sec)

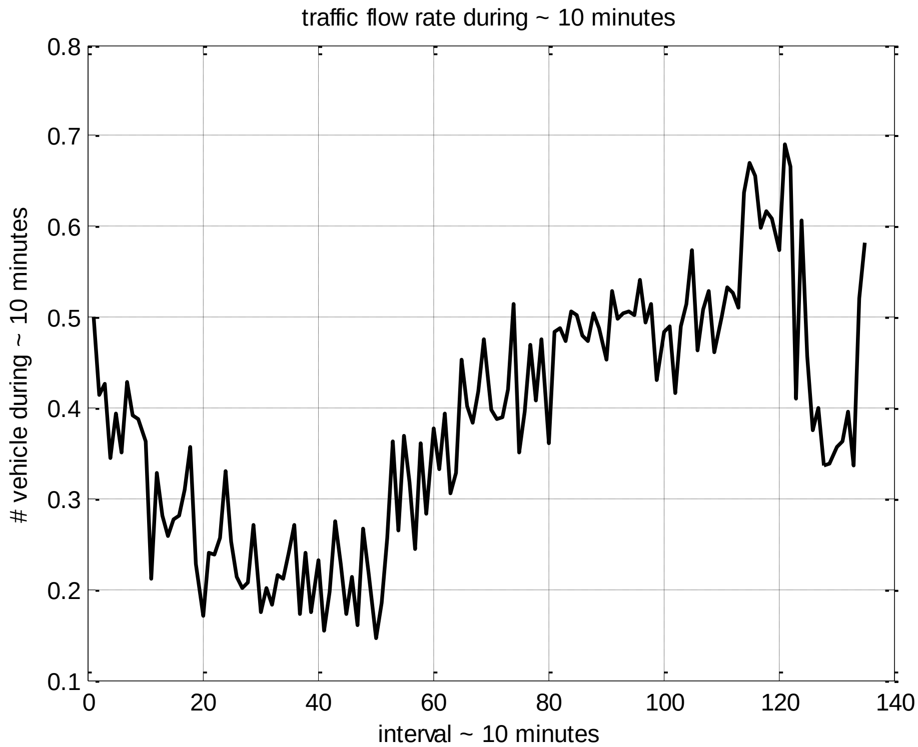

Figure.6.2. Long update interval (10 minute)

A major challenge in stochastic control is the design of chance constrained optimal control law in each sample time step *k*. This implies that we can calculate an estimate of the current state using past observations, and the find the value of u_k that minimizes a cost, while ensuring that certain constraints are satisfied with a given probability. This requires the solution at each time k of non-convex stochastic program, that is computationally intractable, except for very special cases. In order to obtain a tractable solution, various sample-based approximation approaches have been considered. They share the significant advantage of coping with generic probability distributions, as long as a sufficient number of random samples can be obtained. Blackmore in [58] uses a finite number of particles to approximates the distribution of the system state of the hybrid model and Blackmore show that the approximate problem can be solved using efficient mixed-integer linear programming techniques.

The main contribution of this chapter is to develop and to demonstrate that combining the joint state-parameter estimation technique of chapter 5 with, adaptive stochastic control can be a good candidate to avoid queue spillback in the critical intersection . In the perspective of the paper of Daganzo [64] that a network's density never be allowed to approach the critical value, the control that we propose is a potential technique to decrease the likelihood of gridlock on the network.

6.1 Uncertainty in traffic flow

To use the stochastic control methods developed in this thesis, a description of the uncertainty of the system is required. Often linear system models with Gaussian noise are used, leading to simple linear calculations. In our specific urban traffic applications (considering also queue-length dynamic at signalized intersection), the model is highly nonlinear and Gaussian assumptions do not hold for the probability distributions of the states.

The mathematical model of traffic flow rate in the successive intersections along the urban link must reflect the random variability due to random perturbations as well as the time of day fluctuations. In order to properly control and coordinate the traffic lights along this urban link it is crucial to select value of the time update interval that is small enough to allow reaction to significant perturbations, but at the same time long enough to be computationally efficient and to avoid reacting to short term random perturbations that have little effect on the overall system performance. Fig.6.1 and Fig.6.2 show the significant differences between short and long update interval. Fig.6.1 shows the evolution of the arrival flow at a signalized intersection, aggregating vehicles arriving in a short update interval, corresponding to one cycle of traffic light (~ 40-120 sec). Fig.6.2 on the other hand shows the same traffic flow with a longer update interval (10 minute). The results confirms that the choice of the update interval is an important factor to define the probability distribution of traffic flow and queue-length used for the feedback control design. One need feedback on the time scale of one cycle or one phases is that this is the fastest time scale at which one can control via traffic lights. So basically it is the technology used - in this case traffic lights - that determines the time scale. The reason one need feedback because the variability at any time scale causes perturbations (like spillbacks) that are so large that cause global performance deterioration. This in turn determines the prediction horizon that should be taken into account for a real-time model-prediction-based feedback controller of the traffic lights in an urban network. This prediction horizon should be a multiple of the time granularity used for the flow rate measurements, so the prediction horizon is one of important topic related to the choice of the time update step.

6.2 Identification

In this thesis, the mode of the traffic operation (e.g., free flowing, congested or faulty) is evaluated by the real-time traffic flow information gathered from detectors that were placed upstream and downstream to determine the queue-length. Traffic operations evaluation methods can be classified into two types: data mining methods and traffic flow parameters estimation based on advanced the identification technique.

In the Chapters 4 and 5, we have developed quantitative parameter identification techniques, developing both offline and online algorithms. These identification algorithms can be executed jointly with state estimation, and must be incorporated in the feedback control loop in order to improve the performance and robustness of the closed loop system. The proposed controllers thus achieve the automatic adaptation of the controllers tuning to changing model parameters.

In this chapter, we only examine the online identification approach to simultaneously performing parameter estimation and online traffic flow and queue length estimation and prediction as discussed in Chapter.5 and integrate it with stochastic control.

6.3 Adaptive Stochastic Control

The schematic diagram of adaptive stochastic control for traffic signal control of a signalized intersection is shown in Fig.6.3. Refer to Fig.2.5.b in Chapter 2, in lane L_1, we collect a set of data about the traffic flow rate of arrival and departure flows and about the signal timing sequence. We simplify the problem formulation by assuming that the signal timing sequence consists of two phases, red and green only, ignoring the yellow phases and possible all red phases, as shown in Fig.2.5.a.

In this chapter, the data collection is used to identify the parameters using the online technique, then we put the estimated values of the parameters into the jump Markov model based traffic flow equations in order to estimate the current state of the traffic flow rate. These estimates can then be used estimate the current queue length and to predict the evolution of the queue length over the prediction horizon using sequential Monte-Carlo method, allowing comparison of the performance for different possible choices of the control values (cycle time, red/green fraction). Sequential Monte Carlo updates are used since the states estimated at the current time step form the basis of the estimation at the next time step. It can be done by generating a number of scenarios, corresponding to different choices of the control values, via a Monte Carlo simulation of the future behavior. Optimization of the performance among these possible scenarios allows selection of a good choice of control values, with small queueing delays on the average and satisfaction of the probabilistic constraints, to be defined below.

We propose a new control algorithm, probabilistically constrained predictive control. The performance of traffic light are defined by the objective function with constraints on inputs and states should be satisfied in the presence of uncertainties. The propagation of probabilistic parameter uncertainties and exogenous disturbances through the system model and the reformulation of probabilistic constraints to computationally tractable expressions are key issues in stochastic control for real application. In this thesis, we use the convex bounding method developed by Nemirovski [7] which draws samples for the uncertain parameters and bounds the chance constraints with a convex function and then uses the samples to evaluate the convex bound. We use the stochastic hybrid model as explained in Chapter 2 in combination with approximate convex optimization in order to determine the optimal green split allocation for each cycle.

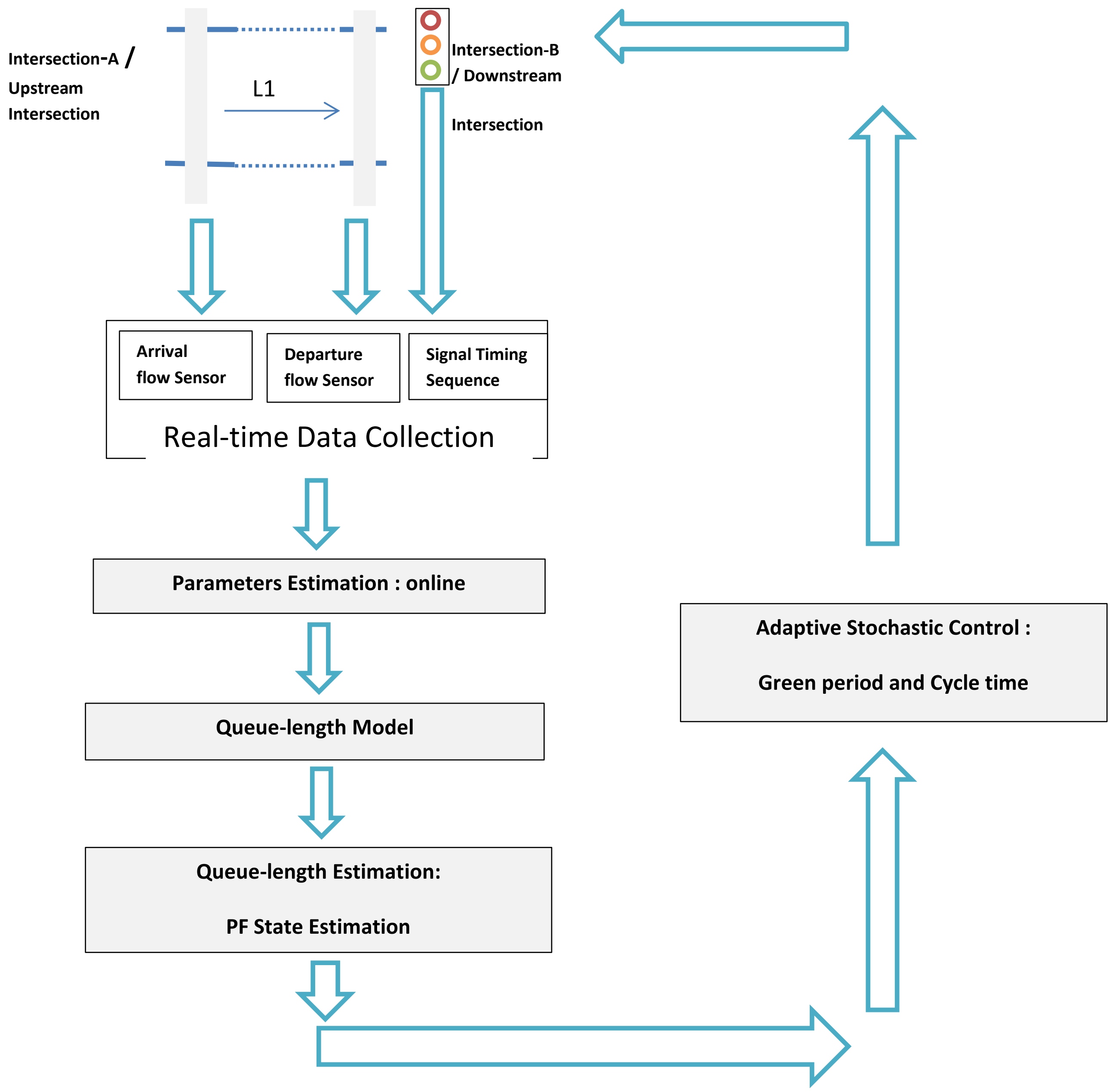

Figure.6.3 Schematic diagram of Adaptive Stochastic Control

6.4 Adaptive Stochastic Predictive Control with Chance Constraint: a Critical Intersection Case

6.4.1 Introduction

As described in chapter 1 that the traditional UTC intend to minimize the traffic flow through the whole network, where the objective function can be represented as the total average delay time experienced by the vehicles in all queues. However, Gayah, in his paper[62], showed that in an extremely congested network adaptive traffic signals might have little to no effect on the network due to downstream congestion and queue spillbacks. Hence, under this traffic conditions, other strategies should be used to mitigate the instability, such as perimeter control, which basically use the unimodal relation between average flow and average density which has been come to be known as the MFD, please refer to Chapter 1 to know the detail.

In this thesis, we follow the idea of Daganzo that the traffic control is developed by integrating an identification and estimation technique approach into a control policy of long queues prior to spillback occurrence to reduce the risk of spillovers through a chance constraint based feedback strategy. In this proposed controller, the risk of spillbacks is ensured by using chance constraint in order to keep the queue length less than threshold to avoid spillback. Its unique feature is that the resulting solution ensures a predefined probability of satisfying the constraints. The solution will lead to an expected optimal value of the cost function by searching for the decision in a feasible region to hold a give threshold level, denoted $0<(1-\delta)<1$. Since δ can be defined by the user, it is possible to select different levels and make a compromise between the cost function value and risk of constraint violation.

In this chapter, we focus on the critical intersection where the traffic lights are set so as to maximize the flow of vehicles along the critical congested link; typically they carry the heavy traffic load, i.e. the traffic flow rates along the roads connecting the adjacent intersections are larger than along the cross roads (as minor road). We will discuss this idea by referring Fig.6.4.

If one considers a critically congested link L_1 in Fig.6.4 to regulate its accumulation to the uncongested state, we need to manipulate the outflow of the critically congested link by applying an efficient control strategy in intersection-I.

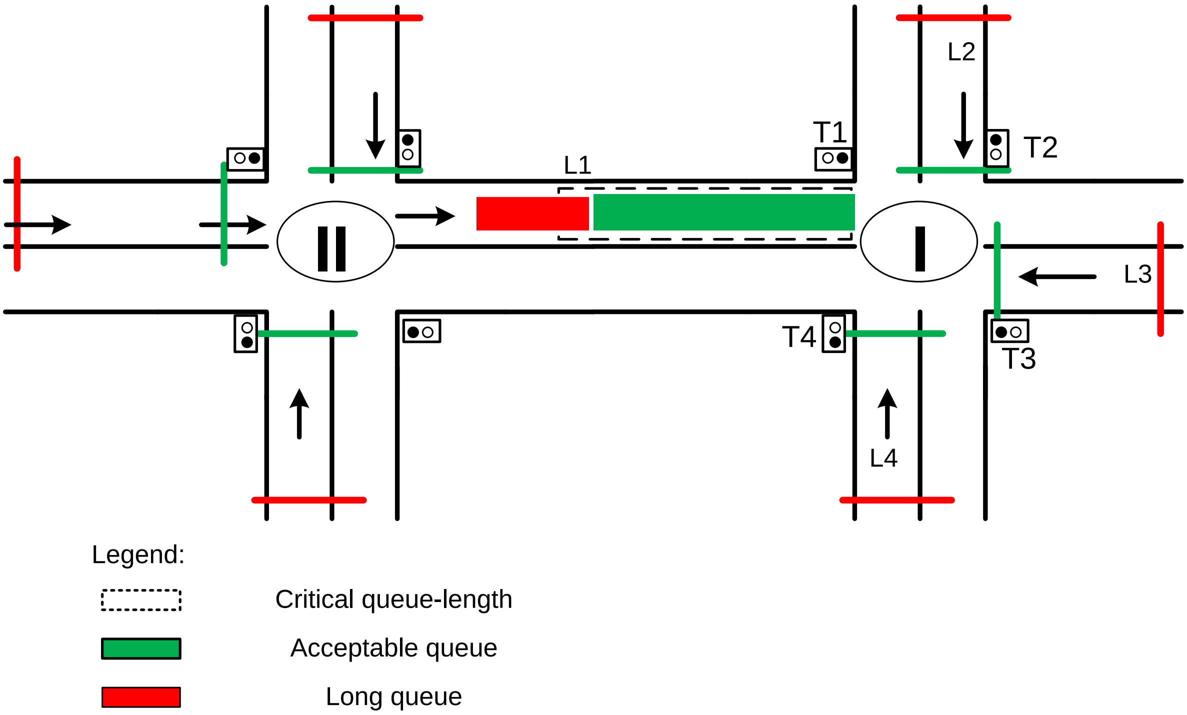

Figure.6.4 Schematic Example of Critical Intersection Case

Thus, the general idea of a good control strategy is to :

(a) increase the green duration of the movement L_1 at the exit intersection (increase the outflow of intersection I) and

(b) to decrease the corresponding green duration at the downstream intersection (intersection II) in order to decrease the inflow at link L_1).

In urban networks with signalized intersections, when demand exceeds capacity queues fail to clear during the allocated green times creating oversaturated traffic conditions and spillbacks occur when growing queues at the downstream link block the arrivals from the upstream link such that vehicle queues cannot discharge at capacity, although the signal phase is green. Spillbacks may also occur when left turning vehicles fill up the available storage length and block the through movements.

In this thesis, for the time being, we merely consider the remedy (a) to increase the outflow of intersection I. In chapter 5 we introduced joint parameter-state estimation approach that detects in real-time which link currently has a long queue larger than critical queue length q_{cr} along one direction of the intersection. We apply this estimator of chapter 5 to several intersections, and only apply the control algorithm to those intersections that are "risky" in the sense that without control the chance constraints are not satisfied.

The following chance constraints must be satisfied in a probabilistic manner in the presence of uncertainties. These chance constraints express the condition that, for the selected control values, over the prediction horizon, the number of vehicles on road L1 remains below the threshold q_{cr} with a probability larger than or equal to the user-defined level of probability 1-δ of holding the constraints will be ensured:

$$p[q_{k+p}^{L1} \leq q_{cr}^{L1}] \geq 1-\delta \tag{6.1}$$

where p=0,1,…,N-1 and N indicates the N-step horizon and set N=3.

As said before that it is possible to select different levels δ and make a compromise between the cost function value and risk of constraint violation. Of course a high confidence threshold to ensure the constraint is always preferred. The knowledge of a maximum value $(1-\delta)^{max}$ is crucial; if a value greater than $(1-\delta)^{max}$ is chosen, the feasible region will be empty.

Remark: In this chapter to simplify the notation, we use *k* to denotes the sampling times t_k.

If this intersection-I indeed constitute a critical intersection then, proper adjustment of the signal timing in that intersection holds the potential to improve traffic condition in the critical link. The signal timing strategy is applied by integrating the state-parameter estimation and adaptive stochastic predictive control with chance constraint (relating to the probability of the queue length being less than the threshold q_{cr}, refer to Fig.6.4).

In this thesis, we tightly integrate stochastic control with joint state-parameter estimation leading to adaptive stochastic predictive control. The basic difference between the convex and non-convex optimization is that in (i) convex optimization there can be only one optimal solution, which is globally optimal or we might prove that there is no feasible solution to the problem, while in (ii) nonconvex optimization may have multiple locally optimal points and it can take a lot of time to identify whether the problem has no solution or if the solution is global. Hence, the efficiency in time of the convex optimization problem is much better.

By integrating the remedy (a) and (b) previously mentioned (which is equivalent to the use of MFD for controller design as proposed by Daganzo [3] and Aboudolas [1]), it will in the future become possible to implement a locally decentralized signal control strategy while ensuring at the same time the global coherence of these strategies along the network. In this manner, we seek to improve traffic on a critical link at a low cost by acting merely on critical intersections, as opposed for instance to network-wide optimization process. . In this thesis we consider as a first step chance constraint local controllers, which minimize the risk of global interactions.

6.4.2 Problem Formulation

We consider the isolated signalized intersection as a switching system with fixed-sequences consists of two phases (A and B) as shown in Fig.2.5 in chapter 2.

We follow the SHM complete model as formulated in equation (5.1) in order to predict future evolution of the state under the influence of selected control values. We proposes to select, as a function of the available observations as summarized by the obtained estimates of the current state and parameter values, a value for the control variable $T_{g,k}$ at sampling instants: k ∈ $[t_{2k},t_{2k+1}]$ so as to minimize an objective function, like (6.2) below, while constraints on the inputs and states should be satisfied in the presence of uncertainties, with chance constraint in equation (6.1). We consider the case when the criterion cost function *J* is the average queue length, positively weighted sum of queue lengths as shown in equation (6.2). Hence we introduce the objective function as follows:

$$J = \sum_{p=1}^{3} [wg_1 \; E[q_{1,k+p}] + wg_2 \; E[q_{2,k+p}] \qquad (6.2)$$

where wg_1 and wg_2 are weighting factor for L_1 and L_2. The reason for selecting a time horizon of 3 steps ahead is because of the sufficient results of traffic flow prediction over this time horizon and also the fact for automatic adaptation we need a quick anticipation and choosing p=3 is reasonable for urban traffic network in the sense that the distance in time between intersections is shorter than the three cycle lengths.

The novelty in the control design proposed in this chapter is to guarantee moreover the security of the critical link, since avoiding spillbacks along the critical link (L_1) is an overriding condition that takes precedence over the performance along the other flow directions(L_2). This cost function (6.2) together with chance constraint in equation (6.1) defines a controller that ensures avoiding the spillback.

This optimization calculates the green signal times for N-step horizon $T_{g,k+p}$, at the beginning of the k-th cycle, but only apply the first value $T_{g,k}$, where p=0,1,2.

Remark: From the perspective of traffic engineering, this cost function can also be seen as a level-of-service (LOS) indicator. LOS is defined in the HCM (Highway Capacity Manual) [95] as a qualitative measure that reflects user perception of quality of service. Delay has been chosen as the reasonable measure for determining signalized intersection LOS and delay is proportional to the integral (or sum in the discrete time model considered here) of the queue lengths justifying the performance measure or LOS under consideration here.

6.4.2.1 Uncertainties

The queue length dynamics at traffic movement L_i strongly depends on the arrival flow and departure flows. Both traffic flow are subject to the uncertainties of the evolution of the traffic modes, and on the stochastic parameters θ_α of AR process.

6.4.2.2 Objective Function and Constraints

Summarizing the above considerations we now formulate the problem of this section formally. This section considers the adaptive stochastic predictive control of the SHM as shown in equation (2.6) – (2.10) while constraints (6.1) on the inputs and states should be satisfied in the presence of uncertainties, with the probabilistic limits calculated (or estimated via scenarios) according to the probability distribution of the uncertainties as specified above.

Note that in this SHM we define:

State variables x are the queue-lengths q_{tk}, modes s_{tk}, and arrival/departure flows in each competing movements (L_1/L_3 or L_2/L_4) as shown in equation (5.1); control variable u is green fraction $T_{g,k}$ whereas measurement variables y are number of passing vehicles per green (red) duration at arrival/ departure sensors location. State variables x are predicted by stochastic hybrid model (2.6-2.10) through joint state-parameter estimation.

Stochastic optimal control problem (6.3) is implemented in this thesis in a N-receding horizon fashion, with N=3. The closed loop control that is applied to the system is defined by the optimal solution switching time sequence t_k, $t_{2(k+1)}$, $t_{3(k+1)}$… (refer to Fig.2.5.a). Under uncertainties in the initial conditions, and uncertainty about the systems parameters, the solution trajectories of stochastic hybrid (6.4 -6.9) may violate the constraints (6.10-6.13). In this thesis, inputs $T_{g,k+p}$ are designed to satisfy (6.10-6.12) such that the chance constraints (6.13) is fulfilled in a probabilistic manner in the presence of these uncertainties. The value of probability distribution of chance constraint can be computed using a convex bounding method which will be discussed in detail in the next section.

In the application of a critical intersection with link L_1 and L_2, we consider the following receding horizon (or model predictive) control problem for a given p number of cycles horizon (not necessarily of equal lengths) and starting time t_0, compute an optimal switching time sequence t_k, $t_{2(k+1)}$, $t_{3(k+1)}$… (refer to Fig.2.7.a) that minimizes the performance criterion *J*, defined in (6.2),can be formulated as follows:

(P.1) $$\min J \tag{6.3}$$

Subject to: $$q_{1,2k+(p+1)} = \max\ [q_{1,2k+p} + (\lambda_{1,2k+p} - \mu_{1,2k+p})T_{g,k}\ ,0] \tag{6.4}$$

$$q_{1,2k+(p+2)} = q_{1,2k+(p+1)} + \lambda_{1,2k+(p+1)}\ T_{r,k} \tag{6.5}$$

$$q_{2,2k+(p+1)} = q_{2,2k+p} + \lambda_{2,2k+p} T_{g,k} \tag{6.6}$$

$$q_{2,2k+(p+2)} = \max\ [q_{2,2k+(p+1)} + (\lambda_{2,2k+(p+1)} - \mu_{2,2k+(p+1)})T_{r,k}\ ,0] \tag{6.7}$$

$$\lambda_{1,2k+p} = \beta_{1,\lambda,2k+p-1}(s_{2k+p-1}) + \gamma_{1,\lambda,2k+p-1}(s_{2k+p-1}) \lambda_{1,2k+p-1} + w_{1,\lambda,2k+p-1} \tag{6.8.a}$$

$$\mu_{1,2k+p} = \beta_{1,\mu,2k+p-1}(s_{2k+p-1}) + \gamma_{1,\mu,2k+p-1}(s_{2k+p-1}) \mu_{1,2k+p-1} + w_{1,\mu,2k+p-1} \tag{6.8.b}$$

$$\lambda_{1,2k+p} = \beta_{1,\lambda,2k+p-1}(s_{2k+p-1}) + \gamma_{1,\lambda,2k+p-1}(s_{2k+p-1}) \lambda_{1,2k+p-1} + w_{1,\lambda,2k+p-1} \tag{6.8.c}$$

$$\lambda_{2,2k+p} = \beta_{2,\lambda,sk+p-1}(s_{2k+p-1}) + \gamma_{2,\lambda,2k+p-1}(s_{2k+p-1}) \lambda_{2,2k+p-1} + w_{2,\lambda,2k+p-1} \tag{6.9.a}$$

$$\lambda_{2,\lambda,2k+p+1} = \beta_{2,\lambda,2k+p}(s_{2k+p}) + \gamma_{2,\lambda,2k+p}(s_{2k+p})\,\lambda_{2,2k+p} + w_{2,\lambda,2k+p} \tag{6.9.b}$$

$$\mu_{2,2k+p+1} = \beta_{2,\mu,2k+p}(s_{2k+p}) + \gamma_{2,\mu,2k+p}(s_{2k+p})\,\mu_{2,2k+p} + w_{2,\mu,2k+p} \tag{6.9.c}$$

for p=0,1,…,N-1 and N indicates the N-step horizon and set N=3;

Remember that the evolution of the modes s_k according to the transition probability matrix (TPM) Π_{ij}= Prob$(s_{t_k} = j \,|\, s_{t_{k-1}} = i, s_{t_{k-2}}, s_{t_{k-3}}, \ldots)$.

Additionally, the following constraints should must always be satisfied:

$$\mathbf{T}_{g,k+p} + \mathbf{T}_{r,k+p} = C \tag{6.10}$$

$$T_{g,\min} \leq \mathbf{T}_{g,k+p} \leq T_{g,\max} \tag{6.11}$$

$$C_{,\min} \leq C \leq C_{\max} \tag{6.12}$$

while the chance constraint limiting the queue-length at movement L_1must be satisfied with probability larger than 1-δ:

$$p[q_{1,k+p} \leq q_{1,cr}] \geq 1 - \delta \tag{6.13}$$

where :

$q_{m,k+p}$: predicted queue length using equation (2.6) - (2.9) in each movement *m* (L_1/L_3 or L_2/L_4) at time instant *k+p* on the corresponding traffic signal sequence at time *t* where *p*=0,1,…,N-1 and N indicates the N-step horizon and set N=3.

$\lambda_{m,2k+p}$, $\lambda_{m,2k+(p+1)}$: arrival flows are detected by arrival sensors as indicated by red-lines in Figure 2.5.b, for each movement *m* (L_1/L_3 or L_2/L_4) during the green period T_g (resp. the red period T_r) on the corresponding traffic signal sequence of the movement *m* as shown in Fig.2.5.a.

$\mu_{m,2k+p}$, $\mu_{m,2k+p+1}$: departure flows are detected by departure sensors as indicated by green-lines in Figure 2.5.b in each movement *m* (L_1/L_3 or L_2/L_4) during the green period T_g (resp. red period T_r) on the corresponding traffic signal sequence of the movement *m* as shown in Fig.2.5.a.

Remember that for p>0 or prediction: arrival flow and departure flow are predicted using state equation (2.10.b), by using the parameters of AR $\theta = \{\beta_1, \gamma_1, \sigma_1^2, \beta_2, \gamma_2, \sigma_2^2, \Pi\}$ at index k. It means that over the prediction horizon, we use the previous value of θ at time k.

$\beta_{m,\lambda,2k+p}(s_{2k+p})$, $\beta_{m,\lambda,2k+p+1}(s_{2k+p+1})$ and $\gamma_{m,\lambda,2k+p}(s_{2k+p})$, $\gamma_{m,\lambda,2k+p+1}(s_{2k+p+1})$: estimated parameters of JMM model (2.10) in modes *s* at time instant 2*k*+*p* for arrival flow λ (departure flow μ) in each movement *m* during the green period T_g (resp. red period T_r).

$w_{m,\lambda,2k+p,}$, $w_{m,\lambda,2k+p+1,}$: an i.i.d. sequence of random variables, E [$w_{m,2k+p}$] = 0 and

$$\mathrm{Var}[\, w_{m,2k+p} \,] = \sigma^2(s_{tk})$$

δ ∈ (0,1] is user-specified parameter chosen according to process requirements: δ =0 corresponds to hard constraints that should hold at all times for all uncertainty realizations; δ < 1 allows for constraint violation with probability δ in order to trade-off control performance with robustness.

C, C_{min}, and C_{max} represent cycle time refer to Fig.2.5.a, lower and upper bound for the cycle time respectively. The term $\mathbf{T_{g,k+p}}$ expresses the green signal time variable of lane L_1 over the N-step horizon. It can be written as $\mathbf{T_{g,k+p}}$= [$T_{g,k}$ $T_{g,k+1}$ … $T_{g,k+p}$]^T. This optimization calculates the green signal times for N-step horizon $\mathbf{T_g}$(k+p), at the beginning of the k-th cycle (or t_{2k} refer to Fig.2.5a), but only apply the first value $T_{g,k}$ and the next optimization loop is started at the (k+1)-th cycle (or $t_{2(k+1)}$ refer to Fig.2.5.a). For each step of the N-step horizon, the summation of green time and red time is equal to the cycle time, which may vary between a lower and upper value or may constant. Each green time value also has a minimum $T_{g,\,min}$ and maximum $T_{g,\,max}$ which is fixed over the horizon.

6.4.2.3 Convex bounding method

Because chance constraint is an important issue in the perspective of avoiding spillbacks along the critical link, hence this section elaborate in detail of the convex bounding method , proposed by Nemirovski [7], allowing the probabilistic inequality constraints (6.13) to be transformed into deterministic constraints. The deterministic constraints strongly depend on estimation of queue-length and the resulting problem is a convex program which may decrease the computational complexity.

The most challenging aspect of solving the optimization program (**P.1**) is in evaluating and satisfying the chance constraints in equation (6.13). We focus on using a sampling technique to approximate the chance constraint of the system by using a finite set of particles to represent the probability distribution of the system. In this thesis, sampling methods sample N_s particles at each time-step from the Gaussian noise source and initial state to obtain the sets

$$
\begin{aligned}
&\text{samples} \quad : \quad \{w_0^{1},\ldots, w_{N-1}^{1},\ldots, w_0^{Ns},\ldots, w_{N-1}^{Ns}\},\\
&\text{noise} \quad : \quad \{\eta_0^{1},\ldots, \eta_{N-1}^{1},\ldots, \eta_0^{Ns},\ldots, \eta_{N-1}^{Ns}\}, \qquad (6.14)\\
&\text{state} \quad : \quad \{x_0^{1},\ldots, x_0^{Ns}\}
\end{aligned}
$$

An approximation of the distribution of the system state, measurement output and control input can then be calculated using this set of samples. Please see equation (5.2.a) and (5.2.b) and we rewrite here to clarify and again we use *k* denotes the sampling times t_k.:

$$x_k^j = f_s(x_{k-1}^j, \theta, \eta_k^j)$$

$$u_k^j = \phi(y_0^j, \cdots, y_k^j)$$

where $\phi(.)$ is mapping from observation to control value, for all $j=1,\ldots,N_s$. Let $\mathbf{X}^j$ be defined as $\mathbf{X}^j=[x_0^j,\ldots, x_N^j]$ and similarly $\mathbf{U}^j=[u_0^j,\ldots, u_N^j]$

There are three different sampling approaches that are commonly employed. The first uses mixed integer programming to approximate the chance constraints by counting the number of constraint violations [58]. The second method enforces the constraints for all of the samples and determines the probability that the chance constraints will be satisfied [77]. The third method bounds the chance constraints with a convex function and then uses the samples to evaluate the convex bound [7]. We apply the third method to a stochastic control with state-parameter estimation.

Typically chance constraints deal with the satisfaction of each constraint separately, requiring each to be satisfied with probability 1- δ_i

$$p[q_{k+p}^{L1} \le q_{cr}^{L1}] \ge 1-\delta \quad \Leftrightarrow \quad p\big(\psi(\mathrm{X},U) \le 0\big) \ge 1-\delta$$

where each ψ is a scalar function and in the case of queue length: $\psi(X,U) = \hat{q}_{k+p} - q_{cr}$.

This thesis use the convex bounding method [7] by approximating the probability distribution by number of samples. This Monte Carlo approach simplifies the evaluation of the chance constraints, which really are upper bounds on the value taken by a probability integral. The Monte Carlo method is an efficient way of calculating this integral. The following subsections explain in detail how we use sampling to handle these chance constraints by using a finite set of particles to represent the probability distribution of the system; this converts the stochastic control problem into a deterministic one.

Consider a single generic individual chance constraint of the form

$$p\big(\psi(\mathrm{X},U) \le 0\,\big) \ge 1-\delta$$

which is equivalent to:

$$p\big(\psi(\mathrm{X},U) > 0\,\big) \le \delta \tag{6.15}$$

The probability distribution in equation (6.15) can be calculated via Monte Carlo methods is of course dependent on the quality of the Monte Carlo approximation.

$$\begin{aligned} p\big(\psi(\mathrm{X},U) > 0\,\big) \;&=\; E\big[\mathbf{1}(\psi(X,U))\big] \\ &\approx\; \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(\psi(X,U) \end{aligned}$$

where **1(.)** is the indicator function defined in equation (2.3).

This indicator function **1**(z) is a nonconvex function, this greatly complicates the evaluation of the chance constraints (6.15) and solution of the optimization problem (6.3). However, by bounding the indicator function by a convex function the optimization program simplifies to a convex program.

Suppose a nonnegative, non-decreasing, convex function $\overline{\psi}_i : R \to R$ can be found such that for any $\alpha,\ \overline{\psi}(z/\alpha) \ge 1(z)$ for all z the following holds:

$$E\,[\,\overline{\psi}\,(\psi(\mathrm{X},U)/\alpha]\ \ge E[\mathbf{1}(\psi(X,U))] = p\,(\psi(X,U) > 0)$$

Consequently, if the following convex constraint is satisfied:

$$E\,[\,\overline{\psi}\,(\psi(\mathrm{X},U)/\alpha]\ \le \delta \tag{6.16}$$

then the original chance constraint in equation (6.15) is guaranteed to hold a fortiori.

The constraint in equation (6.16) holds for any α, but different choices of α may lead to a better approximation of the original chance constraint in equation (6.15). Equation (6.16) can be written as by following the detail in [7], reducing the conservativeness of the method

$$E[\alpha\overline{\psi}(\psi(\mathrm{X},U)/\alpha] \leq \alpha\delta \tag{6.17}$$

The left-hand side of equation (6.17) is a perspective function $\alpha\overline{\psi}(z/\alpha)$ that is convex in z and α for α >0.

Definition.6.1 [16]:

If f: $\mathbf{R}^{\mathbf{n}}\rightarrow\mathbf{R}$ then the *perspective* of f is the function g: $\mathbf{R}^{\mathbf{n+1}}\rightarrow\mathbf{R}$ defined by

$$g(x,t) = t\ f(x/t)$$

with domains

$$\mathbf{dom}\ g = \{(x,t) \mid x/t \in \mathbf{dom}\ f, t>0\}$$

As shown in [16], the perspective of a convex function is also convex, therefore $\alpha\overline{\psi}(\psi(\mathrm{X},U/\alpha)$ is convex in **X**, U and α for α >0. Consequently, the constraint in equation (6.17) is convex. By deriving the optimal α^* providing the best approximation by solving (6.18) can reduce the conservativeness of the method:

$$\inf_{\alpha>0} E[\alpha\overline{\psi}(\psi(\mathrm{X},U)/\alpha] - \alpha\delta \leq 0 \tag{6.18}$$

Now that the parameter α of the convex constraint used to bound the original chance constraint has been selected the next step is to determine what form of function to use for $\overline{\psi}$. The restrictions on the functions, as stated previously, are that it needs to be a convex function and $\overline{\psi}(z) \geq 1(z)$ for all z. Several candidate for the function are considered in [74]:

a. Markov : $\psi(z) = [1+z]_+$

b. Chebyshev : $\psi(z) = [1+z]_+^2$

c. Traditional Chebyshev : $\psi(z) = (1+z)^2$

d. Chernoff/Bernstein : $\psi(z) = e^z$

where $[.]_+ = \max(.,0)$. Each function places a different penalty on the severity of constraint violation. The best approximation is given by the function that is closest to indicator function.

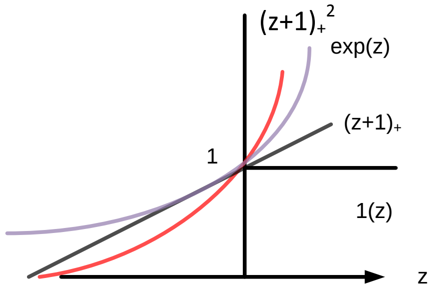

Figure.6.5. Several convex bounds to approximate the indicator function **1**(z)

In this thesis, we use the traditional Chebyshev function as indicated red curve in Fig.6.5. One advantage of using a smooth generating function, such as the traditional Chebyshev, is that we can write explicitly the optimality condition setting the gradient to 0 by evaluating the expression in (6.19); otherwise; one must resort to an uncertainty sampling method which is incurs a higher computational cost.

Using the traditional Chebyshev function, the bound is obtained:

$$E\,[\alpha\,(\psi(\mathrm{X},U)/\alpha+1)^2\,] \le \alpha\delta \tag{6.19}$$

Which aftter expanding terms and simplifying can be written as

$$2\,E[\,\psi(X,U)]+\frac{1}{\alpha}E[\psi(X,U)^2\,]+\alpha(1-\delta)\le 0 \tag{6.20}$$

Minimizing the previous equation over α gives $\alpha^* = \left(\frac{1}{1-\delta}E\left[\psi(X,U)^2\right]\right)^{1/2}$ which finally yields the constraint

$$E[\psi(X,U)]+((1-\delta)E[\psi(X,U)^2)^{1/2} \le 0 \tag{6.21}$$

It is important to note that this approximate conservative constraint only depends upon the first and second moments of the function $\psi(X,U)$ which can be computed from the convex bounding method via the samples allowing the constraints to be transformed into deterministic constraints. Note that the probabilistic inequality constraint depends upon the first and second moments of the function $\psi(X,U)=\hat{q}-q_{cr}$ which it can be computed using state estimation PF as described in Chapter 3, where q_{cr} is critical queue length as noted at Fig.6.4.

In summary, the following problems need to be addressed to efficiently solve **(P.1)**

Problem.1. Propagation of probabilistic uncertainties of state dynamic are propagating through stochastic hybrid state dynamics (6.4)-(6.9). The problem of propagation over the prediction horizon is addressed by using the previous value of θ at time k that can be achieved using online joint state-parameter estimation.

Problem.2. Efficient evaluation of the chance-constraints (6.13). Since the probability distribution of the chance constraints (6.13) may not be a convex function, it is difficult to include them in the optimization program. The convex bounding method uses a suitable conservative, convex approximation for the probability distribution of the chance constraints as defined above. The resulting problem is a convex program which may decrease the computational complexity.

Remark :

- **(a)** We assume that the chance constraint only applies to the queue length on road L_1 as indicated q_1 in equation (6.13), but the optimization for the traffic control law also takes into account the queue length on L_2 (the minor road) through involving the queue length dynamics on L_2 in equation (6.6) and (6.7) . Please see L_1 and L_2 as indicated in Fig.6.4.
- **(b)** in the future extension of the method for controlling traffic lights, if one wants to consider a complete network, then one needs to consider joint chance constraints, and this requires multidimensional integration which makes the sampling approach a lot harder. This joint chance constraint will be valuable for coordinating to multiple intersection through multi agent system approach.

Using stochastic hybrid model Eq.(6.4-6.9) describing the evolution of the variables q_{tk}, T_g, w and n, over the prediction horizon $t_1, t_2, \ldots, t_{2N}$ the value of probability distribution of chance constraint (6.13) can be computed by propagating the uncertainty in q_k, T_g, w and n through a convex bounding method as was mentioned in previous section. Hence, the estimation of probability distribution of the chance constraint is simply an uncertainty propagation problem.

6.5 Simulation Studies

The aim of this experiment described in this section is to demonstrate how one can control efficiently for the case of a critical road link where queues that may spill back upstream of critical intersection I (please see Fig.6.4) are created with fixed time signal, when the inflow grows too large. In the simulation studies below, we proposed a signal control strategy that acts only on the exit intersection along the arterial. We use the adaptive stochastic with chance constraint explained in the previous sections. This chance constraint needs the choice of a threshold value δ to decide whether a link is considered as congested or not, based on a queue-length state-estimation strategy that detects queues that exceed the capacity of the corresponding approach at the downstream intersection.

We consider problem **(P.1)** for the critical intersection on the saturated link, with objective function as shown in (6.2). The simulation studies are based on one signalized intersection with competing links and considering link L_1 as a major road and link L_2 as minor road as shown in Fig.6.4 The parameters of the traffic condition are listed below:

a. Cycle time is fixed to C= 90s for that intersection
b. Synthetic traffic flow data at the inflow nodes of the network with random Gaussian noise are generated with aim to show changing traffic flow condition from one mode to another mode over the cycle index (please note in this chapter we use *k* denotes the sampling times t_k):
    a. arrival flow rate during red period :
        i. $\lambda_{1,2k+1}$ (movement L_1)
            i. with mean 0.4 and variance 0.01 during time index 1-400
            ii. with mean 0.3 and variance 0.01 during time index 401-800
        ii. $\lambda_{2,2k}$ (movement L_2)
            i. with mean 0.3 and variance 0.02 during time index 1-400
            ii. with mean 0.2 and variance 001 during time index 401-800
    b. arrival flow rate during green period:
        i. $\lambda_{1,2k}$ (movement L_1)
            i. with mean 0.3 and variance 0.01 during time index 1-400
            ii. with mean 0.2 and variance 0.02 during time index 401-800
        ii. $\lambda_{2,2k+1}$ (movement L_2)

i. with mean 0.4 and variance 0.02 during time index 1-400

ii. with mean 0.3 and variance 0.01 during time index 401-800

c. departure flow rate during green period:

i. $\mu_{1,2k}$ (movement L_1)

a. with mean 0.8 and variance 0.02 during time 1-800.

ii. $\mu_{2,2k+1}$ (movement L_2)

a. with mean 0.5 and variance 0.02 during time 1-400

b. with mean 0.4 and variance 0.02 during time 401-800

c. Allowable green period is 45 sec $<T_g<$ 70 sec

d. Initial random queue length with mean 5 vehicles and variance (1 veh)^2

e. The probability of the queue length on road L_1 must at all times satisfy the constraint that is below the threshold q_{cr} =15 vehicles, with a probability (1-δ) that is 90 % during the whole simulation. Note that the chance constraint applies only for critical link road L_1.

f. The prediction horizon is selected as N=3. This selection is based on the performance assessment on online joint state and parameter estimation that is good performance in the prediction horizon up to N=3.

Note: Gaussian assumption is not realistic, but still acceptable because the average values of the traffic flows (in this simulation) turn out always to be positive because the standard deviation are small compare to the mean value, which in turn generates positive values of traffic flows.

To solve online optimal control problem **(P.1)**, the coefficients of the AR process and the transition probabilities of the mode process s_k are estimated by joint parameter estimation at each discrete time instant using both new measurements and also decision variable $T_{g,k}$. Please see algorithm-1 in table.6.2

For online implementation, problem **(P.1)** is embedded in a receding horizon algorithm. Adaptive stochastic MPC is implemented in MATLAB and the optimization is based on subroutine *fmincon* the sequential optimization strategy. Note that the algorithm in table 6.1 needs algorithm in table.5.3 for state and parameter estimation.

Table. 6.1 Adaptive stochastic mpc

Algorithm of (Receding horizon implementation of Adaptive Stochastic MPC
Input: 1. Initial time step k 2. Feasible initial control input T_g 3. Initial state $\{x1_k\} = \{q_{1,k}, (\lambda_{1,2k}, s_{2k}), (\lambda_{1,2k+1}, s_{2k+1}), (\mu_{1,2k}, s_{2k})\}$ $\{x2_k\} = \{q_{2,k}, (\lambda_{2,2k}, s_{2k}), (\lambda_{2,2k+1}, s_{2k+1}), (\mu_{2,2k}, s_{2k})\}$ $\{x_k\} = \{x1_k \; x2_k\}$ 4. Uncertainty description of parameter vector $\theta_\alpha(s_k) = \{\beta_\alpha(s_k), \gamma_\alpha(s_k), \sigma^2_\alpha(s_k), \Pi_\alpha\}$ where α is a generic flow (it can be $\lambda_{1,2k}$, $\lambda_{1,2k+1}$, $\mu_{1,2k}$ for movements L_1 and L_2)
At discrete time instant k: 1). Use x_1 and Tg(1) to carry out simulation of the SHM using N samples drawn from optimal-tuning kernel smoothing approach. This step is critical step when the traffic condition change: jump to another mode. 2). Do state estimation and prediction using OTPF with N=3,see table.5.3 in chapter 5 3). Solve the deterministic optimal control problem **(P.1)** to determine optimal control policy. For each optimization iteration, this requires repeated parameter estimation at step 1 and also state prediction/ estimation at step 2, such that the (chance) constraints (6.21) are fulfilled in probabilistic manner in the presence of uncertainties. 4. Apply the first element Tg to the stochastic hybrid system (start from step 1 at each time instant k)

Using the parameters listed above, we simulate the evolution of queue length different control loops viz. adaptive stochastic mpc with chance constraint (*asmpc-cc*), adaptive stochastic mpc with cost function only considering queue-length in L_1 (*asmpc-diff J*) and fixed green duration (*fixed*) T_g=45 secs. Both the *asmpc-cc* and the *asmpc-diff J* use stochastic control algorithm along with joint state-parameter estimation. The *asmpc-diff J* considers all the constraint equation (6.10)-(6.12), but does not include the probabilistic constraint (6.13) and use a different cost function as follows:

$$J = \sum_{p=1}^{3} [wg_1 \, E[q_{1,k+p}]] \tag{6.22}$$

The evolution of queue length on link L_1 (major road) at a signalized intersection for case **(P1)** is shown in Fig.6.6- Fig.6.8:

(a) The *asmpc-cc* fulfills the constraints on the upper limit of queue length in major road L_1 less than q_{cr} =15. However, there are queue lengths more than 15 but with probability of the occurrence less than 10 %. This result fulfill the chance constraint condition as indicated in the condition (e). Despite this control value T_g for the link/movement L_1 is increasing but still it is able to fulfill the constraint on green period to stay in the set of allowable green period (45 sec <T_g< 70 sec) as shown in Fig.6.6-Fig.6.8.

(b) The *asmpc-diff J* has cost function (6.22) that only considering L_1. In this case we give a highest priority to movement in L_1 and the controller generate the maximum green period T_g =70 and this makes the queue length in major road L_1 has a lowest value compared to another controller as shown in Fig.6.6. Consequently the queue length in minor road L_2 reaches highest value as shown in Fig.6.8 with T_g=20 sec.

(c) In the fixed control profile, the result, as expected, shows the queue length in major road L_1 has highest value compared to the others.

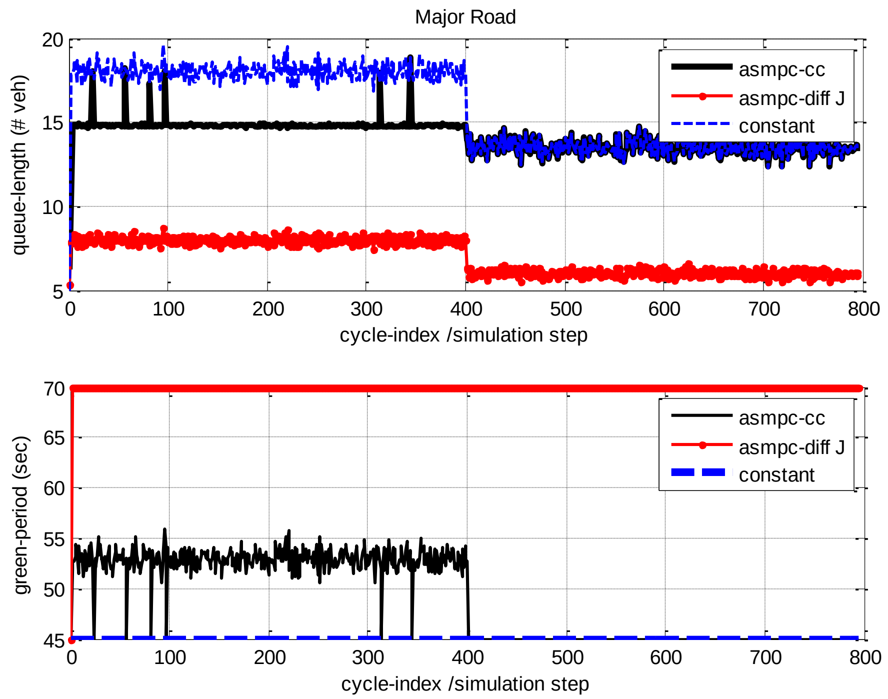

Figure.6.6 Queue-length evolution (upper) and green period adapting (lower) for L_1

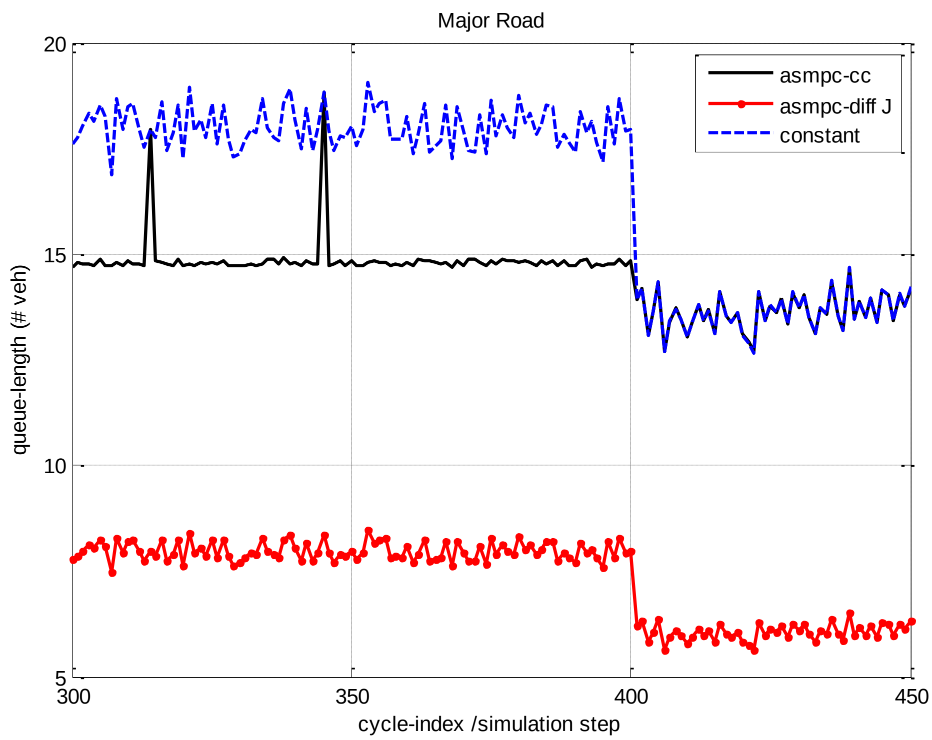

Figure.6.7 Queue-length evolution for L_1 (zoom in between index 300-450)

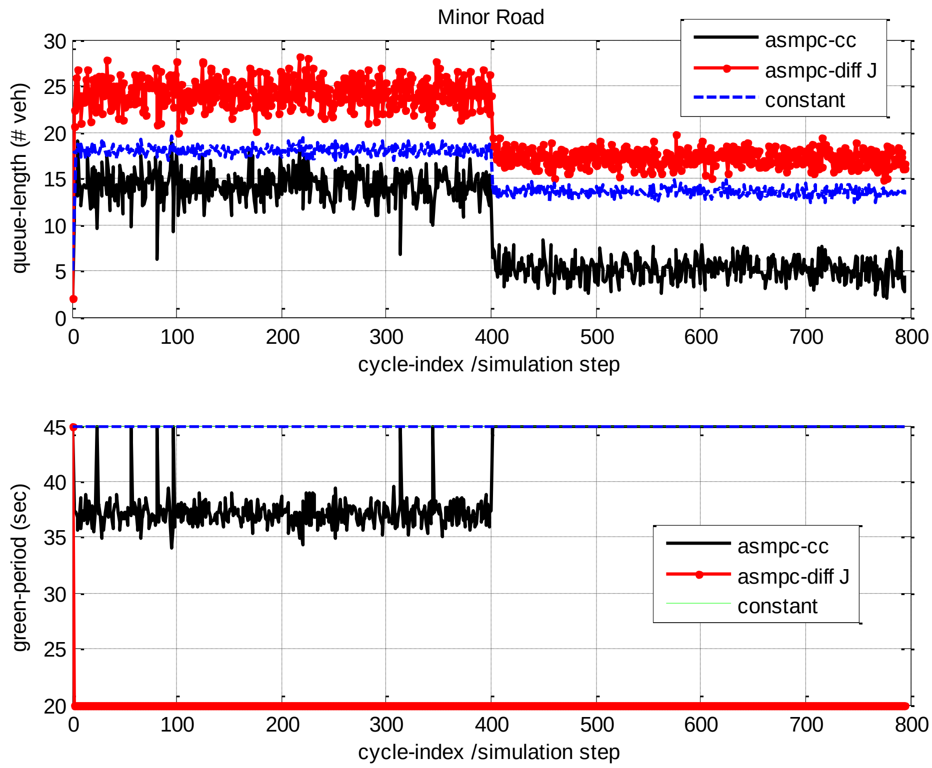

Figure.6.8 Queue-length evolution (upper) and green period adaptation (lower) for L_4

To evaluate the performance of the *asmp-cc* in more general case, we examine by considering the longer simulation in order to evaluate better the probability of the chance constraint being satisfied. Monte Carlo simulation of the closed-loop system is performed using 800 random vectors generated from the Gaussian pdfs of the traffic flows. Fig.6.6 and Fig.6.8 reveal its ability in shaping the probability distribution of system states, as well as ensuring the fulfillment of state constraints in a stochastic setting. From the whole simulation, the result fulfills the requirement that the probability constraint of the queue length is below the threshold upper limit of 90 %. This is a good indication that the convex approximation of chance constraint using convex bounding method is successfully implemented in this case.

As we know that performance of the *asmpc-cc* and the *asmpc-diff J* strongly depend on the quality of the state prediction. The performance of joint state-parameter prediction is shown in Fig.6.9 by using a measure of *relative error* (RE):

$$relative\ \ error\ (\%) = \frac{(\hat{x} - x)}{x} \times 100\% \tag{6.23}$$

where term $\hat{x}$ in equation (6.23) indicates the results of estimator, 1-predicor and 2-predictor for arrival flow $\lambda_{1,2k+1}$. The RE measure allows us to compare the performance of each estimators/prediction in the percentage form in the same graph.

The result in Fig.6.9 shows that the relative error of joint state-parameter estimation is less than 10%, especially excluding the initial time and the transition time around index 20 at the time when traffic flow intensity is changing rapidly

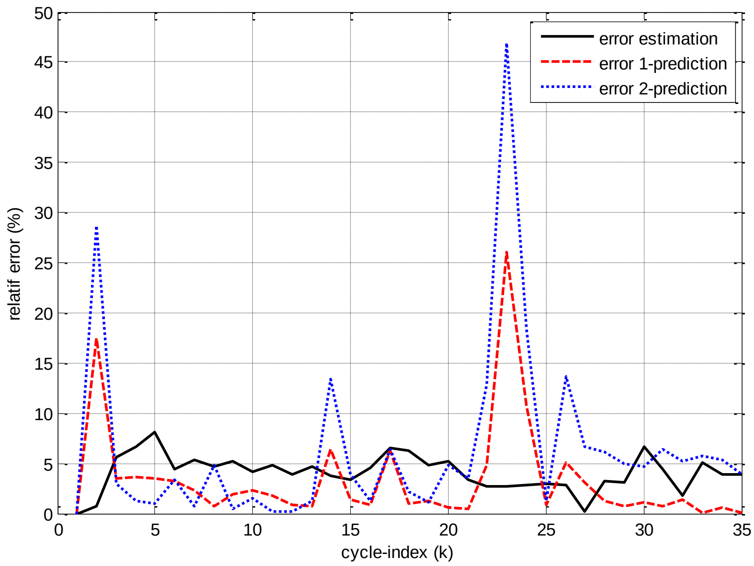

Figure 6.9 Performance of joint state-parameter estimator

6.6 Conclusions

This chapter proposes the adaptive stochastic control by integrating with joint state-parameter estimation technique as proposed in Chapter 5 into the control loop. The convex bounding method developed by Nemirovski [7] is used to approximate the individual chance constraint on excessive queue lengths that might lead to spillback. This convexification enables the successfully implementation of the stochastic control for avoiding queue spillback in the critical intersection. The joint state-parameter estimation is used to detect in real-time the link with long queue larger than critical queue length q_{cr}.

The control that we propose is a potential technique to decrease the likelihood of gridlock on the network. One of the future extensions for decreasing the likelihood of gridlock is by imposing *a joint* chance constraint (in our application, a joint chance constraint can be represented as a set of constraint on several incoming links each imposing limits on the chance of a queue length exceeding a critical value. It is then an important challenge to find globally optimal control inputs by decentralized optimization problem under stochastic uncertainty with a coupling through a joint chance constraint. This is an interesting topic for further research.

7
Conclusions and future works

7.1 Conclusions

This results presented in this thesis have developed an adaptive stochastic control, integrating a joint state-parameter estimation algorithm with a risk constrained stochastic control for the control of an urban traffic signal control. Using a stochastic hybrid model framework, it is possible to predict the queue-length evolution at a signalized intersection, using past observation of traffic flow in the neighborhood of the intersection on the model represents the interaction between the selected traffic light sequences, arrival/departure traffic flow, and the queue lengths. This interaction involves both event-driven dynamics and time-driven dynamics. The event-driven dynamics is dictated by green-red light switches, by switching of the traffic mode of operation, and by events causing some queue lengths to switch from positive value to zeros or vice versa. Since the discrete event and discrete time dynamics of such models are coupled, partially observable and stochastic, this is a very challenging problem in the field of data-driven estimation and control. In general, the traffic flow can be modeled as jump Markov model which is a multi–mode model and in each mode is modelled by first-order autoregressive (AR) model.

Because the ultimate aim is to reduce the queue length in each of the incoming lanes, some of which are connected to the nearby adjacent intersection, then the good choice of the signal traffic sequence is crucial in order to fulfill the aim of minimizing delays and avoiding spillbacks. Different possible selections of the signal traffic sequence are compared, so as to develop a feedback control strategy where the objective function can be represented as the total average delay time experienced by the vehicles in all queues, ensuring the risk is small of having queues that are so long that they cause deterioration of the performance upstream.

Parameter estimation are used to determine the unknown parameters of the stochastic hybrid model. The parameters of the AR process take different values depending on the mode of traffic operation – free flowing, congested or faulty – making this a hybrid stochastic process. Mode switching occurs according to a first-order Markov chain. This thesis proposes both an offline technique and an online technique for estimating the transition matrix of this Markovian mode process and the parameters of the AR models describing flow rates for each mode. The offline technique uses an expectation-maximization (EM) technique while the online technique utilize a particle filter approach. Both techniques have been validated both using actual traffic flow and VISSIM traffic simulator as discussed in previous chapters.

Online joint state-parameter estimation is based on the key idea of looking at the most likely mode as an unknown system parameter. The system is assumed to follow the dynamics of this most likely mode. A particle filtering approach is used for jointly estimating the transition matrix of the traffic modes, the parameters of the autoregressive (AR) model describing the evolution of the traffic flow rates, as well as for estimating and predicting the traffic flow rates and the queue length. The method has been made computationally feasible by utilizing an optimal-tuning kernel smoothing approach to estimate the parameters of the first order AR model in combination with a Dirichlet distribution approach to estimate the transition probability matrix of the first-order Markov chain representing the traffic flow modes.

In this thesis, we merely use the online approach or joint state-parameter estimation to integrate with the adaptive stochastic control and the performance of controller is defined by the objective function with constraints on the inputs and states that should be satisfied in the presence of uncertainties. The identified parameters from parameter estimation, along with cycle length and green period, are used as an input, together with the new observations on arrival/departure flows, in order to estimate the queue-length. In this thesis we have shown that this possible with an accuracy which is sufficient both for feedback control and to detects queues that exceed the capacity of the corresponding approach and may block the upstream intersection. Therefore, the accurate state estimation is useful to predict the effect of control decision over a sufficiently long prediction horizon.

The proposed stochastic control is demonstrated using a micro-simulated case study to give satisfactory results. In fact we found better results than those obtained with others controllers, both a fixed controller and an adaptive stochastic with deterministic constraints. The proposed controller is able to fulfill all the constraints including the chance constraint which is that the probability of the queue length on road L_1 is ensured below of some predetermined threshold during the whole simulation.

There are a set of limitation in this approach as follows:

a. we use observations on traffic flow only;
b. Gaussian assumption in modeling of traffic flow might possibly to a negative estimate of traffic flow;
c. cycle length is a constant
d. our SHM can be classified as unguarded transition.

7.2 Future works

In this section, we list some open problems that can be elaborated in further research and that will help removing the limitations that are still limiting the practical application of the methods proposed in this thesis.. We can classify the future research topics under three main headers:

7.2.1 Interesting improvements of the fluid flow model approach

a. Fluid flow model for another application

It may be interesting to see how our fluid flow approach, including detection of zero crossings of queues can be used for other system such as manufacturing system [72].

b. Fluid flow model for perimeter control

The fluid flow approach that we used in this thesis in order to define the queue length evolution can be applied to larger urban network in order to be able to support into perimeter control design. This model is currently under development in order to estimate the queue length and traffic flow for the case of large network in city of Surabaya, Indonesia which cover almost 60 signalized intersections. This will then be used as a basis tool for developing perimeter control for this large urban area.

c. Stochastic hybrid model with Guarded Transition

As discussed in chapter 4 an important improvement to our unguarded SHM would be to include guarded transition, where the switching between different modes of operation depends on the queue length and on the traffic flow state. The stability condition of the AR model can be used to characterize *the guard condition* [76] in our stochastic hybrid model.

7.2.2 Possible improvements to the parameter estimation algorithms

a. Offline Technique :

Offline EM technique can be potentially extended to assist an algorithm for traffic estimation, particularly for urban roadway sections, with missing data, representing failure of detector. One possible approach is to extend the flow model with an ARIMA-based approach [73]. This could also be extended taking into account the correlation between the flow in the section of interest and its adjacent stream approaches (upstream and/or downstream).

In the context of developing stochastic hybrid model with guarded transition, one possible approach is to follow the spirit of a method that is proposed by Santana in his paper [76].

b. Online Technique : Particle filter approach

In our online joint state-parameter estimation technique, one of parameter that we estimated is the transition probabilities matrix (TPM). This TPM is estimated based on the most likely mode in OTPF filter along with Dirichlet distribution. The different approach proposed by Mihaylova in [71] by using *Changepoint approach* with auxiliary particle filter (APF) leads to adaptive parameter estimation. This approach assumes that the set of possible values of the TPM is a priori given. It is interesting that by making this assumption, we may compare to the performance between OTPF and APF for joint state-parameter estimation.

7.2.3 Adaptive Stochastic Control with Chance Constraint

The perimeter control along with the proposed control in this thesis will be developed to the case of large network in city of Surabaya, Indonesia which covers almost 60 signalized intersections. In this case, we would like to propose traffic signal control in congested urban arterials by integrating a clustering approach (through state-estimation as we propose in this thesis) into a feedback control policy avoiding the occurrence of long queues, that might cause spillback reducing the risk that congestion could spread globally. First, we plan to introduce an arterial clustering approach that detects in real-time the links with long queues along one direction of the arterial, clustering them together if they are consecutive and then identifying the entrance and exit intersections of each cluster. These intersections indeed constitute critical junctions and therefore, proper adjustment of the signal timing settings in those intersections has the potential to improve traffic conditions in the whole arterial. The adjustment can be realized by implementing a traffic signal control strategy that acts not only on the exit intersection as thesis did, but also to acts on entrance intersections along the arterial. This can be performed by using adaptive stochastic with chance constraint proposed in this thesis. Thus, it will enable implementation of a locally smaller-sized (and computationally simpler) decentralized traffic signal control strategies while ensuring at the same time the global coherence of these strategies along the arterial. In this framework, we will seek to improve traffic on arterial at a very low cost by acting merely on critical intersections, as opposed for instance to network-wide optimization strategies line SCATS or SCOOT. Hence, the purpose of our approach is to develop a signal control strategy based on the arterial clustering approach that enables to act only locally on specific intersections. Including an advanced detection of oversaturated states and a specific focus on queue spillovers prevention, it can lead to a computationally feasible significant reduction of congestion and thus improves the network traffic conditions.

The one possibility that could be developed is to optimize the cycle-length in the supervisor level of a network.

7.3 Summary of Contributions

The contribution of this thesis can be found in the list of publications as bellows:

1. **HY Sutarto**, René Boel and Endra Joelianto " *EM-Parameter estimation for stochastic hybrid model applied to urban traffic flow estimation*", IET Control Theory and Applications, Volume 9, Issue 11, 2015
2. **HY Sutarto** and Endra Joelianto, "*Expectation-Maximization Based Parameter Identification for HMM of Urban Traffic Flow*", Int.J.Appl.Math.Stat;Vol.53;Issue No.2. 2015
3. **HY Sutarto**, René Boel and A.Nugroho "*On-line Bayesian State-Parameter Estimation in Stochastic Hybrid Model Applied to Queue-Length Estimation*" (2015-Under-reviewed)
4. **HY Sutarto** and Endra Joelianto, "*Modeling, Identification, Estimation and Simulation of Urban Traffic Flow in Jakarta and Bandung*", Journal of Mechatronics, Electrical Power, and Vehicular Technology, Vol 6 No 1,2015
5. **HY Sutarto** and René Boel, "*Adaptive Stochastic Control with Chance Constraints for Urban Traffic Signal Control : A Stochastic Hybrid Model Approach*" , Preprints

6. Renato,Vasquez, **HY Sutarto**, René Boel and Manuel Silva, ”*Hybrid Petri net model of a traffic intersection in a urban network*”, IEEE Multi-conference on Systems and Control, Yokohama, Japan, sept 2010.
7. **HY Sutarto** , Endra Joelianto and Taufik.S Sumardi, “*Estimation and Prediction of road traffic flow using particle filter for real-time traffic control*”, 2nd IEEE Conference on Control, Systems & Industrial Informatics, 2013

EU Project Meeting and National Meetings

1. **H.Y Sutarto** and René Boel ,” *Hybrid Automata Model Approach for Coordinating Traffic Signal Control*”, 29th Benelux Meeting on Systems and Control, Heeze-Netherland, 2010
2. **H.Y Sutarto** and René Boel ,” *Road traffic modeling with fluid Petri nets*” 2nd Project-Meeting DISC (Distributed Supervisory Control of Large Plants- EU- FP7.ICT program), Ghent, Belgium ,April 2009
1. **H.Y Sutarto** and René Boel, ” *Fluid flow models for road traffic control*” 3rd Project-Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), Universidad Zaragoza, Spain, Sept 2009
2. **H.Y Sutarto** and René Boel, ” *Modeling, estimation and control of road traffic using fluid flow model*” 4th Project- Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), INRIA- Rennes, France, March 2010
3. **H.Y Sutarto** and René Boel*, ” Application of Infinitesimal Perturbation Analysis for Coordinating road traffic network through Fluid Flow model*”, 5th Project- Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), TU-Berlin, Germany, Sept 2010
4. **H.Y Sutarto** and René Boel,” *Queue size estimation for signalized intersections* “, 6th Project Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), CWI, The Netherlands, March 2011
5. **H.Y Sutarto** and René Boel, “*Statistical validation of fluid flow models and their use for control of urban traffic* ”, 7th Project Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), Institute of Mathematics, Brno, Czech, Nov, 2011

6. **H.Y Sutarto** and René Boel ,"Perturbation analysis and sample-path optimization : stochastic flow models of urban traffic networks case", *Interuniversity Attraction Pole IAP VI/4 DYSCO Study Day,2009*
7. **H.Y Sutarto** and René Boel," Coordinating Road Traffic Network", 11de FirW Doctoraatssymposium, Universiteit Gent,Dec 2010

Bibliography

1. Aboudolas, Konstantinos and Nikolas Geroliminis, ”Perimeter and boundary flow control in multi-reservoir heterogeneous networks”, Transportation Research Part B 55 (2013) 265–281.
2. Nikolas Geroliminis, Jack Haddad, and Mohsen Ramezani, ”Optimal Perimeter Control for Two Urban Regions With Macroscopic Fundamental Diagrams: A Model Predictive Approach” IEEE Transaction on Intelligent Transportation Systems, Vol 14, issue 1, March 2013.
3. Daganzo, C.F, “Urban gridlock: Macroscopic modeling and mitigation approaches,” Transp. Res. B: Methodol., vol. 41, no. 1, pp. 49–62, Jan. 2007.
4. Kumar, P.R, and Pravin Varaiya, ”Stochastic Systems: Estimation, Identification and Control”, Prentice Hall, 1986
5. Blackmore, Lars and M. Ono, ”Convex chance constrained predictive control without sampling,” in Proceedings of the AIAA Guidance, Navigation, and Control Conference, (Chicago, Illinois), August 2009.
6. D. van Hessem, “Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation. PhD thesis, Delft University of Technology, 2004.
7. Nemirovski, A and A. Shapiro, ”Convex approximations of chance constrained programs,” SIAM Journal of Optimization, vol. 17, no. 4, pp. 969-996, 2006
8. Anuj Sharma, D.M Bullock and J.A Bonneson,” Input-Output and Hybrid Techniques for Real-Time Prediction of Delay and maximum Queue-Length at Signalized Intersections”, Transportation Research Board Journal of the Transportation Research Board No. 2035, 2007, pp 69-80
9. Ilya Ioslovich, Jack Haddad, Per-Olof Gutman and David Mahalel,” Optimal traffic control synthesis for an isolated intersection”, Control Engineering Practice 19, (2011),900-911
10. Daikaki, C, Papageorgiou, M and Kostas Aboudolas, ”A Multivariables regulator approach for traffic-responsive network-wide signal control”, Control engineering Practice 10,(2002), 183-195

11. Yafeng Yin, ”Robust optimal traffic signal timing”, Transportation Research part B 42(2008) 911-924
12. Haddad, J, Bart de Schutter, David Mahalel and Per-Olof Gutman,”Steady-State and N-stages Control for Isolated Controlled Intersections”, Proceeding American Control Conference, 2009, St. Louis, MO, USA, June 2009, pp. 2843-2848.
13. Nicolae Marinica and René Boel ,”Platoon based model for urban traffic control”, Proceedings of the American Control Conference. p.6563-6568, 2012
14. Sutarto,HY, René Boel and Endra Joelianto ”EM-Parameter estimation for stochastic hybrid model applied to urban traffic flow estimation”, IET Control Theory and Applications, Volume 9, Issue 11, 2015
15. Vazquez, C.Renato; HY Sutarto, René Boel and Manuel Silva,”Hybrid Petri net model of a traffic intersection in a urban network”, IEEE Multi-conference on Systems and Control, Yokohama, Japan, sept 2010.
16. Stephen Boyd and L. Vandenberghe, ”Convex Optimization”, Cambridge, England: Cambridge University Press, 2004.
17. Simon J.D. Prince, ”Computer vision: models, learning and inference”, Cambridge University Press,2012
18. Sutarto, HY, Endra Joelianto and Taufik.S Sumardi, ”Estimation and Prediction of road traffic flow using particle filter for real-time traffic control”, 2nd IEEE Conference on Control, Systems & Industrial Informatics, 2013
19. M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, ”A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” IEEE Trans. on Signal Processing, vol50, pp. 174-188, 2002.
20. Liu, J. and M. West (2001),” Combined parameter and state estimation in simulation-based filtering”, In A. Doucet, N. de Freitas, and N. Gordon (Eds.), Sequential Monte Carlo in practice, pp. 197-223. New York: Springer-Verlag.
21. Chen, T., J. Morris, and E. Martin (2005),”Particle filters for state and parameter estimation in batch processes”, Journal of Process Control 15, 665-673.
22. J. Hu, J. Lygeros, and S. Sastry, ”Towards a theory of stochastic hybrid systems,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, N. Lynch and B. H. Krogh, Eds. New York: Springer Verlag, 2000, vol. 1790, pp. 160–173.

23. Davis, M. H. A. (1984). "Piecewise-Deterministic Markov Processes: A General Class of Non-Diffusion Stochastic Models". *Journal of the Royal Statistical Society. Series B (Methodological)* 46 (3): 353–388.
24. Wardi,Y, Egerstedt,M and Hale,M ,"Switched-mode systems: gradient-descent algorithms with Armijo step sizes , "Discrete Event Dyn Syst , 25:571–599, 2015
25. Sutarto, HY, René Boel, "Modeling and control of road traffic using fluid flow model", Rennes - Fourth DISC Meeting, 2010.
26. Julvéz, J. and Boel, R, "A Continuous Petri Net Approach for Model Predictive Control of Traffic Systems", IEEE Transactions on Systems Man and Cybernetics – Part.A Systems and Humans, August, 2010
27. I. Ljung , "System Identification: Theory for the user", System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition, 1999
28. Thomas B Schön, Adrian Wills and Brett Ninness, "System Identification of Nonlinear State-Space Models", Automatica Vol 47, Issue 1, Jan 2011
29. Z. Ghaharamani and S.T Rowels, "Learning nonlinear dynamical systems using an EM algorithm", In Advances in Neural Information Processing Systems, volume 11, MIT Press,1999.
30. Dempster, A., Laird, N., Rubin, D.B, 'Maximum likelihood from incomplete data via the EM algorithm', *J. R. Stat. Soc.*, 1977, **39**, pp. 1–38
31. Hamilton, J, "Analysis of time series subject to changes in regime", J. Econometrics., 1990, 45, pp. 39–70
32. Kim, C.J, "Dynamics linear model with Markov-switching', J. Econometrics, 1994, 60, pp. 1–22
33. Moon, T.K, "The expectation-maximization algorithm", IEEE Signal Process. Mag., 1996, 13, (6), pp. 47–60
34. Wu, C.F, "On the convergence properties of the EM algorithm", Ann. Stat.,1983, 11, pp. 95–103
35. Tafazoli, S., Sun, X, "Hybrid system state tracking and fault detection using particle filters", *IEEE Trans. Control Syst. Technol.*, 2006, **14**, (6), pp. 1078–1087
36. de Oliveira, L.B., Camponogara, E, "Multi-agent model predictive control of signaling split in urban traffic networks", *Transp. Res. C*, 2010, **18**, (1), pp. 120–139
37. Aboudolas, K., Papageorgiou, M., Kouvelas, A., Kosmatopoulos, E, "A rolling horizon quadratic programming approach to the signal control problem in largescale congested urban road networks", Transp. Res. C, 2010, 18, (5), pp. 680–694

38. Sutarto, HY and René Boel, “Statistical validation of fluid flow models and their use for control of urban traffic ”, 7th Project Meeting DISC (Distributed Supervisory Control of Large Plants- EU-FP7.ICT program), Institute of Mathematics, Brno, Czech, Nov, 2011
39. Sutarto, HY and Endra Joelianto, ”Expectation-Maximization Based Parameter Identification for HMM of Urban Traffic Flow”, . Int.J.Appl.Math.Stat;Vol.53;Issue No.2. 2015
40. Sutarto,HY and Endra Joelianto,” Modeling, Identification,Estimation and Simulation of Urban Traffic Flow in Jakarta and Bandung”, Journal of Mechatronics, Electrical Power, and Vehicular Technology, Vol 6 No 1,2015
41. Sutarto, HY, ”Modeling and Simulation of Queue-length on Signalized Intersection using Discrete Event Simulator” (in Bahasa Indonesia), Jurnal Telematika, Vol.9 no 2, Institut Teknologi Harapan Bangsa,(http://journal.ithb.ac.id/index.php/telematika)
42. Wang, Y., Papageorgiou, M., Messmer A., Coppola, P., Tzimitsi, A and Nuzollo, A,” An adaptive freeway traffic state estimator”, *Automatica* 45(1): 10-24, 2009.
43. Mihaylova, L., Boel, R and Hegyi, A,”Freeway Traffic Estimation within Particle Filtering Framework”, *Automatica* 43(2): 290-300, 2007
44. Carvalho C.M and Lopes HF, ”Simulation-based sequential analysis of Markov switching stochastic volatility models”, Computational Statistics and Data Analysis 51, 4526–4542,2007
45. Tulsyan, A; Huang,B; Gopaluni, R.B and Forbes, J.F, ”On simultaneous on-line state and parameter estimation in non-linear state-space models”, Journal of Process Control.23(4), 2013
46. Ristic, B; Arulampalam,S and Gordon,N, ”*Beyond the Kalman filter: particle filter for tracking applications”,* Artech House Boston, 2004
47. Chen, Z, ”Bayesian filtering from Kalman filters to particle filters and beyond”, Communication Research Laboratory.McMaster University, Hamilton, Canada, Tech.Rep, 2003
48. Xue, Y and Runolfsson, T, ”Efficient estimation of hybrid systems with application to tracking. International Journal of System Science, DOI:10.1080/00207721.2011.5666-41, 2011
49. Blom, H.A.P and Bloem, E.A,”Exact bayesian and particle filtering of stochastic hybrid system”, IEEE Transactions on Aerospace and Electronic Systems,43(1), pages 55-70, 2007

50. Kantas, N., Doucet, A., Singh, S and Maciejowski, J,"An overview of sequential Monte Carlo methods for parameter estimation on general state-space models," in Proceedings of the 15th IFAC Symposium on System Identification, 2009
51. Tulsyan, A., Huang, B., Gopaluni, R.B., and Forbes, J.F, "Bayesian identification of non-linear state-space models: Part II - Error Analysis", In: *Proceedings of the IFAC 10th International Symposium on Dynamics and Control of Process Systems (DyCoPS),Mumbai, India*, 2013
52. Chib, S, "Calculating posterior distributions and model estimates in Markov mixture models", Journal of Econometrics 75,p 79-97, 1996
53. Frigyik, B.A; Kapila, A and Gupta, M.R," Introduction to the Dirichlet Distribution and Related Processes", Technical Report Number UWEETR-2010-0006, Department of Electrical Engineering, University of Washington, 2010
54. Nicolae Emanuel Marinica, " Distributed Estimation and Control of Interacting Hybrid Systems for Traffic Applications", PhD Thesis Universiteit Gent Belgium, 2014
55. Doyle,J.C, Glover,K, Khargonekar,P and Francis, B," State Space Solution to standard H_2 and H_∞ optimal control problem", IEEE Trans-Automatic Control, Aug, 1989
56. Apkarian,P, Gahinet,P, "A Convex Characterization of Gain Scheduled H_∞ Controllers", IEEE Trans.Automatic Control, vol.40, no.5, pp853-864,1995
57. Sutarto, HY, Budiyono, A, Joelianto, E, and Hiong, GT," Switched Linear Control of a Model Helicopter", 9th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 2006
58. Blackmore, Lars, Masahiro, Ono, Bektassov Askar, and Williams, B.C,"A probabilistric particle-control approximation of chance constrained stochastic predictive control", IEEE Trans on Robotics, Vol. 26, No 3, June 2010.
59. Oldewurtel, F, et al, "Adaptively constrained stochastic model predictive control for closed loop constraint satisfaction", Proc of American Control Conference, 2013.
60. Paulson, J.A, Streif,S and Mesbah, A,"Stability for Receding-horizon Stochastic Model Predictive Control", Proc of American Control Conference, 2015
61. Ekbatani, M.K, Papageorgiou, Knoop, V.L,"Controller design for gating traffic control in presence of time-delay in urban road networks", Transportation Research Procedia 7, 651-668, 2015
62. Gayah, V.V, Gao,X, Nagle,A.S, "On the impacts of locally adaptive signal control on urban network stability and the Macroscopic Fundamental Diagram", Transportation Research Part B: Methodological, Vol 70, pp 255-268, Dec 2014.

63. Godfrey, J, ”The mechanism of a road network” Traffic Engineering and Control 11, 323-327, 1969
64. Daganzo, C., Gayah, V., Gonzales, E, ”Macroscopic Relations of Urban Traffic Variables: Bifurcations, multivaluedness and instability”, Transportation Research Part B 45, 278-288, 2011
65. Biegler, Lorentz T, ”Nonlinear Programming”, MOS-SIAM Series on Optimization, 2010
66. Douc, R., Cappe, O., and Moulines, E., “Comparison of Resampling Schemes for Particle Filtering,” International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, pp. 64–69, Sept. 2005.
67. M. P. Wand and M. C. Jones*, “*Kernel Smoothing*”*. London, U.K.: Chapman & Hall, 1995
68. Ripley, B.D, ”Stochastic Simulation”, John Wiley and Son, 1987
69. Doucet, A., Freitas, N., and Gordon, N., eds., “Sequential Monte Carlo Methods in Practice”, Springer-Verlag, 2001.
70. Vitus, M.P and Tomlin, C.J, ”A Hybrid Method for Chance Constrained Control in Uncertain Environments”, Proceedings of the 51th IEEE Conference on Decision and Control, Maui, Hawaii, December, 2012.
71. Nemeth, C., Fearnhead, P., Mihaylova, L, ”Sequential Monte Carlo Methods for state and parameter estimation in abruptly Changing Environments”, IEEE Transaction on Signal Processing, Vol 62, Issue 5, 2013.
72. Lefeber E. & Rooda J.E, ”Controller design for switched linear systems with setups”, *Physica A*, vol. 363, 48–61, 2006.
73. Chang, Tang-Hsien., Chueh, C.H., and Yang, L.K, ”Dynamic Traffic prediction for insufficient data roadways via automatic control theories”, Control Engineering Practice,19, 1479-1499, 2011.
74. Summers,T.H.,Warrington,J.,Morari, M.,Lygeros,J,” Stochastic optimal power flow based on convex approximations of chance constraints”, Proceedings of Power Systems Computation Conference, Poland, 2014
75. Gordon, N. J., Salmond, D. J., and Smith, A. F. M., “Novel approaches to nonlinear/non-Gaussian Bayesian state estimation,” IEE Proceedings-F, vol. 140,pp. 107–113, Apr 1993.

76. Pedro Santana, Spencer Lane, Eric Timmons, Brian Williams, Carlos Forster, “Learning Hybrid Models with Guarded Transitions”, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
77. Calafiore, G.C and M. C. Campi, “The scenario approach to robust control design," IEEE Transactions on Automatic Control, vol. 51, pp. 742-753, May, 2006.
78. Millan, J.M, ”Online Discrete Event Control of Hybrid Systems”, PhD Thesis Memorial University of Newfoundland, 2006
79. Schön, Thomas B,” Solving Nonlinear State Estimation Problems Using Particle Filters – An Engineering Perspective”, Technical report no.: LiTH-ISY-R-2953, from Automatic Control at Linköpings universitet, 2010.
80. Vlahogianni, E.I.,Karlaftis,M.G.,&Golias,J.C.(2008),”Temporal evolution of short-term urban traffic flow: a nonlinear dynamics approach”, Computer-Aided Civil and Infrastructure Engineering, 23, 536–548
81. Karlaftis, M.G.,&Vlahogianni,E.I.(2009),”Memory properties and fractional integration intransportationtime-series” Transportation Research PartC, 17, 444–453.
82. SCATS 6, ,”Functional Description Manual”, http://www.aldridgetrafficcontrollers.com.au/ArticleDocuments/47/ATC_An_Introduction_To_The_New_Generation_Scats_6%20_5_%203_.pdf.aspx
83. The SCOOT Urban Traffic Control System Website, http://www.scoot-utc.com/
84. VISSIM 5.10-03 COM Interface Manual, PTV Vision, 2008
85. Trafficon, Traffic Video Detection Cameras Website, http://www.traficon.be/
86. Geroliminis, N., Haddad, J., Ramezani, M.,” Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: a model predictive approach”, IEEE Transactions on Intelligent Transportation Systems 14 (1), 348–359,2013
87. Haddad,Jack and Shraiber Arie, ”Robust perimeter control design for an urban region”, Trsnaportation research Part B 68 (2014), pp 315-332
88. M. Morari, J. H. Lee,”Model predictive control: past, present and future”, Computers & Chemical Engineering 23 (4-5) ,667–682,1999
89. A. Bemporad, M. Morari, ”Robust model predictive control: A survey”, in: A. Garulli, A. Tesi (Eds.), Robustness in Identification and Control, Springer, Berlin, London, New York, pp. 207–226, 1999.
90. S. Garatti, M. C. Campi, ”Modulating robustness in control design: Principles and algorithms”, IEEE Control Systems Magazine 33 (2), pp 36-51, 2013

91. A. Schwarm, M. Nikolaou, ”Chance-constrained model predictive control”, AIChE Journal 45, 1743–1752, 1999
92. P. Li, M. Wendt, G. Wozny, ”Robust model predictive control under chance constraints”, Computers & Chemical Engineering 24 (2),829–834, 1999
93. D. H. V. Hessem, O. H. Bosgra, ”A full solution to the constrained stochastic closed-loop MPC problem via state and innovations feedback and its receding horizon implementation”, in: Proc. 42nd IEEE Conference on Decision and Control (CDC), Maui, pp. 929–934, 2003
94. Roess, Roger P, Prassas Elena S, and McShane William R,” Traffic Engineering”, Pearson Education International, Third Edition, 2004
95. Transport research board ,” Highway capacity manual”. fifth edition, TRB, 2010
96. Sanderson A,”Traffic Signal Design Terminology”, available online: http://www.traffic-signal-design.com/terminology_main.htm, 2016
97. Wardrop, J.G, ”Journey speed and flow in central urban areas” *Traffic Engineering and Control*, 9(11):528-532, 1968
98. Zahavi, Y, ” Traffic performance evaluation of road networks by the relationship. Parts I and II”, *Traffic Engineering and Control*, 14(5 and 6):228-231,292-293, 1972
99. Herman, R. and Prigogine, I,” A two-fluid approach to town”, Science, 204:148-151, 1979
100. Geroliminis, N., Daganzo, C.F.,”Existence of urban-scale macroscopic fundamental diagrams: some experimental findings”, Transportation Research Part B: Methodological 42 (9), 759–770, 2008
101. Buisson, C. and Ladier, C.,” Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams”, Transportation Research Record: Journal of the Transportation Research Board, 2124:127–136, 2009
102. Aboudolas K, Papageorgiou M, Kosmatopoulos E, ”Store-and-forward based methods for the signal control problem in large-scale congested urban road networks”, Transport Res C;17:163–174, 2009a
103. Aboudolas K, Papageorgiou M, Kouvelas A, Kosmatopoulos E,”A rolling-horizon quadratic programming approach to the signal control problem in large-scale congested urban road networks”. Transport Res C;17:680–694, 2009b
104. B. D. 0. Anderson and J. B. Moore, ”*Optimal Filtering”***.** Englewood Cliffs, NJ, Prentice-Hall, 1979.

105. Varaiya, Pravin,"The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections", in S.V. Ukkusuri and K. Ozbay (eds.), *Advances in Dynamic Network Modeling in Complex Transportation Systems*, Complex Networks and Dynamic Systems 2, DOI 10.1007/978-1-4614-6243-9 2, © Springer Science+Business Media New York 2013